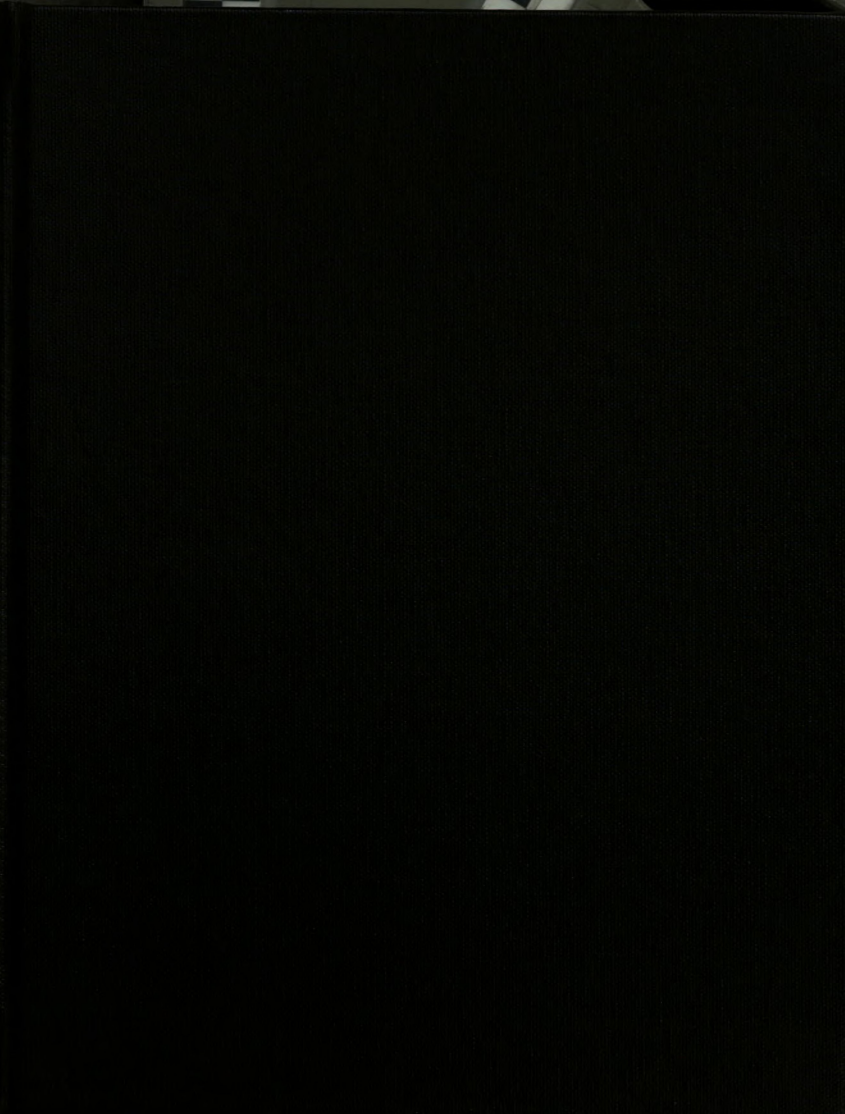






3 1293 02058 1876

This is to certify that the
thesis entitled
Development, Application, and Evaluation
of the Archaeological Site Prediction
System

presented by

Jessica Jo Dolanski

has been accepted towards fulfillment
of the requirements for

M.A. degree in Geography

A handwritten signature in dark ink, appearing to read "David S. Brown", written over a horizontal line.

Major professor

Date 10/21/97

SUPPLEMENTARY

MATERIAL

DIGITAL MEDIA CENTER

in _____

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.

TO AVOID FINES return on or before date due.

MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

SUPPLEMENTARY
DIGITAL MEDIA CENTER
MATERIAL in _____

DEVELOPMENT, APPLICATION, AND EVALUATION OF
THE ARCHAEOLOGICAL SITE PREDICTION SYSTEM

By

Jessica Jo Dolanski

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF ARTS

Department of Geography

1997

ABSTRACT

DEVELOPMENT, APPLICATION, AND EVALUATION OF THE ARCHAEOLOGICAL SITE PREDICTION SYSTEM

By

Jessica Jo Dolanski

Archaeological site prediction models depict regions of the landscape that have archaeological potential. Traditional models were developed manually by archaeologists, while contemporary methods of site prediction use Geographic Information Systems (GIS). This technology expands the capabilities of site prediction modeling, but few archaeologists possess the needed computer skills to run a GIS. Consequently, site prediction models are being developed by GIS technicians rather than archaeologists.

This thesis develops, applies, and evaluates a site prediction system with a user-friendly interface that allows archaeologists to analyze their data and create site prediction models. The Archaeological Site Prediction System (ASPS) was developed by customizing ArcView 3.0 and consists of six components: set-up, data manipulation, deductive modeling, inductive modeling, evaluation, and help/information. Exploratory data analysis was performed with Athienou Archaeological Project data. Professional archaeologists and geographers evaluated the ASPS. Results indicate that the ASPS is adequate for the development of contemporary site prediction models.

Copyright by
Jessica Jo Dolanski
1997

To my parents
Daniel and Patricia Dolanski
and
in memory of
grandma G.J. and great-grandpa Wiese.

ACKNOWLEDGEMENTS

It is often said that a thesis is a personal journey that one must travel on her own, in her own time, and with her own strength. I disagree. My thesis could not have been completed without the guidance, patience, and encouragement of several individuals. To these individuals, I owe my sincere gratitude.

Thank you to my advisor, Daniel Brown, and my committee members, Jon Burley, and Richard Groop. Your insights, suggestions, and individual geographic perspectives were essential to the completion of my thesis. Together, you transformed an archaeologist into a geographer.

Thank you to my parents. Mom, throughout this endeavor you have challenged me to persevere, comforted me when I was discouraged, and served as my faithful editor; dad, you have been my voice of experience, my ever-present listener, and my cherished “Must See TV” study break. My thesis could not have been completed without the love and inspiration I have received from you both. I am proud to have you as my parents, and privileged to have you as my friends.

Thank you to Derek. I hold you in my highest esteem. You have endured my frustrations, supported me in my weakest hours, and somehow managed to find the countless “bugs” that eluded me. You are my sweetheart. E & B. A & E.

Thank you to grandma G.J. and great-grandpa Wiese. These individuals, through their lives, taught me that learning is a life-long adventure. They showed me that wisdom does not come from university degrees, but from a lifetime of experiences, good books, and meaningful discussion. To these wise souls, I owe my contentment in being both a university student and a student of life.

And finally, thank you to the Athienou Archaeological Project. I greatly appreciate the use of your data and the opportunities afforded me through a summer in Cyprus. May the Archaeological Site Prediction System be instrumental in the discovery of the archaic town in Malloura.

TABLE OF CONTENTS

List of Tables	ix
List of Figures	x
I. Introduction and Background	1
Statement of Problems and Objectives	1
Site Prediction Modeling	2
Traditional Site Prediction Methods	5
Contemporary Site Prediction Methods	5
Relevant Trends in GIS	6
II. Study Area and Data Acquisition	9
Regional and Geologic Setting	10
Excavations and Surveys to Date	11
Athienou Archaeological Project and the Archaeological Site Prediction System	12
Description of Data	14
Manipulation of Data	15
III. Methods for Creating, Applying, and Evaluating the Archaeological Site Prediction System	19
Design Elements for the Archaeological Site Prediction System	20
Customization of ArcView 3.0	23

Application of the Archaeological Site Prediction System to the Case Study Data	26
Procedure for Evaluating the Archaeological Site Prediction System.....	28
IV. Results and Discussion	31
The Archaeological Site Prediction System and Results from the Exploratory Data Analysis	31
Set-Up Component.....	33
Data Preparation Component.....	35
Deductive Modeling Component.....	36
Inductive Modeling Component	40
Evaluation Component.....	46
Help and Information Component	50
Analysis and Discussion of Professional Evaluations	52
Archaeologist Evaluations	52
Geographer Evaluations.....	63
V. Conclusions	67
Appendix A: List of Map Sources	69
Appendix B: ArcView 3.0 Scripts.....	71
Appendix C: Professionals who contributed to the Evaluation of the Archaeological Site Prediction System	170
Appendix D: Survey and Questionnaires for Professional Evaluations	172
Appendix E: Help Pages from the Archaeological Site Prediction System	174
Appendix F: List of Software and Programming Resources Used.....	203
Bibliography	204

LIST OF TABLES

Table 1: Athienou Archaeological Project Sites	12
Table 2: Map Resources for the Archaeological Site Prediction System	14
Table 3: Ratings for Computer and GIS Skills	29
Table 4: Comparison of Deductive Models at 10 M and 125 M Resolutions	50
Table 5: Positive Aspects of the Archaeological Site Prediction System.....	53
Table 6: Negative Aspects/Limitations of the Archaeological Site Prediction System	55
Table 7: Suggested Improvements for the Archaeological Site Prediction System	56
Table 8: Perceived Impact the Archaeological Site Prediction System and Similar Systems Will Have on the Field of Archaeology.....	57
Table 9: Results of the Geographer Interviews.....	63

LIST OF FIGURES

Figure 1: Reference Map of Cyprus and Study Area.....	9
Figure 2: Malloura Valley and Survey Area.....	10
Figure 3: Structure of the Archaeological Site Prediction System	21
Figure 4: Sample Message Box	21
Figure 5: Sample List Box	22
Figure 6: Sample Input Box.....	22
Figure 7: Components and Sub-commands of the Archaeological Site Prediction System (ASPS)	23
Figure 8: Pull-down Menu and Buttons Added to the View Screen.....	32
Figure 9: Sample View, Following the Add Grids Command.....	34
Figure 10: Sample View, Following the Distance From... Command.....	36
Figure 11: Sample View of a Deductive Site Prediction Model (10 M).....	38
Figure 12: Sample View of a Deductive Site Prediction Model (125 M).....	39
Figure 13: Geology Attribute Table with One Binary Field per Geology Type.....	40
Figure 14: Geology Attribute Table with One Binary Field for Combining Several Geology Types	41
Figure 15: Sample Input Text File	42
Figure 16: Portion from an Arc/Info Logistic Regression Report	45
Figure 17: Sample View of an Inductive Site Prediction Model	46

Figure 18: Sample Histogram for an Inductive Model	47
Figure 19: Sample Histogram for a Deductive Model (10 M).....	49
Figure 20: Sample Histogram for a Deductive Model (125 M).....	50
Figure 21: Help Topics for the Archaeological Site Prediction System	51

CHAPTER ONE

INTRODUCTION AND BACKGROUND

Archaeological site prediction is employed by archaeologists and planners to organize future excavations and protect potential cultural resources. It requires the creation of a spatial model to depict regions of the landscape that have archaeological potential. Geographic Information Systems (GIS), a computer-based tool for overlaying, visualizing, and statistically analyzing multiple layers of data, can assist in site prediction. This thesis describes the development, application, and assessment of a GIS-based site prediction system.

Statement of Problem and Objectives

Traditional site prediction models were mental exercises performed by the archaeologist and based upon his or her *a priori* knowledge of the environmental and social factors that influenced early human settlement. While these models employed the archaeologist's expert knowledge, they were limited by the investigator's inability to mentally process several environmental and cultural factors simultaneously (Binford, 1989). Contemporary GIS-based methods of site prediction overcome many of these limitations. In its current state, however, GIS is technically complex and requires its users to have advanced technical training. Few archaeologists possess the needed skills

and must delegate the task of creating the site prediction model to a GIS technician. The GIS technician has the ability to develop a composite site prediction, but often lacks the archaeological expertise to develop a precise and accurate model. Therefore, the goal of this research is to bridge the technological gap between archaeologists and GIS through the creation of a user-friendly archaeological site prediction system.

This research supports three objectives:

1. Develop a site prediction system with a user-friendly interface that will allow archaeologists, with minimal GIS computer expertise, to analyze their data and create accurate and precise GIS site prediction models.
2. Apply the site prediction system to a set of case study data provided by the Athienou Archaeological Project.
3. Evaluate the site prediction system in terms of its technical performance and benefits to archaeological research.

It is hoped that the development of this archaeological site prediction system will allow archaeologists to use their expert knowledge and the capabilities of GIS technology for successful archaeological investigation. This process may ultimately save time and money in future surveys and excavations. Additionally, planners would have a system for making educated decisions that will protect potential cultural resources from human interference.

Site Prediction Modeling

As defined by Warren (1990, p. 201), "Predictive models are tools for projecting known patterns or relationships into unknown times or places." Archaeological research

has established that locations of early human settlements follow patterns with respect to the physical and social environment (Willey, 1953; Binford, 1982; Shermer and Tiffany, 1985; Binford, 1989; Ebert, 1992; Lock and Harris, 1992). “All cultural types - ranging from simple hunter-gatherers and early farmers to advanced agriculturists and urban peoples - respond to environmental conditions in siting their activity areas or settlements” and most cultural types also respond to social infrastructure such as, “roads, central places, and markets” (Brandt et.al., 1992, p. 269). Therefore, known patterns can be employed to predict the locations of currently unidentified archaeological sites.

There are two types of modeling used in archaeological site prediction: deductive and inductive. Deductive modeling is based upon the investigator’s preexistent assumptions pertaining to the resources necessary for site location. These resources may be ranked or weighted in accordance with the investigator’s theories. Inductive modeling is based upon empirical observation and depends heavily on statistical analysis.

Theoretically, these two types of modeling are distinct, however, in practice, archaeologists often utilize both methods (van Leusen, 1995; Warren, 1990a).

Both deductive and inductive models are grounded in settlement theory. It is assumed that people obey certain laws and principles when creating settlements (Doxiadis, 1970; Wood, 1978). These principles include the supposition that human activity is limited to a certain range of locational and environmental situations. Humans seek to minimize their distance from cultural and environmental resources, minimize the effort required to attain these resources, and optimize protective space (Doxiadis, 1970; Johnson, 1977; Wood, 1978; Hunt, 1992; Marble, 1996). The extent of human

settlement, therefore, can be defined as “the distance man wants to go or can go in the course of his daily life” (Doxiadis, 1970, p. 395).

Settlement theory guides archaeologists in their selection of factors that influence site location. Therefore, Hassan et.al. (n.d.) argues that site prediction models must include the following six factors: avoidance of environmental hazards, proximity to life-sustaining resources, transportation costs and accessibility, security/defense, social factors, and aesthetics (c.f., Kohler and Parker, 1986). Due to the unique aspects of every archaeological investigation, the emphasis placed on each of these factors will vary. For example, historical research may focus on social and cultural variables such as access to roads or religious centers. Conversely, prehistoric research may focus on environmental variables such as elevation, slope, and distance to water, since less is known about the socio-cultural aspects of this time period. Archaeological site prediction literature cites a variety of plausible site location factors (Shermer and Tiffany, 1985; Kohler and Parker, 1986; Charnichael, 1990; Hasenstab and Resnick, 1990; Wood, 1990b; Bradnt, 1992).

Site prediction models have proven effective for reducing the labor and financial costs of archaeological investigation (Kohler and Parker, 1986; Brandt et.al., 1992). In a research setting, models assist in the identification of potential sites, thereby reducing the need for extensive archaeological survey (Brandt et.al., 1992). In a cultural resources management setting, prediction models are used to locate regions of archaeological potential. Measures can then be taken to protect possible cultural resource sites from modern construction projects and looters (Kohler and Parker, 1986; Brandt et.al., 1992; van Leusen, 1995).

Traditional Site Prediction Methods

Traditional study of human settlement patterns was an overwhelming task that required manual tabulation, organization, and mental processing of massive amounts of archaeological spatial data (Kohler and Parker, 1986; Binford, 1989). As a result, most archaeological research was based upon the expert's experience and intuitions rather than formal site prediction models (Lock and Harris, 1992). An early example of settlement pattern research comes from a study by Willey (1953) in the Virú Valley, Peru (c.f., Thomas, 1989). The project was an extensive undertaking and produced large quantities of data. Willey (1953) attempted to synthesize this information on a series of unwieldy maps with different symbols signifying different site types (Thomas, 1989). While this method of investigation utilized the archaeologist's expertise, it was cumbersome.

Contemporary Site Prediction Methods

The advent of the computer and, more specifically, GIS, has alleviated the difficulty of spatial data management and provided the analytical, statistical and visualization tools necessary for creating complex site prediction models (Marble, 1990; Savage, 1990; Warren, 1990; Lock and Harris, 1991; Williams, 1997). Presupposing high quality data, "GIS technology makes use of the inherent precision of computers, and allows for the accurate positioning of sites and artifacts in space" (Miller, 1995, p. 320). Despite its benefits, archaeologists are reluctant to adopt new technology (Baker, 1988; Doran, 1988). Consequently, the majority of the contemporary prediction models are actually developed by a GIS technician rather than archaeologists (e.g., Dann and Yerkes, 1994; Jacobson, 1994). This dependence on GIS technicians can significantly affect the

validity of prediction model results. As Harris and Lock (1995, p. 355) remind, “GIS is not an objective, value neutral, unbiased technology.” The results are dependent upon the decisions of the user and, “although the mathematics involved in the modeling may be impressive, some of its ingredients will likely be based on the personal opinions of the researcher involved” (Luesen, 1996, p.181). If the user is a GIS technician, then the site prediction model no longer incorporates the archaeologist’s knowledge and intuition, and may, therefore, suffer inaccuracies.

Relevant Trends in GIS

As GIS technology continues to mature, its associated hardware, software, data products, and users continue to experience new innovations. While the field of GIS is currently expanding in many directions, two prevailing trends lend themselves to the creation of an expert-guided, archaeological site prediction system. These trends are object-oriented programming and increased user-friendliness.

Object-oriented programming is the process of assigning code and/or data to an object, such as a button, pull-down menu, or icon. When this object is activated, the assigned code and/or data is called, and a specific function is performed (Pinson and Wiener, 1990; Worboys, 1994; Razavi, 1995). This method of programming is very flexible and relatively simple to understand (Lee and Chin, 1995; Berry, 1996b).

Consequently, object-oriented programming has been implemented by several graphics and GIS packages (Wilson, 1996). This allows the GIS interface and system functions to be customized to meet the user’s needs and skill level. An object-oriented programming

language produced by Environmental Systems Research Institute, Inc. (ESRI), called Avenue, allows for the customization of ArcView 3.0.

In August, 1995, the magazine, *Geo Info Systems*, published a showcase edition that evaluated current trends in GIS software, hardware, data, and services. This edition included statements by the nine members of the journal's Editorial Advisory Board. Under the heading of *Software Trends*, Bruce Joffe (Bossler et. al., 1995, p.12), founding principal of GIS Consultants, remarked that, "functionality is becoming less important than ease of use. This is because the software has developed to such a high level of capability. Fewer benchmark tests need to confirm whether 'the software can do it,' but are now focusing on 'how does the user do it with the software.'" This sentiment has been voiced by others (e.g., Barlaz, 1995; Kvamme, 1995; Obermeyer, 1995). GIS vendors are beginning to reflect this trend toward user-friendly software through the adoption of windows-based packages with iconic buttons and pull-down menus, mouse-driven interaction with the graphical components of the system, and simplified commands that require fewer arguments. The result of this increased ease of use is that "GIS will lose its unique identity as a separate technology or discipline" and will become integrated into all fields that maintain a spatial component, including Archaeology (Dangermond et.al., 1996, p. 42).

While user-friendly systems have the potential to disseminate GIS to the non-expert, they can pose specific limitations on functionality. As Obermeyer (1995, p. 78) warns, "what organizations gain in easier-to-use, general purpose geographic information systems, comes at the cost of having models that are not specifically tailored to meet their unique needs." Therefore, this thesis intends to employ both current trends in the

development of a GIS archaeological site prediction system. It will apply object-oriented programming in the creation of a system that provides the necessary tools to the archaeologist in a user-friendly environment.

The remainder of this thesis chronicles the development, application, and evaluation of the Archaeological Site Prediction System. Chapter Two describes the case study data provided by the Athienou Archaeological Project. Details of the data characteristics, their manipulation, and the roles they served in this research are discussed. Chapter Three outlines the design elements used to create the Archaeological Site Prediction System, the employment of the case study data, and the evaluation format of informal interviews with professional archaeologists and geographers. Chapter Four discusses the results of these processes and finally, Chapter Five summarizes and concludes this research.

CHAPTER TWO

STUDY AREA AND DATA ACQUISITION

The study area of the Athienou Archaeological Project (AAP) is located in the Malloura Valley of central Cyprus (Figures 1 and 2). The project is directed by Dr. Michael K. Toumazou of Davidson College and his colleagues Dr. Richard W. Yerkes of Ohio State University and Dr. P. Nick Kardulias of Kenyon College. AAP's archaeological investigations consist of both systematic survey and excavation. Spatial information derived from these investigations will be used as case study data to test and evaluate the Archaeological Site Prediction System.

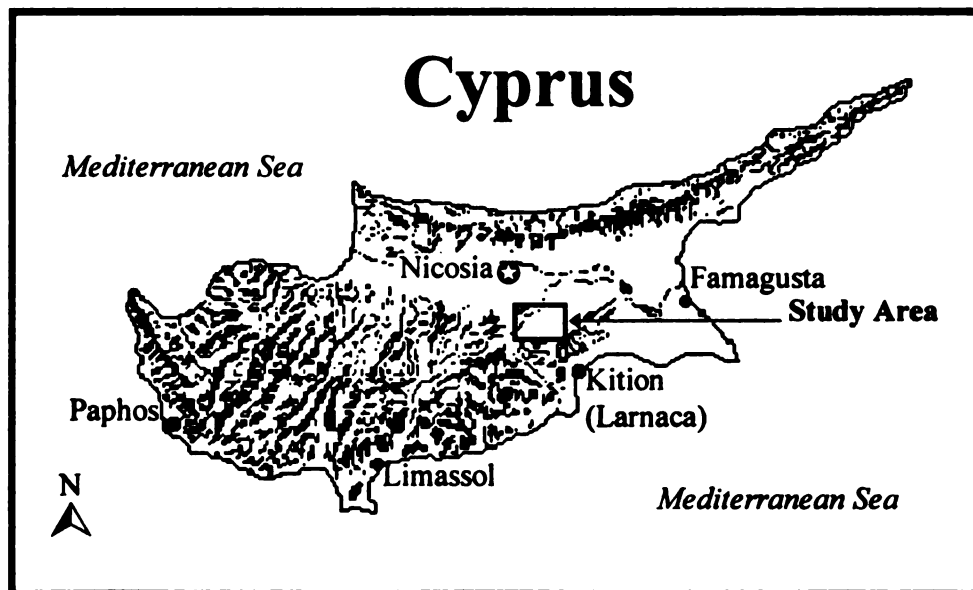


Figure 1. Reference map of Cyprus and study area (Toumazou et.al., 1995).

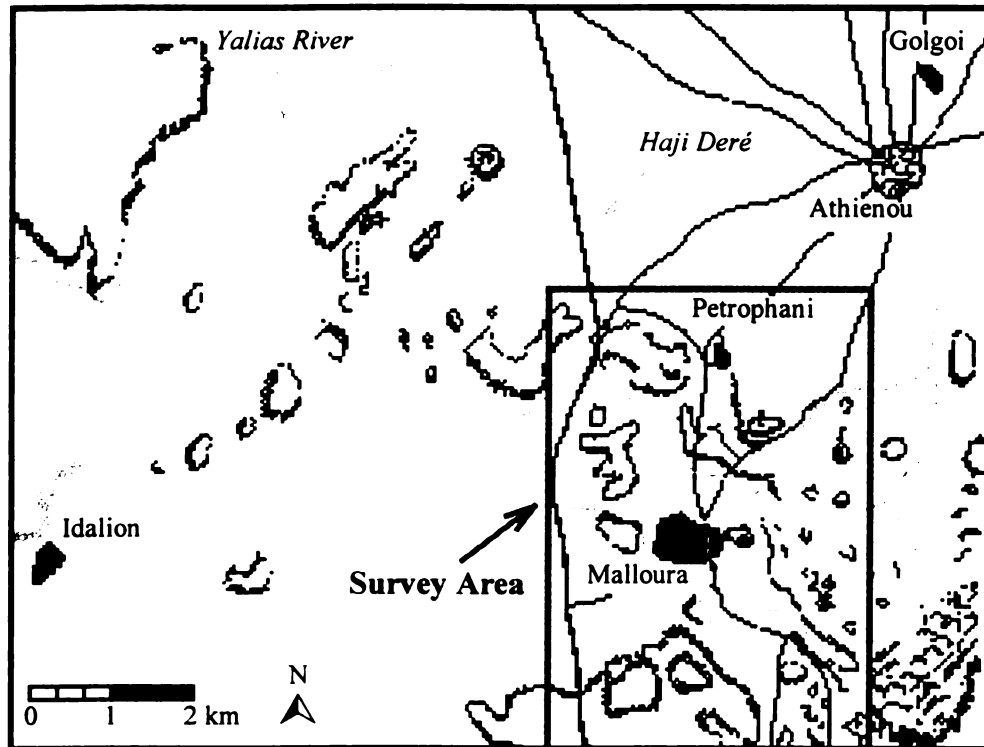


Figure 2. Malloura Valley and survey area (Toumazou et.al., 1995).

Regional and Geologic Setting

The Malloura Valley lies at approximately $35^{\circ} 01' N$ latitude and $33^{\circ} 30' E$ longitude, which places it about halfway between the capitol city of Nicosia and the port city of Larnaca. This region of Cyprus consists primarily of limestone capped hills, and valleys filled with alluvium and colluvium overlying marls and chinks. The Malloura Valley lies in the driest part of the island and receives approximately 38 cm of rainfall during the Winter wet season (Christodoulou, 1959; Toumazou et.al., 1995). This rainfall produces several intermittent streams that flow north and west to join the Haji Deré River or south and east to drain into the Larnaca Bay. The Haji Deré River flows north and east

to join the Yalias River, which eventually drains into the Famagusta Bay (Toumazou et.al., 1995).

Excavations and Surveys to Date

Since 1990, the AAP has been dedicated to the examination of “long term cultural change in the Malloura Valley” (Toumazou, et.al. 1995, p. 2). The entire AAP study area is approximately 20 km² in size; however, because of nearby military encampments, only 13.5 km² have been systematically surveyed. Both survey and excavation endeavors have been productive. To date, thirty archaeological sites have been located within the study area. These archaeological site designations were based upon the existence of standing structures or high artifact densities. Several time periods and settlement types are represented. The surveys have located “prehistoric lithic workshops, tomb groups dating from the Cypro-Archaic through the Roman periods, small Classical/Hellenistic habitation sites and look-out posts utilized from the Late Roman through Ottoman times” (Toumazou et.al., 1995, p. 1). In addition, excavation has exposed a sanctuary that dates to the Cypro-Archaic through the Roman periods, two buildings of Late Roman/Early Byzantine age, Venetian period structures and several tombs (Herscher, 1995; Toumazou et.al., 1995). Refer to Table 1 for a complete listing of the AAP sites and their ages.

Table 1. Athienou Archaeological Project sites.

Site #	Site Name	Site Type	Cultural Period	Approximate Dates
1	Malloura	sanctuary/structures	Archaic-Venetian	750BC-1489AD
2	Magara Tepesi	tomb group	Archaic-Roman	750BC-330AD
3	Diamond Field	mudbrick structure	Recent	post 1878AD
4	Sklinikos North	stone structure	Recent	post 1878AD
5	Sklinikos	artifact scatter	Unknown	
6	Jinarotos East	kiln/furnace	Roman-Byzantine	30BC-1191AD
7	Khirin Kourats	tomb/storage pit	Classical-Roman	475BC-330AD
8	Kokkinokafallia	cut blocks/dump	Recent	post 1878AD
9	Chakmaklik I	chert workshop	Early Prehistoric	3900-5500BC
10	Chakmaklik II	chipping station	Recent	post 1878AD
11	Chakmaklik III	small chert scatter	Prehistoric	pre 1050BC
12	Petrophani	village	Prehistoric, Recent	pre 1050BC, post 1878AD
13	Petrophani South	mudbrick structure	Recent	post 1878AD
14	Mandra tou Photi	artifact scatter	Unknown	
15	Mavri East	chert workshop	Unknown	
16	Mavrous Northeast	settlement	Roman-Byzantine	30BC-1191AD
17	Khania East	settlement	Recent	post 1878AD
18	Koutsopetrika East	stone workshop	Recent	post 1878AD
19	Mandra tou Photi South	stone hut	Recent	post 1878AD
20	Acropolis Southeast	millstone and cistern	Byzantine	330-1191AD
21	Chakmaklik IV	chert scatter	Unknown	
22	Asprovounos	artifact scatter	Roman, Recent	30BC-330AD, post 1887AD
23	Dhkyovounon	artifact scatter	Roman, Recent	30BC-330AD, post 1887AD
24	Mesovounos	artifact scatter	Prehistoric, Roman, Recent	pre 1050BC, 30BC-330AD, post 1878AD
25	Platia Laxia	artifact scatter	Roman, Recent	30BC-330AD, post 1887AD
26	Kryon Neron West	artifact scatter	Roman, Recent	30BC-330AD, post 1887AD
27	Kryon Neron North	settlement	Archaic, Roman	750-475BC, 30BC-330AD
28	Chakmaklik V	small chert scatter	Prehistoric	pre 1050BC
29	Zeitini Tarlasi	artifact scatter	Roman-Byzantine, Turkish	30BC-1191AD, 1571-1878AD
30	Trakhonas	settlement	Classical-Byzantine	475BC-1191AD

Athienou Archaeological Project and the Archaeological Site Prediction System

The Archaeological Site Prediction System's case study data were extracted from the Athienou Archaeological Project for four reasons. First, the AAP offers a diversity of archaeological settings. The existence of survey and excavation data, multiple site types, and a wide range of site ages provide an excellent foundation for a complete examination of the Archaeological Site Prediction System. Secondly, the AAP data are appropriate in terms of quality and format. The cultural and environmental information is in hard copy,

large scale maps that can be easily transferred to digital format. Thirdly, the data are readily accessible. The project supervisors are interested in expanding their research area through the use of a GIS prediction model and have generously offered the use of their spatial data for this case study. Fourthly, the utilization of AAP data has the potential to expand the capacity of site prediction modeling. Most GIS site prediction modeling has been performed in the United States, where site types, ages, and densities are vastly different from their European counterparts (Kohler and Parker, 1986; Brandt et.al., 1992). Therefore, a system that utilizes Cypriot data may provide new insights into European archaeology and GIS site prediction modeling.

The Athienou Archaeological Project directors have two goals for the Archaeological Site Prediction System. First, they expect to use this system to locate the Archaic town that supported the previously discovered Archaic sanctuary and burial tombs (Referred to as sites one and two in Table 1). Secondly, the project directors intend to use the Archaeological Site Prediction System and their digital data to facilitate GIS communications with the Cypriot government and other archaeological projects (Toumazou, 1996). Within the next five years, the Cypriot government plans to complete its "Cyprus GIS" project. The database resulting from this project will contain country-wide political, environmental, and cultural data at a scale of 1:2,000 and will allow archaeologists and government officials to join efforts in the expansion of archaeological awareness and site preservation (Hadjiraftis, 1996). In addition to their governmental relations, the AAP directors anticipate that GIS technology will allow archaeologists from various projects to share their knowledge of cultural finds and patterns. At this time, the Sydney Cyprus Survey Project located in Mitsero, Cyprus and headed by Dr.

Bernard Knapp, is the only other Cypriot archaeological project using a GIS (Knapp, 1995; Knapp, 1996). It is hoped, however, that future Cypriot archaeological investigations will consider this technology for their predictions, excavations, and surveys.

Description of Data

The Archaeological Site Prediction System will use eight layers of information: a digital elevation model (DEM) created from six topographic maps, slope angle and slope aspect derived from the DEM, bedrock geology, groundwater, calcareous soils, streams, and the location of Athienou Archaeological Project's sites. With the exception of the derived coverages, all data were provided in hard copy format. For a complete listing of available maps, see Table 2. For a listing of the map source information, see Appendix A.

Table 2. Map resources for the Archaeological Site Prediction System.

Map Title	Map Scale	Date of Publication
Cyprus Topographic Series	1:5000	1977
Hydrologic Map of Cyprus	1:250,000	1970
Calcareous Soils of Cyprus	1:250,000	1987
Geological Map of Cyprus	1:250,000	1995

The geology layer represents the Malloura Valley's several chalk hills and limestone capped mesas; the soils layer, as a reflection of the geologic setting, consists of varying degrees of soils; and the hydrology layer exhibits regions of underlying confined

and unconfined groundwater aquifers. Due to the Malloura Valley's dry climate, all streams are intermittent. The sites layer contains the thirty AAP archaeological sites as points in UTM coordinates.

Manipulation of Data

Each of the hard copy maps was converted to digital form through digitizing into Arc/Info, a GIS package produced by Environmental Systems Research Institute, Inc. (ESRI). The hydrology, soils, and geological maps were digitized as polygon coverages. The intermittent streams were digitized as arc coverages from the six topographic maps and appended together to create the streams layer.

In order to maintain consistency between all coverages and permit modeling calculations in the Archaeological Site Prediction System, the data were converted from vector to raster format. The raster conversion for the groundwater, geology, soils, streams, and sites coverages was accomplished through Arc/Info's *polygrid*, *linegrid*, and *pointgrid* commands. A 10 m grid resolution was used, thereby generating a grid composed of 646 rows and 726 columns. The selection for this cell size was based on Tobler's rules for spatial data resolution and relevant archaeological literature.

According to Tobler (1988), resolution of mapped data in meters is equivalent to $(\frac{1}{2} * (\text{the denominator of the map scale} / 1,000))$, assuming that the smallest physical mark that can be made on a map is 0.5 mm. Based on this equation, the elevation, slope angle, slope aspect, and streams coverages have a resolution of 2.5 m, while the hydrology, soils, and groundwater maps have a resolution of 125 m.

In addition to the spatial data considerations, the archaeological data must also be evaluated when selecting a grid cell size. A site prediction model can be deemed ineffective if the pixel size is too small, thereby overburdening the computer system, or if the pixel size is too large for the data, thereby leading to misinterpretations and errors of omission (Ebert et. al, 1996). Kvamme (1990) recommends using a cell size that approximates the average site size to avoid these potential problems. Due to the nature of the AAP data, “average site size” is not easily calculated. Six of the AAP’s thirty sites are actually large areas composed of several smaller sites. For example, Malloura (referred to as Site 1 in Table 1) is a conglomerate of multiple structures spread throughout an area of 260,000 m². If the AAP’s six large sites are omitted from the calculations, the average size of the remaining twenty-four sites is about 136 m². Based upon these factors, a cell size of 10 m was the most reasonable for this data set. Although this resolution may be overly fine for the three small-scale coverages (soils, groundwater, and geology) it facilitates the identification of smaller archaeological sites.

The conversion of topographic contours to a digital elevation model was a fairly complex process. Research performed by Eklundh and Mårtensson (1995) demonstrated that DEM’s could be interpolated from digitized point samples, rather than the traditional time and labor intensive method of contour digitizing, without a loss of accuracy. Therefore, a composite sampling strategy was adopted for the Malloura Valley data. In accordance with Makorovic’s (1977) composite sampling guidelines, elevation points were selected based on both a progressive grid and “very important points.” A 100 meter grid was superimposed on the topographic maps. This interval was chosen for its logical divisions on the topographic map sheets and its consistency with the Eklundh and

Mårtensson (1995) research. Their research utilized a 2 km sampling interval on 1:50,000 scale maps, which creates a point sample to map scale ratio of 1:25. The topographic maps used for the AAP elevation data were sampled at a 100 m interval on 1:5000 scale maps, which produced a point sample to map scale ratio of 1:50. Therefore, the AAP data were sampled at an interval that was twice as fine as the Eklundh and Mårtensson data and were thereby expected to meet or exceed their results. Elevation points were digitized on the contour line nearest each grid intersection. If the elevation change between two adjacent sample points (based upon four directions or a rook's case) exceeded 10 meters (equivalent to a 5.7° slope), additional points were entered midway between the two original samples. This procedure was performed in order to increase sampling density where slopes were moderate to steep. In addition to this progressive technique, all benchmarks, peaks and pits (local maxima and minima, respectively) were selectively entered to define significant features on the landscape. This composite method yielded approximately 1,000 sample points for each topographic map sheet.

Once the point coverages were digitized, they were checked for accuracy. First, a histogram of the sample points was checked for significant outliers or blatant data entry errors. Secondly, the points were converted into a triangulated irregular network (TIN) which was used to produce a contour map at a 4 m interval. This contour map was then visually compared with the original topographic map sheets to determine whether significant data had been omitted. Observed errors were then corrected and the individual elevation coverages were appended into one coverage.

The point elevation data were interpolated to form a 10 m resolution DEM through the use of Hutchinson's (1989) algorithm, implemented as Arc/Info's *topogrid*

function. This algorithm assumes that sinks (local minima) that impede drainage are rare in nature and can be eliminated conservatively when constructing the DEM. The procedure uses the sample point data and the streams coverage to produce a grid of elevation that recognizes fundamentals of hydrology and enforces them over the sample data. The tolerances required by the *topogrid* function were set according to the suggestions of the software vendor and include the root mean square (RMS) residual of the data, the accuracy and density of the sample points (tol1), and the limit placed on the amount of modification allowed when removing a sink (tol2). The three recommended values were: an RMS of zero for point data, a tol1 of $\frac{1}{2}$ the contour interval and a tol2 equal to the contour interval (Environmental Systems Research Institute, Inc, 1992). Since the tol1 and tol2 point data arguments were not specified by ESRI, the *topogrid* tolerances used to create the AAP elevation grid were based upon the six original topographic maps and set to 2 m and 4 m, respectively.

Upon completion of the DEM, the slope angle and slope aspect grids were created. These were derived from the DEM using the Arc/Info *slope* and *aspect* commands, respectively.

The extent of each of the six, original topographic maps was approximately 7.7 km². When joined together, the grids cover a contiguous area of approximately 46 km². This combined grid was used as the bounding polygon for the study area. The geology, groundwater, and soils grids were each clipped to match this boundary. The final resultant data were eight 46 km² grids at 10 m resolution that have been projected into the UTM coordinate system.

CHAPTER THREE

METHODS FOR CREATING, APPLYING, AND EVALUATING

THE ARCHAEOLOGICAL SITE PREDICTION SYSTEM

The Archaeological Site Prediction System was developed in the Spatial Analyst Extension of ArcView 3.0. This GIS software package, produced by Environmental Systems Research Institute, Inc., was selected for three reasons. First, ArcView maintains an object-oriented programming language that allows for customization. This language, named Avenue, provides the programmer with the tools to alter system commands, create user-friendly dialog boxes, and develop new functions. Secondly, ArcView meets many needs for the archaeologist. Since archaeology departments are generally not well funded, the cost of purchasing a Unix workstation and learning a new operating system and corresponding software is often beyond the financial and temporal means of the archaeologist (Kvamme, 1995). ArcView can be run on PC machines, is relatively inexpensive, and retains many of the Windows conventions, reducing the time required to learn the software. Thirdly, ArcView communicates well with its powerful cohort, Arc/Info. Therefore, specific functions that cannot be employed by ArcView, can be accessed using Arc/Info.

Design Elements for the Archaeological Site Prediction System

The primary objective of this research was to develop an archaeological site prediction system with a user-friendly interface that would allow archaeologists, with minimal computer expertise, to analyze their data and produce GIS site prediction models. Therefore, the creation of this system required that several design decisions be considered prior to implementation. ArcView already possesses a user-friendly structure that includes pull-down menus and iconic buttons. The Archaeological Site Prediction System has incorporated this structure through a pull-down menu that assists the archaeologist with the development and analysis of site prediction models. This pull-down menu allows the user to enter their data into the project, manipulate these data, and create and evaluate site prediction models.

In his book Human-Computer Interface Design Guidelines, Brown (1988) lists several recommendations for successful interactive computer software. These recommendations include (a) using logical terminology and graphics, (b) reducing the need for memorization by offering prompts before tasks are performed, and (c) employing “clickable” menu bars and lists for data entry. Each of these suggestions has been applied to the Archaeological Site Prediction System. First, the layout of the commands in the pull-down menu is designed to replicate the order in which the commands should be followed (Figure 3). In addition to the logical structure, terminology that is consistent with ArcView conventions and other Windows-based software, has been maintained. Secondly, in accordance with Brown’s recommendation for prompts, several “message boxes” have been placed throughout the Archaeological Site Prediction System (Figure 4). When successive steps were deemed complicated or

not intuitive, message boxes that supplied reminders, helpful hints, and prompts that direct the next course of action, were provided for the user. Thirdly, because methods of data entry that rely heavily on alpha-numeric entries from the keyboard are prone to incorrect punctuation, misspellings, improper word spacing, and other typographical errors, list boxes that dictate available filenames, field variables, or command parameters have been employed (Figure 5). When list boxes are inappropriate for the data required, input boxes with prompts that assist the user with the entry have been utilized (Figure 6).

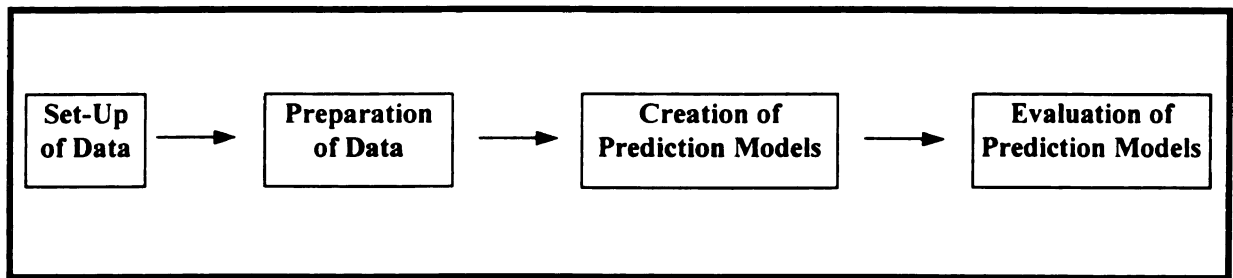


Figure 3. Structure of the Archaeological Site Prediction System.

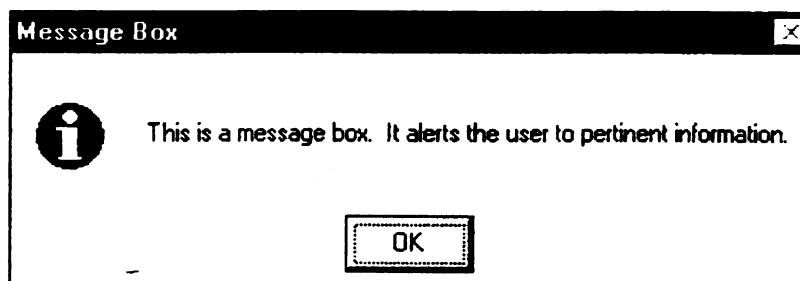


Figure 4. Sample message box.

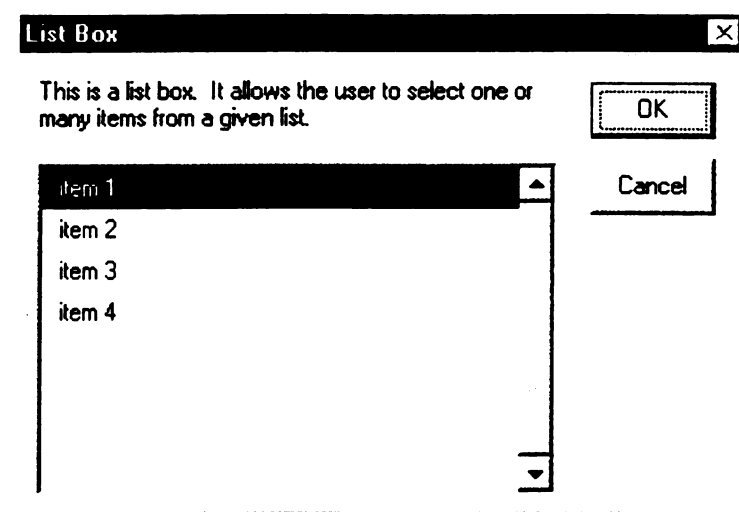


Figure 5. Sample list box.

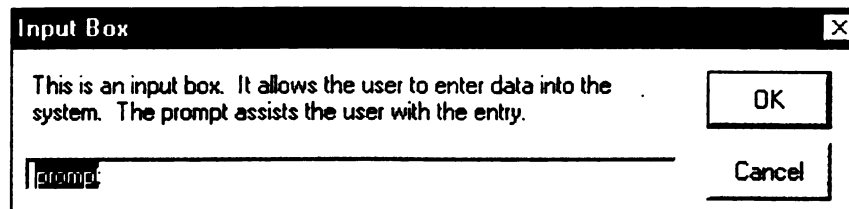


Figure 6. Sample input box.

As defined by Ravden and Johnson (1989, p. 15), a successful user-interface is one that “enables the end-user to access and make use of the facilities and functions which the system provides, and to carry out the tasks for which it has been designed. It provides the user with information about the system, about what it does, and about what the user can and should do. It enables the user to learn about the system and to build an understanding of how it works.” It is hoped that the design elements recommended by Brown (1988) and the implementation of ArcView will facilitate each of these desired

functions and promote successful interaction between the archaeologist and the Archaeological Site Prediction System.

Customization of ArcView 3.0

The Archaeological Site Prediction System was developed as a single pull-down menu in the View screen of ArcView. This menu is comprised of six components and their sub-commands (Figure 7). All programming, with the exception of one Arc Macro Language (AML) program in Arc/Info, utilized Avenue. The global variables, Avenue scripts, and AML that were produced in the customization process can be viewed in Appendix B.

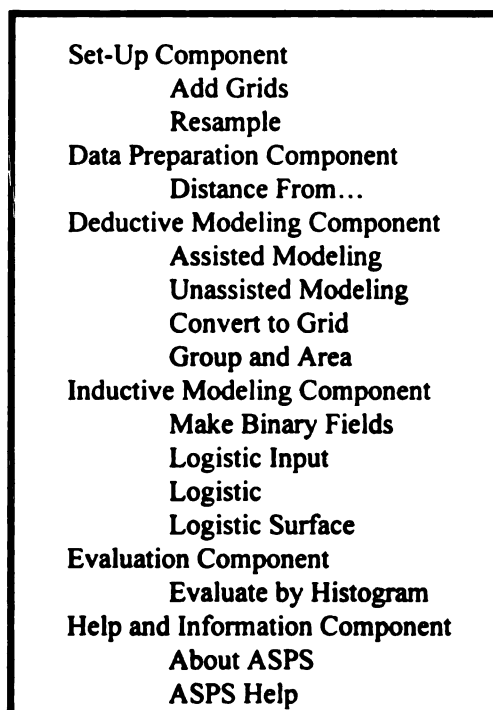


Figure 7. Components and sub-commands of the Archaeological Site Prediction System (ASPS).

As the name implies, the set-up component is designed to assist the user with the initial aspects of establishing the data within the Archaeological Site Prediction System. A series of input boxes have been created to aid the user with the input of the directory path and eight basic grids: elevation, slope angle, slope aspect, streams, groundwater, geology, soils, and sites. These eight data layers were selected for their correspondence with the AAP data, and their prevalence in archaeological site prediction literature (e.g., Schermer and Tiffany, 1985; Brandt et.al., 1992; Charmichael, 1990; Warren, 1990b). In addition to these eight basic grids, supplementary data are accommodated through a set of general prompts. In addition to entering the data, the set-up component provides the user with the option of resampling. This function was designed to encourage archaeologists to consider the issue of resolution and provide a means for altering the pixel size of their grids.

It has been demonstrated that the locations of sites are directly related to distance from pertinent cultural and natural resources, such as streams, roads, or other sites (e.g., Kohler and Parker, 1986). Therefore, the data preparation component has been developed, permitting the archaeologist to create distance surfaces from any feature grid (e.g., streams, roads, etc.).

The Archaeological Site Prediction System offers two components for creating prediction models. The first component consists of a deductive approach that utilizes the archaeologist's *a priori* knowledge to develop a set of environmental and cultural factors that influence site location. These factors are employed to form a predictive query expression that is used in the creation of a binary deductive modeling grid. This grid illustrates areas that are likely to have archaeological potential, based upon the parameters

entered into the query expression. Once the deductive model has been created, contiguous groups of grid cells can be classified into individual groups whose areas are then calculated.

The second component for creating site prediction models consists of an inductive approach, using logistic regression. Logistic regression analysis employs binary, interval, and ratio independent data to create a probability surface for the dependent variable, which is archaeological sites in this case (Warren, 1990a). By examining the values of the independent variables at locations known to be sites and at locations known not to be sites, logistic regression calculates the probability for sites to exist at currently unsampled locations. The importance of this method has been demonstrated in both geographic literature (Pereira and Itami, 1991; Brown, 1994; Shields et.al., 1996) and archaeological site prediction literature (Kohler and Parker, 1986; Carmichael, 1990; Warren, 1990b; van Leusen, 1996). The Archaeological Site Prediction System allows the user to create binary grids, make a logistic input file, run logistic regression (provided the user has access to Arc/Info), and build a logistic surface.

The evaluation component provides a visual method for evaluating the deductive and inductive site prediction models. The deductive model can be evaluated using the process recommended by Brandt et.al. (1992). In this method, the number of known sites that fall within the query area and the number of random sites that fall within the query area are compared using a histogram display. The inductive model can also be evaluated through the use of a histogram. For this type of model, a simple bar graph of how many pixels fall within each category, representing a probability level, is displayed to further illustrate the aspects of the surface.

The help component of the Archaeological Site Prediction System was created with the Microsoft Windows '95 Help authoring Kit. This software allows the programmer to develop help files that replicate the Microsoft conventions. Help files were created to accompany each component of the Archaeological Site Prediction System and loaded into ArcView 3.0. A brief information page that provides name and contact information for the author of the Archaeological Site Prediction System is also included.

Application of the Archaeological Site Prediction System to the Case Study Data

Case study data from the AAP were used throughout the customization process to test each command for continuous loops, inappropriate prompts, and other programming oversights. These data served a valuable role in error-tracking, however, they were not yet applied to create or analyze site prediction models. Therefore, in order to fully evaluate the durability and functionality of the Archaeological Site Prediction System, an exploratory data analysis (EDA) investigation was conducted. EDA facilitates the examination of the data and software by using the results of each step in the prediction process as a guide to the next step (Breckenridge, 1983). Since this method capitalizes on unrestrained exploration of the data, the user has the potential to discover new insights and unforeseen patterns (Williams et.al, 1990; Fotheringham, 1992).

The goal of the Archaeological Site Prediction System was to allow archaeologists to explore their own data. Therefore, site prediction models for field use will be designed by the AAP archaeologists. The objective of the EDA performed with this system was to test procedures for generating site prediction models. Two concerns guided this EDA investigation: can viable site prediction models be created using the

Archaeological Site Prediction System and how does resampling the data to a coarser resolution affect the results of a deductive prediction model? The EDA process followed the general outline displayed in Figure 3. First, the AAP data were added to the project. Secondly, these data were resampled to 125 m, to facilitate comparison between resultant models at a 10 m pixel size and at a size that adheres to Tobler's rules for resolution of spatial data. Thirdly, feature data were manipulated to determine distances from the streams. Fourthly, the data were used to create deductive site prediction models at 10 m and 125 m resolutions and inductive site prediction models at 10 m resolution. Finally, the resultant models were evaluated.

A stronger correlation between site locations and attributes of the environment is generally observed when the sites are of the same age and/or cultural period (van Leusan, 1996). Therefore, a subset of the AAP data was utilized for EDA. The data selected were from the Roman occupation of Cyprus (30 BC - 330 AD). These data were chosen because they represent the most numerous site type in the study area and because the Roman occupation has been well documented in archaeological and historical literature. According to archival records, nucleated settlements under private ownership dominated the Cypriot landscape during the highly populated Roman occupation (Christodoulou, 1959; Karouzis, 1977). Both historical and archaeological investigations have unearthed the presence of "look-out posts" or points from which approaching enemies could be observed. (Christodoulou, 1959; Toumazou et.al, 1995). Additionally, it has been noted that several archaeological sites dating to this time period have been located on high plateaus or steep slopes (Symeonoglou, 1972; Hadjisavvas, 1977; Toumazou et.al., 1995). These factors have lead archaeologists and historians to believe that settlement

location during the Roman Empire was based upon defense and proximity to fresh water, a scarce resource in Cyprus (Christodoulou, 1959; Adovasio, 1975).

Procedure for Evaluating the Archaeological Site Prediction System

The Archaeological Site Prediction System was evaluated by nine professional archaeologists who represented the three primary divisions of archaeology: Public Archaeology (state government), Cultural Resources Management (private contractors), and academia (see Appendix C). Informal interviews were conducted with small groups composed of one to four archaeologists. It was hypothesized that prior experience with computers and/or knowledge of GIS would have a significant impact on the types of responses given. Therefore, each archaeologist was asked to complete a short survey pertaining to their skill levels (see Appendix D). These surveys were then used to assign computer and GIS proficiency ratings to each archaeologist. Each rating system was scaled from one to five, with one representing no experience and five representing expertise in that area (see Table 3). Once the surveys were completed, the archaeologists were informed that an audio recording would be made of the interview. Next, the professionals were asked to view a demonstration of the Archaeological Site Prediction System. Discussion and questions were encouraged throughout this presentation. Upon completion of the demonstration, the archaeologists were permitted time to experiment with the Archaeological Site Prediction System.

Table 3. Ratings for computer and GIS skills.

Rating	Computer Skills	GIS Skills
1	Uncomfortable around computers, has very little or no experience with this technology.	Unfamiliar with the term GIS, has never considered using GIS for archaeological endeavors.
2	Has limited computer experience, most comfortable with office software and/or e-mail.	Familiar with some concepts of GIS, has no experience with GIS software.
3	Is comfortable around computers, able to multi-task and maneuver through many software packages and more than one operating system.	Understands GIS concepts, has limited experience with GIS software and no formal training.
4	Maintains considerable hardware and software knowledge, able to manipulate many operating systems, may know a programming language.	Able to apply many GIS concepts, has had formal GIS training through a university or place of employment.
5	Computer expert.	GIS expert.

The intention of this evaluation process was to elicit feedback on four issues that pertain to the Archaeological Site Prediction System: the positive aspects of the system, the negative aspects/limitations of the system, suggestions for how the system could be improved, and how this system and similar user-friendly GIS interfaces will affect the field of archaeology. If these four issues were not addressed through informal discussions during the demonstration process, a standardized questionnaire was administered at the completion of the demonstration (see Appendix D). All comments offered by the archaeologists were recorded and categorized into one of these four issue groups. These comments were later paraphrased (all direct quotes are placed within quotation marks), transcribed into tabular format, and analyzed with respect to the archaeologist's computer and GIS skill levels. If the same comment was made by two or

more archaeologists with similar computer and GIS skills, it was tabulated just once. Upon completion of the interviews, the responses were examined and the results summarized.

In addition to the archaeologist assessments, five geographers with GIS experience were asked to evaluate the Archaeological Site Prediction System (see Appendix C). These interviews were conducted and recorded using the same method described above. However, the geographers were not required to complete the computer and GIS skill level survey. The objectives for the geographer interviews resemble those discussed above, with the exception that issue number four, which pertains to the impact the Archaeological Site Prediction System will have on the field of archaeology, was not addressed. See Appendix D for the standardized questionnaire used for the geographer evaluations.

The archaeologist and geographer survey results were qualitatively analyzed and used to draw conclusions about questions related to the four issue areas. The following chapter will display and discuss the tabulations for each of the four issue groups: the positive aspects, the negative aspects/limitations, suggested improvements, and potential applications to the field of archaeology.

CHAPTER FOUR

RESULTS AND DISCUSSION

This research sustained three objectives: the development of a user-friendly site prediction system, the application of this system to a set of case study data, and the evaluation of this system through professional interviews. Each of these objectives has been met through the creation and testing of the Archaeological Site Prediction System.

The Archaeological Site Prediction System and Results from the Exploratory Data Analysis

The Archaeological Site Prediction System (ASPS) was developed as a single pull-down menu that was added to ArcView's View Screen in the format of a menu bar and five tool buttons that were added to the button bar (Figure 8). The pull-down menu gives the user access to the sub-commands associated with the six components of the system. The five tool buttons are activated when particular sub-commands are running, and allow the user to graphically interact with the view.

Exploratory data analysis (EDA) was performed on the Archaeological Site Prediction System to examine its functionality and durability. Through this process, several viable prediction models for locating Roman period sites were developed. These models are not presented as optimum for the AAP data, rather they are used to demonstrate procedures for generating archaeological site prediction models.

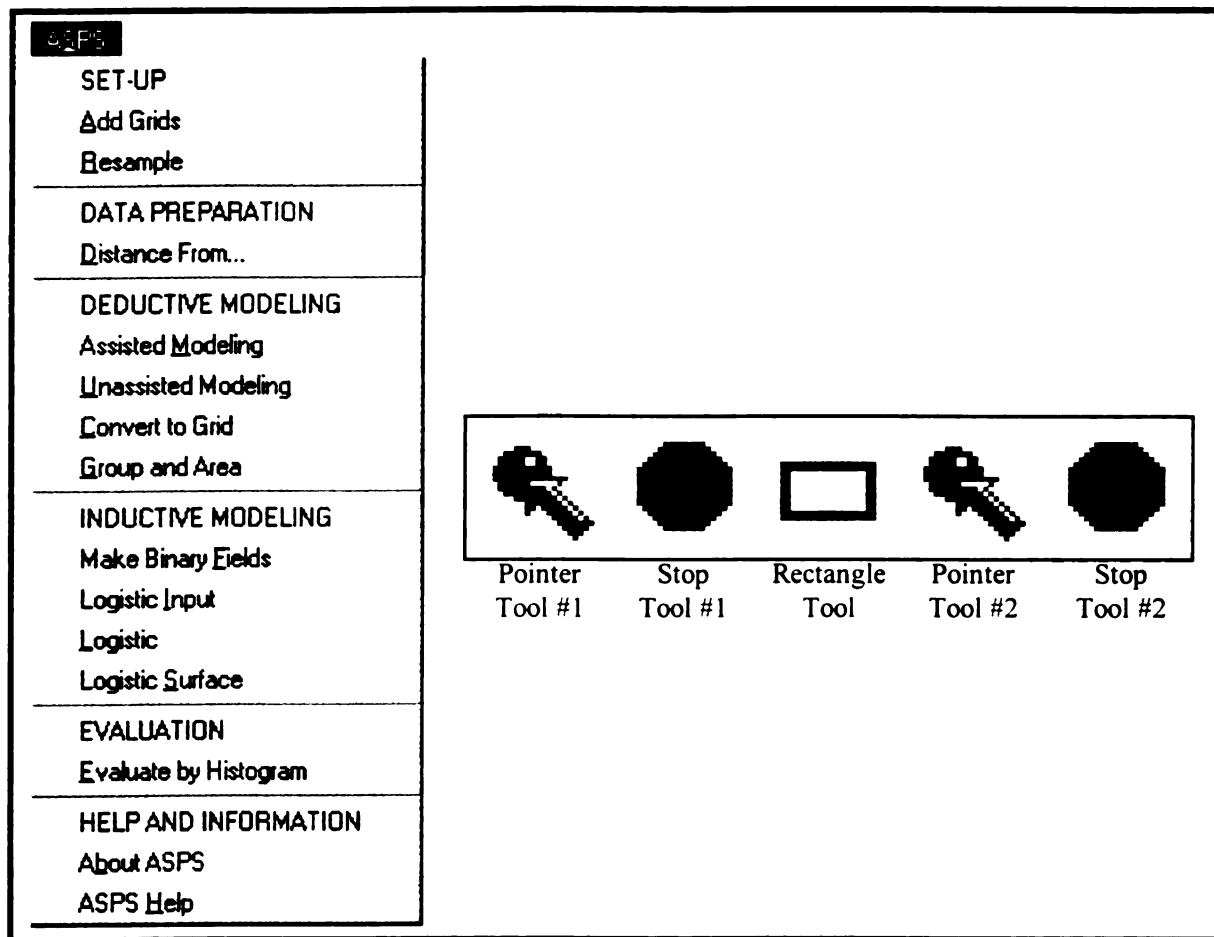


Figure 8. Pull-down menu and buttons added to the View Screen.

The following section provides a detailed illustration of the Archaeological Site Prediction System and application to the study area. The structural design and functional capabilities of each component and its sub-commands are discussed. In addition to this discussion, screen captures, taken during the EDA process, have been included to illustrate aspects of the system. Pertinent results and observations drawn from the EDA procedure are simultaneously addressed.

a
th
an
cit

Set-Up Component

The set-up component is comprised of two sub-commands: *add grids*, and *resample*. Together, these commands assist the user with the initial aspects of site prediction modeling. The *add grids* function permits the user to enter data into the ArcView project and View. These data must be in Arc/Info grid format and all located in the same directory. A series of input boxes have been programmed to obtain directory information, grid names, and the fields that should be used as guidelines for legend creation. Grid names for eight specific themes are requested: elevation, slope angle, slope aspect, streams, groundwater, geology, soils, and sites. If the user enters a valid grid name for one of these eight grids, a legend is created and a global variable is assigned for tracking purposes. Legends are tailored to the theme and are assigned either unique hues for categorical data, or a single hue with a gradation of values for numerical data. Once the eight grid names are entered, the user is permitted to enter the names of additional data layers through a set of generalized input boxes. Since the details of these supplementary data are unknown, the user must specify whether the legend is categorical or numerical in nature. There is no limit to the number of supplementary grids that can be entered through this generalized format. A sample View, containing the AAP elevation, streams, and sites grids is displayed in Figure 9.

Resample, the second sub-command in the set-up component, allows the user to alter the resolution of one, many, or all of their grids. A list box permits the user to select the grid(s) to be resampled, a message box alerts the user to the current resolution(s), and an input box prompts for the new cell size. The user may resample the data through either a nearest neighbor, bilinear interpolation, or cubic convolution interpolation

method. If one of the eight specific grids is processed using *resample*, then the global variable associated with that grid is assigned to the resampled grid. As with most of the grids created in the Archaeological Site Prediction System, the resampled grids are automatically converted to permanent grids, and are placed in the directory path previously specified in the *add grids* command.

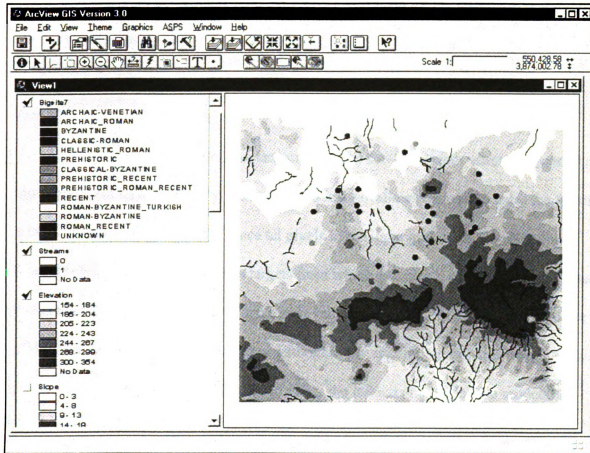


Figure 9. Sample view, following the *Add Grids* command.

In accordance with the EDA objectives, the elevation, slope angle, streams, and groundwater grids were resampled to a resolution of 125 m using the cubic convolution

and nearest neighbor interpolation methods. In the cubic convolution interpolation method, resampled cells are assigned a value equal to the weighted distance average of their nearest sixteen neighbors, thereby creating a smooth surface. Since smooth surfaces better represent the landscape, cubic convolution interpolation was used for the continuous elevation and slope angle data. In the nearest neighbor interpolation method, resampled cells are assigned a value equal to their nearest data point. This method is most appropriate for categorical data, therefore, it was used for the streams and groundwater grids.

Data Preparation Component

The data preparation component contains one sub-command: *distance from*. An input box, with its default set to the streams coverage, allows the user to enter the feature grid to calculate distance from. Since all pixels in the input grid, with the exception of those labeled “no data,” are used as targets from which distance is calculated, provisions were made to permit the user to assign a zero value as “no data.” The resultant distance grid is converted to a permanent grid and stored in the appropriate data directory. At this point, the distance grid is ready for use in either the inductive or deductive modeling components.

Figure 10 illustrates a distance grid, in meters, created from the 10 m resolution AAP streams coverage. The streams have been placed over the grid for enhanced visualization. The *distance from...* function was also applied to the 125 m resolution streams grid. These distance grids were employed in the deductive and inductive modeling processes.

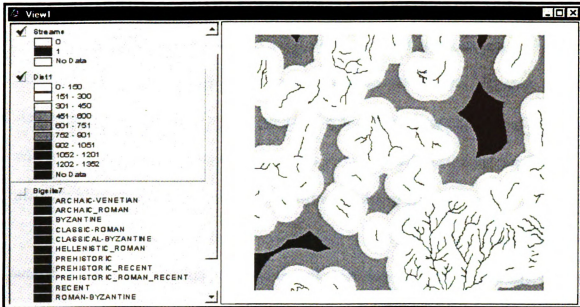


Figure 10. Sample view, following the *Distance From...* command.

Deductive Modeling Component

The deductive modeling component provides the tools necessary to create a site prediction model that is derived from a query of the data. The user may choose the desired data parameters and create a query expression with either the *unassisted* or *assisted modeling* sub-commands. The *assisted modeling* function employs list and input boxes to prompt the user through the development of each segment in a query expression. Elements of this process include the selection of a grid to be queried, assessment of whether this grid is categorical or numerical in nature, and the definition of suitable values to be used in the analysis. In addition to the generalized prompts, the *assisted modeling* command employs the global variables associated with the eight basic grids and offers customized prompts when appropriate. There are no restrictions on the

number of segments allowed in a query, therefore the user may repeat this process to create an expression of unlimited length. Unfortunately, ESRI has not released the programming code to allow this query expression to be run directly (Kopp, 1997).

Therefore, a message box displays the query expression and instructs the user to type this statement into the *unassisted model* builder. It is hoped that future releases of ArcView will provide access to the necessary code, thus allowing the deductive site prediction model to be produced entirely within the *assisted modeling* command.

All queries are ultimately run in the “query builder dialog box,” located in the *unassisted modeling* function therefore, the advanced user may use this function directly to develop the deductive model without the intervention of information boxes and prompts. Since the model is constructed through the query builder dialog box, an unalterable system script, the resultant binary grid cannot be automatically transformed to a permanent grid. Therefore, the user must run the *convert to grid* command.

In the EDA process, several deductive site prediction models were developed for both the 10 m and 125 m AAP data. Figure 11 displays a sample deductive site prediction model that was created from the analysis of four 10 m grids. The areas shown in gray are deemed to have archaeological site potential, based upon these parameters: slope > 4°, elevation > 200 m, distance to streams < 750 m, and groundwater that was neither discontinuous nor poor in quality. Figure 12 displays a sample deductive site prediction model created with the 125 m grids and the same parameters as Figure 11. Since the EDA process used a Roman period subset of the AAP sites data, the factors selected for these deductive prediction models were based upon the defensive and fresh water needs of that time period. It should be noted that Figures 11 and 12 display models

based on just one possible set of parameters. By altering the grids used, modifying the values in the query expression, or selecting a range of values (e.g., $4^{\circ} < \text{slope} < 15^{\circ}$), the possible combinations for deductive site prediction models are unlimited.

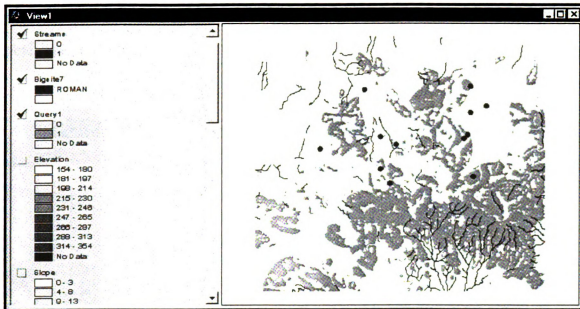


Figure 11. Sample view of a deductive site prediction model (10 m).

Permanent deductive modeling grids may be used to run the *group and area* command. This function identifies contiguous groups of grid cells in the query. The user is asked whether cells that touch on the diagonal should be used in this identification process, thereby defining these contiguous groups as rook's case (based on four directions) or king's case (based on eight directions). These contiguous groups are classified into individual regions, they are assigned a unique identifier, and their areas are calculated. The group and area grids are converted to permanent, integer grids to facilitate future exploration and queries of this data. However, ArcView 3.0 integer grids

are limited to values of 1,000,000 or less. Therefore, the numerical values in the integer area grid are divided by 100 in order to avoid software errors. The user is alerted to this limitation in a message box and is reminded to scale the area values accordingly when utilizing them for further calculations. Additionally, the temporary floating point area grid is not deleted from the legend, thereby providing the user with a reference for the actual areas.

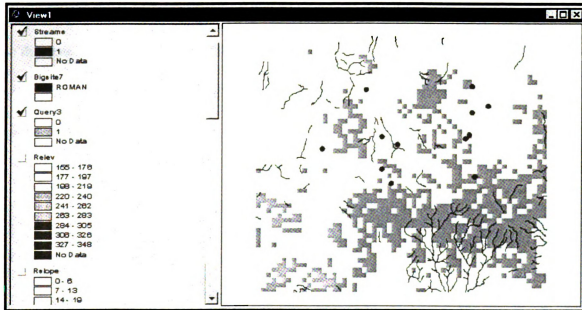


Figure 12. Sample view of a deductive site prediction model (125 m).

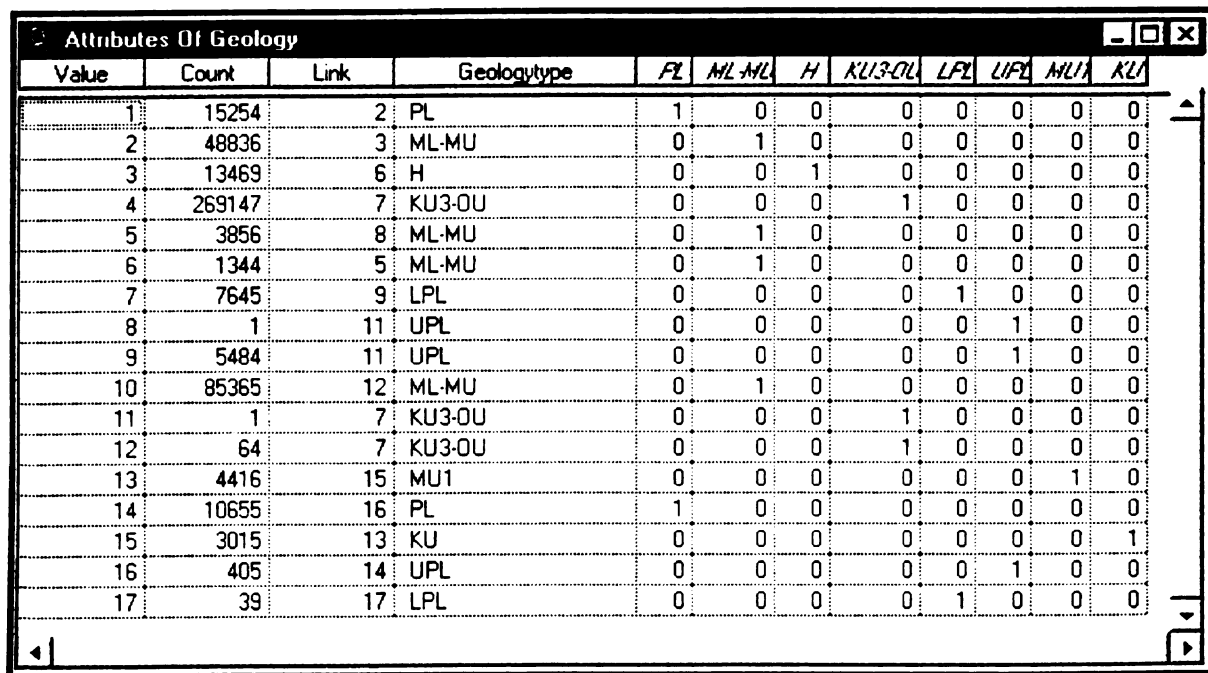
During the EDA process, the *group and area* command was run on the deductive site prediction models displayed in Figures 11 and 12. The model at a 10 m resolution produced 845 unique groups of cells and a total area of approximately 14.0 km² for locations predicted to have archaeological potential. The model at a 125 m resolution

produced 132 unique groups of cells and a total area of approximately 12.6 km² for locations predicted to have archaeological potential.

Inductive Modeling Component

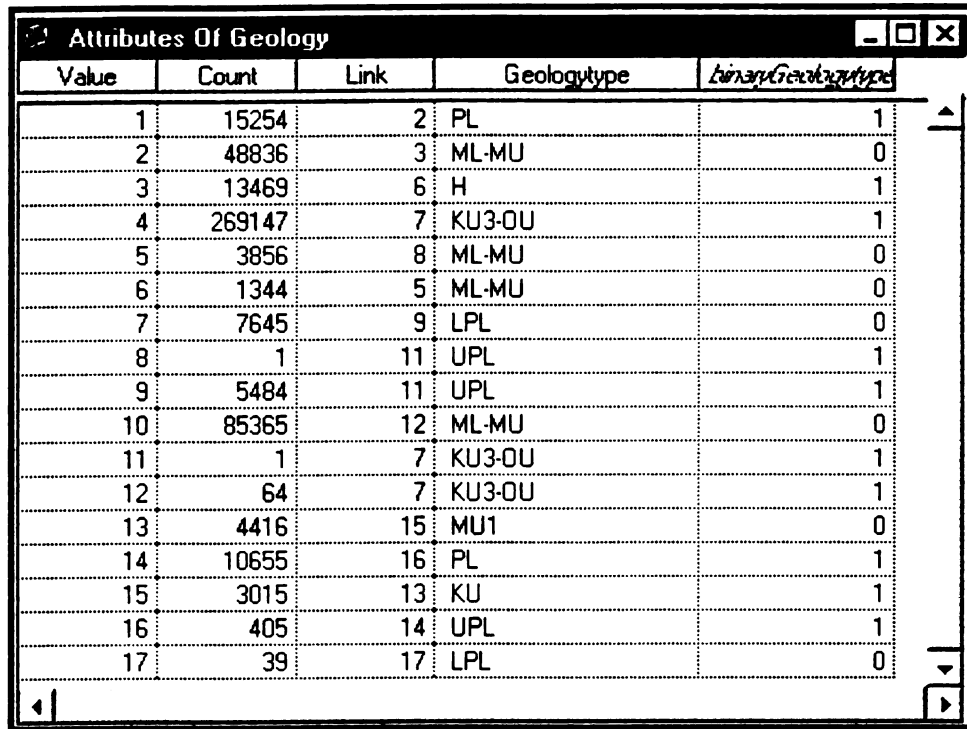
The inductive modeling component is comprised of four sub-commands: *make binary grids*, *logistic input*, *logistic*, and *logistic surface*. In order to be used in a logistic regression equation, categorical data, such as geology or soil types, must be converted to dummy variables. The *make binary grids* command assists the user with this task.

Binary fields for independent categorical data can be created by adding (0/1) information to the specified grid's attribute table. A binary field may be created for each category (Figure 13) or, a single field may be created for a combined analysis of all the categories (Figure 14).



Value	Count	Link	Geologytype	FL	ML-ML	H	KU3-OU	LPL	LIP	MU1	KU1
1	15254	2	PL	1	0	0	0	0	0	0	0
2	48836	3	ML-MU	0	1	0	0	0	0	0	0
3	13469	6	H	0	0	1	0	0	0	0	0
4	269147	7	KU3-OU	0	0	0	1	0	0	0	0
5	3856	8	ML-MU	0	1	0	0	0	0	0	0
6	1344	5	ML-MU	0	1	0	0	0	0	0	0
7	7645	9	LPL	0	0	0	0	1	0	0	0
8	1	11	UPL	0	0	0	0	0	1	0	0
9	5484	11	UPL	0	0	0	0	0	1	0	0
10	85365	12	ML-MU	0	1	0	0	0	0	0	0
11	1	7	KU3-OU	0	0	0	1	0	0	0	0
12	64	7	KU3-OU	0	0	0	1	0	0	0	0
13	4416	15	MU1	0	0	0	0	0	0	1	0
14	10655	16	PL	1	0	0	0	0	0	0	0
15	3015	13	KU	0	0	0	0	0	0	0	1
16	405	14	UPL	0	0	0	0	0	1	0	0
17	39	17	LPL	0	0	0	0	1	0	0	0

Figure 13. Geology attribute table with one binary field per geology type.



Value	Count	Link	Geologytype	binaryGeologytype
1	15254	2	PL	1
2	48836	3	ML-MU	0
3	13469	6	H	1
4	269147	7	KU3-OU	1
5	3856	8	ML-MU	0
6	1344	5	ML-MU	0
7	7645	9	LPL	0
8	1	11	UPL	1
9	5484	11	UPL	1
10	85365	12	ML-MU	0
11	1	7	KU3-OU	1
12	64	7	KU3-OU	1
13	4416	15	MU1	0
14	10655	16	PL	1
15	3015	13	KU	1
16	405	14	UPL	1
17	39	17	LPL	0

Figure 14. Geology attribute table with one binary field for combining several geology types.

Once the categorical data have been converted to binary format, the second command, *logistic input*, may be employed to create a text file needed to run the logistic function. This text file contains the dependent variable (site/non-site), its (x,y) coordinates, and the values of the independent variables at that location (Figure 15). The independent variables can be any ordinal or binary grid data that the user has selected from a customized list box. These independent variables are assigned global variables for future tracking and prompting purposes. The locations of known sites and their coordinates may be entered graphically, tabularly, from a grid, or from a point coverage. If the user selects the graphical method, the pointer tool #1 and stop tool #1 buttons (Figure 8) are activated and the user is prompted to enter site locations with the mouse. If

the tabular method is chosen, the user is prompted for the Northing and Easting coordinates of each site location. Finally, if either the grid or point coverage methods are elected, the user is prompted to enter the grid/coverage name and select whether all sites or a subset of the sites should be used for the logistic input.

Input.txt

1	545104	3876187	431	197	1
1	546197	3877682	306	180	1
1	546584	3876464	196	202	1
1	547000	3876270	600	209	3
1	546612	3875661	304	230	3
1	546834	3875301	363	242	9
0	547876	3874975	202	259	7
0	544953	3874628	598	236	4
0	545742	3874305	152	246	2
0	545900	3877085	252	210	1
0	545330	3874083	602	232	8
0	549818	3874606	180	321	7

Independent Variables

Northing (y-coordinate)

Easting (x-coordinate)

Site (1)/Non-site (0)

Figure 15. Sample input text file.

Once the known sites and their appropriate independent variable values have been entered into the text file, the rectangle tool button is activated and the user is prompted to draw the boundary for the surveyed area. Next, ArcView selects a random non-site location from within this boundary using its random number generator, and asks the user whether this location should be added to the text file. Until ESRI releases the programming code to conduct a query, thereby allowing a buffer to be placed around

known sites, this method ensures that random site locations are not placed upon known sites. ArcView repeats this process until the number of random sites equals the number of known sites.

Based on the defensive and fresh water needs of the Roman period, the distance from water, elevation, and slope angle grids were used as the logistic input for the EDA process. These data were entered through the tabular input method.

At this time, ArcView 3.0 cannot perform logistic regression analysis. Therefore, the third sub-command, *logistic*, is merely a message box alerting the user to this limitation and providing instructions for running logistic regression in Arc/Info. It is hoped that future releases of ArcView will be equipped with a more powerful statistics package and that logistic regression will be functional within this software. The user must type the command *&run logistic* at the Arc: prompt to run a short AML that has been designed to ask the user for the text file created in the logistic input command and to perform the logistic regression procedure in Arc/Info's Grid package.

The logistic regression procedure creates a report that is divided into three sections (Figure 16). The first section contains the coefficients used in the logistic regression equation:

$$(1 / (1 + e^{-(a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots a_nx_n)}))$$

“where a_0 , a_1 , a_2 , etc. are the coefficients from the regression and x_1 , x_2 , x_3 , etc. are the corresponding independent grids from which these coefficients were derived” (ESRI, 1992). Using the example table from Figure 16 and the independent grids used in the EDA logistic input, the logistic equation would be:

$$1/(1 + e^{-(3.204 + (0.003 * \text{distance to water grid}) + (-0.019 * \text{elevation grid}) + (-0.013 * \text{slope angle grid}))}).$$

The second section displays the z value, or actual dependent variable value, and the z error, or the difference between the actual and predicted value, for each of the known site and random location input points. Figure 16 provides z errors for eight of the points used in the EDA input file. The user can determine the calculated probability that a point in the list is actually a site by subtracting the z error value from one. For example, the z error value for point id #1 in Figure 16 indicates that there is a 68.5% chance of classifying point #1 (a known site) as an archaeological site. Therefore, the higher the z error for known sites, the less likely a point will be classified as an archaeological site. Section three provides summary statistics for the logistic regression input points. A lower root mean squared (RMS) error, suggests that the predicted values more closely resemble actual site/non-site findings. The chi-square value provides a test of significance through the use of a chi-square table. For binomial logistic regression, with two possible outcomes, the degrees of freedom is one. In Figure 16, the chi-square value of 4.771 is higher than 3.841, the critical chi-square value at a 0.05 significance level and one degree of freedom. This suggests that the model displayed in Figure 16 provides a statistically significant level of explanation of site/non-site patterns. It should be noted that other models, with different sets of independent variables, may also produce significant levels of explanation. Unfortunately, the Arc/Info logistic procedure does not provide a means of evaluating the contribution of each variable to the model. Therefore, each model should be evaluated using the procedure described below to determine its usefulness to archaeological site prediction modeling.

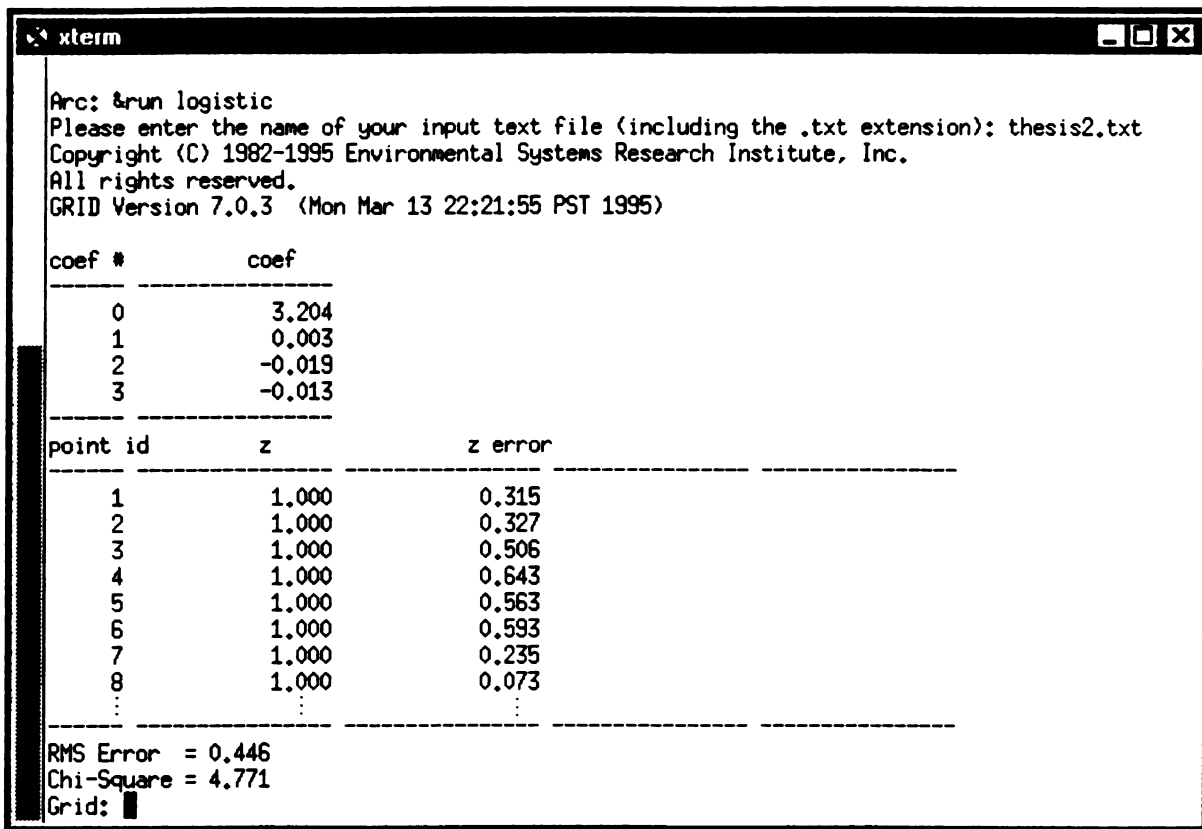


Figure 16. Portion from an Arc/Info logistic regression report.

ArcView and Arc/Info cannot directly communicate. Therefore, in order to create a logistic surface (i.e., probability surface), the user must record the coefficients from the logistic regression report in Arc/Info (Figure 16) and enter them into an input box that is initiated by the *logistic surface* command in the Archaeological Site Prediction System. This command prompts the user for the coefficients associated with each of the independent grids, based upon the global variables established in the *logistic input*. ArcView will compute the logistic surface from these inputs and display it as a permanent grid in the View screen.

During the EDA process, a logistic surface was created from the model illustrated in Figure 16, the Roman period sites, and the 10 m elevation, slope, and distance from

streams grids (Figure 17). This surface illustrates probability for archaeological sites. According to the model in Figure 17, sites are more likely to occur in areas of darker hue values (refer to the legend for Surface2).

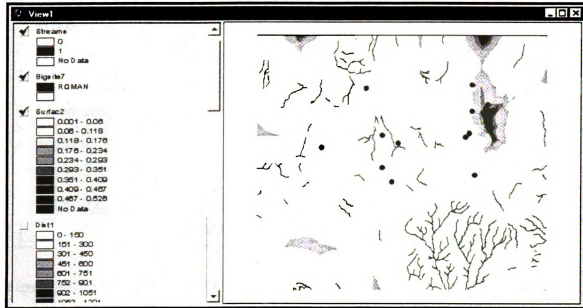


Figure 17. Sample view of an inductive site prediction model.

Evaluation Component

The evaluation component allows the user to visually evaluate their inductive and deductive site prediction models through the use of histograms. In the creation of a histogram for an inductive model, a simple bar graph of how many pixels fall within each category of predicted probability is displayed (Figure 18). In addition to the histogram, inductive models can be evaluated through visual examination. By looking at the logistic surface (e.g., Figure 17) and comparing the predicted locations of sites with the actual locations of sites, the model can be assessed for accuracy. A satisfactory inductive model

would be reflected as a histogram with few highly probable site locations and a surface that demonstrates that the archaeological sites fall within this area.

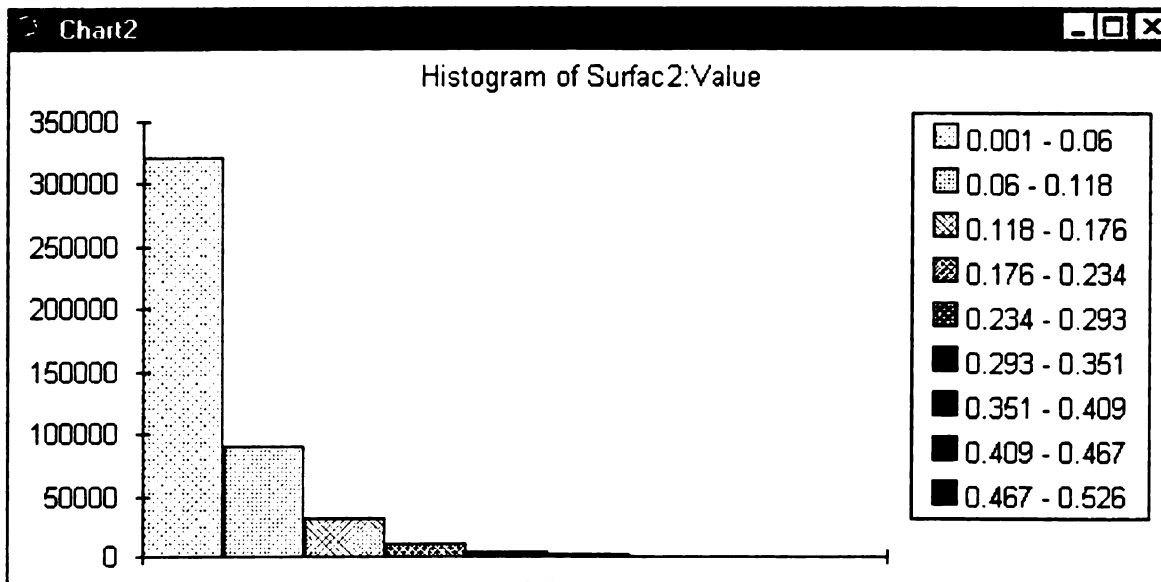


Figure 18. Sample histogram for an inductive model.

Figure 18 corresponds to the surface created in Figure 17. While this histogram meets the objective of “few highly probable site locations,” examination of Figure 17 reveals that many known Roman sites do not fall within these highly probable areas. These results could have numerous explanations. Four are addressed here. First, there may not be a relationship between Roman sites in the Malloura Valley and elevation, slope, and distance to streams. Perhaps other independent variables should be explored. Second, the dependent variable may need further exploration. The logistic surface in Figure 17 illustrates that some Roman sites were better predicted for than others. With the exception of Site 27, if the site type is considered (Table 1), artifact scatters (Sites 22-

26, 29) appear to be better predicted than settlements and tomb groups (Sites 1, 2, 6, 7, 27, 30). Perhaps site functionality should be addressed in logistic modeling, as well as site age and cultural affiliation. Third, the sample size may be too small to create a viable statistical model. Unfortunately, this is a dilemma that archaeologists will always face. The purpose of this modeling procedure is to locate new sites, if large sample sizes already existed, the need for prediction would be diminished. Fourth, while logistic regression has received high acclaim from the archaeological world, it possesses inherent flaws. For example, logistic regression assumes that the known sites are independent from one another. When three sites, in a close proximity, are identified as a “settlement,” “tomb group,” and a “kiln,” this assumption may be invalid, thereby skewing the results of the logistic regression analysis.

In the creation of a histogram for a deductive model, the user is prompted to enter the name of the query grid and the locations of the known sites. These locations can be entered graphically, tabularly, from a grid, or from a point coverage (see *logistic input* command for further explanation). ArcView selects a set of random locations using its random number generator, equal in number to the set of known sites, and determines whether these specified and random locations fall within the query area. The results are displayed in a chart format. A satisfactory deductive model would have a histogram in which many of the known sites fall within the query area and few of the random sites fall within the query area.

Figure 19 is the histogram created for the deductive model in Figure 11 and demonstrates that four of the twelve Roman sites were correctly predicted. Figure 20 is the histogram created from the deductive model in Figure 12 and indicates that three of

the Roman sites were correctly predicted. Although the histograms indicate that the models did not predict many sites, the deductive models used were precise enough to eliminate many random locations from the query areas. Further exploration of the data by altering the grids used and modifying the values in the query expressions, may yield models with improved relationships between site locations and query areas.

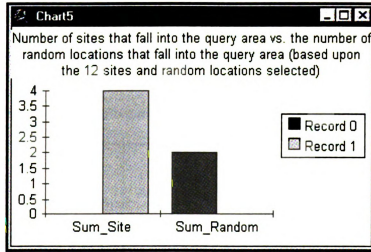


Figure 19. Sample histogram for a deductive model (10 m).

Table 4 examines the deductive models illustrated in Figures 11 and 12. It compares the total query area and the number of Roman sites that fall within that area at 10 m and 125 m resolutions. In a visual comparison, Figure 12 possesses less detail than Figure 11, however, alteration of the resolution has had little impact on the models' results. It should be noted that further data exploration and model creation may produce vastly different outcomes.

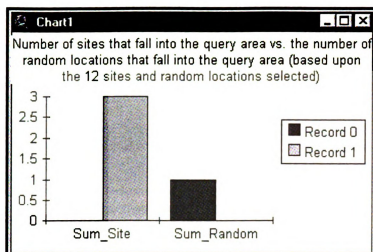


Figure 20. Sample histogram for a deductive model (125 m).

Table 4. Comparison of deductive models at 10 m and 125 m resolutions.

	Total Query Area in km ²	Number of Roman Sites Located in the Query Area
10 m Resolution Model	14.0	4
125 m Resolution Model	12.6	3

Help and Information Component

This component is comprised of two items that provide the user with help and information about the Archaeological Site Prediction System. The *about ASPS* function displays the name and contact information for the author of the system. The *ASPS help* function gives the user assistance with each of the Archaeological Site Prediction

System's components, commands, and button tools (Figure 21). See Appendix E for a complete listing of the help files.

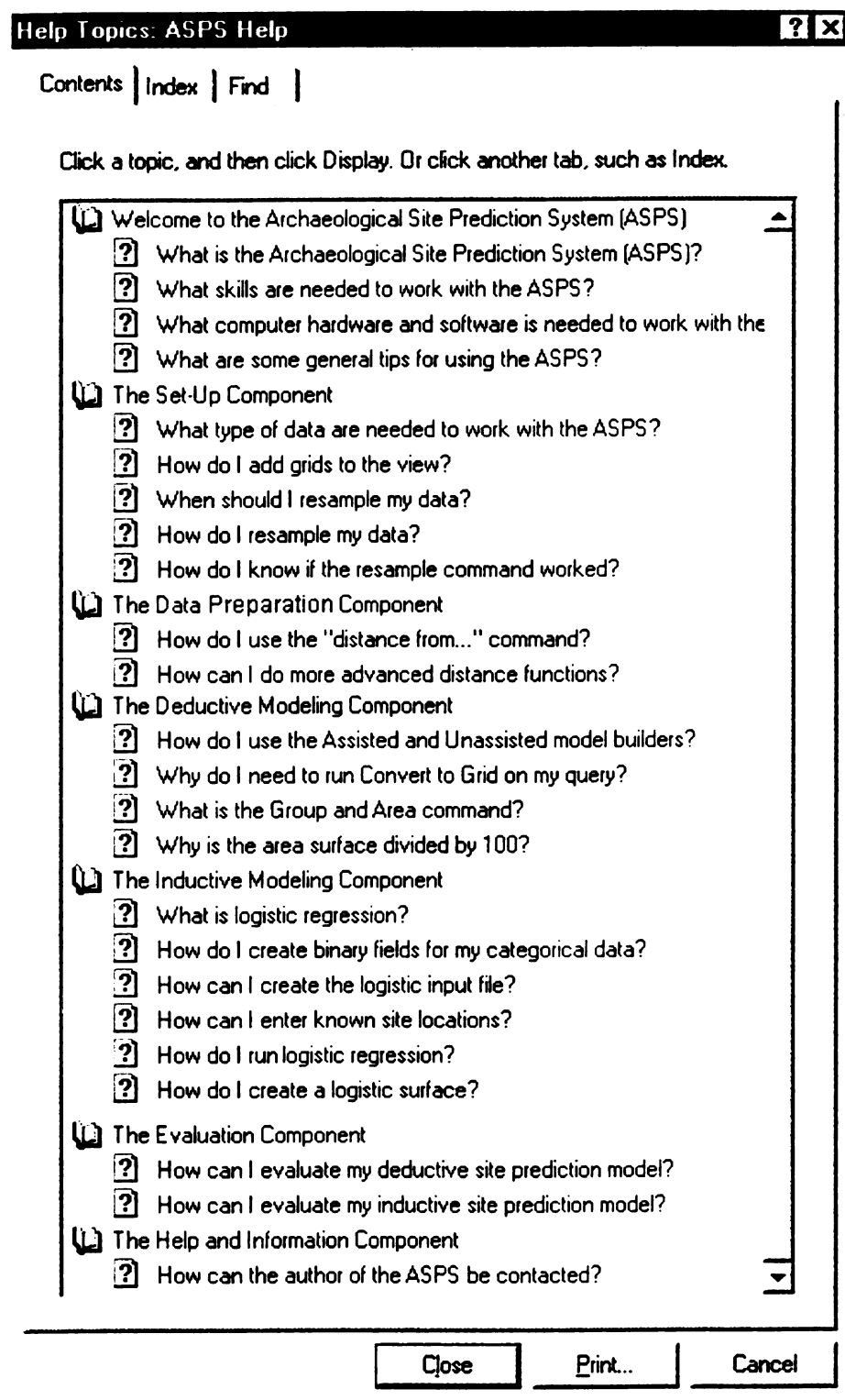


Figure 21. Help topics for the Archaeological Site Prediction System.

Analysis and Discussion of Professional Interviews

The professional interviews were intended to evaluate four aspects of the Archaeological Site Prediction System: positive features, negative features/limitations, ways in which the system could be improved, and perceived impact this system and similar customized GIS systems will have on the field of Archaeology. Although the responses elicited addressed each of these aspects, the proper categorization for the tabulation process was not always apparent. For example, a comment such as “I would use this system for my site predictions” has ramifications for both the positive aspects of the Archaeological Site Prediction System and the potential impact this system will have on the field of Archaeology. Therefore, categorization guidelines were developed as follows. If the archaeologist projected himself or herself as the user and discussed how he or she would employ this system, the comment was classified as a positive aspect. However, if the comment was not personalized and referred to archaeologists in general, it was classified as a perceived impact on the field of archaeology. Similarly, a comment such as “this system does not address the problem of getting the data into Arc/Info grid format, I would like to see a digitizing component added to the system” addresses both a limitation and a suggestion. In these instances, the comments were broken into two separate thoughts and placed into the limitations and suggestions categories, respectively.

Archaeologist Evaluations

The results of the archaeologist interviews have been tabulated by the four Archaeological Site Prediction System issues and by GIS and computer proficiency

ratings of commenting individuals (Tables 5-8). Summaries and discussion of the responses follow the tabular results.

Table 5. Positive aspects of the Archaeological Site Prediction System.

Comments - Positive Aspects	GIS Proficiency Rating	Computer Proficiency Rating
I have never considered using GIS for my archaeological endeavors, but this system looks interesting and useful to site prediction modeling.	1/2	2
A graphical representation of site data is useful.	1/2	2
I see analysis potential for this system - my historic farmsteads are easy to find (through oral histories/maps), but I could use this system to determine whether site patterns/site size is dependent on certain environmental/cultural variables.	1/2	2
I would like to use this system at a finer (artifact-level) resolution to examine intra-site dynamics.	1/2	2
I think the <i>unassisted model builder</i> is appropriate for beginning ArcView users, I like it better than the <i>assisted model builder</i> .	1/2	2
Based upon knowledge of other sites locations, this system would be good for locating sites in remote areas that have not had extensive archaeological study (e.g. the Upper Peninsula of Michigan).	1/2	2
Learning GIS by yourself is almost impossible, even with a manual, but a simple program like this alleviates much of the difficulty.	2	2
I would like to use this system to examine inter-site relationships - we often focus on our one little study area and do not examine how it fits into the landscape as a whole.	2	2
The value of being able to visualize and quickly manipulate data should not be underestimated - making these maps by hand would be awful, and is, therefore, rarely done.	2	2
It is helpful to have a system that is already compiled for us, where all we have to do is enter the data.	2	2
This system is very flexible - I like that I can enter any cultural or environmental variables that I want.	2	2
I would definitely consider using this system and other user-friendly GIS systems.	2	2
The visual display for evaluating data is quite useful.	2/3	3
I appreciate that the temporary grids are automatically converted to permanent grids.	2/3	3
I would use this system to evaluate sites on the different lake level coastlines by querying on elevation.	2/3	3
The eight grids prompted for in the <i>add grids</i> command address the primary variables that I would be interested in.	2/3	3

Table 5 (cont).	GIS Proficiency Rating	Computer Proficiency Rating
This system appears to be flexible in terms of data handling (e.g. stream order can be added to the grid's table)	2/3	3
I like that this system has both inductive and deductive modeling	2/3	3
While I do not use Arc/Info, I could certainly use the logistic input file in my statistics software to run logistic regression and other statistics.	2/3	3
This system is better than the database software we are currently using because it allows for graphical representation of the sites and provides a means for identifying site characteristics.	2/3	3/4
I am amazed by all of the prompts and help comments.	2/3	3/4
The terminology and GIS concepts are at an appropriate difficulty level.	3	3
The order of the pull-down menu is very helpful - it reduces memorization of command names and the order in which they should be followed.	3	3
I would like to use this system at a finer resolution to examine relationships that may exist within the ceramics data.	3	3
This system accommodates a wide range of user skills - the help component and prompts assist the novice user, without hindering the advanced user.	4	3/4
The option of using the <i>assisted</i> or <i>unassisted model builder</i> increases flexibility and allows for differing user skill levels.	4	3/4
I would consider using this system for my archaeological site prediction modeling - its standardized format would ensure that all models utilized the same methods.	4	3/4
This system makes GIS easier, even if you have GIS experience. I like that the pull-down menu reduces the need to memorize commands.	4	3/4
I like how the pull-down menu replicates the order in which the commands should be followed.	4	3/4
"Its a good thing."	4/5	3/4
I like the <i>group/area</i> function - it is helpful to know the size of particular regions.	4/5	3/4
I would like to run a prediction model based on site functionality - I could do this by simply adding functionality to the appropriate grid table.	4/5	3/4

Table 6. Negative aspects/limitations of the Archaeological Site Prediction System.

Comments - Negative Aspects/Limitations	GIS Proficiency Rating	Computer Proficiency Rating
Data entry is a problem. I would like to scan my data in, but this system has no provisions for such a task.	1	3
The GIS terminology is somewhat confusing (e.g. while I understand the concept of resolution, to me, <i>resample</i> means "going into the field and getting new information or a different sample," not anything relating to cell size)	1/2	2
Data entry is a problem. For small areas, I could make maps by hand with colored pencils and topographic quads faster than I could digitize/scan in the data.	1/2	2
While I would consider using this system, it is not completely intuitive, I would have to set aside some time to learn it.	1/2	2
This system does not allow for cluster analysis.	2	2
This system does not account for edge effect problems at the boundaries of the surveyed areas - this could affect the validity of the random locations selected during the logistic input process, thereby affecting the results of the logistic surface.	2/3	3
This system treats sites as point data, not areal data, which is not always correct.	2/3	3
In the <i>distance from...</i> command, if you convert zero data to "no data" you will lose the actual locations of "no data."	2/3	3
My biggest concern is data - we have Michigan's 18,000 sites in two other databases, I do not want to enter this data AGAIN.	2/3	3/4
This system assumes the data is correct and precise - a lot of Michigan's sites are not like this. (e.g. we have sites that were reported by hikers that are "somewhere in this 40 acre area").	2/3	3/4
My biggest problem is getting the data in - this system does not help me digitize, project, or add the tabular information to my data.	4	3/4
The logistic regression function is only in Arc/Info, which means that fewer archaeologists will have access to it and even if they do have access to the software, they may not have the courage to run Arc/Info, even with an AML.	4	3/4
This system does not have any built-in hydrologic modeling tools.	4	3/4
This system does not help the user with adding and editing data in the tables.	4/5	3/4

Table 7. Suggested improvements for the Archaeological Site Prediction System.

Comments - Suggested Improvements	GIS Proficiency Rating	Computer Proficiency Rating
I would like to see a manual or tutorial that explains this system's capabilities and how to use its components.	1	3
A tutorial with sample data and a definition of GIS terms would be helpful.	1/2	2
GIS terminology should be defined (e.g. definition of grid, different interpolation methods)	2	2
A printed manual, even just the printed help pages, would be useful - we are just so used to referring to a book, not the help screen.	2	2
I would like to see cluster analysis statistics added to the system.	2	2
In addition to the group/area command, which breaks down all of the query areas into individual groups, it would be nice to have a sum of all the query area (e.g. how much total area meets the query criteria).	2/3	3
A written manual would definitely be helpful.	2/3	3/4
I would like to see an evaluation of which variables play the strongest role in the decision making process (e.g. I recently saw a GIS site prediction demonstration that looked great and predicted nearly all the sites, but the study area was in the Appalachian Mts. and the primary site determination factor was slope. Anyone could tell you that the sites would be located on the plateaus and flat lands, not on the Mt. slopes. That prediction system was useless.)	2/3	3/4
Terminology should be better defined, even simple terms like "area."	2/3	3/4
An introductory tutorial on ArcView should accompany this system.	2/3	3/4
All terms should be defined in the help component, including the statistical and interpolation terms.	3	3
I would like to see sample graphics or screen captures in the help component to give an idea of what to expect from each command.	3	3
It would be nice to have some hydrologic modeling tools added to the system.	4	3/4
It would be nice to have a tutorial to introduce ArcView and the Archaeological Site Prediction System to the archaeologist.	4	3/4
I would like to see a component that aids the user in entering data (e.g., digitizing, projecting, adding tabular data to the grid table)	4	3/4
It would be nice to have some assistance with adding and editing data in a table.	4/5	3/4

Table 8. Perceived impact the Archaeological Site Prediction System and similar systems will have on the field of Archaeology.

Comments - Perceived Impact on the Field of Archaeology	GIS Proficiency Rating	Computer Proficiency Rating
This system will be useful at the Phase I survey stage, if the data is already in digital format, to provide archaeologists with a quick overview of where they should concentrate their limited time and resources.	1/2	2
This system would also be useful for long term (5+ years) projects where in depth analysis of the landscape is desired.	1/2	2
I think user-friendly software will definitely encourage archaeologists to consider GIS, because "change is hard," and any encouragement is useful.	1/2	2
This system will definitely influence CRM (cultural resources management) archaeology. It would assist in project planning (e.g. determine how much money to bid, people to allocate, etc. based on the potential for sites).	1/2	2
This system would definitely help archaeologists with computer-phobia (like me) break into GIS.	2	2
"This system will not address the people who do not understand or want to use GIS or computers, but it will be perfect for the 'middle range' group of people...the computer literate, analytic types who are not GIS literate"	2/3	3
Archaeologist cannot afford GIS, therefore, user-friendly systems will not have a significant impact on the field.	2/3	3
The reason I have not explored GIS is time. Archaeologists simply do not have the time to devote to learning new software, user-friendliness will help, but it still takes time.	2/3	3/4
People who understand computers and can effectively communicate with people who don't understand them are few and far between. This and similar software may begin to bridge this gap.	2/3	3/4
We are not ready for any GIS software, the volume of data at the Bureau of History is overwhelming. "There is so much labor involved in getting this stuff in that I am not thinking too much yet about getting this stuff out."	2/3	3/4
I would like to use this sort of software for the State of Michigan's collections, but it will not be within our financial or time budget for several years, and by then all of this software will be out-dated.	2/3	3/4
This system has a lot of potential for doing a lot of interesting archaeological tasks.	2/3	3/4
The average archaeologist may not know enough about site prediction modeling to use this and similar systems to their potential.	3	3
Archeologists need someone/something to teach them GIS - it is not something that will be learned on one's own. This system is a good "stepping-stone" into GIS for archaeologists who are interested.	3	3
User-friendly software is an excellent way to learn "introductory" GIS. The archaeologist who is interested in advanced GIS will seek out the next level of GIS (e.g. Arc/Info) when he or she is ready.	4	3/4

Table 8 (cont).	GIS Proficiency Rating	Computer Proficiency Rating
The Archaeological Site Prediction System has great potential for the field of archaeology.	4/5	3/4
There will always be archaeologists who are "afraid of computers" or who feel that "computers should not be used for 'real archaeology,'" however, there are also archaeologists who are looking for a user-friendly way, like this, to learn GIS.	4/5	3/4

There is an obvious discrepancy between the number of positive comments versus the number of negative comments. It should be noted that the professional archaeologists interviewed were known to the Archaeological Site Prediction System's developer prior to the system's evaluation. Therefore, the results may be slightly biased in the positive direction due to this prior relationship. Despite this bias, several valuable conclusions can be drawn from the comments offered by the professional archaeologists.

Nearly all of the archaeologists interviewed stated that they would consider using the Archaeological Site Prediction System. They immediately projected their own data into the system to identify remotely located sites in Michigan's Upper Peninsula, search for patterns at an artifact-level resolution, analyze characteristics of historic farmstead data, etc. The positive interest and projected uses, by archaeologists of all GIS and computer skill levels indicate that the Archaeological Site Prediction System may have a wide audience of potential users. An archaeologist with minimal GIS skills commented, "There are a whole batch of computer literate types out there that don't want to spend the time to become GIS experts. I am fairly computer literate, but I don't have the time to sit down and go through a whole series of GIS courses and workshops to become proficient at setting up my own system. With a system like this available, I think it will allow

people like me to efficiently interface with GIS.” Additionally, a more advanced user noted, “This is a helpful tool for archaeologists experienced with GIS because the procedure is standardized and is a good starting point for additional GIS work.”

In addition to their desire to use the Archaeological Site Prediction System, several archaeologists expressed a fondness for particular commands or components of the system. The popular features included the ability to use either the *assisted* or *unassisted model builder*, the fact that temporary grids were automatically converted to permanent grids, the flexibility of the deductive and inductive modeling processes, the design and order of the pull-down menu, and the presence of the *group/area* function.

One potential hindrance of the Archaeological Site Prediction System is the fact that logistic regression cannot be run in ArcView. While one archaeologist did cite this as a limitation of the system, other archaeologists were undaunted. Since logistic regression cannot be run in ArcView, a logistic input file to be used in Arc/Info, is created in the Archaeological Site Prediction System. It was felt that the creation of this text file actually increased the flexibility of the Archaeological Site Prediction System by producing an output that can be taken to any commercial statistics package and used to run a variety of functions, including logistic regression.

Although the comments pertaining to the negative aspects/limitations of the system were much fewer in number, they illuminated potential problems with the Archaeological Site Prediction System. The primary issue, addressed by archaeologists of all skill levels, was the lack of assistance in data entry. Concern was voiced over scanning, digitizing, projecting, and tabulating data. Future modifications to the Archaeological Site Prediction System may recognize this current deficiency by adding a

seventh component entitled “data entry.” This component would be positioned before “set-up” in the pull-down menu and would permit the user to employ the digitizing features of ArcView, import data from a conventional database or spreadsheet, convert vector data to Arc/Info grid format, select a valid projection and corresponding measurement unit, and enter tabular data into grid tables.

Two archaeologists expressed a need for hydrologic modeling and cluster analysis functions. ArcView currently maintains a hydrologic modeling extension. Therefore, these desired functions could be incorporated into the Archaeological Site Prediction System without difficulty. However, due to the limited statistical package offered by ArcView 3.0, inclusion of cluster analysis algorithms could prove more difficult. Arc/Info is capable of grouping data into a specified number of clusters based on their attributes using the *isocluster* command. However, the statistics needed for testing the assumptions of this algorithm, performing cluster analysis with an unspecified number of clusters, and evaluating cluster significance are not available. Therefore, the *isocluster* command could be utilized in Arc/Info to cluster the data, but a text file containing the necessary variables would need to be created in order to perform further cluster analysis in a commercial statistics package. This would allow cluster analysis to be handled much like logistic regression in the current Archaeological Site Prediction System.

In addition to increased GIS functionality, modification of two functions was also desired. First, it was recognized that by converting zero data to “no data” in the *distance from...* command, actual locations of “no data” would be obscured. For example, surveyed regions that contain no artifacts would be confused with areas that had not yet been surveyed. Therefore, this command should include additional prompts that would

allow the user to treat the feature from which distance is calculated as a subset of the grid. This process would eliminate the need to convert zero data to “no data” by running *distance from...* on the subset of the grid rather than the entire surface. Secondly, it was felt that edge effect issues should be addressed when selecting random locations from within the survey area. Assuming that ESRI will provide the necessary programming code to run a query, a buffer could be placed around the survey boundary to exclude these areas as potential random site locations.

The treatment of sites was deemed another limitation of the Archaeological Site Prediction System. In this system, sites are interpreted as points. Unfortunately, with the exception of findspots (locations where a single artifact is discovered), most sites are areas, not points. While allowing for areal sites would require significant reconstruction of the logistic and evaluation components, this issue could be acknowledged for the remaining components if the sites were digitized as polygon features and treated as a groups of cells rather than assigned one grid cell. Commands such as *summarize zone*, which allow summary relationships to be calculated between particular grids and feature themes/grids (i.e. sites), could be applied to these areal sites to allow for further analysis.

Despite the enthusiasm for the Archaeological Site Prediction System and the claims that it was user-friendly, the number one suggestion was to develop a hard copy manual to accompany this system. Supplementary recommendations associated with this suggestion included a tutorial for the Archaeological Site Prediction System, a tutorial for ArcView, and sample data, graphics, and screen captures. It is hoped that a manual/tutorial can be produced prior to the distribution of this system to the archaeological community.

Increased definition of GIS and the Archaeological Site Prediction System's terminology was also a prevalent suggestion. While terms that are used by other commercial software, such as query, were not deemed inappropriate, technical terms that describe the interpolation methods and logistic regression were deemed confusing. Many definitions have already been incorporated into the ASPS help component or are a part of ArcView help. Remaining problematic terms could be added to the ASPS help and any manual/tutorial that is developed.

It was agreed that GIS is beneficial to several types of archaeology and that it should be run by knowledgeable archaeologists. As one archaeologist commented, "While sharing between the social science fields is good and has always happened, analysis has to be in the hands of the archaeologist...The archaeologist is needed to evaluate whether correlations are based on cultural traits or random phenomena. Is the size of a historic farmstead based on Finnish or Dutch influences or is it simply proportional to family size?" Despite the concurrence on the need for GIS, the ensuing responses pertaining to the impact the Archaeological Site Prediction System and similar systems would have on archaeology were divergent in nature. Several archaeologists were optimistic that user-friendly systems would encourage other archaeologists to explore GIS. As stated by one of the interviewees, "I think archaeologists are ready for an easy-to-use GIS." Other responses were less encouraging, citing that data entry issues, cost, and the time required to learn a new system, even if it is user-friendly, will continue to hinder the impact GIS will have on the field of archaeology. While it is hoped that the optimistic position prevails, the other responses suggest the need for creating GIS systems that simplify data entry, use cheaper software, and increase user-friendliness.

Geographer Evaluations

The results of the geographer interviews have been tabulated by positive aspects, negative aspects/limitations, and suggested improvements. Summaries and discussion of these results follow Table 9.

Table 9. Results of the geographer interviews.

Comments - Positive Aspects
This system simplifies many raster tools and deals with concepts that are not always addressed (e.g. resampling).
This is an interesting application.
This system seems like a good application of GIS.
A lot of potential errors appear to have been second-guessed (e.g. the order of the pull-down menu, availability of prompts, etc.). This should reduce the time needed to learn the system.
I like the pull-down menu and its logical order, these are easier to remember than multiple tool buttons or commands.
Comments - Negative Aspects/Limitations
This system creates a lot of grids. If left unmanaged, they will quickly clutter and fill a user's hard drive.
This system does not allow you to determine the distance between two features on the view. You should be able to click on one object on the view and then on a second object to determine how far apart they are.
I am apprehensive as to how well this can actually penetrate the archaeological field - my experience with archaeologists is that those who will use GIS already are and the rest will never respond, regardless of user-friendliness.
Comments - Suggested Improvements
A manual would be a good addition to the system, because many people do not think to use help menus while they are working.
I would like to see a "slide show" of screen captures that would demonstrate each component of the Archaeological Site Prediction System.
A data entry module would probably be a useful first step for introducing GIS to archaeologists.
You should be able to overlay grids (e.g. have four ideas as to where sites are located, create four models, overlay them, and assess overlapping area as the locations most likely to be a site).
What if sites are areas, not points? You should be allowed to place a buffer around the sites and assign the averaged data attributes for that area.

Table 9 (cont).
The random number generator should be examined and possibly re-programmed, my experience is that ESRI products do not have the best random number generators.
I am more inclined to see patterns in a visual/graphical display, rather than in a chart (histogram), therefore, more visual functions would be useful.
I would like to be able to visually compare different site prediction models.
More visualization of the data would be useful (e.g. hillshading).
It would be nice to see further analysis of the data (e.g. basic histograms of the data to see natural breaks and trends, and brushing to link locations on the map with the histogram).
It would be easier to enter grids into the project through a browse button, thereby reducing the need for memorization of grid names.
Indention may assist with differentiating the title of the component with its sub-commands, in addition to the capitalization.

The geographers interviewed concurred that the Archaeological Site Prediction System is an interesting application of GIS that addresses several user-friendly issues. Simplified commands, the logical order of the pull-down menu, and the availability of prompts were cited as positive aspects of the system.

While few negative aspects/limitations of the Archaeological Site Prediction System were noted by the geographers, two merit further discussion. First, it was recognized that this system does not provide for the management or deletion of unwanted grids. To avoid filling excessive hard drive space, future modifications of the Archaeological Site Prediction System will need to address this issue. Grid management may be accomplished through a *delete themes* option. In this option, a command would allow the user to delete a grid theme from the view. At that time, a secondary function would ask the user whether the grid should be deleted from the hard drive simultaneously.

The second negative aspect/limitation that requires further discussion pertains to the lack of an interactive tool that would permit the user to ascertain distance between two features on the view. ArcView 3.0 maintains a distance function that allows for this desired ability to graphically interact with the view, however, the map projection and units must be previously defined. Therefore, if the proposed “data entry” component is incorporated into the Archaeological Site Prediction System, thereby establishing the projection and units of the data, the distance tool could also be employed.

The geographer interviews produced several valuable suggestions for future improvements to the Archaeological Site Prediction System. Two of these recommendations mimicked those of the archaeologists: the creation of a manual/tutorial, and the development of a “data entry” component to assist archaeologists with inputting and organizing their data. In addition to these suggestions, a desire for functions that would allow for further analysis of the data and visualization of resultant site prediction models was expressed. Future modifications to the Archaeological Site Prediction System may accommodate these requests by incorporating tools for creating histograms of the data, producing summaries of the data with regard to other grids, and visualizing the data through hillshading. Additionally, map algebra functions would be accessible, providing the tools to overlay grids. These overlays could be used to visually compare the differences between site prediction models, obtain the summed area of multiple models, or determine the “most probable” site locations by calculating where several models overlap.

The archaeologist and geographer interviews provided a valuable assessment of the Archaeological Site Prediction System. Their insights highlighted aspects of this

system that are well designed and fulfill their intended objectives. Concurrently, their evaluations revealed aspects of this system that require further enhancement and additional programming.

CHAPTER FIVE

CONCLUSIONS

This research demonstrated that a user-friendly system could be developed to assist archaeologists with the creation of a site prediction models. The Archaeological Site Prediction System was produced through the customization of ArcView 3.0 and consists of six components: set-up, data preparation, deductive modeling, inductive modeling, evaluation, and help/information. Together, these components guide the archaeologist through the tasks of entering their data into the system, resampling their data, calculating distance from particular features in their data, creating deductive and inductive site prediction models, and evaluating these models through histograms.

In addition to the development of the Archaeological Site Prediction System, the research illustrated that this system could be successfully employed to create and evaluate viable site prediction models. Case study data, from the Athienou Archaeological Project, was used to test the system through an exploratory data analysis (EDA) process. The EDA process demonstrated that viable site prediction models can be created and evaluated with the Archaeological Site Prediction System and that for this application of EDA, resampling at different resolutions had little affect on the deductive model results.

Finally, this research evaluated the Archaeological Site Prediction System using interviews with several professional archaeologists and geographers. These interviews

elicited feedback pertaining to the positive aspects, negative aspects/limitations, suggestions for, and the impact of the Archaeological Site Prediction System. This feedback was tabulated and evaluated. Archaeologists indicated that the system met their needs and was user-friendly, however, its weaknesses included lack of assistance for data entry and availability of a users manual. Geographers agreed that this was a suitable application of GIS, but concurred with the need for a users manual and recommended that the system be more graphically oriented.

Considerations for future research pertaining to the Archaeological Site Prediction System can be classified into three categories. First, several of the archaeologist and geographer recommendations should be accommodated by the system. Items such as a component that eases data entry, commands that allow for further exploration/visualization of the data through histograms, zonal statistics, and map algebra, and a manual/tutorial with well defined terminology would be developed. Second, the Archaeological Site Prediction System would be converted to an extension that could be loaded into the ArcView software. This process would involve the encryption of the Avenue scripts to create an unalterable system for distribution. Third, the Archaeological Site Prediction System would be extended to the archaeological community via a web site. It is hoped that this software, provided gratis, will encourage the creation of GIS site prediction models by archaeologists.

The Archaeological Site Prediction System has the potential to put the control of archaeological site prediction modeling back into the hands of the archaeologist. This system will have achieved its goal if it replaces the geographer by placing the archaeologist in the role of the GIS specialist.

APPENDICES

APPENDIX A

LIST OF MAP SOURCES

Topographic Maps:

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 31/XXVII*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 31/XXVIII*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 31/XXIX*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 40/III*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 40/IV*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Cyprus Government's Department of Lands and Surveys and the British Government's Ministry of Overseas Development, 1977. *Cyprus 1:5000 Topographical Series, 40/V*. Published for the Cyprus Government by the Directorate of Overseas Surveys.

Additional Maps:

Government of Cyprus, Geological Survey Department, 1970. *Hydrological Map of Cyprus*. Printed by D. Couvas and Sons LTD. Limassol, Cyprus.

APPENDIX A

Government of Cyprus, Geological Survey Department, Ministry of Agriculture and Natural Resources, Department of Agriculture, 1987. *Calcareous Soils of Cyprus*. Nicosia, Cyprus.

Government of Cyprus, Geological Survey Department, Ministry of Agriculture and Natural Resources and Environment, 1995. *Geological Map of Cyprus*. Nicosia, Cyprus.

APPENDIX B

ARCVIEW 3.0 SCRIPTS

AddGrid Script

'This script allows the user to add their grid data as themes to an ArcView View.

```
theView = av.getActiveDoc
```

'Get directory path for data

```
_dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory Path",  
"C:\data")
```

```
if (_dirPath <> nil) then
```

'Get elevation data, make legend, and assign global variable

```
elevGrid = msgBox.input ("Please input the name of the elevation grid.", "Elevation Coverage",  
"elevation")
```

```
if (elevGrid <> nil) then
```

```
    elevDir = _dirPath.asString + "\" + elevGrid.asString
```

```
    srcElev = grid.makeSrcName (elevDir.asString)
```

```
    if (srcElev = nil) then
```

```
        theAnswer = msgBox.yesNo ("You have entered an invalid directory path or elevation grid. Do you  
want to reenter the information?", "Improper Input", true)
```

```
        if (theAnswer = true) then
```

```
            while (srcElev = nil)
```

```
                _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory  
Path", _dirPath.asString)
```

```
                if (_dirPath <> nil) then
```

```
                    elevGrid = msgBox.input ("Please input the name of the elevation grid.", "Elevation Coverage",
```

```
"elevation")
```

```
                    if (elevGrid <> nil) then
```

```
                        elevDir = _dirPath.asString + "\" + elevGrid.asString
```

```
                        srcElev = grid.makeSrcName (elevDir.asString)
```

```
                    else
```

```
                        exit
```

```
                    end
```

```
                else
```

```
                    exit
```

```
                end
```

```
            end
```

```
            theGrid = grid.make (srcElev)
```

```
            theGtheme = gTheme.make (theGrid)
```

```
            theView.addTheme (theGtheme)
```

```
            elevItem = msgBox.input ("What is the name of the field that contains your elevation information?",  
"Elevation Field", "elev-id")
```

```
            theLegend = theGtheme.getLegend
```

```
            theLegend.setLegendType (#LEGEND_TYPE_COLOR)
```

APPENDIX B

```

theLegend.natural (theGtheme, elevItem.asString, 9)
theLegend.setPrecision (-1)
theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
for each scheme in theColorSchemes
  if (scheme.getName = "Red Monochromatic") then
    theColorScheme = scheme
  end
end
theLegend.getSymbols.randomSavedSymbols (theColorScheme)
theGtheme.updateLegend
theGtheme.setVisible (true)
else
end
else
  theGrid = grid.make (srcElev)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  elevItem = msgBox.input ("What is the name of the field that contains your elevation information?",
"Elevation Field", "elev-id")
  theLegend = theGtheme.getLegend
  theLegend.setLegendType (#LEGEND_TYPE_COLOR)
  theLegend.natural (theGtheme, elevItem.asString, 9)
  theLegend.setPrecision (-1)
  theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
  for each scheme in theColorSchemes
    if (scheme.getName = "Red Monochromatic") then
      theColorScheme = scheme
    end
  end
  theLegend.getSymbols.randomSavedSymbols (theColorScheme)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
end
_elevation = srcElev.asString
else
end

'Get slope data, make legend, and assign global variable
slopeGrid = msgBox.input ("Please input the name of the slope grid.", "Slope Coverage", "slope")
if (slopeGrid <> nil) then
  slopeDir = _dirPath.asString + "\" + slopeGrid.asString
  srcSlope = grid.makeSrcName (slopeDir.asString)
  if (srcSlope = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid directory path or slope grid. Do you want
to reenter the information?", "Improper Input", true)
    if (theAnswer = true) then
      while (srcSlope = nil)
        _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)

```

APPENDIX B

```

if (_dirPath <> nil) then
  slopeGrid = msgBox.input ("Please input the name of the slope grid.", "Slope Coverage", "slope")
  if (slopeGrid <> nil) then
    slopeDir = _dirPath.asString + "\" + slopeGrid.asString
    srcSlope = grid.makeSrcName (slopeDir.asString)
  else
    exit
  end
else
  exit
end
end
theGrid = grid.make (srcSlope)
theGtheme = gTheme.make (theGrid)
theView.addTheme (theGtheme)
slopeItem = msgBox.input ("What is the name of the field that contains your slope information?",
"Slope Field", "slope-id")
theLegend = theGtheme.getLegend
theLegend.setLegendType (#LEGEND_TYPE_COLOR)
theLegend.natural (theGtheme, slopeItem.asString, 9)
theLegend.setPrecision (-1)
theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
for each scheme in theColorSchemes
  if (scheme.getName = "Green Monochromatic") then
    theColorScheme = scheme
  end
end
theLegend.getSymbols.randomSavedSymbols (theColorScheme)
theGtheme.updateLegend
theGtheme.setVisible (true)
else
end
else
  theGrid = grid.make (srcSlope)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  slopeItem = msgBox.input ("What is the name of the field that contains your slope information?",
"Slope Field", "slope-id")
  theLegend = theGtheme.getLegend
  theLegend.setLegendType (#LEGEND_TYPE_COLOR)
  theLegend.natural (theGtheme, slopeItem.asString, 9)
  theLegend.setPrecision (-1)
  theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
  for each scheme in theColorSchemes
    if (scheme.getName = "Green Monochromatic") then
      theColorScheme = scheme
    end
  end
  theLegend.getSymbols.randomSavedSymbols (theColorScheme)

```

APPENDIX B

```

    theGtheme.updateLegend
    theGtheme.setVisible (true)
end
_slope = srcSlope.asString
else
end

```

'Get aspect data, make legend, and assign global variable

```

aspectGrid = msgBox.input ("Please input the name of the aspect grid.", "Aspect Coverage", "aspect")
if (aspectGrid <> nil) then
    aspectDir = _dirPath.asString + "\" + aspectGrid.asString
    srcAspect = grid.makeSrcName (aspectDir.asString)
    if (srcAspect = nil) then
        theAnswer = msgBox.yesNo ("You have entered an invalid directory path or aspect grid. Do you want
to reenter the information?", "Improper Input", true)
        if (theAnswer = true) then
            while (srcAspect = nil)
                _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
                if (_dirPath <> nil) then
                    aspectGrid = msgBox.input ("Please input the name of the aspect grid.", "Aspect Coverage",
"aspect")
                    if (aspectGrid <> nil) then
                        aspectDir = _dirPath.asString + "\" + aspectGrid.asString
                        srcAspect = grid.makeSrcName (aspectDir.asString)
                    else
                        exit
                    end
                else
                    exit
                end
            end
            theGrid = grid.make (srcAspect)
            theGtheme = gTheme.make (theGrid)
            theView.addTheme (theGtheme)
            aspectItem = msgBox.input ("What is the name of the field that contains your aspect information?",
"Aspect Field", "aspect-id")
            theLegend = theGtheme.getLegend
            theLegend.setLegendType (#LEGEND_TYPE_COLOR)
            theLegend.interval (theGtheme, aspectItem.asString, 8)
            theLegend.setPrecision (0)
            theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
            for each scheme in theColorSchemes
                if (scheme.getName = "precipitation") then
                    theColorScheme = scheme
                end
            end
            theLegend.getSymbols.randomSavedSymbols (theColorScheme)
            theGtheme.updateLegend

```

APPENDIX B

```

    theGtheme.setVisible (true)
  else
  end
else
  theGrid = grid.make (srcAspect)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  aspectItem = msgBox.input ("What is the name of the field that contains your aspect information?",
"Aspect Field", "aspect-id")
  theLegend = theGtheme.getLegend
  theLegend.setLegendType (#LEGEND_TYPE_COLOR)
  theLegend.interval (theGtheme, aspectItem.asString, 8)
  theLegend.setPrecision (0)
  theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORRAMP)
  for each scheme in theColorSchemes
    if (scheme.getName = "precipitation") then
      theColorScheme = scheme
    end
  end
  theLegend.getSymbols.randomSavedSymbols (theColorScheme)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
end
_aspect = srcAspect.asString
else
end

'Get soils data, make legend, and assign global variable
soilGrid = msgBox.input ("Please input the name of the soil grid.", "Soil Coverage", "soils")
if (soilGrid <> nil) then
  soilDir = _dirPath.asString + "\" + soilGrid.asString
  srcSoil = grid.makeSrcName (soilDir.asString)
  if (srcSoil = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid directory path or soils grid. Do you want
to reenter the information?", "Improper Input", true)
    if (theAnswer = true) then
      while (srcSoil = nil)
        _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
        if (_dirPath <> nil) then
          soilGrid = msgBox.input ("Please input the name of the soil grid.", "Soil Coverage", "soils")
          if (soilGrid <> nil) then
            soilDir = _dirPath.asString + "\" + soilGrid.asString
            srcSoil = grid.makeSrcName (soilDir.asString)
          else
            exit
          end
        else
          exit
        end
      end
    end
  end

```

APPENDIX B

```

        end
    end
    theGrid = grid.make (srcSoil)
    theGtheme = gTheme.make (theGrid)
    theView.addTheme (theGtheme)
    theLegend = theGtheme.getLegend
    soilItem = msgBox.input ("What is the name of the field that contains your soils information?", "Soil
Field", "soiltype")
    theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
    theLegend.unique (theGtheme, soilItem.asString)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
else
end
else
    theGrid = grid.make (srcSoil)
    theGtheme = gTheme.make (theGrid)
    theView.addTheme (theGtheme)
    theLegend = theGtheme.getLegend
    soilItem = msgBox.input ("What is the name of the field that contains your soils information?", "Soil
Field", "soiltype")
    theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
    theLegend.unique (theGtheme, soilItem.asString)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
end
_soil = srcSoil.asString
else
end

```

'Get groundwater data, make legend, and assign global variable

```

grH2OGrid = msgBox.input ("Please input the name of the groundwater grid.", "Groundwater Coverage",
"grh2o")
if (grH2OGrid <> nil) then
    grH2ODir = _dirPath.asString + "\" + grH2OGrid.asString
    srcGrH2O = grid.makeSrcName (grH2ODir.asString)
    if (srcGrH2O = nil) then
        theAnswer = msgBox.yesNo ("You have entered an invalid directory path or groundwater grid. Do
you want to reenter the information?", "Improper Input", true)
        if (theAnswer = true) then
            while (srcGrH2O = nil)
                _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
                if (_dirPath <> nil) then
                    grH2OGrid = msgBox.input ("Please input the name of the groundwater grid.", "Groundwater
Coverage", "grh2o")
                    if (grH2OGrid <> nil) then
                        grH2ODir = _dirPath.asString + "\" + grH2OGrid.asString
                        srcGrH2O = grid.makeSrcName (grH2ODir.asString)

```

APPENDIX B

```

    else
        exit
    end
else
    exit
end
end
theGrid = grid.make (srcGrH2O)
theGtheme = gTheme.make (theGrid)
theView.addTheme (theGtheme)
theLegend = theGtheme.getLegend
grH2OItem = msgBox.input ("What is the name of the field that contains your groundwater
information?", "Groundwater Field", "aquifertype")
theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
theLegend.unique (theGtheme, grH2OItem.asString)
theGtheme.updateLegend
theGtheme.setVisible (true)
else
end
else
    theGrid = grid.make (srcGrH2O)
    theGtheme = gTheme.make (theGrid)
    theView.addTheme (theGtheme)
    theLegend = theGtheme.getLegend
    grH2OItem = msgBox.input ("What is the name of the field that contains your groundwater
information?", "Groundwater Field", "aquifertype")
    theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
    theLegend.unique (theGtheme, grH2OItem.asString)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
end
_grH2O = srcGrH2O.asString
else
end

'Get geology data, make legend, and assign global variable
geolGrid = msgBox.input ("Please input the name of the geology grid.", "Geology Coverage", "geology")
if (geolGrid <> nil) then
    geolDir = _dirPath.asString + "\" + geolGrid.asString
    srcGeol = grid.makeSrcName (geolDir.asString)
    if (srcGeol = nil) then
        theAnswer = msgBox.yesNo ("You have entered an invalid directory path or geology grid. Do you
want to reenter the information?", "Improper Input", true)
        if (theAnswer = true) then
            while (srcGeol = nil)
                _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
                if (_dirPath <> nil) then

```

APPENDIX B

```

    geolGrid = msgBox.input ("Please input the name of the geology grid.", "Geology Coverage",
"geology")
    if (geolGrid <> nil) then
        geolDir = _dirPath.asString + "\" + geolGrid.asString
        srcGeol = grid.makeSrcName (geolDir.asString)
    else
        exit
    end
    else
        exit
    end
    end
    theGrid = grid.make (srcGeol)
    theGtheme = gTheme.make (theGrid)
    theView.addTheme (theGtheme)
    theLegend = theGtheme.getLegend
    geolItem = msgBox.input ("What is the name of the field that contains your geology information?",
"Geology Field", "geologytype")
    theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
    theLegend.unique (theGtheme, geolItem.asString)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
    else
    end
    else
        theGrid = grid.make (srcGeol)
        theGtheme = gTheme.make (theGrid)
        theView.addTheme (theGtheme)
        theLegend = theGtheme.getLegend
        geolItem = msgBox.input ("What is the name of the field that contains your geology information?",
"Geology Field", "geologytype")
        theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
        theLegend.unique (theGtheme, geolItem.asString)
        theGtheme.updateLegend
        theGtheme.setVisible (true)
    end
    _geol = srcGeol.asString
    else
    end

```

'Get streams data, make legend, and assign global variable

```

streamGrid = msgBox.input ("Please input the name of the streams grid.", "Streams Coverage",
"streams")
if (streamGrid <> nil) then
    streamDir = _dirPath.asString + "\" + streamGrid.asString
    srcStream = grid.makeSrcName (streamDir.asString)
    if (srcStream = nil) then
        theAnswer = msgBox.yesNo ("You have entered an invalid directory path or streams grid. Do you
want to reenter the information?", "Improper Input", true)

```

APPENDIX B

```

if (theAnswer = true) then
  while (srcStream = nil)
    _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
    if (_dirPath <> nil) then
      streamGrid = msgBox.input ("Please input the name of the stream grid.", "Stream Coverage",
"streams")
      if (geolGrid <> nil) then
        streamDir = _dirPath.asString + "\" + streamGrid.asString
        srcStream = grid.makeSrcName (streamDir.asString)
      else
        exit
      end
    else
      exit
    end
  end
  theGrid = grid.make (srcStream)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  theLegend = theGtheme.getLegend
  streamItem = msgBox.input ("What is the name of the field that contains your streams information?",
"Stream Field", "streams")
  theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
  theLegend.unique (theGtheme, streamItem.asString)
  theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORSCHEME)
  for each scheme in theColorSchemes
    if (scheme.getName = "The High Seas") then
      theColorScheme = scheme
    end
  end
  theLegend.getSymbols.randomSavedSymbols (theColorScheme)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
else
  end
else
  theGrid = grid.make (srcStream)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  theLegend = theGtheme.getLegend
  streamItem = msgBox.input ("What is the name of the field that contains your streams information?",
"Stream Field", "streams")
  theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
  theLegend.unique (theGtheme, streamItem.asString)
  theColorSchemes = symbolList.getPreDefined (#SYMLIST_TYPE_COLORSCHEME)
  for each scheme in theColorSchemes
    if (scheme.getName = "The High Seas") then
      theColorScheme = scheme
    end
  end
end

```

APPENDIX B

```

    end
  end
  theLegend.getSymbols.randomSavedSymbols (theColorScheme)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
end
_stream = srcStream.asString
else
end

'Get site data, make legend, and assign global variable
siteGrid = msgBox.input ("Please input the name of the sites grid.", "Sites Coverage", "sites")
if (siteGrid <> nil) then
  siteDir = _dirPath.asString + "\" + siteGrid.asString
  srcSite = grid.makeSrcName (siteDir.asString)
  if (srcSite = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid directory path or sites grid. Do you want to
reenter the information?", "Improper Input", true)
    if (theAnswer = true) then
      while (srcSite = nil)
        _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
        if (_dirPath <> nil) then
          siteGrid = msgBox.input ("Please input the name of the site grid.", "Site Coverage", "sites")
          if (siteGrid <> nil) then
            siteDir = _dirPath.asString + "\" + siteGrid.asString
            srcSite = grid.makeSrcName (siteDir.asString)
          else
            exit
          end
        else
          exit
        end
      end
      theGrid = grid.make (srcSite)
      theGtheme = gTheme.make (theGrid)
      theView.addTheme (theGtheme)
      theLegend = theGtheme.getLegend
      siteItem = msgBox.input ("What is the name of the field that contains your site information?", "Site
Field", "sitetype")
      theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
      theLegend.unique (theGtheme, siteItem.asString)
      theGtheme.updateLegend
      theGtheme.setVisible (true)
    else
    end
  else
  end
  theGrid = grid.make (srcSite)
  theGtheme = gTheme.make (theGrid)

```

APPENDIX B

```

theView.addTheme (theGtheme)
theLegend = theGtheme.getLegend
siteItem = msgBox.input ("What is the name of the field that contains your site information?", "Site
Field", "sitetype")
theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
theLegend.unique (theGtheme, siteItem.asString)
theGtheme.updateLegend
theGtheme.setVisible (true)
end
_site = srcSite.asString
else
end

'Get additional coverages, determine whether data is categorical or numerical, and make
appropriate legend
i = 0
while (msgBox.yesNo ("Would you like to add an additional grid theme to the view?", "Additional
Grids", true))
  newGrid = msgBox.input ("Please enter the name of your grid.", "New Coverage", "")
  if (newGrid <> nil) then
    newDir = _dirPath.asString + "\" + newGrid.asString
    srcNew = grid.makeSrcName (newDir.asString)
    if (srcNew = nil) then
      theAnswer = msgBox.yesNo ("You have entered an invalid directory path or grid name. Do you want
to reenter the information?", "Improper Input", true)
      if (theAnswer = true) then
        while (srcNew = nil)
          _dirPath = msgBox.input ("Please enter the directory path for the location of your data.", "Directory
Path", _dirPath.asString)
          if (_dirPath <> nil) then
            newGrid = msgBox.input ("Please input the name of the grid.", "New Coverage", "")
            if (newGrid <> nil) then
              newDir = _dirPath.asString + "\" + newGrid.asString
              srcnew = grid.makeSrcName (newDir.asString)
            else
              exit
            end
          else
            exit
          end
        end
        theGrid = grid.make (srcNew)
        theGtheme = gTheme.make (theGrid)
        theView.addTheme (theGtheme)
        newItem = msgBox.input ("What is the name of the field that contains your grid information?",
"Grid Field", "")
        legendList = {"Unique Value", "Graduated Color"}
        newLegend = msgBox.listAsString (legendList, "Does your grid theme require a unique value
legend or a graduated color legend?", "Legend Types")

```

APPENDIX B

```

if (newLegend = "Unique Value") then
  theLegend = theGtheme.getLegend
  theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
  theLegend.unique (theGtheme, newItem.asString)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
elseif (newLegend = "Graduated Color") then
  theLegend = theGtheme.getLegend
  theLegend.setLegendType (#LEGEND_TYPE_COLOR)
  theLegend.interval (theGtheme, newItem.asString, 9)
  theGtheme.updateLegend
  theGtheme.setVisible (true)
else
  end
else
  end
else
  theGrid = grid.make (srcNew)
  theGtheme = gTheme.make (theGrid)
  theView.addTheme (theGtheme)
  newItem = msgBox.input ("What is the name of the field that contains your grid information?", "Grid
Field", "")
  legendList = {"Unique Value", "Graduated Color"}
  newLegend = msgBox.listAsString (legendList, "Does your grid theme require a unique value legend
or a graduated color legend?", "Legend Types")
  if (newLegend = "Unique Value") then
    theLegend = theGtheme.getLegend
    theLegend.setLegendType (#LEGEND_TYPE_UNIQUE)
    theLegend.unique (theGtheme, newItem.asString)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
  elseif (newLegend = "Graduated Color") then
    theLegend = theGtheme.getLegend
    theLegend.setLegendType (#LEGEND_TYPE_COLOR)
    theLegend.interval (theGtheme, newItem.asString, 9)
    theLegend.setPrecision (-1)
    theGtheme.updateLegend
    theGtheme.setVisible (true)
  else
    end
  end
else
  end
end
end
end

```

APPENDIX B

AddXY Script

'This script adds (x,y) fields to an Ftab, these values can then be extracted and used in the "logInput" script.

```
theView = av.getActiveDoc
```

'Check to see if View is projected

```
theProjection = theView.getProjection
```

```
project_flag = theProjection.isNull.not
```

'Check for ability to edit _theFtab2

```
edit_state = _theFtab2.isEditable
```

```
if (_theFtab2.canEdit) then
```

```
  _theFtab2.setEditable (TRUE)
```

```
  if ((_theFtab2.canAddFields).not) then
```

```
    msgBox.info ("Cannot add fields.", "")
```

```
    exit
```

```
  end
```

```
else
```

```
  msgBox.info ("Cannot edit table.", "")
```

```
  exit
```

```
end
```

'Determine whether the Ftheme is a polygon coverage or a point coverage

```
if (((_theFtheme2.getSrcName.getSubName = "point") or (_theFtheme2.getSrcName.getSubName = "polygon")).not) then
```

```
  msgBox.error ("An error has occurred.", "Error")
```

```
  exit
```

```
end
```

'Add (x,y) fields - overwrite if already exist

```
_theFtab2.setEditable (TRUE)
```

```
xExists = (_theFtab2.findField ("X-coord") = nil).not
```

```
yExists = (_theFtab2.findField ("Y-coord") = nil).not
```

```
if (xExists or yExists) then
```

```
  if (xExists) then
```

```
    _theFtab2.removeFields ({_theFtab2.findField ("X-coord")})
```

```
  end
```

```
  if (yExists) then
```

```
    _theFtab2.removeFields ({_theFtab2.findField ("Y-coord")})
```

```
  end
```

```
end
```

```
x = field.make ("X-coord", #FIELD_LONG, 18, 0)
```

```
y = field.make ("Y-coord", #FIELD_LONG, 18, 0)
```

```
_theFtab2.addFields ({x,y})
```

'Add coordinate information to (x,y) fields

APPENDIX B

```
if (_theFtheme2.getSrcName.getSubName = "point") then
  if (project_flag) then
    _theFtab2.calculate ("[shape].returnProjected (theProjection).getX", x)
    _theFtab2.calculate ("[shape].returnProjected (theProjection).getY", y)
  else
    _theFtab2.calculate ("[shape].getX", x)
    _theFtab2.calculate ("[shape].getY", y)
  end
end
else
  if (project_flag) then
    _theFtab2.calculate ("[shape].returnCenter.returnProjected (theProjection).getX", x)
    _theFtab2.calculate ("[shape].returnCenter.returnProjected (theProjection).getY", y)
  else
    _theFtab2.calculate ("[shape].returnCenter.getX", x)
    _theFtab2.calculate ("[shape].returnCenter.getY", y)
  end
end
end
```

APPENDIX B

AddXY2 Script

'This script adds (x,y) fields to an Ftab, these values can then be extracted and used in the “logInput” and “histogram” scripts.

```
theView = av.getActiveDoc
```

'Check to see if View is projected

```
theProjection = theView.getProjection
```

```
project_flag = theProjection.isNull.not
```

'Check for ability to edit _theFtab2

```
edit_state = _theFtab2.isEditable
```

```
if (_theFtab2.canEdit) then
```

```
  _theFtab2.setEditable (TRUE)
```

```
  if ((_theFtab2.canAddFields).not) then
```

```
    msgBox.info ("Cannot add fields.", "")
```

```
    exit
```

```
  end
```

```
else
```

```
  msgBox.info ("Cannot edit table.", "")
```

```
  exit
```

```
end
```

'Determine whether the Ftheme is a polygon coverage or a point coverage

```
if (((_theFtheme2.getSrcName.getSubName = "point") or (_theFtheme2.getSrcName.getSubName = "polygon")).not) then
```

```
  msgBox.error ("An error has occurred.", "Error")
```

```
  exit
```

```
end
```

'Add (x,y) fields - overwrite if already exist

```
_theFtab2.setEditable (TRUE)
```

```
xExists = (_theFtab2.findField ("X-coord") = nil).not
```

```
yExists = (_theFtab2.findField ("Y-coord") = nil).not
```

```
if (xExists or yExists) then
```

```
  if (xExists) then
```

```
    _theFtab2.removeFields ({_theFtab2.findField ("X-coord")})
```

```
  end
```

```
  if (yExists) then
```

```
    _theFtab2.removeFields ({_theFtab2.findField ("Y-coord")})
```

```
  end
```

```
end
```

```
x = field.make ("X-coord", #FIELD_LONG, 18, 0)
```

```
y = field.make ("Y-coord", #FIELD_LONG, 18, 0)
```

```
_theFtab2.addFields ({x,y})
```

'Add coordinate information to (x,y) fields

APPENDIX B

```
if (_theFtheme2.getSrcName.getSubName = "point") then
  if (project_flag) then
    _theFtab2.calculate ("[shape].returnProjected (theProjection).getX", x)
    _theFtab2.calculate ("[shape].returnProjected (theProjection).getY", y)
  else
    _theFtab2.calculate ("[shape].getX", x)
    _theFtab2.calculate ("[shape].getY", y)
  end
else
  if (project_flag) then
    _theFtab2.calculate ("[shape].returnCenter.returnProjected (theProjection).getX", x)
    _theFtab2.calculate ("[shape].returnCenter.returnProjected (theProjection).getY", y)
  else
    _theFtab2.calculate ("[shape].returnCenter.getX", x)
    _theFtab2.calculate ("[shape].returnCenter.getY", y)
  end
end
end
```

APPENDIX B

Convert Script

'This script converts the active theme to a permanent grid, using an ArcView system script (recommended use: after running the "unassisted modeling" script).

```
theView = av.getActiveDoc
```

'Determine whether one grid is active

```
theAnswer = msgBox.yesNo ("The theme to convert should be active and all other themes should be inactive. Do you want to continue with the conversion?", "Continue Conversion", true)
```

```
if (theAnswer = false) then
```

```
  exit
```

```
else
```

```
  gridList = list.make
```

```
  for each t in theView.getActiveThemes
```

```
    if (t.is (gTheme)) then
```

```
      theGrid = t.getGrid
```

```
      theName = theGrid.getSrcName.asString
```

```
      gridList.add (theName.asString)
```

```
    end
```

```
  end
```

```
  gridCount = gridList.count
```

```
  if (gridCount = 0) then
```

```
    msgBox.info ("You need to activate the grid to be converted.", "Improper Grid")
```

```
    exit
```

```
  elseif (gridCount > 1) then
```

```
    msgBox.info ("You have more than one grid activated.", "Improper Grid")
```

```
    exit
```

```
  elseif (gridCount = 1) then
```

'Convert temporary file to permanent grid

```
  _themeName = "query"
```

```
  av.run ("makeGrid2", "")
```

```
  for each t in theView.getActiveThemes
```

```
    theView.deleteTheme (t)
```

```
  end
```

```
end
```

```
end
```

APPENDIX B

DistanceFrom Script

'This script calculates a grid of distance from a specified feature in another grid.

```
theView = av.getActiveDoc
```

'Get stream coverage name as possible feature grid

```
if (_stream2 <> nil) then
  strm = _stream2.asString
elseif (_stream <> nil) then
  strm = _stream.asString
else
  strm = "Stream Coverage"
end
```

'Get the grid with the feature that distance will be calculated from

```
theDistGrid = msgBox.input ("Please enter the grid that contains the feature to calculate distance from.",
"Distance From Grid", strm.asString)
```

```
if (theDistGrid <> nil) then
```

```
  if (_dirPath <> nil) then
```

```
    theFile = _dirPath + "\"
```

```
    theSrcName = grid.makeSrcName (theFile.asString + theDistGrid.asString)
```

```
    if (theSrcName = nil) then
```

```
      reEnter = msgBox.yesNo ("You have entered an invalid grid name. Would you like to reenter the
information?", "Reenter Data", True)
```

```
      if (reEnter <> false) then
```

```
        while (theSrcName = nil)
```

```
          directory = msgBox.input ("Please enter the directory path for location of your grid.", "Directory
Path", _dirPath.asString)
```

```
          if (directory <> nil) then
```

```
            theGrid = msgBox.input("Please enter the grid that contains the feature to calculate distance from.",
```

```
"Distance From Grid", strm.asString)
```

```
            if (theGrid <> nil) then
```

```
              theFile = directory.asString + "\" + theGrid.asString
```

```
              theSrcName = grid.makeSrcName (theFile.asString)
```

```
            else
```

```
              exit
```

```
            end
```

```
          else
```

```
            exit
```

```
          end
```

```
        end
```

```
      else
```

```
        exit
```

```
      end
```

```
    else
```

```
      end
```

```
    else
```

APPENDIX B

```

    directory = msgBox.input ("Please enter the directory path for location of your grid.", "Directory Path",
_dirPath.asString)
    if (directory <> nil) then
        theGrid = msgBox.input ("Please enter the grid that contains the feature to calculate distance from.",
"Distance From Grid", strm.asString)
        if (theGrid <> nil) then
            theFile = directory.asString + "\" + theGrid.asString
            theSrcName = grid.makeSrcName (theFile.asString)
            if (theSrcName = nil) then
                msgBox.error ("You have entered an invalid grid name or directory path.", "Invalid Entry")
                exit
            end
        else
            exit
        end
    else
        exit
    end
end
else
    exit
end
end

```

'Make grid and run eucDistance

```

theGrid = grid.make (theSrcName)
theAnswer = msgBox.yesNo ("Distance will be calculated from zero values. Set zero values to no data?",
"Zero to No Data", True)
if (theAnswer = "False") then
    distGrid = theGrid.eucDistance (nil, nil, nil)
else
    noDataGrid = (theGrid = 0.asGrid).setNull (theGrid)
    distGrid = noDataGrid.eucDistance (nil, nil, nil)
end
theGtheme = gTheme.make (distGrid)
for each t in theView.getThemes
    t.setActive (False)
end
_themeName = "Dist"
theGtheme.setName ("Dist From" ++theSrcName.asString)
theView.addTheme (theGtheme)
theGtheme.setActive (True)
for each t in theView.getActiveThemes
    if (t <> nil) then
        theGrid = t.getGrid
        intGrid = theGrid.int
        intGtheme = gTheme.make (intGrid)
        intGtheme.setName ("Dist From" ++theSrcName.asString)
        theView.addTheme (intGtheme)
        theView.deleteTheme (t)
    end
end

```

APPENDIX B

```
    intGtheme.setActive (True)
else
    exit
end
end
```

```
'Convert temporary grid into a permanent grid
av.run ("makeGrid2", "")
theView.deleteTheme (intGtheme)
```

APPENDIX B

Enable Script

'This script controls whether the user can access the tools to graphically input known sites into a text file for the "logInput" script.

```
theView = av.getActiveDoc  
enabled = FALSE
```

```
'Enable or disable the input and stop tools  
if ( _theInput = "Graphical with the mouse") then  
    SELF.setEnabled (TRUE)  
elseif ( _theInput = "finished") then  
    SELF.setEnabled (FALSE)  
else  
    SELF.setEnabled (FALSE)  
end
```

APPENDIX B

Enable2 Script

'This script controls whether the user can access the tool to draw a bounding area rectangle for the "logInput" script.

```
theView = av.getActiveDoc  
enabled = FALSE
```

```
'Enable or disable the rectangle tool  
if ( _theRecInput = "Draw Rectangle") then  
    SELF.setEnabled (TRUE)  
elseif ( _theRecInput = "finished") then  
    SELF.setEnabled (FALSE)  
else  
    SELF.setEnabled (FALSE)  
end
```

APPENDIX B

Enable3 Script

'This script controls whether the user can access the tools to graphically input known sites into a **table** for the "histogram" script.

```
theView = av.getActiveDoc
enabled = FALSE

'Enable or disable the input and stop tools
if (theInput2 = "Graphical with the mouse") then
    SELF.setEnabled (TRUE)
elseif (theInput2 = "finished") then
    SELF.setEnabled (FALSE)
else
    SELF.setEnabled (FALSE)
end
```

APPENDIX B

GroupArea Script

'This script groups connected cells of a given value, assigns them a unique value, and calculates their area (recommended use: after the creation of a deductive model).

```

theView = av.getActiveDoc

'Get the grid to be grouped
theAnswer = msgBox.yesNo ("The grid to be grouped should be active. All other grids should be inactive.
Do you want to continue?", "Continue Group", true)
if (theAnswer = false) then
    exit
else
    gridList = list.make
    for each t in theView.getActiveThemes
        if (t.is (gTheme)) then
            theGrid = t.getGrid
            theName = theGrid.getSrcName.asString
            gridList.add (theName.asString)
        end
    end
    gridCount = gridList.count
    if (gridCount = 0) then
        msgBox.info ("You need to activate the grid to be grouped.", "Improper Grid")
        exit
    elseif (gridCount > 1) then
        msgBox.info ("You have more than one grid activated.", "Improper Grid")
        exit
    end

'Group cells of like values and create a permanent grid of grouped regions
elseif (gridCount = 1) then
    theAnswer = msgBox.yesNo ("Do you want like cells that touch on the diagonal to be considered
connected?", "Diagonal or Orthogonal Connectivity", true)
    if (theAnswer = true) then
        diagProp = true
    elseif (theAnswer = false) then
        diagProp = false
    end
    groupGrid = theGrid.regionGroup (diagProp, false, nil)
    groupTheme = gTheme.make (groupGrid)
    for each t in theView.getThemes
        t.setActive (False)
    end
    _themeName = "Group"
    groupTheme.setName ("Grouped" ++ theName.asString)
    theView.addTheme (groupTheme)
    groupTheme.setActive (True)

```

APPENDIX B

```

    av.run ("makeGrid2", "")
    theView.deleteTheme (groupTheme)
end

'Determine area for each of the individual regions and create a permanent grid of area
    regionArea = groupGrid.zonalGeometry (#GRID_GEOMDESC_AREA)
    areaTheme = gTheme.make (regionArea)
    for each t in theView.getThemes
        t.setActive (False)
    end
    __themeName = "Area"
    areaTheme.setName ("Areas for" ++theName.asString)
    theView.addTheme (areaTheme)
    areaTheme.setActive (True)
    theGrid = areaTheme.getGrid
    theIntGrid = theGrid.int
    theIntGrid2 = theIntGrid / 100
    intGtheme = gTheme.make (theIntGrid2)
    theView.addTheme (intGtheme)
    areaTheme.setActive (False)
    intGtheme.setActive (True)
    av.run ("makeGrid2", "")
    theView.deleteTheme (intGtheme)
end
msgBox.warning ("The theme " +areaTheme.asString++ "contains the correct area values. The area grid
that was created exceeds the limitations for an integer grid, therefore its values that have been divided by
100. Use this reduced integer grid for future queries on area.", "Integer Grid Exceeded Limits")

'Query on area - the development of the query string runs correctly, however, the actual query
builder does not run properly, therefore, entire function has been commented out

'Determine whether user desires to query their area grid
'qArea = msgBox.yesNo ("Do you want to query your area theme?", "Query Area", True)
    'if (qArea <> False) then
        'for each t in theView.getActiveThemes
            'if (t <> nil) then
                'theGrid = t.getGrid
                'theInputSrc = theGrid.getSrcName
                'theVtab = theGrid.getVtab
            'else
                'msgBox.info ("You do not have your Area grid in the view and active.", "Cannot Query")
                'exit
            'end
        'end
    'else
        'exit
    'end

```

APPENDIX B

```

• Determine whether areas to be queried are greater, lesser, or between two values, and assist user in developing the query string
'optionsList = {"Greater", "Lesser", "Between Two Values"}
'greaterLesser = msgBox.choiceAsString (optionsList, "Do you want to query for locations greater than (>) or lesser (<) than a particular area or in between two areas (> and <)?", "Greater/Lesser")
'if (greaterLesser = "Greater") then
    'greater = msgBox.input ("Please enter the minimum area, in squared units, that you desire for your query.", "Greater", "250")
    'if (greater > nil) then
        'expr = "[" + theName.asString + "]" > " + greater.asString + ")"
    'else
        'exit
    'end
'elseif (greaterLesser = "Lesser") then
    'lesser = msgBox.input ("Please enter the maximum area, in squared units, that you desire for your query.", "Greater", "250")
    'if (lesser > nil) then
        'expr = "[" + theInputSrc.asString + "]" < " + lesser.asString + ")"
        'theQueryGrid = theGrid.test (expr.asString)
        'queryArea = gTheme.make (theQueryGrid)
        'queryArea.setVisible (true)
    'else
        'exit
    'end
'elseif (greaterLesser = "Between Two Values") then
    'greater = msgBox.input ("Please enter the minimum area, in squared units, that you desire for your query.", "Greater", "250")
    'if (greater > nil) then
        'expr1 = "[" + theName.asString + "]" > " + greater.asString + ")"
    'else
        'exit
    'end
    'lesser = msgBox.input ("Please enter the maximum area, in squared units, that you desire for your query.", "Greater", "250")
    'if (lesser > nil) then
        'expr2 = "[" + theName.asString + "]" < " + lesser.asString + ")"
    'else
        'exit
    'end
    'expr = expr1.asString ++ "and" ++ expr2.asString
'end

• runs query builder - not currently functioning
'theBitmap = gridVtab.getSelection
'gridVtab.query (expr, theBitmap, #VTAB_SELTYPE_NEW)
'gridVtab.setSelection (theBitmap)
'gridVtab.updateSelection

```

APPENDIX B

Help Script

'This script opens and runs the help segment for the Archaeological Site Prediction System.

system.execute ("winhelp.exe C:\data\asps\help.hlp")

APPENDIX B

Histogram Script

'This script allows for the evaluation of deductive or inductive site prediction models through the use of histograms.

```
theView = av.getActiveDoc
```

```
'Get model type
```

```
optionsList = {"Deductive Model", "Inductive Model"}
```

```
theModel = msgBox.listAsString (optionsList, "Do you want to evaluate a model created through the  
deductive modeling procedure or the inductive modeling procedure?", "Model Type")
```

```
'Check for directory path
```

```
if (_dirPath = nil) then
```

```
    _dirPath = msgBox.input ("Please enter the directory path for your data.", "Directory Path", "C:\data")
```

```
end
```

```
'Create histogram for a deductive model
```

```
if (theModel = "Deductive Model") then
```

```
    _theQueryFile = msgBox.input ("What is the name of your query grid?", "Query Grid", "query")
```

```
    if (_theQueryFile = nil) then
```

```
        exit
```

```
    else
```

```
        aSrcName = _dirPath + "\" + _theQueryFile.asString
```

```
        theSrcName = grid.makeSrcName (aSrcName.asString)
```

```
        if (theSrcName = nil) then
```

```
            theAnswer = msgBox.yesNo ("You have entered an invalid grid name. Would you like to re-enter the  
query grid?", "Invalid Entry", TRUE)
```

```
            if (theAnswer = true) then
```

```
                while (theSrcName = nil)
```

```
                    _theQueryFile = msgBox.input ("What is the name of your query grid?", "Query Grid", "query")
```

```
                    if (_theQueryFile = nil) then
```

```
                        exit
```

```
                    else
```

```
                        aSrcName = _dirPath + "\" + _theQueryFile.asString
```

```
                        theSrcName = grid.makeSrcName (aSrcName.asString)
```

```
                    end
```

```
                end
```

```
            else
```

```
                exit
```

```
            end
```

```
        end
```

```
    end
```

```
'Determine how known sites will be entered
```

```
inputOptions = {"Graphical with the mouse", "Tabular with UTM coordinates", "Taken from the grid that  
contains the site data", "Taken from a point coverage that contains site data"}
```

APPENDIX B

```

    _theInput2 = msgBox.listAsString (inputOptions, "Please select a method for entering your site
locations.", "Input Method")

```

'Graphical selection of points

```

    if (_theInput2 = "Graphical with the mouse") then
        msgBox.info ("Click on the ""red pointer"" tool on the toolbar to begin entering site locations. Press on
the ""S"" stop tool on the toolbar to finish entering known site locations.", "Enter Known Points
Graphically")

```

'Tabular entry of coordinates

```

    elseif (_theInput2 = "Tabular with UTM coordinates") then
        av.run ("makeVtab", "")
        enter = "yes"
        while (enter = "yes")
            xCoor = msgBox.input ("Please enter the Easting coordinate for your site.", "Easting Coordinate", "")
            if (xCoor = nil) then
                exit
            elseif (xCoor <> nil) then
                yCoor = msgBox.input ("Please enter the Northing coordinate for your site.", "Northing Coordinate",
                "")
                pt = point.make (xCoor.asNumber, yCoor.asNumber)
                newPt = point.make (xCoor.asNumber, (yCoor.asNumber - 180))

```

'Put marker symbol on screen

```

    s = basicMarker.make
    s.setStyle (#BASICMARKER_STYLE_PATTERN)
    s.setColor (color.getBlack)
    s.setSize(28)
    d = theView.getDisplay
    d.beginClip
    d.drawPoint (newpt, s)
    d.endClip
    d.Flush
    av.GetProject.SetModified(true)

```

'Get the x and y values and sets format

```

    i = "site"
    x = xCoor.asNumber
    y = yCoor.asNumber
    x.setFormat ("ddddddd")
    y.setFormat ("ddddddd")

```

'Create a record in theVtab2 and add coordinate information

```

    recNum = _theVtab2.addRecord
    _theVtab2.setValueString (_theSite2, recNum, i.asString)
    _theVtab2.setValueString (_xCoor, recNum, x.asString)
    _theVtab2.setValueString (_yCoor, recNum, y.asString)

```

'Determine whether the point falls within the query area and add information to theVtab2

APPENDIX B

```

aSrcName = _dirPath + "\" + _theQueryFile.asString
theSrcName = grid.makeSrcName (aSrcName.asString)
theGrid = grid.make (theSrcName)
theGtheme = gTheme.make (theGrid)
num = theGtheme.returnCellValue (pt)
theVtab = theGtheme.getVtab
theField = "Site"
theFieldName = _theVtab2.findField (theField.asString)
_theVtab2.setValueString (theField, recNum, num.asString)
end
enter = "no"
end
av.run ("randomPts2", "")

```

'Take points from a grid coverage

```

elseif (_theInput2 = "Taken from the grid that contains the site data") then
  av.run ("makeVtab", "")

```

'Get site file name and make point coverage from grid

```

if (_site2 <> nil) then
  theFile = _site2.asString
elseif (_site <> nil) then
  theFile = _site.asString
else
  theFile = "Sites"
end
theSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
theFile.asString)
if (theSiteFile <> nil) then
  theGtheme = theView.findTheme (theSiteFile)
  if (theGtheme.is (Gtheme).not) then
    theGtheme = nil
  end
  if (theGtheme = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid grid name. Would you like to re-enter the
Site grid?", "Invalid Entry", TRUE)
    if (theAnswer = true) then
      while (theGtheme = nil)
        aSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
theFile.asString)
        if (aSiteFile = nil) then
          exit
        else
          theGtheme = theView.findTheme (aSiteFile.asString)
        end
      end
    else
      exit
    end
  end

```

APPENDIX B

```

    end
  else
    exit
  end
  theFile = av.getProject.makeFileName ("cover4", "")
  theFtab = theGtheme.exportToFtab (theFile)
  _theFtheme2 = Ftheme.make (theFtab)
  _theFtab2 = _theFtheme2.getFtab
  av.run ("addXY2", "")

```

'Determine whether all values should be used as sites

```

  theVtab = theGtheme.getVtab
  theSitesNumber = theVtab.getNumRecords
  theAnswer = msgBox.yesNo ("Do you want to use all" ++theSitesNumber.asString++ "sites as the logistic
input?", "Keep All Sites", TRUE)

```

'If only a portion of the values desired, get field and categories desired and add appropriate data to the Vtab

```

  if (theAnswer <> TRUE) then
    theFieldList = theVtab.getFields
    theField = msgBox.listAsString (theFieldList, "Which field contains the information that decides
whether a site will be kept?", "Select Field")
    if (theField <> nil) then
      theList = list.make
      for each i in theVtab
        theValue = theVtab.returnValueString (theField, i)
        theList.add (theValue)
      end
      theList.removeDuplicates
    end
    theUsableSites = msgBox.multiListAsString (theList, "Please select the sites to be used in the logistic
input. You may choose multiple sites by depressing the shift key while you make your selection.",
"Desired Categories")
    for each v in _theFtab2
      xVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("X-coord")), v)
      yVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("Y-coord")), v)
      theGridcode = _theFtab2.returnValueNumber ((_theFtab2.findField ("Gridcode")), v)
      pt = point.make (xVal, yVal)
      theNum = theGridcode - 1
      theNewField2 = theVtab.findField (theField.asString)
      theDecidingField = theVtab.returnValueString (theNewField2, theNum)
      for each s in theUsableSites
        if (theDecidingField.asString = s.asString) then

```

'Create a record in the _theVtab2 and add coordinate information

```

  p = "site"
  recNum = _theVtab2.addRecord
  _theVtab2.setValueString (_theSite2, recNum, p.asString)
  _theVtab2.setValueString (_xCoord, recNum, xVal.asString)

```

APPENDIX B

```

    _theVtab2.setValueString (_yCoord, recNum, yVal.asString)

'Determine whether the point falls within the query area and add field information
    aSrcName = _dirPath + "\" + _theQueryFile.asString
    theSrcName = grid.makeSrcName (aSrcName.asString)
    theGrid = grid.make (theSrcName)
    theGtheme2 = gTheme.make (theGrid)
    num = theGtheme2.returnCellValue (pt)
    theField3 = "Site"
    theFieldName = _theVtab2.findField (theField3.asString)
    _theVtab2.setValueString (theFieldName, recNum, num.asString)
end
end
end
else

'Use all values as sites
for each v in _theFtab2
    xVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("X-coord")), v)
    yVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("Y-coord")), v)
    pt = point.make (xVal, yVal)

'Create a record in the _theVtab2 and add coordinate information
    p = "site"
    recNum = _theVtab2.addRecord
    _theVtab2.setValueString (_theSite2, recNum, p.asString)
    _theVtab2.setValueString (_xCoord, recNum, xVal.asString)
    _theVtab2.setValueString (_yCoord, recNum, yVal.asString)

'Determine whether the point falls within the query area and add field information
    aSrcName = _dirPath + "\" + _theQueryFile.asString
    theSrcName = grid.makeSrcName (aSrcName.asString)
    theGrid = grid.make (theSrcName)
    theGtheme3 = gTheme.make (theGrid)
    num = theGtheme3.returnCellValue (pt)
    theField4 = "Site"
    theFieldName = _theVtab2.findField (theField4.asString)
    _theVtab2.setValueString (theFieldName, recNum, num.asString)
end
end
theTable = table.make (_theVtab2)

'Get random points
    av.run ("randomPts2", "")

'Take points from a site coverage
elseif (_theInput2 = "Taken from a point coverage that contains site data") then
    av.run ("makeVtab", "")

```

APPENDIX B

'Get the site file

```

theSiteFile = msgBox.input ("What is the name of the coverage that contains your sites?", "Site Grid",
"Sites")
if (theSiteFile <> nil) then
  _theFtheme2 = theView.findTheme (theSiteFile)
  if (_theFtheme2.is (Ftheme).not) then
    _theFtheme2 = nil
  end
  if (_theFtheme2 = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid grid name. Would you like to re-enter the
Site grid?", "Invalid Entry", TRUE)
    if (theAnswer = TRUE) then
      while (_theFtheme2 = nil)
        aSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
"Sites")
        if (aSiteFile = nil) then
          exit
        else
          _theFtheme2 = theView.findTheme (aSiteFile)
        end
      end
    else
      exit
    end
  end
  _theFtab2 = _theFtheme2.getFtab
  av.run ("addXY2", "")

```

'Determine whether all points should be used as sites

```

theSitesNumber = _theFtab2.getNumRecords
theAnswer = msgBox.yesNo ("Do you want to use all" ++theSitesNumber.asString++ "sites as the logistic
input?", "Keep All Sites", TRUE)

```

'If only a portion of the points desired, get field and categories desired and add appropriate data to the text file

```

j = 1
if (theAnswer <> TRUE) then
  theFieldList = _theFtab2.getFields
  theField = msgBox.listAsString (theFieldList, "Which field contains the information that decides
whether a site will be kept?", "Select Field")
  if (theField <> nil) then
    theList = list.make
    for each i in _theFtab2
      theValue = _theFtab2.returnValueString (theField, i)
      theList.add (theValue)
    end
  end
end

```

APPENDIX B

```

end
theList.removeDuplicates
end
theUsableSites = msgBox.multiListAsString (theList, "Please select the sites to be used in the logistic
input. You may choose multiple sites by depressing the shift key while you make your selection.",
"Desired Categories")
for each v in _theFtab2
  xVal = _theFtab2.returnValueString ((_theFtab2.findField ("X-coord")), v)
  yVal = _theFtab2.returnValueString ((_theFtab2.findField ("Y-coord")), v)
  theField3 = _theFtab2.findField (theField.asString)
  theDecidingField = _theFtab2.returnValueString (theField3, v)
  pt = point.make (xVal.asNumber, yVal.asNumber)
  for each s in theUsableSites
    if (theDecidingField.asString = s.asString) then

'Create a record in the _theVtab2 and add coordinate information
      i = "site"
      recNum = _theVtab2.addRecord
      _theVtab2.setValueString (_theSite2, recNum, i.asString)
      _theVtab2.setValueString (_xCoord, recNum, xVal.asString)
      _theVtab2.setValueString (_yCoord, recNum, yVal.asString)

'Determine whether the point falls within the query area and add field information
      aSrcName = _dirPath + "\" + _theQueryFile.asString
      theSrcName = grid.makeSrcName (aSrcName.asString)
      theGrid = grid.make (theSrcName)
      theGtheme = gTheme.make (theGrid)
      num = theGtheme.returnCellValue (pt)
      theVtab = theGtheme.getVtab
      theField2 = "Site"
      theFieldName = _theVtab2.findField (theField2.asString)
      _theVtab2.setValueString (theFieldName, recNum, num.asString)
    end
  end
end
else

'Use all points as sites
  for each v in _theFtab2
    xVal = _theFtab2.returnValueString ((_theFtab2.findField ("X-coord")), v)
    yVal = _theFtab2.returnValueString ((_theFtab2.findField ("Y-coord")), v)
    pt = point.make (xVal.asNumber, yVal.asNumber)

'Create a record in the _theVtab2 and add coordinate information
    i = "site"
    recNum = _theVtab2.addRecord
    _theVtab2.setValueString (_theSite2, recNum, i.asString)
    _theVtab2.setValueString (_xCoord, recNum, xVal.asString)
    _theVtab2.setValueString (_yCoord, recNum, yVal.asString)
  end
end

```

APPENDIX B

'Determine whether the point falls within the query area and add field information

```

aSrcName = _dirPath + "\" + _theQueryFile.asString
theSrcName = grid.makeSrcName (aSrcName.asString)
theGrid = grid.make (theSrcName)
theGtheme = gTheme.make (theGrid)
num = theGtheme.returnCellValue (pt)
theVtab = theGtheme.getVtab
theField2 = "Site"
theFieldName = _theVtab2.findField (theField2.asString)
_theVtab2.setValueString (theFieldName, recNum, num.asString)
end
end

```

'Get random points

```

av.run ("randomPts2", "")
else
  exit
end

```

'Create histogram of values for an inductive model surface

```

elseif (theModel = "Inductive Model") then
  theFile = msgBox.input ("What is the name of your surface grid?", "Surface Grid", "surface")
  aSrcName = _dirPath + "\" + theFile.asString
  theSrcName = grid.makeSrcName (aSrcName.asString)
  if (theSrcName <> nil) then
    theGrid = grid.make (theSrcName)
    theGtheme = gTheme.make (theGrid)
    theChart = theGtheme.histogram (TRUE, value, TRUE)
    theChart.getWin.activate
  else
    msgBox.error ("You have entered an invalid surface grid", "Invalid Entry")
  end
end

elseif (theModel = nil) then
  exit
end

```

APPENDIX B

Information Script

'This script runs the Archaeological Site Prediction System banner and opens an information box containing author information.

```
theView = av.getActiveDoc
```

'Open banner

```
file1 = fileName.make ("C:\data\cyprus.tif")  
msgBox.banner (file1, 5, "Archaeology Site Prediction System")
```

'Open Info box, following banner

```
i = 1  
for each i in 1..10000  
  i = i + 1  
end  
msgBox.info ("The Archaeological Site Prediction System was developed by Jessica Dolanski. All  
questions or comments should be directed to her at: dolansk3@pilot.msu.edu", "System Information")
```

APPENDIX B

Logistic Script

'This script alerts the user to what is needed in order to run logistic regression in Arc/Info and to what is needed in order to create the logistic surface in ArcView.

msgBox.Report ("In order to run the logistic regression procedure in Arc/Info, you will need the name of text file you created in logistic input (" + _textFile.asString+ "). To run the logistic aml, you will need to change into the appropriate working directory and open Arc/Info. Type *&run logistic* at the Arc prompt to run the logistic regression statistic. At the completion of the logistic procedure, Arc/Info will produce an output chart of coefficient numbers (coef #) and coefficients (coef). You will need to record these values to create the logistic surface in ArcView.", "Logistic Information")

APPENDIX B

LogisticInput Script

'This script allows the user to create the input file needed to run logistic regression in Arc/Info. This input file can be created through graphical methods, tabular methods, taken from a grid of site locations or from a point file's .PAT.

```
theView = av.getActiveDoc
```

'Logistic regression disclaimer

```
msgBox.info ("Logistic Regression is not currently available for ArcView. If you have access to Arc/Info, you may use Logistic Input to create the text file needed to run the logistic regression statistic in Arc/Info.", "Logistic Disclaimer")
```

'Check to see that the directory path is valid

```
if (_dirPath = nil) then
  _dirPath = msgBox.input ("Please enter the directory path for your data.", "Directory Path", "C:\data")
else
end
```

'Get possible themes and allow user to select the ones to be used as independent variables

```
possibleThemes = list.make
for each t in theView.getThemes
  if (t.is (gTheme)) then
    theGrid = t.getGrid
    theName = theGrid.getSrcName.asString
    possibleThemes.add (theName.asString)
  end
end
_selectedThemes = msgBox.multiListAsString (possibleThemes, "Please select the grid(s) to be used as independent variables in the logistic regression function. You may choose multiple grids by clicking on many grid names.", "Select Grids")
```

'Get the field that contains the logistic information for each grid in _selectedThemes

```
if (_selectedThemes <> nil) then
  _selectedItems = list.make
  for each t in _selectedThemes
    theFieldsAlias = list.make
    theTheme = theView.findTheme (t.asString)
    theVtab = theTheme.getVtab
    theFields = theVtab.getFields
    for each i in theFields
      theAlias = i.getAlias
      if (theAlias = nil) then
        theFieldsAlias.add (i.asString)
      else
        theFieldsAlias.add (theAlias.asString)
      end
    end
  end
end
```

APPENDIX B

```

theFieldsAlias.removeDuplicates
theField = msgBox.listAsString (theFieldsAlias, "Please enter the field name that contains the
binary/ordinal data for the" ++t.asString++ "grid.", "Input Field Name")
if (theField <> nil) then
    _selectedItems.add (theField.asString)
else
    exit
end
end
else
    exit
end

```

'Get a file name for the logistic input file

```

theFile = msgBox.input ("Please enter a name for your logistic input file. You will be using this file in the
Arc/Info logistic aml.", "Logistic Input File Name", "input")
if (theFile <> nil) then
    _textFile = theFile.asString + ".txt"
else
    exit
end

```

'Determine how the user will enter the known locations

```

inputOptions = {"Graphical with the mouse", "Tabular with UTM coordinates", "Taken from the grid that
contains the site data", "Taken from a point coverage that contains site data"}
_theInput = msgBox.listAsString (inputOptions, "Please select a method for entering the known site
locations that will be used as the dependent variables in the logistic regression function.", "Input Method")

```

'Graphical selection of points

```

if (_theInput = "Graphical with the mouse") then
    msgBox.info ("Click on the ""red pointer"" tool on the toolbar to begin entering known site locations.
Press on the ""S"" stop tool on the toolbar to finish entering known site locations.", "Enter Known Points
Graphically")

```

'Tabular entry of coordinates

```

elseif (_theInput = "Tabular with UTM coordinates") then
    av.run ("makeTxt", "")
    enter = "yes"
    while (enter = "yes")
        xCoor = msgBox.input ("Please enter the Easting coordinate for your site.", "Easting Coordinate", "")
        if (xCoor <> nil) then
            yCoor = msgBox.input ("Please enter the Northing coordinate for your site.", "Northing Coordinate",
            "")
            pt = point.make (xcoor.asNumber, ycoor.asNumber)
            newPt = point.make (xCoor.asNumber, (yCoor.asNumber - 180))

```

'Put marker symbol on screen

```

s = basicMarker.make
s.setStyle (#BASICMARKER_STYLE_PATTERN)

```

APPENDIX B

```
s.setColor (color.getBlack)
s.setSize(28)
d = theView.getDisplay
d.beginClip
d.drawPoint (newPt, s)
d.endClip
d.Flush
av.GetProject.SetModified(true)
```

'Create a string to hold the coordinate information

```
i = 1
fileString = i.asString++xCoor.asString++ yCoor.asString
```

'Add data into dependent variable fields

```
k = 0
for each t in _selectedThemes
  theGtheme = theView.findTheme (t.asString)
  num = theGtheme.returnCellValue (pt)
  theVtab = theGtheme.getVtab
  theField = _selectedItems.get (k)
  if (theField.asString = "Value") then
    fileString = fileString++num.asString
    k = k + 1
  else
    theField2 = theVtab.findField (theField.asString)
    newNum = num - 1
    theValue = theVtab.returnValueString (theField2, newNum)
    fileString = fileString++theValue.asString
    k = k + 1
  end
end
```

'Add fileString to the text file

```
_theFile2.writeELT (fileString)
else
  enter = "no"
end
end
```

'Get random points

```
av.run ("stop", "")
```

'Take points from a grid coverage

```
elseif (_theInput = "Taken from the grid that contains the site data") then
  av.run ("makeTxt", "")
```

'Get site file name and make point coverage from grid

```
if (_site2 <> nil) then
  theFile = _site2.asString
```

APPENDIX B

```

elseif (_site <> nil) then
  theFile = _site.asString
else
  theFile = "Sites"
end
theSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
theFile.asString)
if (theSiteFile <> nil) then
  theGtheme = theView.findTheme (theSiteFile)
  if (theGtheme.is (Gtheme).not) then
    theGtheme = nil
  end
  if (theGtheme = nil) then
    theAnswer = msgBox.yesNo ("You have entered an invalid grid name. Would you like to re-enter the
Site grid?", "Invalid Entry", TRUE)
    if (theAnswer = true) then
      while (theGtheme = nil)
        aSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
theFile.asString)
        if (aSiteFile = nil) then
          exit
        else
          theGtheme = theView.findTheme (aSiteFile)
        end
      end
    else
      exit
    end
  end
else
  exit
end
theFtab = theGtheme.exportToFtab ("Cover".asFileName)
_theFtheme2 = Ftheme.make (theFtab)
_theFtab2 = _theFtheme2.getFtab
av.run ("addXY", "")

```

'Determine whether all points should be used

```

j = 1
theVtab = theGtheme.getVtab
theSitesNumber = theVtab.getNumRecords
theAnswer = msgBox.yesNo ("Do you want to use all" ++theSitesNumber.asString++ "sites as the logistic
input?", "Keep All Sites", TRUE)

```

'Use subset of data, get appropriate information, and add it to the text file

```

if (theAnswer <> TRUE) then
  theFieldList = theVtab.getFields
  theField = msgBox.listAsString (theFieldList, "Which field contains the information that decides
whether a site will be kept?", "Select Field")

```

APPENDIX B

```

if (theField <> nil) then
  theList = list.make
  for each i in theVtab
    theValue = theVtab.returnValueString (theField, i)
    theList.add (theValue)
  end
  theList.removeDuplicates
end
theUsableSites = msgBox.multiListAsString (theList, "Please select the sites to be used in the logistic
input. You may choose multiple sites by depressing the shift key while you make your selection.",
"Desired Categories")
thePointList = list.make
for each i in _theFtab2
  xVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("X-coord")), i)
  yVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("Y-coord")), i)
  theGridcode = _theFtab2.returnValueNumber ((_theFtab2.findField ("Gridcode")), i)
  pt = point.make (xVal, yVal)
  xVal2 = xVal
  yVal2 = yVal
  xVal2.setFormat ("ddddddd")
  yVal2.setFormat ("ddddddd")
  theNum = theGridcode - 1
  theNewField2 = theVtab.findField (theField.asString)
  theDecidingField = theVtab.returnValueString (theNewField2, theNum)
  for each s in theUsableSites
    if (theDecidingField.asString = s.asString) then
      fileString = j.asString++xVal2.asString++yVal2.asString
      k = 0
      l = 1
      for each t in _selectedThemes
        theGtheme = theView.findTheme (t.asString)
        num = theGtheme.returnCellValue (pt)
        theGthemeVtab = theGtheme.getVtab
        theVtabField = _selectedItems.get (k)
        if (theVtabField.asString = "Value") then
          fileString = fileString++num.asString
          k = k + 1
        else
          theVtabField2 = theGthemeVtab.findField (theVtabField.asString)
          newNum = num - 1
          theValue = theGthemeVtab.returnValueString (theVtabField2, newNum)
          fileString = fileString++theValue.asString
          k = k + 1
        end
      end
      _theFile2.writeELT (fileString)
    end
  end
end
end
end

```

APPENDIX B

'Use all data, get appropriate information, and write it to text file

```

else
  for each i in _theFtab2
    xVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("X-coord")), i)
    yVal = _theFtab2.returnValueNumber ((_theFtab2.findField ("Y-coord")), i)
    pt = point.make (xVal, yVal)
    xVal2 = xVal
    yVal2 = yVal
    xVal2.setFormat ("ddddddd")
    yVal2.setFormat ("ddddddd")
    fileString = j.asString++xVal2.asString++yVal2.asString
    k = 0
    l = 1
    for each t in _selectedThemes
      theGtheme = theView.findTheme (t.asString)
      theGthemeVtab = theGtheme.getVtab
      num = theGtheme.returnCellValue (pt)
      theVtabField = _selectedItems.get (k)
      if (theVtabField.asString = "Value") then
        fileString = fileString++num.asString
        k = k + 1
      else
        theVtabField2 = theGthemeVtab.findField (theVtabField.asString)
        newNum = num - 1
        theValue = theGthemeVtab.returnValueString (theVtabField2, newNum)
        fileString = fileString++theValue.asString
        k = k + 1
      end
    end
    _theFile2.writeELT (fileString)
  end
end
end

```

'Get random points

```
av.run ("stop", "")
```

'Take points from a site coverage

```

elseif (_theInput = "Taken from a point coverage that contains site data") then
  av.run ("makeTxt", "")

```

'Get the site file

```

theSiteFile = msgBox.input ("What is the name of the coverage that contains your sites?", "Site Grid",
"Sites")
if (theSiteFile <> nil) then
  _theFtheme2 = theView.findTheme (theSiteFile)
  if (_theFtheme2.is (Ftheme).not) then
    _theFtheme2 = nil
  end

```

APPENDIX B

```

if (_theFtheme2 = nil) then
  theAnswer = msgBox.yesNo ("You have entered an invalid grid name. Would you like to re-enter the
Site grid?", "Invalid Entry", TRUE)
  if (theAnswer = TRUE) then
    while (_theFtheme2 = nil)
      aSiteFile = msgBox.input ("What is the name of the grid that contains your sites?", "Site Grid",
"Sites")
      if (aSiteFile = nil) then
        exit
      else
        _theFtheme2 = theView.findTheme (aSiteFile)
      end
    end
  else
    exit
  end
end
else
  exit
end
end
_theFtab2 = _theFtheme2.getFtab
av.run ("addXY2", "")

```

'Determine whether all points should be used as sites

```

theSitesNumber = _theFtab2.getNumRecords
theAnswer = msgBox.yesNo ("Do you want to use all" ++theSitesNumber.asString++ "sites as the logistic
input?", "Keep All Sites", TRUE)

```

'Use subset of data, get field and categories desired and add appropriate data to the text file

```

j = 1
if (theAnswer <> TRUE) then
  theFieldList = _theFtab2.getFields
  theField = msgBox.listAsString (theFieldList, "Which field contains the information that decides
whether a site will be kept?", "Select Field")
  if (theField <> nil) then
    theList = list.make
    for each i in _theFtab2
      theValue = _theFtab2.returnValueString (theField, i)
      theList.add (theValue)
    end
    theList.removeDuplicates
  end
  theUsableSites = msgBox.multiListAsString (theList, "Please select the sites to be used in the logistic
input. You may choose multiple sites by depressing the shift key while you make your selection.",
"Desired Categories")
  thePointList = list.make
  for each i in _theFtab2
    xVal = _theFtab2.returnValueString ((_theFtab2.findField ("X-coord")), i)
    yVal = _theFtab2.returnValueString ((_theFtab2.findField ("Y-coord")), i)

```

APPENDIX B

```

theField2 = _theFtab2.findField (theField.asString)
theDecidingField = _theFtab2.returnValueString (theField2, i)
pt = point.make (xVal.asNumber, yVal.asNumber)
for each s in theUsableSites
  if (theDecidingField.asString = s.asString) then
    fileString = j.asString++xVal.asString++yVal.asString
    k = 0
    l = 1
    for each t in _selectedThemes
      theGtheme = theView.findTheme (t.asString)
      theGthemeVtab = theGtheme.getVtab
      num = theGtheme.returnValue (pt)
      theVtabField = _selectedItems.get (k)
      if (theVtabField.asString = "Value") then
        fileString = fileString++num.asString
        k = k + 1
      else
        theVtabField2 = theGthemeVtab.findField (theVtabField.asString)
        newNum = num - 1
        theValue = theGthemeVtab.returnValueString (theVtabField2, newNum)
        fileString = fileString++theValue.asString
        k = k + 1
      end
    end
  end
  _theFile2.writeELT (fileString)
end
end
end

```

'Use all data, get appropriate information, and add it to text file

```

else
  for each i in _theFtab2
    xVal = _theFtab2.returnValueString ((_theFtab2.findField ("X-coord")), i)
    yVal = _theFtab2.returnValueString ((_theFtab2.findField ("Y-coord")), i)
    theField2 = _theFtab2.findField (theField.asString)
    pt = point.make (xVal.asNumber, yVal.asNumber)
    fileString = j.asString++xVal.asString++yVal.asString
    k = 0
    l = 1
    for each t in _selectedThemes
      theGtheme = theView.findTheme (t.asString)
      theGthemeVtab = theGtheme.getVtab
      num = theGtheme.returnValue (pt)
      theVtabField = _selectedItems.get (k)
      if (theVtabField.asString = "Value") then
        fileString = fileString++num.asString
        k = k + 1
      else
        theVtabField2 = theGthemeVtab.findField (theVtabField.asString)

```

APPENDIX B

```
    newNum = num - 1
    theValue = theGthemeVtab.returnValueString (theVtabField2, newNum)
    fileString = fileString++theValue.asString
    k = k + 1
  end
end
  _theFile2.writeELT (fileString)
end
end

'Get random points
av.run ("stop", "")

else
  exit
end
```

APPENDIX B

LogisticSurface Script

'This script creates the logistic surface using the coefficients obtained from the logistic regression procedure in Arc/Info.

```
theView = av.getActiveDoc
numGrids = _selectedThemes.count
i = 1
```

'Get first coefficient

```
theCoef = msgBox.input ("Please enter the constant value (coef #0).", "Constant Coefficient", "")
if (theCoef = nil) then
  exit
end
```

'Get remaining coefficients and make surface

```
theEquation = nil
for each t in _selectedThemes
  if ( _dirpath <> nil) then
    aSrcName = _dirPath + "\" + t.asString
  else
    _DirPath = msgBox.input ("What is the directory path for your grid data?", "Directory Path", "C:\data")
    aSrcName = _DirPath.asString + "\" + t.asString
  end
  theSrcName = grid.makeSrcName (aSrcName.asString)
  theGrid = grid.make (theSrcName)
  coef = msgBox.input ("Please enter the coefficient for the" ++t.asString++ "grid. It should be coef #"
+i.asString++ ".", "Input Coefficients", "")
  if (coef <> nil) then
    if (i = 1) then
      theEquation = theGrid * coef.asNumber
    elseif (i <> 1) then
      theEquation = theEquation + (theGrid * coef.asNumber)
    end
    i = i + 1
  elseif (coef = nil) then
    exit
  end
end
theGrids = theEquation * -1
theSurface = 1.asGrid / ((theGrids.exp) + 1)
theGtheme = gTheme.make (theSurface)
for each t in theView.getThemes
  t.setActive (False)
end
```

'Convert the surface to permanent grid

```
_themeName = "Surface"
```

APPENDIX B

```
theView.addTheme (theGtheme)  
theGtheme.setActive (True)  
av.run ("makeGrid2", "")  
theView.deleteTheme (theGtheme)
```

APPENDIX B

MakeBinary Script

'This script assists the user in creating binary (1/0) fields for categorical data. This must be run prior to using categorical data in the logistic regression function.

```
theView = av.getActiveDoc
theVtab = nil
theVtab2 = nil
```

'Check for directory path

```
if (_dirPath = nil) then
  _dirPath = msgBox.input ("Please enter the directory path for your data", "Directory Path", "C:\data")
end
```

'Get available themes and add to a list

```
theThemesList = list.make
for each t in theView.getThemes
  if (t.is (gTheme)) then
    theGrid = t.getGrid
    theSrcName = theGrid.getSrcName.asString
    theThemesList.add (theSrcName)
  end
end
```

'Get the categorical grid and its Vtab

```
theData = msgBox.listAsString (theThemesList, "Please select the grid that contains categorical data to be
converted to binary data.", "Select Grid")
if (theData = nil) then
  exit
else
  theGtheme = theView.findTheme (theData)
  theVtab = theGtheme.getVtab
  theVtab.unjoinAll
end
```

'Get categorical field and method for creating the binary field(s)

```
theFieldList = theVtab.getFields
theField = msgBox.listAsString (theFieldList, "Please select the field that contains the categorical data to
be converted to binary data.", "Select Field")
if (theField = nil) then
  exit
end
theCategories = list.make
for each i in theVtab
  theValue = theVtab.returnValueString (theField, i)
  theCategories.add (theValue)
end
```

APPENDIX B

```

theCategories.removeDuplicates
theOptionsList = {"Create binary field for each category", "Create one binary field for all categorical
data"}
theOption = msgBox.listAsString (theOptionsList, "Please select a method for creating the binary fields.",
"Binary Fields")
theVtab.setEditable (TRUE)

```

'One field for each category

'Make new Vtab to hold binary data, clone the categorical field, and add categorical data

```

if (theOption = "Create binary field for each category") then
  theName = theField.asString
  theVtab2 = Vtab.makeNew (theName.asFileName, dbase)
  theVtab2.setEditable (TRUE)
  theFieldClone = theVtab.findField (theName.asString)
  theFieldClone2 = theFieldClone.clone
  theVtab2.addFields ({theFieldClone2})
  for each k in theCategories
    l = theVtab2.addRecord
    theVtab2.setValueString (theFieldClone2, l, k.asString)
  end

```

'Add field for each category and fill in appropriate binary data

```

for each v in theCategories
  theNewField = field.make (v.asString, #FIELD_BYTE, 1, 0)
  theNewFieldList = {theNewField}
  theVtab2.addFields (theNewFieldList)
  theNewField.setEditable (TRUE)
  for each j in theVtab2
    theField2 = theVtab2.findField (theFieldClone2.asString)
    theCategory = theVtab2.returnValueString (theField2, j)
    if (theCategory.asString = v.asString) then
      theVtab2.setValueNumber (theNewField, j, 1)
    else
      theVtab2.setValueNumber (theNewField, j, 0)
    end
  end
end
theVtab.join (theFieldClone, theVtab2, theFieldClone2)

```

'Multiple categories in each field

'Make Vtab to hold binary data, clone the categorical data field, and add categorical data

```

elseif (theOption = "Create one binary field for all categorical data") then
  theName = theField.asString
  theVtab2 = Vtab.makeNew (theName.asFileName, dbase)
  theVtab2.setEditable (TRUE)
  theFieldClone = theVtab.findField (theName.asString)
  theFieldClone2 = theFieldClone.clone
  theVtab2.addFields ({theFieldClone2})
  for each k in theCategories

```

APPENDIX B

```

l = theVtab2.addRecord
theVtab2.setValueString (theFieldClone2, l, k.asString)
end

```

'Add field to hold binary data and add appropriate data

```

theNewFieldName = "binary" +theName.asString
theNewField = field.make (theNewFieldName.asString, #FIELD_BYTE, 1, 0)
theNewFieldList = ({theNewField})
theVtab2.addFields (theNewFieldList)
theNewField.setEditable (TRUE)
for each v in theCategories
  binary = nil
  while (binary = nil)
    binary = msgBox.input ("What value should be assigned to the" ++v.asString++ "category? (0/1)",
"Binary Input", "0")
    if (binary = "0") then
      for each j in theVtab2
        theField2 = theVtab2.findField (theFieldClone2.asString)
        theCategory = theVtab2.returnValueString (theField2, j)
        if (theCategory.asString = v.asString) then
          theVtab2.setValueNumber (theNewField, j, 0)
        end
      end
    elseif (binary = "1") then
      for each j in theVtab2
        theField2 = theVtab2.findField (theFieldClone2.asString)
        theCategory = theVtab2.returnValueString (theField2, j)
        if (theCategory.asString = v.asString) then
          theVtab2.setValueNumber (theNewField, j, 1)
        end
      end
    else
      msgBox.info ("Your entry is not a binary entry. You must enter a ""1"" or a ""0"". ", "Improper
Input")
      binary = nil
    end
  end
end
theVtab.setEditable (TRUE)
theVtab.join (theFieldClone, theVtab2, theFieldClone2)
else
  exit
end

```

APPENDIX B

MakeGrid Script

'This script converts temporary grids to permanent grids using an ArcView system script. This script is called by the resample command. Appropriate global variables are assigned to the permanent grids.

```
theView = av.GetActiveDoc
theProject = av.getProject
```

'Get the directory path

```
if (_dirPath <> nil) then
  theFileName = _dirPath + "\"
  newDir = theProject.setWorkDir (theFileName.asFileName)
end
```

'Loop through each active theme

```
firstTime = TRUE
for each t in theView.GetActiveThemes
  needDelete = FALSE
  if (firstTime) then
    def = av.getProject.MakeFileName("r" + _srcGridName.asString, "")
  else
    def = FileName.GetCWD.MakeTmp("r" + _srcGridName.asString, "")
  end
```

'Convert selected features of a FTheme to Grid

```
if (t.Is(FTHEME)) then
  anFTab = t.GetFTab
```

'Set extent and cell size for conversion if not already set

```
ae = theView.GetExtension(AnalysisEnvironment)
box = Rect.Make(0@0,1@1)
cellSize = 1
if ((ae.GetExtent(box) <> #ANALYSISENV_VALUE) or (ae.GetCellSize(cellSize) <>
#ANALYSISENV_VALUE)) then
  ce = AnalysisPropertiesDialog.Show(theView, TRUE, "Conversion Extent:" ++ t.GetName)
  if (ce = NIL) then continue end
  ce.GetCellSize(cellSize)
  ce.GetExtent(box)
end
```

'Make list of fields

```
fl = {}
for each f in anFTab.GetFields
  if (f.IsVisible and (f.IsTypeNumber or f.IsTypeString)) then
    fl.Add(f)
  end
end
```

APPENDIX B

```
aField = MsgBox.List(fl, "Pick field for cell values:", "Conversion Field :" ++ t.GetName)
if (aField = NIL) then continue end
```

'Do conversion

```
aPrj = theView.GetProjection
aGrid = Grid.MakeFromFTab(anFTab, aPrj, aField, {cellSize, box})
```

'Convert ITheme to Grid

```
elseif (t.Is(ITHEME)) then
  anImage = t.GetSrcName.GetFileName
  if (ImgCat.IsValidFileName(anImage)) then continue end
  aGrid = Grid.MakeFromImage(anImage, 0)
```

'Convert selected features of a GTheme to Grid

```
elseif (t.Is(GTHEME)) then
  aVTab = t.GetGrid.GetVTab
  if (aVTab = NIL) then
    tmpFile = av.GetProject.GetWorkDir.MakeTmp("ctg", "")
    ok = Grid.CopyDataSet(t.GetGrid.GetSrcName.GetFileName, tmpFile)
    if (ok.Not) then
      MsgBox.Error("Error during conversion", "Convert to Grid : " ++ t.GetName)
      continue
    end
    gridSrc = Grid.MakeSrcName(tmpFile.AsString)
    aGrid = Grid.Make(gridSrc)
    needDelete = TRUE
  else
    if (aVTab.GetNumSelRecords > 0) then
      aGrid = t.GetGrid.ExtractSelection
    else
      tmpFile = av.GetProject.GetWorkDir.MakeTmp("ctg", "")
      ok = Grid.CopyDataSet(t.GetGrid.GetSrcName.GetFileName, tmpFile)
      if (ok.Not) then
        MsgBox.Error("Error during conversion", "Convert to Grid : " ++ t.GetName)
        continue
      end
      gridSrc = Grid.MakeSrcName(tmpFile.AsString)
      aGrid = Grid.Make(gridSrc)
      needDelete = TRUE
    end
  end
else
  continue
end
```

'Obtain output data source name and add it to a view if user wants

```
if (aGrid.Is(LIST)) then
  if (aGrid.Count = 0) then
    MsgBox.Error(t.GetName ++ "could not be converted to a grid", "Conversion Error")
```

APPENDIX B

```

    continue
end
bandNumber = 1
for each band in aGrid
    if (band.HasError) then
        MsgBox.Error("Band" ++ bandNumber ++ "of" ++ t.GetName ++ "could not be converted to a grid",
"Conversion Error")
        bandNumber = bandNumber + 1
        continue
    end
    if (bandNumber > 1) then
        def = FileName.GetCWD.MakeTmp("r" + _srcGridName.asString, "")
    end
    aFN = SourceManager.PutDataSet(GRID, "Convert" ++ t.getName ++ ": Band" ++
bandNumber.AsString ++ "of" ++ aGrid.Count.AsString, def, TRUE)
    if (aFN = NIL) then
        bandNumber = bandNumber + 1
        continue
    end
    status = Grid.GetVerify
    Grid.SetVerify(#GRID_VERIFY_OFF)
    if (band.SaveDataSet(aFN).Not) then
        Grid.SetVerify(status)
        bandNumber = bandNumber + 1
        continue
    end
    Grid.SetVerify(status)
    if (MsgBox.YesNo("Add grid as theme to the view?", "Convert to Grid : " ++ t.GetName, true).Not) then
        bandNumber = bandNumber + 1
        continue
    end
    gthm = GTheme.Make(band)
    theView.AddTheme(gthm)
    theView.GetWin.Activate
    bandNumber = bandNumber + 1
end
else
    if (aGrid.HasError) then
        MsgBox.Error(t.GetName ++ "could not be converted to a grid", "Conversion Error")
        continue
    end
    aFN = SourceManager.PutDataSet(GRID, "Convert " + t.getName ++ "to grid", def, TRUE)
    if (aFN = NIL) then continue end
    if (t.Is(ITHEME)) then
        theSrcName = aGrid.GetSrcName
        theFileName = theSrcName.GetFileName
        oldColorFileName = FileName.Make(theFileName.AsString + ".clr")
        if (oldColorFileName.IsFile) then
            newColorFileName = FileName.Make(aFN.AsString + ".clr")

```

APPENDIX B

```

    File.Copy(oldColorFileName,newColorFileName)
    File.Delete(oldColorFileName)
end
end
status = Grid.GetVerify
Grid.SetVerify(#GRID_VERIFY_OFF)
if (aGrid.SaveDataSet(aFN).Not) then
    Grid.SetVerify(status)
    continue
end
Grid.SetVerify(status)
if (needDelete) then
    Grid.DeleteDataSet(tmpFile)
end
gthm = GTheme.Make(aGrid)
if (t.Is(FTHEME) and (aGrid.GetVTab <> NIL)) then
    theVTab = aGrid.GetVTab
    if (aField.IsTypeNumber) then
        toField = theVTab.FindField("Value")
    else
        toField = theVTab.FindField("S_Value")
        theLegend = gthm.GetLegend
        theLegend.Unique(gthm,"S_Value")
        gthm.UpdateLegend
    end
    if (anFTab.IsBase and anFTab.IsBeingEditedWithRecovery.Not) then
        if (MsgBox.YesNo("Join feature attributes to grid?", "Attribute Join : " ++ t.GetName, FALSE)) then
            theVTab.Join(toField, anFTab, aField)
        end
    end
end
if (MsgBox.YesNo("Add grid as theme to the view?", "Convert to Grid : " ++ t.GetName, true).Not) then
    continue end
theView.AddTheme(gthm)
theView.GetWin.Activate
end
firstTime = FALSE

'Assign appropriate global variables
if (_srcGridName.asString = _elevation.asString) then
    _elevation2 = aGrid.getSrcName
elseif (_srcGridName.asString = _slope.asString) then
    _slope2 = aGrid.getSrcName
elseif (_srcGridName.asString = _aspect.asString) then
    _aspect2 = aGrid.getSrcName
elseif (_srcGridName.asString = _soil.asString) then
    _soil2 = aGrid.getSrcName
elseif (_srcGridName.asString = _grH2O.asString) then
    _grH2O2 = aGrid.getSrcName

```

APPENDIX B

```
elseif (_srcGridName.asString = _geol.asString) then
  _geol2 = aGrid.getSrcName
elseif (_srcGridName.asString = _stream.asString) then
  _stream2 = aGrid.getSrcName
elseif (_srcGridName.asString = _site.asString) then
  _site2 = aGrid.getSrcName
else
  end
end
```

APPENDIX B

MakeGrid2 Script

'This script converts temporary grids to permanent grids using an ArcView system script. This script is called by the distance from, group and area, surface, and convert scripts.

```
theView = av.GetActiveDoc
theProject = av.getProject
```

'Get directory path information

```
if (_dirPath <> nil) then
  theFileName = _dirPath + "\"
  newDir = theProject.setWorkDir (theFileName.asFileName)
end
```

'Determine grid name prefix

```
if (_themeName.asString = "query") then
  name = "query"
elseif (_themeName.asString = "dist") then
  name = "dist"
elseif (_themeName.asString = "group") then
  name = "group"
elseif (_themeName.asString = "area") then
  name = "area"
elseif (_themeName.asString = "surface") then
  name = "surface"
else
  name = "grid"
end
```

'Loop through each active theme

```
firstTime = TRUE
for each t in theView.GetActiveThemes
  needDelete = FALSE
  if (firstTime) then
    def = av.getProject.MakeFileName(name.asString, "")
  else
    def = FileName.GetCWD.MakeTmp(name.asString, "")
  end
```

'Convert selected features of a FTheme to grid

```
if (t.Is(FTHEME)) then
  anFTab = t.GetFTab
```

'Set extent and cell size for conversion if not already set

```
ae = theView.GetExtension(AnalysisEnvironment)
box = Rect.Make(0@0,1@1)
cellSize = 1
```

APPENDIX B

```

    if ((ae.GetExtent(box) <> #ANALYSENV_VALUE) or (ae.GetCellSize(cellSize) <>
#ANALYSENV_VALUE)) then
        ce = AnalysisPropertiesDialog.Show(theView, TRUE, "Conversion Extent:" ++ t.GetName)
        if (ce = NIL) then continue end
        ce.GetCellSize(cellSize)
        ce.GetExtent(box)
    end

```

'Make a list of fields and do conversion

```

    fl = {}
    for each f in anFTab.GetFields
        if (f.IsVisible and (f.IsTypeNumber or f.IsTypeString)) then
            fl.Add(f)
        end
    end
    aField = MsgBox.List(fl, "Pick field for cell values:", "Conversion Field :" ++ t.GetName)
    if (aField = NIL) then continue end
    aPrj = theView.GetProjection
    aGrid = Grid.MakeFromFTab(anFTab, aPrj, aField, {cellSize, box})

```

'Convert ITheme to Grid

```

    elseif (t.Is(ITHEME)) then
        anImage = t.GetSrcName.GetFileName
        if (ImgCat.IsValidFileName(anImage)) then continue end
        aGrid = Grid.MakeFromImage(anImage, 0)

```

'Convert selected features of a GTheme to Grid

```

    elseif (t.Is(GTHEME)) then
        aVTab = t.GetGrid.GetVTab
        if (aVTab = NIL) then
            tmpFile = av.GetProject.GetWorkDir.MakeTmp("ctg", "")
            ok = Grid.CopyDataSet(t.GetGrid.GetSrcName.GetFileName, tmpFile)
            if (ok.Not) then
                MsgBox.Error("Error during conversion", "Convert to Grid : " ++ t.GetName)
                continue
            end
            gridSrc = Grid.MakeSrcName(tmpFile.AsString)
            aGrid = Grid.Make(gridSrc)
            needDelete = TRUE
        else
            if (aVTab.GetNumSelRecords > 0) then
                aGrid = t.GetGrid.ExtractSelection
            else
                tmpFile = av.GetProject.GetWorkDir.MakeTmp("ctg", "")
                ok = Grid.CopyDataSet(t.GetGrid.GetSrcName.GetFileName, tmpFile)
                if (ok.Not) then
                    MsgBox.Error("Error during conversion", "Convert to Grid : " ++ t.GetName)
                    continue
                end
            end

```

APPENDIX B

```

    gridSrc = Grid.MakeSrcName(tmpFile.AsString)
    aGrid = Grid.Make(gridSrc)
    needDelete = TRUE
end
end
else
    continue
end

```

'Obtain output data source name and add it to a view if user wants

```

if (aGrid.Is(LIST)) then
    if (aGrid.Count = 0) then
        MsgBox.Error(t.GetName ++ "could not be converted to a grid", "Conversion Error")
        continue
    end
    bandNumber = 1
    for each band in aGrid
        if (band.HasError) then
            MsgBox.Error("Band" ++ bandNumber ++ "of" ++ t.GetName ++ "could not be converted to a grid",
"Conversion Error")
            bandNumber = bandNumber + 1
            continue
        end
        if (bandNumber > 1) then
            def = FileName.GetCWD.MakeTmp("query","")
        end
        aFN = SourceManager.PutDataSet(GRID,"Convert" ++ t.getName ++ ": Band" ++
bandNumber.AsString ++ "of" ++ aGrid.Count.AsString,def,TRUE)
        if (aFN = NIL) then
            bandNumber = bandNumber + 1
            continue
        end
        status = Grid.GetVerify
        Grid.SetVerify(#GRID_VERIFY_OFF)
        if (band.SaveDataSet(aFN).Not) then
            Grid.SetVerify(status)
            bandNumber = bandNumber + 1
            continue
        end
        Grid.SetVerify(status)
        if (MsgBox.YesNo("Add grid as theme to the view?","Convert to Grid :"+t.GetName,true).Not) then
            bandNumber = bandNumber + 1
            continue
        end
        gthm = GTheme.Make(band)
        theView.AddTheme(gthm)
        gthm.setActive (True)
        theView.GetWin.Activate
        bandNumber = bandNumber + 1
    end
end

```

APPENDIX B

```

end
else
if (aGrid.HasError) then
    MsgBox.Error(t.GetName ++ "could not be converted to a grid", "Conversion Error")
    continue
end
aFN = SourceManager.PutDataSet(GRID,"Convert " + t.getName ++"to grid", def,TRUE)
if (aFN = NIL) then continue end
if (t.Is(ITHEME)) then
    theSrcName = aGrid.GetSrcName
    theFileName = theSrcName.GetFileName
    oldColorFileName = FileName.Make(theFileName.AsString+".clr")
    if (oldColorFileName.IsFile) then
        newColorFileName = FileName.Make(aFN.AsString+".clr")
        File.Copy(oldColorFileName,newColorFileName)
        File.Delete(oldColorFileName)
    end
end
status = Grid.GetVerify
Grid.SetVerify(#GRID_VERIFY_OFF)
if (aGrid.SaveDataSet(aFN).Not) then
    Grid.SetVerify(status)
    continue
end
Grid.SetVerify(status)
if (needDelete) then
    Grid.DeleteDataSet(tmpFile)
end
gthm = GTheme.Make(aGrid)
if (t.Is(FTHEME) and (aGrid.GetVTab <> NIL)) then
    theVTab = aGrid.GetVTab
    if (aField.IsTypeNumber) then
        toField = theVTab.FindField("Value")
    else
        toField = theVTab.FindField("S_Value")
        theLegend = gthm.GetLegend
        theLegend.Unique(gthm,"S_Value")
        gthm.UpdateLegend
    end
    if (anFTab.IsBase and anFTab.IsBeingEditedWithRecovery.Not) then
        if (MsgBox.YesNo("Join feature attributes to grid?", "Attribute Join :" ++ t.GetName, FALSE)) then
            theVTab.Join(toField, anFTab, aField)
        end
    end
end
if (MsgBox.YesNo("Add grid as theme to the view?","Convert to Grid :"+t.GetName,true).Not) then
continue end
theView.AddTheme(gthm)
gthm.setActive (true)

```

APPENDIX B

```
    theView.GetWin.Activate  
end  
firstTime = FALSE  
end
```

APPENDIX B

MakePt Script

This script records the user's graphically selected (x, y) locations for known sites and adds them to a text file and to the view. This script is used for the known locations in the logisticInput command.

```
theView = av.getActiveDoc
```

'Retrieve the user's point

```
pt = theView.getDisplay.returnUserPoint
if (pt <> NIL) then
  xcoor = pt.getX
  ycoor = pt.getY
  newpt = point.make (xcoor, (ycoor - 180))
```

'Put marker symbol on screen

```
s = basicMarker.make
s.setStyle (#BASICMARKER_STYLE_PATTERN)
s.setColor (color.getBlack)
s.setSize(28)
d = theView.getDisplay
d.beginClip
d.drawPoint (newpt, s)
d.endClip
d.Flush
av.GetProject.SetModified(true)
```

'Get the x and y values and set format

```
i = 1
x = xcoor
y = ycoor
x.setFormat ("ddddddd")
y.setFormat ("ddddddd")
```

'Create a string to hold the coordinate information

```
fileString = i.asString++x.asString++ y.asString
else
  exit
end
```

'Add data into dependent variable fields

```
k = 0
p = point.make (xcoor, ycoor)
for each t in _selectedThemes
  theGtheme = theView.findTheme (t.asString)
  num = theGtheme.returnCellValue (p)
  theVtab = theGtheme.getVtab
  theField = _selectedItems.get (k)
  if (theField.asString = "Value") then
```

APPENDIX B

```
    fileString = fileString++num.asString
    k = k + 1
else
    theField2 = theVtab.findField (theField.asString)
    newNum = num - 1
    theValue = theVtab.returnValueString (theField2, newNum)
    fileString = fileString++theValue.asString
    k = k + 1
end
end

_theFile2.writeELT (fileString)
```

APPENDIX B

MakePt2 Script

'This script records the user's graphically selected (x, y) locations for known sites and adds them to a table and to the view. This script is used for known sites in the histogram command.

```
theView = av.getActiveDoc
```

```
'Retrieve the user's point
pt = theView.getDisplay.returnUserPoint
if (pt <> NIL) then
  xCoor = pt.getX
  yCoor = pt.getY
  newpt = point.make (xCoor, (yCoor - 200))
```

```
'Put marker symbol on screen
s = basicMarker.make
s.setStyle (#BASICMARKER_STYLE_PATTERN)
s.setColor (color.getBlack)
s.setSize(28)
d = theView.getDisplay
d.beginClip
d.drawPoint (newpt, s)
d.endClip
d.Flush
av.GetProject.SetModified(true)
```

```
'Get the x and y values and sets format
i = "site"
x = xCoor
y = yCoor
x.setFormat ("dddddd")
y.setFormat ("dddddd")
```

```
'Create a record in theVtab2 and add coordinate information
recNum = _theVtab2.addRecord
_theVtab2.setValueString (_theSite2, recNum, i.asString)
_theVtab2.setValueString (_xCoord, recNum, x.asString)
_theVtab2.setValueString (_yCoord, recNum, y.asString)
```

```
'Determine whether the point falls within the query area and add information to theVtab2
aSrcName = _dirPath + "\" + _theQueryFile.asString
theSrcName = grid.makeSrcName (aSrcName.asString)
theGrid = grid.make (theSrcName)
theGtheme = gTheme.make (theGrid)
num = theGtheme.returnCellValue (pt)
theVtab = theGtheme.getVtab
theField = "Site"
theFieldName = _theVtab2.findField (theField.asString)
```

APPENDIX B

```
_theVtab2.setValueString (theFieldName, recNum, num.asString)
```

APPENDIX B

MakeTxt Script

'This script creates the text file named _textFile used to hold the logistic input data. It rewrites any previously created logistic input text file with the same name.

'Set up the file

```

if (_dirPath <> nil) then
  theTextFile = _dirPath + "\" + _textFile.asString
  theFileName = (theTextFile.asFileName)
  _theFile2 = lineFile.make (theFileName, #FILE_PERM_WRITE)
else
  _dirPath = msgBox.input ("Please enter the directory path for your data.", "Directory Path", "C:\data")
  if (_dirPath <> nil) then
    theTextFile = _dirPath + "\" + _textFile.asString
    theFileName = (theTextFile.asFileName)
    _theFile2 = lineFile.make (theFileName, #FILE_PERM_WRITE)
  else
    msgBox.error ("You have entered an invalid directory path", "Invalid Entry")
    exit
  end
end
end

```

APPENDIX B

MakeVtab Script

'This script creates the table, named _theVtab2, used to hold the evaluate by histogram data. It rewrites any previously created evaluate by histogram table.

'Make Vtab and add fields

```
theTable = ("histo.dbf".asFileName)
_theVtab2 = vTab.makeNew (theTable,Dbase )
_theVtab2.setEditable (TRUE)
_theSite2 = field.make ("Sitetype", #FIELD_CHAR, 6, 0)
_xCoord = field.make ("xCoord", #FIELD_LONG, 7, 0)
_yCoord = field.make ("yCoord", #FIELD_LONG, 8, 0)
_site = field.make ("Site", #FIELD_LONG, 2, 0)
_random = field.make ("Random", #FIELD_LONG, 2, 0)
_theVtab2.addFields ({_theSite2, _xCoord, _yCoord, _site, _random})
```

APPENDIX B

Query Script

'This script opens ArcView's map calculator dialog box to allow queries to be run.

```
theView = av.GetActiveDoc
```

```
msgBox.info ("If you need help in building your visual model, use Assisted Modeling.", "Query")
```

```
msgBox.info ("You will need to run Convert to Grid on your query theme before you run Group and  
Area.", "Query")
```

```
GTheme.MakeQueryMap (theView, "Map Query ")
```

APPENDIX B

QueryAssist Script

'This script assists the user in creating a query expression that can be run by the query command.

```
theView=av.getActiveDoc
expr1 = nil
expr4 = nil
expr7 = nil
expr8 = nil
expr10 = nil
expr12 = nil
expression = nil
j = 15
k = 50
```

'Get theme to add to query

```
exprList = list.make
i = 1
While (msgBox.yesNo("Would you like to add a theme to your query?", "Add Theme", true))
  themeList = list.make
  for each t in theView.getThemes
    if (t.is (gTheme)) then
      theGrid = t.getGrid
      theName = theGrid.getSrcName.asString
      themeList.add (theName.asString)
    end
  end
  theTheme = msgBox.listAsString(themeList, "Select a theme to add to your query", "Possible Themes")
```

'Design queries based on the global variables for the specific grids

'Query for the elevation theme

```
if (theTheme = _elevation.asString) then
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  theFieldCount = fieldList.count
  if (theFieldCount <> 2) then
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
      exit
    end
  else
    theField = "Value"
  end
  optionsList = {"Greater Than", "Less Than", "Between Two Values"}
```

APPENDIX B

```

    greaterlesser = msgBox.ChoiceAsString(optionsList, "In your query, will you be specifying elevations
greater than or less than a particular value, or in between two specific values?", "Greater/Lesser Box")
    if (greaterlesser = "Greater Than") then
        greater = msgBox.input ("Please enter the minimum elevation that you desire for your query.", "Greater
Box", "200")
        if (greater <> nil) then
            expr1 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + greater.asString+ ")"
        elseif (greater = nil) then
            exit
        end
    elseif (greaterlesser = "Less Than") then
        lesser = msgBox.input ("Please enter the maximum elevation that you desire for your query.", "Lesser
Box", "250")
        if (lesser <> nil) then
            expr1 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + lesser.asString+ ")"
        elseif (lesser = nil) then
            exit
        end
    elseif (greaterlesser = "Between Two Values") then
        greater = msgBox.input ("Please enter the minimum elevation that you desire for your query.", "Greater
Box", "200")
        if (greater <> nil) then
            expr2 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + greater.asString+ ")"
        elseif (greater = nil) then
            exit
        end
        lesser = msgBox.input ("Please enter the maximum elevation that you desire for your query.", "Lesser
Box", "250")
        if (lesser <> nil) then
            expr3 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + lesser.asString+ ")"
        elseif (lesser = nil) then
            exit
        end
        expr1 = "(" + expr2.asString++ "and" ++ expr3.asString+ ")"
    elseif (greaterlesser = nil) then
        end
    if (expr1.asString <> nil) then
        exprList.add (expr1.asString)
    else
        end
end

```

'Query for the resampled elevation theme

```

elseif (theTheme = _elevation2.asString) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)
    theVtab = theGrid.getVtab
    fieldList = theVtab.getFields
    theFieldCount = fieldList.count
    if (theFieldCount <> 2) then

```

APPENDIX B

```

    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
        exit
    end
else
    theField = "Value"
end
optionsList = {"Greater Than", "Less Than", "Between Two Values"}
greaterLesser = msgBox.ChoiceAsString(optionsList, "In your query, will you be specifying elevations
greater than or less than a particular value, or in between two specific values?", "Greater/Lesser Box")
if (greaterLesser = "Greater Than") then
    greater = msgBox.input ("Please enter the minimum elevation that you desire for your query.", "Greater
Box", "200")
    if (greater <> nil) then
        expr1 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + greater.asString+ ")"
    elseif (greater = nil) then
        exit
    end
elseif (greaterLesser = "Less Than") then
    lesser = msgBox.input ("Please enter the maximum elevation that you desire for your query.", "Lesser
Box", "250")
    if (lesser <> nil) then
        expr1 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + lesser.asString+ ")"
    elseif (lesser = nil) then
        exit
    end
elseif (greaterLesser = "Between Two Values") then
    greater = msgBox.input ("Please enter the minimum elevation that you desire for your query.", "Greater
Box", "200")
    if (greater <> nil) then
        expr2 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + greater.asString+ ")"
    elseif (greater = nil) then
        exit
    end
    lesser = msgBox.input ("Please enter the maximum elevation that you desire for your query.", "Lesser
Box", "250")
    if (lesser <> nil) then
        expr3 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + lesser.asString+ ")"
    elseif (lesser = nil) then
        exit
    end
    expr1 = "(" + expr2.asString++ "and" ++ expr3.asString+ ")"
elseif (greaterLesser = nil) then
end
if (expr1.asString <> nil) then
    exprList.add (expr1.asString)
else
end

```

APPENDIX B

'Query for the slope theme

```

elseif (theTheme = _slope.asString) then
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  theFieldCount = fieldList.count
  if (theFieldCount <> 2) then
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
      exit
    end
  else
    theField = "Value"
  end
  optionsList = {"Greater Than", "Less Than"}
  greaterLesser = msgBox.ChoiceAsString(optionsList, "In your query, will you be specifying slopes
greater than or less than a particular value?", "Greater/Lesser Box")
  if (greaterLesser = "Greater Than") then
    greater = msgBox.input ("Please enter the minimum slope, that you desire for your query.", "Greater
Box", "5")
    if (greater <> nil) then
      expr4 = "(" + theSrcName.asString++ "." ++ theField.asString+ " > " + greater.asString+ ")"
    elseif (greater = nil) then
      exit
    end
  elseif (greaterLesser = "Less Than") then
    lesser = msgBox.input ("Please enter the maximum slope, that you desire for your query.", "Lesser
Box", "10")
    if (lesser <> nil) then
      expr4 = "(" + theSrcName.asString++ "." ++ theField.asString+ " < " + lesser.asString+ ")"
    elseif (lesser = nil) then
      exit
    end
  end
  if (expr4.asString <> nil) then
    exprList.add (expr4.asString)
  else
    end

```

'Query for the resampled slope theme

```

elseif (theTheme = _slope2.asString) then
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  theFieldCount = fieldList.count

```

APPENDIX B

```

if (theFieldCount < 2) then
  theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
  if (theField = nil) then
    exit
  end
else
  theField = "Value"
end
optionsList = {"Greater Than", "Less Than"}
greaterLesser = msgBox.ChoiceAsString(optionsList, "In your query, will you be specifying slopes
greater than or less than a particular value?", "Greater/Lesser Box")
if (greaterLesser = "Greater Than") then
  greater = msgBox.input ("Please enter the minimum slope, that you desire for your query.", "Greater
Box", "5")
  if (greater < nil) then
    expr4 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + greater.asString+ ")"
  elseif (greater = nil) then
    exit
  end
elseif (greaterLesser = "Less Than") then
  lesser = msgBox.input ("Please enter the maximum slope, that you desire for your query.", "Lesser
Box", "10")
  if (lesser < nil) then
    expr4 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + lesser.asString+ ")"
  elseif (lesser = nil) then
    exit
  end
end
if (expr4.asString < nil) then
  exprList.add (expr4.asString)
else
  end
end

```

'Query for the aspect theme

```

elseif (theTheme = _aspect.asString) then
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  theFieldCount = fieldList.count
  if (theFieldCount < 2) then
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
      exit
    end
  else
    theField = "Value"
  end
end

```

APPENDIX B

```

end
aspect1 = msgBox.input("Please enter the minimum degree for your aspect query. North is 0 or 360
degrees.", "Aspect1 Box", "170")
if (aspect1 <> nil) then
    expr5 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + aspect1.asString + ")"
elseif (aspect1 = nil) then
    exit
end
aspect2 = msgBox.input("Please enter the maximum degree for your aspect query. North is 0 or 360
degrees.", "Aspect1 Box", "190")
if (aspect2 <> nil) then
    expr6 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + aspect2.asString+ ")"
elseif (aspect2 = nil) then
    exit
end
expr7 = "(" + expr5.asString++ "and" ++ expr6.asString+ ")"
if (expr7.asString <> nil) then
    exprList.add (expr7.asString)
else
end

```

'Query for the resampled aspect theme

```

elseif (theTheme = _aspect2.asString) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)
    theVtab = theGrid.getVtab
    fieldList = theVtab.getFields
    theFieldCount = fieldList.count
    if (theFieldCount <> 2) then
        theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
        if (theField = nil) then
            exit
        end
    else
        theField = "Value"
    end
    aspect1 = msgBox.input("Please enter the minimum degree for your aspect query. North is 0 or 360
degrees.", "Aspect1 Box", "170")
    if (aspect1 <> nil) then
        expr5 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" > " + aspect1.asString + ")"
    elseif (aspect1 = nil) then
        exit
    end
    aspect2 = msgBox.input("Please enter the maximum degree for your aspect query. North is 0 or 360
degrees.", "Aspect1 Box", "190")
    if (aspect2 <> nil) then
        expr6 = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" < " + aspect2.asString+ ")"
    elseif (aspect2 = nil) then

```

APPENDIX B

```

    exit
end
expr7 = "(" + expr5.asString++ "and" ++expr6.asString+ ")"
if (expr7.asString <> nil) then
    exprList.add (expr7.asString)
else
end

```

'Query for the soil theme

```

elseif (theTheme = _soil.asString ) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)
    theVtab = theGrid.getVtab
    fieldList = theVtab.getFields
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
        exit
    end
    numRecs = theVtab.getNumRecords
    numRecs2 = numRecs - 1
    optionsList = list.make
    for each r in 0..numRecs2
        option = theVtab.returnValueString (theField, r)
        optionsList.add (option)
    end
    optionsList.removeDuplicates
    soils = msgBox.multiListAsString (optionsList, "Please select the desired soil type. You may select
more than one soil by holding the shift key down while making your selection.", "Soils Box")
    if (soils = nil) then
        exit
    end
    for each s in soils
        expr9 = "(" + theSrcName.asString++ "." ++theField.asString+ "]" = "+s.asString.quote+ ")"
        if (expr8 = nil) then
            expr8 = expr9.asString
        else
            expr8 = expr9.asString++ "and" ++expr8.asString
        end
    end
    if (expr8.asString <> nil) then
        exprList.add (expr8.asString)
    else
    end
end

```

'Query for the resampled soil theme

```

elseif (theTheme = _soil2.asString ) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)

```

APPENDIX B

```

theVtab = theGrid.getVtab
fieldList = theVtab.getFields
theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
if (theField = nil) then
    exit
end
numRecs = theVtab.getNumRecords
numRecs2 = numRecs - 1
optionsList = list.make
for each r in 0..numRecs2
    option = theVtab.returnValueString (theField, r)
    optionsList.add (option)
end
optionsList.removeDuplicates
soils = msgBox.multiListAsString(optionsList, "Please select the desired soil type. You may select more
than one soil by holding the shift key down while making your selection.", "Soils Box")
if (soils = nil) then
    exit
end
for each s in soils
    expr9 = "[" + theSrcName.asString++ "." ++ theField.asString++ "]" = "+s.asString.quote+ ")"
    if (expr8 = nil) then
        expr8 = expr9.asString
    else
        expr8 = expr9.asString++ "and" ++ expr8.asString
    end
end
if (expr8.asString <> nil) then
    exprList.add (expr8.asString)
else
    end

```

'Query for the geology theme

```

elseif (theTheme = _geol.asString) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)
    theVtab = theGrid.getVtab
    fieldList = theVtab.getFields
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
        exit
    end
    numRecs = theVtab.getNumRecords
    numRecs2 = numRecs - 1
    optionsList = list.make
    for each r in 0..numRecs2
        option = theVtab.returnValueString (theField, r)

```

APPENDIX B

```

    optionsList.add (option)
end
optionsList.removeDuplicates
geol = msgBox.multiListAsString(optionsList, "Please select the desired geology type. You may select
more than one geology type by holding the shift key down while making your selection.", "Geology Box")
if (geol = nil) then
    exit
end
for each g in geol
    expr11 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" = " + g.asString.quote+ ")"
    if (expr10 = nil) then
        expr10 = expr11.asString
    else
        expr10 = expr11.asString++ "and" ++ expr10.asString
    end
end
if (expr10.asString <> nil) then
    exprList.add (expr10.asString)
else
end

```

'Query for the resampled geology theme

```

elseif (theTheme = _geol2.asString) then
    theSrcName = grid.makeSrcName (theTheme.asString)
    theGrid = grid.make (theSrcName)
    theVtab = theGrid.getVtab
    fieldList = theVtab.getFields
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
        exit
    end
    numRecs = theVtab.getNumRecords
    numRecs2 = numRecs - 1
    optionsList = list.make
    for each r in 0..numRecs2
        option = theVtab.returnValueString (theField, r)
        optionsList.add (option)
    end
    optionsList.removeDuplicates
    geol = msgBox.multiListAsString(optionsList, "Please select the desired geology type. You may select
more than one geology type by holding the shift key down while making your selection.", "Geology Box")
    if (geol = nil) then
        exit
    end
    for each g in geol
        expr11 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" = " + g.asString.quote+ ")"
        if (expr10 = nil) then
            expr10 = expr11.asString

```

APPENDIX B

```

else
  expr10 = expr11.asString++ "and" ++expr10.asString
end
end
if (expr10.asString <> nil) then
  exprList.add (expr10.asString)
else
  end
end

```

'Query for the groundwater theme

```

elseif (theTheme = _grH2O.asString) then
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
  if (theField = nil) then
    exit
  end
  numRecs = theVtab.getNumRecords
  numRecs2 = numRecs - 1
  optionsList = list.make
  for each r in 0..numRecs2
    option = theVtab.returnValueString (theField, r)
    optionsList.add (option)
  end
  optionsList.removeDuplicates
  grH2O = msgBox.multiListAsString(optionsList, "Please select the desired groundwater type. You may
select more than one groundwater type by holding the shift key down while making your selection.",
"Groundwater Box")
  if (grH2O = nil) then
    exit
  end
  for each gr in grH2O
    expr13 = "[" +theSrcName.asString++ "." ++theField.asString+ "]" = " +gr.asString.quote+ ")"
  if (expr12 = nil) then
    expr12 = expr11.asString
  else
    expr12 = expr13.asString++ "and" ++expr12.asString
  end
end
if (expr12.asString <> nil) then
  exprList.add (expr12.asString)
else
  end
end

```

'Query for the resampled groundwater theme

```

elseif (theTheme = _grH2O2.asString) then

```

APPENDIX B

```

theSrcName = grid.makeSrcName (theTheme.asString)
theGrid = grid.make (theSrcName)
theVtab = theGrid.getVtab
fieldList = theVtab.getFields
theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
if (theField = nil) then
  exit
end
numRecs = theVtab.getNumRecords
numRecs2 = numRecs - 1
optionsList = list.make
for each r in 0..numRecs2
  option = theVtab.returnValueString (theField, r)
  optionsList.add (option)
end
optionsList.removeDuplicates
grH2O = msgBox.multiListAsString(optionsList, "Please select the desired groundwater type. You may
select more than one groundwater type by holding the shift key down while making your selection.",
"Groundwater Box")
if (grH2O = nil) then
  exit
end
for each gr in grH2O
  expr13 = "[" + theSrcName.asString++ "." ++ theField.asString+ "]" = "+gr.asString.quote+ ")"
  if (expr12 = nil) then
    expr12 = expr11.asString
  else
    expr12 = expr13.asString++ "and" ++ expr12.asString
  end
end
if (expr12.asString <> nil) then
  exprList.add (expr12.asString)
else
  end
elseif (theTheme = nil) then
  exit

```

'Query for a user defined theme

'Determine whether the theme is categorical data or numerical data

```

else
  theSrcName = grid.makeSrcName (theTheme.asString)
  theGrid = grid.make (theSrcName)
  theVtab = theGrid.getVtab
  fieldList = theVtab.getFields
  dataType = {"Categorical", "Numerical"}
  theData = msgBox.listAsString (dataType, "Are you querying categorical or numerical data?", "Data
Types")

```

APPENDIX B

'Query for categorical data

```

if (theData = "Categorical") then
  fieldList = theVtab.getFieldList
  theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
  if (theField = nil) then
    exit
  end
  numRecs = theVtab.getNumRecords
  numRecs2 = numRecs - 1
  optionsList = list.make
  for each r in 0..numRecs2
    option = theVtab.returnValueString (theField, r)
    optionsList.add (option)
  end
  optionsList.removeDuplicates
  theQuery = msgBox.multiListAsString(optionsList, "Please select the desired"++ theTheme.asString++
"type. You may select more than one type by holding the shift key down while making your selection.",
"Query Box")
  if (theQuery = nil) then
    exit
  end
  k = 1
  theExpr = nil
  expr = nil
  for each x in theQuery
    'expr = "expr" + k.asString
    expr = "(" + theSrcName.asString++ "." ++ theField.asString+ "]" = "+x.asString.quote+ ")"
    if (k = 1) then
      theExpr = expr.asString
    else
      theExpr = expr.asString++ "and" ++ theExpr.asString
    end
    k = k + 1
  end
  if (theExpr.asString <> nil) then
    exprList.add (theExpr.asString)
  end
end

```

'Query for the numerical data

```

elseif (theData = "Numerical") then
  theFieldCount = fieldList.count
  if (theFieldCount <> 2) then
    theField = msgBox.listAsString (fieldList, "Select the field that contains the data you would like to
query.", "Possible Fields")
    if (theField = nil) then
      exit
    end
  else

```

APPENDIX B

```

    theField = "Value"
end
theExpr = nil
anExpr = nil
expr = nil
optionsList = {"Greater Than", "Less Than", "Between Two Values"}
greaterlesser = msgBox.ChoiceAsString(optionsList, "In your query, will you be specifying values
greater than or less than a particular value, or in between two specific values?", "Greater/Lesser Box")
if (greaterlesser = "Greater Than") then
    greater = msgBox.input ("Please enter the minimum" ++theTheme.asString++ "value, that you desire
for your query.", "Greater Box", "")
    if (greater <> nil) then
        theExpr = "(" + theSrcName.asString++ "." ++theField.asString+ "]" > " +greater.asString+ ")"
    elseif (greater = nil) then
        exit
    end
elseif (greaterlesser = "Less Than") then
    lesser = msgBox.input ("Please enter the maximum" ++theTheme.asString++ "value, that you desire
for your query.", "Lesser Box", "")
    if (lesser <> nil) then
        theExpr = "(" + theSrcName.asString++ "." ++theField.asString+ "]" < " +lesser.asString+ ")"
    elseif (lesser = nil) then
        exit
    end
elseif (greaterlesser = "Between Two Values") then
    greater = msgBox.input ("Please enter the minimum" ++theTheme.asString++ "value that you desire
for your query.", "Greater Box", "")
    if (greater <> nil) then
        anExpr = "(" + theSrcName.asString++ "." ++theField.asString+ "]" > " +greater.asString+ ")"
    elseif (greater = nil) then
        exit
    end
    lesser = msgBox.input ("Please enter the maximum" ++theTheme.asString++ "value that you desire for
your query.", "Lesser Box", "")
    if (lesser <> nil) then
        expr = "(" + theSrcName.asString++ "." ++theField.asString+ "]" < " +lesser.asString+ ")"
    elseif (lesser = nil) then
        exit
    end
    theExpr = "(" + anExpr.asString++ "and" ++expr.asString+ ")"
elseif (greaterlesser = nil) then
end
if (theExpr.asString <> nil) then
    exprList.add (theExpr.asString)
else
end
elseif (theData = nil) then
    exit
end

```

APPENDIX B

```
end
end
```

'List of each expression used in the query - it is hoped that this list will be used to run the query in the future, without the use of the query builder dialog box

```
'msgBox.ListAsString(exprList, "List of expressions used in the query", "List Box")
```

'Create a string of the expressions used in the query and prompt user to write it down for use in the unassisted modeling command

```
expression = nil
```

```
For Each i in exprList
```

```
  If (expression = nil) then
```

```
    expression = i.asString
```

```
  Else
```

```
    expression = expression.asString++ "and" ++i.asString
```

```
  end
```

```
end
```

```
msgBox.info("Please write this expression down and type it into the ""Unassisted Modeling"" query  
builder." +NL +NL +expression.asString, "Query Expression")
```

APPENDIX B

RandomPt Script

'This script obtains the rectangular area to be used as bounds for the random non-sites needed for logistic input. It obtains the appropriate number of random points using the ArcView random number generator and assigns their data to the text file.

```
theView = av.getActiveDoc
```

'Get the user's rectangle and draw it on the view

```
theRect = theView.getDisplay.returnUserRect
theGraphic = graphicShape.make (theRect)
theView.getGraphics.unselectAll
theGraphic.setSelected (FALSE)
theView.getGraphics.add (theGraphic)
_theRecInput = "finished"
```

'Get the rectangle's dimensions

```
maxH = theRect.getTop
minH = theRect.getBottom
maxW = theRect.getRight
minW = theRect.getLeft
```

'Make random (x,y) locations, place on view and determine if the user will keep them

```
theNum = _theFile2.getSize
for each r in 1..theNum
  xVal = number.makeRandom (minW, maxW)
  yVal = number.makeRandom (minH, maxH)
  x = xVal
  y = yVal
  x.setFormat ("ddddddd")
  y.setFormat ("ddddddd")
  pt = point.make (xVal, yVal)
  g = graphicShape.make (pt)
  theView.getGraphics.unselectAll
  g.setSelected (TRUE)
  aSym = symbol.make (#SYMBOL_MARKER)
  aSym.setColor (color.getRed)
  aSym.setSize (30)
  newPt = point.make (xVal, (yVal - 180))
  g.setOrigin (newPt)
  g.setSymbol (aSym)
  theView.getGraphics.add (g)
  theAnswer = msgBox.yesNo ("Do you want to keep this point?", "Keep Random", TRUE)
  if (theAnswer = TRUE) then
    aSym.setColor (color.getBlack)
    g.setSymbol (aSym)
    g.setSelected (FALSE)
  g.invalidate
```

APPENDIX B

'Add data into coordinate fields and dependent variable fields

```

i = 0
k = 0
fileString = i.asString++x.asString++y.asString
for each t in _selectedThemes
  theGtheme = theView.findTheme (t.asString)
  num = theGtheme.returnCellValue (pt)
  theVtab = theGtheme.getVtab
  theField = _selectedItems.get (k)
  if (theField.asString = "Value") then
    fileString = fileString++num.asString
    k = k + 1
  else
    theField2 = theVtab.findField (theField.asString)
    newNum = num - 1
    theValue = theVtab.returnValueString (theField2, newNum)
    fileString = fileString++theValue.asString
    k = k + 1
  end
end
end

```

'If previous random point is not kept, make new random (x,y) locations, place on view and determine whether user will keep them

```

elseif (theAnswer = FALSE) then
  thePt = "bad"
  theView.getGraphics.clearSelected
  while (thePt = "bad")
    xVal = number.makeRandom (minW, maxW)
    yVal = number.makeRandom (minH, maxH)
    x = xVal
    y = yVal
    x.setFormat ("ddddddd")
    y.setFormat ("ddddddd")
    pt = point.make (xVal, yVal)
    g = graphicShape.make (pt)
    theView.getGraphics.unselectAll
    g.setSelected (TRUE)
    aSym = symbol.make (#SYMBOL_MARKER)
    aSym.setColor (color.getRed)
    aSym.setSize (30)
    g.setSymbol (aSym)
    newPt = point.make (xVal, (yVal - 180))
    g.setOrigin (newPt)
    theView.getGraphics.add (g)
    anAnswer = msgBox.yesNo ("Do you want to keep this point?", "Keep Random", TRUE)
  if (anAnswer = FALSE) then
    theView.getGraphics.clearSelected
    thePt = "bad"
  end
end

```

APPENDIX B

```

elseif (anAnswer = TRUE) then
  aSym.setColor (color.getBlack)
  g.setSymbol (aSym)
  g.setSelected (FALSE)
  g.invalidate
  thePt = "good"

```

'Add data into coordinate fields and dependent variable fields

```

  i = 0
  k = 0
  fileString = i.asString++x.asString++y.asString
  for each t in _selectedThemes
    theGtheme = theView.findTheme (t.asString)
    num = theGtheme.returnCellValue (pt)
    theVtab = theGtheme.getVtab
    theField = _selectedItems.get (k)
    if (theField.asString = "Value") then
      fileString = fileString++num.asString
      k = k + 1
    else
      theField2 = theVtab.findField (theField.asString)
      newNum = num - 1
      theValue = theVtab.returnValueString (theField2, newNum)
      fileString = fileString++theValue.asString
      k = k + 1
    end
  end
end
end
end
end
_theFile2.writeELT (fileString)
end
_theFile2.close

```

'Update the view

```

m = 1
for each i in 1..10000
  m = m + 1
end
theView.getGraphics.SelectAll
theView.getGraphics.clearSelected
theView.invalidate

```

APPENDIX B

RandomPt2 Script

'This script makes random points to compare with the known site locations for the evaluate by histogram function and creates the histogram.

```
theView = av.getActiveDoc
```

'Get the extent of the query grid

```
aSrcName = _dirPath + "\" + _theQueryFile.asString
theSrcName = grid.makeSrcName (aSrcName.asString)
theGrid = grid.make (theSrcName)
theGtheme = Gtheme.make (theGrid)
theRect = theGrid.getExtent
maxH = theRect.getTop
minH = theRect.getBottom
maxW = theRect.getRight
minW = theRect.getLeft
```

'Make random (x,y) locations and fill appropriate data into _theVtab2

```
theNum = _theVtab2.getNumRecords
for each r in 1..theNum
  xVal = number.makeRandom (minW, maxW)
  yVal = number.makeRandom (minH, maxH)
  pt = point.make (xVal, yVal)
  x = xVal
  y = yVal
  x.setFormat ("ddddddd")
  y.setFormat ("ddddddd")
  i = "random"
```

'Create a record in theVtab2 and add coordinate information into _theVtab2

```
recNum = _theVtab2.addRecord
_theVtab2.setValueString (_theSite2, recNum, i.asString)
_theVtab2.setValueString (_xCoord, recNum, x.asString)
_theVtab2.setValueString (_yCoord, recNum, y.asString)
```

'Determine whether the point falls within the query area and add information to _theVtab2

```
aSrcName = _dirPath + "\" + _theQueryFile.asString
theSrcName = grid.makeSrcName (aSrcName.asString)
theGrid = grid.make (theSrcName)
theGtheme = gTheme.make (theGrid)
num = theGtheme.returnCellValue (pt)
theVtab = theGtheme.getVtab
theField = "random"
theFieldName = _theVtab2.findField (theField.asString)
_theVtab2.setValueString (theFieldName, recNum, num.asString)
end
```

APPENDIX B**'Update the view**

```

m = 1
for each i in 1..100
  m = m + 1
end
theView.invalidate

```

'Make the chart

```

theNumRecords = _theVtab2.getNumRecords
theNumRecords2 = theNumRecords/2
theFieldName = _theVtab2.findField ("site")
theFieldName2 = _theVtab2.findField ("random")
theSumField = _theVtab2.findField ("Sitetype")
theSumVtab = _theVtab2.summarize ("Summary".asFileName, dBase, theSumField, {theFieldName,
theFieldName2}, {#VTAB_SUMMARY_SUM, #VTAB_SUMMARY_SUM})
aFieldName = theSumVtab.findField ("Sum_Site")
aFieldName2 = theSumVtab.findField ("Sum_Random")
theFieldList = list.make
theFieldList.add (aFieldName)
theFieldList.add (aFieldName2)
theChart = chart.make (theSumVtab, theFieldList)
theTitle = theChart.getTitle
theTitle.setName ("Number of sites that fall into the query area vs. the number of random locations that fall
into the query area (based upon the" ++theNumRecords2.asString++ "sites and random locations
selected)")
theChart.getWin.activate

```

APPENDIX B

Resample Script

This script allows users to resample their grids using nearest neighbor, bilinear interpolation, or cubic convolution interpolation methods. It converts the resampled temporary grids to permanent grids and reassigns appropriate global variables.

```
theView = av.getActiveDoc
```

'Get grids and sourcenames

```
gridList = list.make
for each t in theView.getThemes
  if (t.is (gTheme)) then
    theGrid = t.getGrid
    theName = theGrid.getSrcName.asString
    gridList.add (theName.asString)
  end
end
gridCount = gridList.count
if (gridCount = 0) then
  msgBox.info ("No grids are available for resampling.", "Grid Availability")
else
```

'Get grids to resample, get their current cell sizes, get resample cell size and resample method

```
gridList.add ("all grids")
selectGrid = msgBox.ChoiceAsString(gridList, "Please select the grid you would like to resample.",
"Select Grids")
if (selectGrid = nil) then
  exit
```

'Resample all grids

```
elseif (selectGrid = "all grids") then
  allGridsList = list.make
  for each t in theView.getThemes
    t.setActive (TRUE)
    if (t.is (gTheme)) then
      theGrid = t.getGrid
      cellSize = theGrid.getCellSize
      allGridsList.add (cellSize.asString)
    end
  end
  allGridsList.RemoveDuplicates
  theCount = allGridsList.count
  if (theCount = 1) then
    msgBox.info ("The current resolution of your grid themes is " + cellSize.asString, "Current Resolution")
  else
    msgBox.listAsString (allGridsList, "Your grid themes are composed of the following cell sizes. For
information about a particular grid, check the Theme Properties Dialog Box.", "Current Resolution")
```

APPENDIX B

```

end
newCellSize = msgBox.input ("Please enter the cell size you would like to resample to.", "New Cell
Size", cellSize.asString)
if (newCellSize <> nil) then
  newCellSizeNum = newCellSize.asNumber
  rMethods = {"Nearest Neighbor", "Bilinear Interpolation", "Cubic Convolution"}
  resampleMethod = msgBox.listAsString (rMethods, "Please select a method for resampling",
"Resample Method")
  if (resampleMethod = nil) then
    exit

'Nearest neighbor method
  elseif (resampleMethod = "Nearest Neighbor") then
    i = 1
    for each t in theView.getActiveThemes
      if (t.is (gTheme)) then
        theGrid = t.getGrid
        _srcGridName = theGrid.getSrcName

'Assign appropriate global variables
        if (_srcGridName.asString = _elevation.asString) then
          _elevationb = theGrid.getSrcName
        elseif (_srcGridName.asString = _slope.asString) then
          _slopeb = theGrid.getSrcName
        elseif (_srcGridName.asString = _aspect.asString) then
          _aspectb = theGrid.getSrcName
        elseif (_srcGridName.asString = _soil.asString) then
          _soilb = theGrid.getSrcName
        elseif (_srcGridName.asString = _grH2O.asString) then
          _grH2Ob = theGrid.getSrcName
        elseif (_srcGridName.asString = _geol.asString) then
          _geolb = theGrid.getSrcName
        elseif (_srcGridName.asString = _stream.asString) then
          _streamb = theGrid.getSrcName
        elseif (_srcGridName.asString = _site.asString) then
          _siteb = theGrid.getSrcName
        else
          end
          theGtheme = "theGtheme" + i.asString
          theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum,
#GRID_RESTYPE_NEAREST))
          t.setActive (False)
          theGtheme.setName ("Resampled" ++t.getName)
          theView.addTheme (theGtheme)
          theGtheme.setActive (True)
          i = i + 1
        elseif (t <> gTheme) then
          exit
        end

```

APPENDIX B

```

end

'Convert to permanent grid
av.run ("makeGrid", "")
for each t in theView.getActiveThemes
  theView.deleteTheme (t)
end

'Bilinear interpolation method
elseif (resampleMethod = "Bilinear Interpolation") then
  i = 1
  for each t in theView.getActiveThemes
    if (t.is (gTheme)) then
      theGrid = t.getGrid
      _srcGridName = theGrid.getSrcName

'Assign appropriate global variables
      if (_srcGridName.asString = _elevation.asString) then
        _elevationb = theGrid.getSrcName
      elseif (_srcGridName.asString = _slope.asString) then
        _slopeb = theGrid.getSrcName
      elseif (_srcGridName.asString = _aspect.asString) then
        _aspectb = theGrid.getSrcName
      elseif (_srcGridName.asString = _soil.asString) then
        _soilb = theGrid.getSrcName
      elseif (_srcGridName.asString = _grH2O.asString) then
        _grH2Ob = theGrid.getSrcName
      elseif (_srcGridName.asString = _geol.asString) then
        _geolb = theGrid.getSrcName
      elseif (_srcGridName.asString = _stream.asString) then
        _streamb = theGrid.getSrcName
      elseif (_srcGridName.asString = _site.asString) then
        _siteb = theGrid.getSrcName
      else
        end
      theGtheme = "theGtheme" + i.asString
      theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum,
#GRID_RESTYPE_BILINEAR))
      t.setActive (False)
      theGtheme.setName ("Resampled" ++t.getName)
      theView.addTheme (theGtheme)
      theGtheme.setActive (True)
      i = i + 1
    elseif (t > gTheme) then
      exit
    end
  end

'Convert to permanent grid

```

APPENDIX B

```

av.run ("makeGrid", "")
for each t in theView.getActiveThemes
  theView.deleteTheme (t)
end

```

'Cubic convolution method

```

elseif (resampleMethod = "Cubic Convolution") then
  i = 1
  for each t in theView.getActiveThemes
    if (t.is (gTheme)) then
      theGrid = t.getGrid
      _srcGridName = theGrid.getSrcName

```

'Assign appropriate global variables

```

  if (_srcGridName.asString = _elevation.asString) then
    _elevationb = theGrid.getSrcName
  elseif (_srcGridName.asString = _slope.asString) then
    _slopeb = theGrid.getSrcName
  elseif (_srcGridName.asString = _aspect.asString) then
    _aspectb = theGrid.getSrcName
  elseif (_srcGridName.asString = _soil.asString) then
    _soilb = theGrid.getSrcName
  elseif (_srcGridName.asString = _grH2O.asString) then
    _grH2Ob = theGrid.getSrcName
  elseif (_srcGridName.asString = _geol.asString) then
    _geolb = theGrid.getSrcName
  elseif (_srcGridName.asString = _stream.asString) then
    _streamb = theGrid.getSrcName
  elseif (_srcGridName.asString = _site.asString) then
    _siteb = theGrid.getSrcName
  else
    end
  theGtheme = "theGtheme" + i.asString
  theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum, #GRID_RESTYPE_CUBIC))
  t.setActive (False)
  theGtheme.setName ("Resampled" ++t.getName)
  theView.addTheme (theGtheme)
  theGtheme.setActive (True)
  i = i + 1
elseif (t <> gTheme) then
  exit
end
end
end

```

'Convert to permanent grid

```

av.run ("makeGrid", "")
for each t in theView.getActiveThemes
  theView.deleteTheme (t)
end

```

APPENDIX B

```

    end
  else
    exit
  end
end

'Resample individual grids
else
  if (_dirPath = nil) then
    theDirPath = msgBox.input ("Please input the directory path for the " ++selectGrid.asString ++"grid.",
"Directory Path", "C:\data")
    if (theDirPath <> nil) then
      _dirPath = theDirPath.asString
    else
      myAnswer = msgBox.yesNo ("You have entered an invalid directory path. Do you want to reenter
the information?", "Improper Input", true)
      if (myAnswer = true) then
        while (theDirPath = nil)
          theDirPath = msgBox.input ("Please input the directory path for the " +selectGrid.asString +
grid.", "Directory Path", "C:\data")
          if (theDirPath <> nil) then
            _dirPath = theDirPath.asString
          else
            exit
          end
        end
      end
    end
  end
  _srcGridName = grid.makeSrcName (_dirPath.asString + "\" +selectGrid.asString)
  If (_srcGridName = nil) then
    myAnswer = msgBox.yesNo ("The grid you have selected is not in the " +_dirPath ++" working
directory. Would you like to reenter the directory path?", "Improper Directory", true)
    if (myAnswer = true) then
      while (_srcGridName = nil)
        theDirPath = msgBox.input ("Please input the directory path for the " +selectGrid.asString +
grid.", "Directory Path", "C:\data")
        if (theDirPath <> nil) then
          dirPath = theDirPath.asString
          _srcGridName = grid.makeSrcName (dirPath.asString + "\" +selectGrid.asString)
        else
          exit
        end
      end
    end
  end
  theGrid = grid.make (_srcGridName)
  cellSize = theGrid.getCellSize
  msgBox.info ("The current resolution of the" ++selectGrid++ "theme is " +cellSize.asString, "Current
Resolution")

```

APPENDIX B

```

newCellSize = msgBox.input ("Please enter the cell size you would like to resample to.", "New Cell
Size", cellSize.asString)
if (newCellSize <> nil) then
  newCellSizeNum = newCellSize.asNumber
  rMethods = {"Nearest Neighbor", "Bilinear Interpolation", "Cubic Convolution"}
  resampleMethod = msgBox.listAsString (rMethods, "Please select a method for resampling",
"Resample Method")
  if (resampleMethod = nil) then
    exit

'Nearest neighbor method
  elseif (resampleMethod = "Nearest Neighbor") then
    theTheme = gTheme.make (theGrid)
    theTheme.setActive (true)
    theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum, #GRID_RESTYPE_NEAREST))
    for each t in theView.getThemes
      t.setActive (False)
    end
    theGtheme.setName ("Resampled" ++theTheme.getName)
    theView.addTheme (theGtheme)
    theGtheme.setActive (True)
    av.run ("makeGrid", "")
    theView.deleteTheme (theGtheme)

'Bilinear interpolation method
  elseif (resampleMethod = "Bilinear Interpolation") then
    theTheme = gTheme.make (theGrid)
    theTheme.setActive (true)
    theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum, #GRID_RESTYPE_BILINEAR))
    for each t in theView.getThemes
      t.setActive (False)
    end
    theGtheme.setName ("Resampled" ++theTheme.getName)
    theView.addTheme (theGtheme)
    theGtheme.setActive (True)
    av.run ("makeGrid", "")
    theView.deleteTheme (theGtheme)

'Cubic convolution method
  elseif (resampleMethod = "Cubic Convolution") then
    theTheme = gTheme.make (theGrid)
    theTheme.setActive (true)
    theGtheme = gTheme.make (theGrid.Resample (newCellSizeNum, #GRID_RESTYPE_CUBIC))
    for each t in theView.getThemes
      t.setActive (False)
    end
    theGtheme.setName ("Resampled" ++theTheme.getName)
    theView.addTheme (theGtheme)
    theGtheme.setActive (True)

```

APPENDIX B

```
'Convert to permanent grid
  av.run ("makeGrid", "")
  theView.deleteTheme (theGtheme)
end
else
  exit
end
end
end
```

APPENDIX B

Stop Script

'This script stops the input of graphically selected (x,y) locations for the logistic input file. It alerts the user to the use of the rectangle tool and calls the enable script.

```
theView = av.getActiveDoc
```

```
msgBox.info ("Click on the ""rectangle"" tool and draw the rectangle to bound the surveyed area. The  
random non-site locations will be taken from within this bounding rectangle.", "Rectangular Boundary")  
_theRecInput = "Draw Rectangle"  
_theInput = "finished"
```

APPENDIX B

Stop2 Script

'This script stops the input of graphically selected (x,y) locations for the histogram function.

```
theView = av.getActiveDoc
```

```
_theInput2 = "finished"  
av.run ("randomPts2", "")
```

APPENDIX B

Global Variables Used in the Scripts

_aspect = source name for aspect grid

_aspect2 = source name for resampled aspect grid

_aspectb = source name for permanent resampled aspect grid

_elevation = source name for elevation grid

_elevation2 = source name for resampled elevation grid

_elevationb = source name for permanent resampled elevation grid

_dirPath = directory path for project data

_geol = source name for geology grid

_geol2 = source name for resampled geology grid

_geolb = source name for permanent resampled geology grid

_grH2O = source name for groundwater grid

_grH2O2 = source name for resampled groundwater grid

_grH2Ob = source name for permanent resampled groundwater grid

_random = field name in the Vtab used by the evaluate by histogram procedure

_selectedItems = list of fields used by independent variables for logistic input

_selectedThemes = list of themes used as the independent variables for logistic input and logistic surface

_slope = source name for slope grid

_slope2 = source name for resampled slope grid

_slopeb = source name for permanent resampled slope grid

_site = source name for site grid

_siteb = source name for permanent resampled slope grid

_soil = source name for soil grid

APPENDIX B

_soil2 = source name for resampled soil grid

_soilb = source name for permanent resampled soil grid

_srcGridName = source name prefix used for creating permanent resampled grids

_stream = source name for stream grid

_stream2 = source name for resampled stream grid

_streamb = source name for permanent resampled stream grid

_textFile = name of the text file created in logistic input

_themeName = prefix used for creating permanent grid names

_theFile2 = file created for grid or point coverage input methods in logistic input

_theFtab2 = name of Ftab used for grid or point coverage input methods in logistic input

_theFtheme2 = theme created for grid or point coverage input methods in logistic input

_theInput = variable that controls whether the pointer and stop tools are activated for logistic input

_theInput2 = variable that controls whether the pointer and stop tools are activated for evaluate by histogram

_theQueryFile = name of the query grid used in evaluate by histogram procedure

_theRecInput = variable that controls whether the rectangle tool is enabled

_theSite2 = field name in the Vtab used by the evaluate by histogram procedure

_theVtab2 = table used by the evaluate by histogram procedure

_xCoord = field name in the Vtab used by the evaluate by histogram procedure

_yCoord = field name in the Vtab used by the evaluate by histogram procedure

APPENDIX B**Logistic AML**

```
/* This AML will set the graphics display, prompt the  
/* user to enter their logistic input file, set the graphics  
/* display, get into grid and run logistic regression.
```

```
&station 9999
```

```
&setvar file := [response 'Please enter the name of your input text file. (include the .txt extension)']
```

```
GRID
```

```
REGRESSION %FILE% LOGISTIC
```

APPENDIX C

PROFESSIONALS WHO CONTRIBUTED TO THE EVALUATION OF THE ARCHAEOLOGICAL SITE PREDICTION SYSTEM

Archaeologists:

anonymous: PhD candidate in Archaeology, State University of New York, Buffalo

Christine Branstner: PhD candidate in Anthropology, Archaeology Specialty, Michigan State University; Lab Manager, Great Lakes Research Associates

Wendy Cavallari: PhD candidate in Archaeology, Ohio State University; Field Supervisor for the Athienou Archaeological Project

Jannice Cottrell: Ethno-Historic Researcher, Aurora Associates

Misty Jackson: PhD candidate in Anthropology, Archaeology Specialty, Michigan State University; Staff Archaeologist, Andrews Cultural Resources

William Lovis: Professor of Archaeology, Michigan State University

Barbara Mead: Assistant State Archaeologist at the Office of the State Archaeologist, Michigan Historical Center

Jennifer Pederson: PhD candidate in Archaeology, Ohio State University

Richard Yerkes: Professor of Archaeology, Ohio State University; Assistant Director for the Athienou Archaeological Project

Geographers:

Stephen Aichele: Masters Candidate in Geography, Michigan State University

Jill Hallden: Masters Candidate in Geography, Michigan State University

Duane Marble: Professor of Geography and Natural Resources, Ohio State University

APPENDIX C

Derek Moy: Masters Candidate in Geography, Eastern Michigan University

Alison Philpotts: PhD Candidate in Geography, Michigan State University

APPENDIX D

SURVEY AND QUESTIONNAIRES FOR PROFESSIONAL INTERVIEWS

Skill Level Survey for Professional Archaeologists

1. Computer ranking:

- A. What type of computer(s)/operating system(s) do you have access to (i.e. Unix workstations, PC's, Windows '95, DOS, etc.)?
- B. How do you rate your computer "literacy" on a scale of one to five (1 being little or no experience and 5 being a computer expert)?
- C. What do you use your computer for (please rank the following - 1 being the least use and 6 being the most common use)?

_____ "Office Software" (word processing, spreadsheets, limited graphics)

_____ E-mail

_____ Exploration of the Internet

_____ Computer Programming

_____ GIS

_____ Other _____

2. GIS ranking:

- A. Do you currently use a GIS in your archaeological profession?

1. Yes

- a. In what capacity?
- b. What GIS software are you using?

2. No

- a. Have you considered using GIS in your archaeological Endeavors? If yes, in what capacity?
- b. What is limiting you from using a GIS (access to software, unfamiliarity with software/GIS technology, cost of learning GIS, etc.)?

APPENDIX D

Questionnaire for Professional Archaeologists

1. What did you like/dislike about the archaeological site prediction system? Do you feel that this system is sufficient for creating a site prediction model?
2. Did you find this system easy to use? Were you able to understand and apply each of its components?
3. Do you feel that increased user-friendly software is needed to encourage archaeologists to use GIS? Do you feel that this software meets a need?
4. Would you consider using this software in your own archaeological endeavors? Would you consider using other user-friendly GIS systems if they were created in a similar fashion? What GIS tools would you like to see in future user-friendly GIS systems?
5. Do you have any closing questions, comments, or suggestions pertaining to this archaeological site prediction system?

Questionnaire for Professional Geographers

1. What did you like/dislike about the archaeological site prediction system? Do you feel that this system appropriately applies the concepts and tools of GIS?
2. Did you find this system easy to use? Were you able to understand and apply each of its components?
3. Can you think of any additional GIS functions that may be useful for this type of GIS analysis?
4. Do you have any closing questions, comments, or suggestions pertaining to this archaeological site prediction system?

APPENDIX E

HELP PAGES FROM THE ARCHAEOLOGICAL SITE PREDICTION SYSTEM

^s ^kIntroduction to the Archaeological Site Prediction System

The Archaeological Site Prediction System (ASPS) is designed to assist archaeologists with the creation and analysis of site prediction models. The ASPS is sub-divided into 6 component groups.

- **Set-Up:** This component assists the user in adding grid data into the project and provides the user with the option of resampling their grids to a different resolution.
- **Data Manipulation:** This component assists the user in the development of a “distance grid.” Each cell in the distance grid is composed of values that represent the distance that cell is from a specified feature in another grid.
- **Deductive Modeling:** This component assists the user in the development of a Deductive Site Prediction Model. The user may create a query expression using the map calculator dialog box or through the assisted query method. Upon the completion of a deductive model, the resultant binary grid may be grouped and its areas calculated for further queries.
- **Inductive Modeling:** This component assists the user in the creation of an Inductive Site Prediction Model using logistic regression. Categorical data may be converted to binary data for use in the logistic regression analysis. An input file, containing the necessary dependent and independent variables can be made and used to run the logistic regression function in Arc/Info. The resultant logistic regression coefficients may be used in ArcView to create a logistic surface.
- **Evaluation:** This component assists the user in creating histograms that can be used to summarize the results of the deductive and inductive site prediction models.
- **Help and Information:** This component provides the user with on-line help and information pertaining to the ASPS.

The Archaeological Site Prediction System was created by Jessica Dolanski. All ASPS systems related questions and comments should be directed to her at: dolansk3@pilot.msu.edu

^s Introduction to the Archaeological Site Prediction System

[#] ASPS Introduction

^k ASPS; ASPS Introduction; Archaeological Site Prediction System; Site Prediction System

APPENDIX E

Skills Needed to Successfully Run the ASPS

The ASPS is designed for the novice GIS user. However, you should have a general understanding of Windows-based software and how to manipulate pull-down menus and buttons. A general knowledge of ArcView is also recommended.

^s Skills needed to successfully run the Archaeological Site Prediction System

[#] ASPS skills

^k ASPS skills; skills; skill level; skill levels

APPENDIX E

*** * * Hardware and Software Requirements of the ASPS**

The optimal hardware for the ASPS is a PC computer with either a Windows® 95 or Windows® NT operating system. In order to be fully operational, the ASPS requires ArcView 3.0 and Arc/Info software produced by Environmental Systems Research Institute, Inc. (ESRI).

The ESRI software is also available for Unix and Macintosh systems, however, the ASPS has not been tested on these platforms.

With the exception of the Logistic component, the ASPS is contained within ArcView software.

*** Hardware and software requirements of the ASPS**

computer requirements

*** computer requirements; computer requirement; hardware; software**

APPENDIX E

^s ^k **Helpful Hints**

- Address problems early. After each command, take the time to look at your result. Ask yourself, does this result make sense? If not, make the necessary corrections before continuing on to the next step.
- Orient yourself to the data. After a new grid has been created (i.e. following the *distance from...*, *unassisted modeling*, *logistic surface*, etc. commands), reacquaint yourself with the data. If you have a feature grid such as roads, streams, or sites, move it to the top of your legend bar so that these features are overlaid on your new grid. This will assist with the orientation process.
- Practice with the system. While the ASPS is designed for the novice user, it will require some practice before you feel comfortable with each of the commands and buttons of the system.
- Experiment with your data. The purpose of the ASPS is to create multiple site prediction models. A single “best model” may not exist for your data, therefore, the further you experiment with your data, the more likely you are to create viable models that will be useful to your archeological endeavors.

^s Helpful hints for the ASPS

^h Helpful Hints

^k helpful hints; hints; suggestions

APPENDIX E

^s ^{*} ^k Data Requirements of the ASPS

In order to enter your data through the *add grids* command in the ASPS, it must meet the following conditions:

- Arc/Info grid format
- Located in one directory

Data that do not meet these conditions (i.e. Arc/Info vector coverages or grid data located in multiple directories) can be entered through the ArcView *add theme* command in the View pull-down menu. Most of the ASPS commands require that the data meet the previously stated conditions, therefore, it is best if your data can be entered through the *add grids* command.

^s Data requirements of the ASPS

^{*} Data

^k data; data requirements

APPENDIX E

^s ^k Adding Grids to the ASPS

The *add grids* command in the ASPS helps you enter data into the project. These data must be in an Arc/Info grid format and located in a single directory. This command prompts you to enter eight specific grids: elevation, slope, aspect, streams, groundwater, geology, soils, and sites. You may enter additional grid data at the conclusion of these prompts.

The *add grids* command follows this order:

- directory path prompt (enter the directory structure for the location of your grid data).
- elevation grid name prompt (enter the name of your elevation grid, if you do not have an elevation grid, press cancel).
- elevation field name prompt (enter the name of the field that contains the elevation data you want to display in the legend).
- the elevation grid is added to the view, the legend is created, and the grid is turned on.
- the slope, aspect, streams, groundwater, geology, soils, and sites grids are prompted for in the same fashion as the elevation grid.
- add more grid data prompt (select “yes” to add more data or “no” to stop entering data).
- grid and field name prompts (enter name of additional grid and field to be displayed in the legend).
- legend type prompt (enter categorical if your data should have a unique-value legend (where every piece of information receives a different color), or numerical if your data should have a ramped legend (where low numbers receive a light color and high numbers receive a dark color in the same hue)).

^s Adding grids to the ASPS

[#] Add Grids

^k add grids; data entry; add data; enter grids

APPENDIX E

^s ^k **Field**

A field can be thought of as a column in a table that contains information. In the example below, the geology grid contains 5 fields: value, count, link, geologytype, and binarygeologytype. Some GIS packages refer to fields as “items.”

Field	Field	Field	Field	Field
↓	↓	↓	↓	↓
Attributes Of Geology				
Value	Count	Link	Geologytype	BinaryGeologytype
1	15254	2	PL	1
2	48836	3	ML-MU	0
3	13469	6	H	1
4	269147	7	KU3-OU	1
5	3856	8	ML-MU	0
6	1344	5	ML-MU	0
7	7645	9	LPL	0
8	1	11	UPL	1
9	5484	11	UPL	1
10	85365	12	ML-MU	0
11	1	7	KU3-OU	1
12	64	7	KU3-OU	1
13	4416	15	MU1	0
14	10655	16	PL	1
15	3015	13	KU	1
16	405	14	UPL	1
17	39	17	LPL	0

^s field pop-up

^k field pop-up

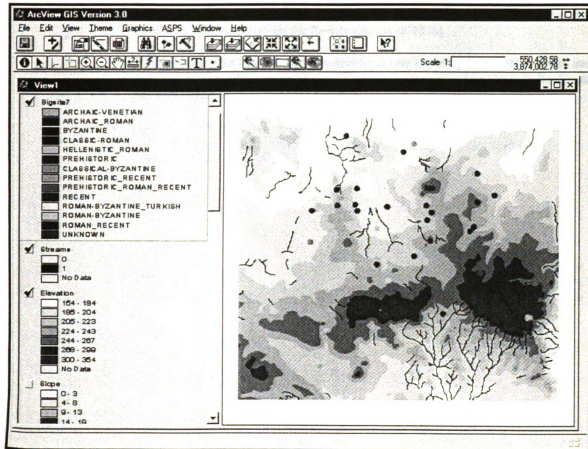
^k field; fields; item; items

APPENDIX E

5.4k Turning a Grid On and Off, Activating a Grid

When a grid is “turned-on,” it has a check mark in the legend and it is visible on in the view. In the example below, the bigsite7, streams, and elevation grids are “turned on.” The slope coverage is “turned-off.” To turn a grid on or off, click in the little gray box to add or remove the check mark.

When a grid theme is activated, a thin gray line appears around its legend. Activated grids can be used for certain ASPS commands and ArcView functions. In the example below, the bigsite7 grid is activated.



⁵ Turning a grid on/off and activation

⁶ legend pop-up

⁷ grid on; grid visible; grid off; grid invisible; grid activated

APPENDIX E

When to Resample

Resampling a grid is the same as changing the resolution or cell size. There are at least three reasons for resampling your data.

- Your data are not the same resolution. The data should all be converted to the resolution of the data that currently has the largest resolution. (e.g. if you have one grid at 10 m and one at 20 m, the 10 m grid should be converted to 20 m)
- You have prior knowledge about the accuracy of the data. (e.g. you have a sites grid at 10 m resolution, however you know that those sites were located through a systematic survey at 20 m spacing, therefore, it may be inappropriate to assume that your sites grid is accurate at the finer resolution of 10 m)
- Your data set is so massive that it is overburdening the computer. Provided that your site prediction models can compromise on detail, you can reduce the time it takes to run commands by increasing the resolution.

^sWhen to resample

^hWhy Resample

^kwhy resample; when to resample

APPENDIX E

***s* * *k* Resampling Grids**

The *resample* command allows you to change the resolution of one or all of your grids.

The *resample* command follows this order:

- grid prompt (select the grid to resample from the list box, or select “all grids” to resample all of your data).
- a message box with the current resolution(s) of your grid(s) appears on the screen.
- new resolution prompt (enter a numerical value for the new cell size).
- interpolation method prompt (select a method for interpolation - nearest neighbor, bilinear interpolation, or cubic convolution).
- conversion to permanent grids (the resampling process creates temporary grids that are not stored in your working directory, the *resample* command automatically converts these temporary grids to permanent grids that are stored in your working directory, you are prompted to enter a new name (if desired) and asked whether you would like this grid added to the view).

***s* Resampling grids**
***s* Resample**
***k* resample; resampling**

APPENDIX E

^s ^{*} ^k **Interpolation Methods**

ArcView 3.0 provides three methods for interpolating a grid: nearest neighbor, bilinear interpolation, and cubic convolution.

nearest neighbor: resampled grid cell is assigned the value of its nearest neighbor, best for categorical data, fastest interpolation method

bilinear interpolation: resampled grid cell is assigned the value of the weighted average of the nearest four neighbors, useful for continuous data

cubic convolution: resampled grid cell is assigned the value of the weighted average of the nearest sixteen neighbors, best for continuous data, most time-consuming interpolation method

* Definitions taken from Arc/Info 7.03 Help

^s **Interpolation methods**

^r **Interpolation Pop-up**

^k **interpolation methods; nearest neighbor; bilinear interpolation; cubic convolution**

APPENDIX E

s # K Verifying Resample Results

Your resampled grids may not have a noticeably different appearance from your original grid data. If you are unsure of whether the resample command worked properly, follow these instructions:

- activate the resampled grid.
- go to ArcView Theme pull-down menu.
- click on theme properties.
- check to see that the cell size is the one that you designated in the *resample* command.

APPENDIX E

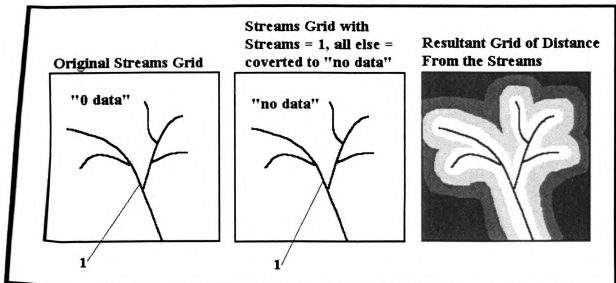
***** Using the Distance From... Command**

The *distance from...* command allows you to create a grid of distance from any feature grid (i.e. roads, or streams grids). A resultant distance grid can be used by either the deductive or inductive modeling components to evaluate how far sites are from certain features.

The *distance from...* command follows this order:

- feature grid prompt (enter the name of the grid from which distance will be calculated from).
- convert zero data to no data prompt (since all pixels, except those labeled "no data" are used in the distance calculations, all pixels that are not part of your feature (zero data) should be converted to "no data").
- distance is calculated from the feature grid.
- conversion to permanent grid (this process creates a temporary grid that is not stored in your working directory, the *distance from...* command automatically converts this temporary grid to a permanent grid that is stored in your working directory, you are prompted to enter a new name (if desired) and asked whether you would like this grid added to the view).

Graphic of the *distance from...* steps:



The experienced user may investigate distance issues further with advanced distance functions.

S Using the distance from... command
Distance From
K distance from; distance; distance calculation; distance calculations

APPENDIX E

^s ^k **Advanced Distance Functions**

Distance issues can be further explored through two advanced distance functions. These functions are not built into the ASPS, therefore, some prior knowledge of ArcView 3.0 is required.

- Calculating distance from only one feature.
If you desire to know the distance from only one stream, road, etc., you will need to edit the grid table. Open the table for the feature grid. Re-code all features to zero, except for the one stream, road, etc. that you wish to use in the distance calculation (do not re-code that feature). Run the distance from command, changing all zero data to no data. The resultant grid will be of distance from the one desired feature.
- Interactively calculating distance from two feature on the view screen.
In order to calculate distance between two features on the screen, you must first establish the projection and map units for the view. You may do this from the ArcView View pull-down menu and by entering the necessary information into the *properties* function. Once these items have been defined, you may use the *ruler* tool, located in the tool button bar to determine distance between various features on your grid.

^s Advanced distance functions

[#] Advanced Distance Functions

^k advanced distance functions; distance functions

APPENDIX E

^s ^k Using the Assisted and Unassisted Model Builders

Deductive query expressions can be created in either the *assisted* or *unassisted modeling* commands. As the name implies, the *assisted* function prompts you through each aspect of the query expression development. The *unassisted* function opens the ArcView query builder dialog box and allows you to create a query expression without assistance. The former method is recommended for the novice user, while the later method is recommended for the advanced user. All queries are actually performed in the *unassisted modeling* function.

The *assisted modeling* function follows this order:

- add grid to the query prompt (press “yes” to add information to the query or “no” to finish adding to your query).
- select grid prompt (select a grid that contains information that you would like to query).
- categorical/numerical prompt (specify whether the grid selected is categorical or numerical data - the format of the next prompt is based upon this information).
- categorical: select the field that contains the variables you would like to query, select the values you would like to use in your query.

numerical: determine whether you would like to query values greater than or less than a particular number, or in between two numbers, enter the value(s) to be used in your analysis.

- repeat this process until you have entered the desired information into your query expression (stop entering information by pressing “no” when prompted to add another theme to your query).
- record the query expression prompt (record the query expression, as typed in the message box, for use in the *unassisted modeling* function).

The *unassisted modeling* function is organized as follows:

- running a query expression from the *assisted modeling* function (type the previously recorded query expression into the model builder dialog box without omitting any punctuation, and press evaluate).
- running a query expression created in the *unassisted modeling* function (enter the components of your query expression into the query builder dialog box and press evaluate).

Upon completion of the model building process, run the *convert to grid* command on your resultant binary query grid

^s Assisted and unassisted modeling

[#] Query

^k query; queries; unassisted modeling; assisted modeling

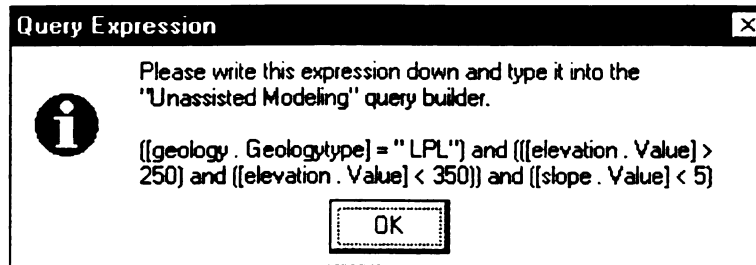
APPENDIX E

^s [#] ^k Query Expression

A query expression is a statement that contains all of the necessary variable and punctuation information needed to run a query in ArcView. At this time, the query expression cannot be run directly from the *assisted modeling* function. Therefore, you must record the query expression for use in the *unassisted modeling* function.

In the example below, you would need to record the following:

```
(([geology.geologytype] = "LPL") and (([elevation.value] > 250) and ([elevation.value] < 350)) and ([slope.value] < 5)
```



The example query is asking the computer to find all of the areas where the geology type is "LPL", the elevation is between 250 and 350 meters, and the slope is less than 5°.

^s Query expression

[#] Query Expression

^k query expression; query expressions

APPENDIX E

^s [#] ^k Convert to Grid

Since the binary query grid created in the *unassisted modeling* function is developed using the ArcView query builder dialog box, the resultant grid is temporary in nature. In order to place this temporary grid into your working directory, you need to convert it to a permanent grid using the *convert to grid* function.

The *convert to grid* command requires that your temporary query grid is activated. If you do not have a grid activated, or you have more than one grid activated, you will be prompted to activate **ONLY** the query grid. The *convert to grid* function will prompt you to enter a new name (if desired) and ask whether you would like this grid added to the view).

^s Convert to grid

[#] Convert to Grid

^k convert to grid; grid conversion; convert to permanent grid

APPENDIX E

^{s # k} Running Group and Area

The *group and area* function identifies contiguous groups of grid cells in the query, classifies them into individual groups, assigns a unique identifier to each, and calculates its area. In order to run this command, you will need a query grid that has been converted to permanent status.

The *group and area* command is organized as follows:

- query grid prompt (enter the name of your permanent, binary query grid).
- the contiguous groups of grid cells are grouped and assigned unique identifiers.
- the area of the unique groups is calculated in the same units as your original data.
- conversion to permanent integer grids (this process creates temporary grids that are not stored in your working directory, the *group and area* command automatically converts these temporary grids to permanent, integer grids that are stored in your working directory, you are prompted to enter new names (if desired) and asked whether you would like these grids added to the view).
- ArcView integer grids are limited to values of 1,000,000 or less, therefore the area grid is divided by 100. The original floating point grid is left in the legend for reference purposes. However, the permanent integer grid should be used for future calculations (be sure to scale your future calculations accordingly - multiply by 100 to obtain the actual area in squared units).

^s Running group and area

[#] Group and Area

^k group and area; group; area

APPENDIX E

§ * * Why Area Grid is Divided by 100

ArcView integer grids are limited to values of 1,000,000 or less. When areas are calculated for large contiguous areas, the area may exceed this value and crash the system. Therefore, the integer area grid is automatically divided by 100.

ArcView does not maintain these limitations for the floating point grids, therefore the temporary area grid created in *group and area* does not affect the system if it exceeds a value of 1,000,000. The temporary area grids are left in the legend for reference purposes. However, the permanent integer grid should be used for future calculations (be sure to scale your future calculations accordingly - multiply by 100 to obtain the actual area in squared units).

§ Why area is divided by 100

* Area Output

* area output; area integer grid

APPENDIX E

^s ^k **Logistic Regression**

By examining the values of binary/ordinal data at locations known to be sites and at locations known NOT to be sites, logistic regression can be used to predict the probability that a site will exist at certain unsampled locations.

This method of analysis requires that that a survey has been previously conducted and that sites and areas that do not contain sites have been identified. It also requires the following inputs:

- dependent variable (i.e. site/non-site): the dependent variable is assigned a 1 if it is a known site, or a 0 if it is a location known NOT to be a site.
- coordinate information: the (x, y) values for the locations of the sites/non-sites.
- independent variables: the binary/ordinal data that is assumed to affect site location.

^s Logistic regression

[#] Logistic Regression

^k logistic regression; statistics

APPENDIX E

^s ^k Creating Binary Fields

Logistic regression uses binary or ordinal data, therefore, all categorical data, used for this procedure, must be converted to binary data. If you are familiar with your data and understand how it affects site location, you may create a single binary field for all categories. Conversely, if you are unfamiliar with how your data affects site location and would like to explore these relationships by developing several site prediction models, you may create a binary field for each category.

The *make binary fields* command follows this order:

- categorical grid prompt (select the categorical grid to add binary field(s) to).
- field prompt (select the field that contains the categorical data to convert).
- number of binary fields prompt (select whether to create one binary field for all categorical data or to create a binary field for each type of categorical data).
- creation of binary fields
 - one field: you will be prompted to enter 1/0 for each category type, a single binary field is created and named according to the category field name (i.e. if the field name is "geologytype", the binary field name would be "binarygeologytype").

field for each category: a field will be created for each category, the name of each field is based upon the categorical data (i.e. if you have three types of roads: dirt, gravel, and paved, you would have three binary fields entitled: dirt, gravel, and paved).

The figure below demonstrates how these binary fields may look in a table.

Single binary field for all categories		One binary field for each category				
<i>category</i>	<i>binarycategory</i>	<i>category</i>	<i>category1</i>	<i>category2</i>	<i>category3</i>	<i>category4</i>
Category 1	0	Category 1	1	0	0	0
Category 2	1	Category 2	0	1	0	0
Category 3	1	Category 3	0	0	1	0
Category 4	0	Category 4	0	0	0	1

^s Creating binary fields

^k Binary Fields

^k binary fields; categorical data

APPENDIX E

^s [#] ^k Creating a Logistic Input File

In order to run logistic regression, you need an input text file that contains the dependent variables (site/non-site), their (x, y) coordinates, and the independent variables (the binary/ordinal data) present at that location.

The logistic input file is created using this process:

- select grids to be used as independent variables (you may select any grids that contain binary/ordinal data - categorical data must be previously converted to binary data using the *make binary fields* command).
- select field that contains the binary/ordinal data to be used for analysis.
- enter a name for the text file to be created (any previously created text files with the same name will be over-written).
- enter known sites (known sites can be entered through graphical or tabular methods, or they can be taken from a grid or point coverage).
- independent data is gathered for the known site locations and entered into text file.
- draw a bounding rectangle for the survey area (click on the activated rectangle tool in the button bar and draw the survey boundary lines on the view - this rectangle serves as the boundary from which random, non-site data will be taken).
- ArcView selects random non-site locations from within the survey boundaries. You are prompted whether to keep each random point. If the random point is “too close” to a known site, you may discard it and ArcView will select another. This process will continue until the number of selected random points equals the number of known sites.
- independent data is gathered for the random non-site locations and entered into the text file.
- the text file created contains the dependent variables, its (x, y) coordinates, and the specified independent variables.

^s Logistic input file

[#] Logistic Input

^k logistic input; input file; text file

input.txt

The diagram illustrates the relationship between four variables. At the bottom left is 'Site (1)/Non-site (0)'. A vertical line from this point goes up to 'Easting (x-coordinate)'. From 'Easting', a horizontal line goes right to 'Northing (y-coordinate)'. From 'Northing', a vertical line goes up to 'Independent Variables'. Additionally, a horizontal line goes directly from 'Site (1)/Non-site (0)' to 'Independent Variables'.

^s Sample text file
Sample Text File
^k sample text file

APPENDIX E

^s * ^k Entering Known Sites

Known sites can be entered in four ways: interactive graphics, tabular data entry, from a grid containing site data, or from a point coverage containing site data.

Interactive Graphics:

If you decide to add your known sites via graphical input with the mouse, two buttons will become activated: the pointer button, and the stop button. To begin entering points, click on the pointer button. Enter your known site locations by clicking on them in the View screen. To finish entering your points, press the stop button.

Tabular Data Entry:

In this method of input, you are prompted for the Northing and Easting coordinate of each site. When you are finished entering all of your known sites, press the cancel button at the next Northing/Easting prompt.

Taken From a Grid:

If you have a grid containing the known site locations, you may use this for input. You will be prompted for the name of the grid. At that time, the grid will be converted to a point coverage (if your sites are areal units, the centroid will be used as the (x, y) location). You will also be prompted as to whether all sites or a subset of the sites should be used for the analysis. If you select a subset, you must also enter which field contains the site information that decides whether it is used (i.e. site name, site type, cultural affiliation, etc.). You will be prompted to choose which sites you wish for the analysis and these will be selected as known sites.

Taken From a Point Coverage:

Similar to the "taken from a grid" option, you may use a point coverage containing site information for input. You will be prompted for the name of the site coverage. You will also be prompted as to whether all sites or a subset of the sites should be used for the analysis. If you select a subset, you must also enter which field contains the site information that decides whether it is used (i.e. site name, site type, cultural affiliation, etc.). You will be prompted to choose which sites you wish for the analysis and these will be selected as known sites.

Note for known sites taken from a grid or point coverage:

Due to the nature of the ASPS programming code, you will have the best success if your data adheres to these two rules: in grid data, make sure the numbers in the value field are incremental (1...n), and with point data, make sure all of your sites are within the extent of your study area and not part of the world polygon. If either of these rules is disregarded, the desired function may not run properly.

^s Entering known sites

* Known Sites

^k known sites; entering known sites

APPENDIX E

^s * Running Logistic Regression

At this time, the ASPS cannot run the logistic regression analysis directly. However, you may use the logistic input file to run the logistic regression function in Arc/Info or in a commercial statistics package.

If you choose to run the logistic regression function in Arc/Info, an AML has been written to assist you with this process. Change into your proper working directory (if necessary) and get into Arc/Info. Type *&run logistic* at the arc prompt. You will be asked to enter the name of the text file created in the *logistic input* command (include the .txt extension). Arc/Info will then run the logistic function. Since ArcView and Arc/Info cannot communicate directly, you must record the resulting coefficients for use in the *logistic surface* command in the ASPS.

```

xterm
Arc: &run logistic
Please enter the name of your input text file (including the .txt extension): thesis2.txt
Copyright (C) 1982-1995 Environmental Systems Research Institute, Inc.
All rights reserved.
GRID Version 7.0.3 (Mon Mar 13 22:21:55 PST 1995)

coef #      coef
-----
0          3.204
1          0.003
2         -0.019
3         -0.013

Record the coef # and its
corresponding coef to create
the logistic surface

point id      z      z error
-----
1          1.000      0.315
2          1.000      0.327
3          1.000      0.506
4          1.000      0.643
5          1.000      0.563
6          1.000      0.593
7          1.000      0.235
8          1.000      0.073
:          :          :

RMS Error = 0.446
Chi-Square = 4.771
Grid:

```

^s Running logistic regression

[#] Run Logistic

^{*} run logistic regression; running logistic regression

APPENDIX E

^s ^k Creating the Logistic Surface

The *logistic surface* command uses the coefficients created when running the logistic regression analysis, the independent grids, and the logistic equation to create a logistic surface. This surface ranges from 0 to 1 with higher values equaling a greater probability for site location. The equation used for this purpose (as defined by Arc/Info help):

$$\text{prediction surface} = 1 / (1 + e^{-(a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n)})$$

where $a_0, a_1, a_2 \dots a_n$ are the coefficients, and $x_1, x_2 \dots x_n$ are the corresponding independent grids

The *logistic surface* command follows this order:

- coefficient prompts (you will be prompted to enter each coefficient created in the logistic regression function).
- the logistic surface will be created.
- conversion to permanent grid (the surface process creates temporary grids that are not stored in your working directory, the *logistic surface* command automatically converts this temporary grid to a permanent grid that is stored in your working directory, you are prompted to enter a new name (if desired) and asked whether you would like this grid added to the view).

^s Creating a logistic surface

[#] Logistic Surface

^k logistic surface; surface; probability surface

APPENDIX E

^{s # K} Deductive Model Evaluation

A deductive site prediction model can be evaluated through the use of a histogram that compares the number of known sites that fall within the query area versus the number of random sites that fall within the query area.

The *evaluate by histogram* function, for a deductive model, follows this order:

- query grid prompt (enter the name of the permanent query grid created through the *unassisted modeling* and *convert to grid* commands).
- enter known sites (known sites can be entered through graphical or tabular methods, or they can be taken from a grid or point coverage).
- ArcView determines whether these known sites fall within the query area and adds the results to a table.
- ArcView selects a set of random locations, equal to the number of known sites, calculates whether these points fall within the query area, and adds these results to a table.
- The results of how many known sites versus how many random locations that fall within the query area are displayed in a chart.

A satisfactory deductive model would be reflected as a histogram where many of the known sites fall within the query area and few of the random sites fall within the query area.

^s Deductive model evaluation

[#] Deductive Evaluation

^K deductive evaluation; deductive model evaluation

APPENDIX E

^{s # k} Inductive Model Evaluation

An inductive site prediction model can be evaluated through the use of a histogram that displays the number of pixels that fall within each logistic surface category.

This *evaluate by histogram* command for the inductive model, requires the name of the logistic surface as input. ArcView creates the appropriate chart from this grid.

A satisfactory inductive model would be reflected as a histogram with few highly probable site locations and a surface that demonstrates that the archaeological sites fall within this area.

^s Inductive model evaluation

[#] Inductive Evaluation

^k inductive evaluation; inductive model evaluation

APPENDIX E

^{s # K} **Author Information**

The Archaeological Site Prediction System was created by Jessica Dolanski. All ASPS systems related questions and comments should be directed to her at:
dolansk3@pilot.msu.edu

^s Author information

[#] Author

^K author; author information

APPENDIX F

LIST OF SOFTWARE AND PROGRAMMING RESOURCES USED

Software:

1. Arc/Info 7.0
Environmental Systems Research Institute, Inc.
Redlands, CA
2. ArcView 3.0
Environmental Systems Research Institute, Inc.
Redlands, CA
3. Microsoft Windows 95 Help Authoring Kit
Microsoft Corporation
Redmond, Washington

Programming Resources:

Razavi, Amir H., J. Alexander, and V. Warwick, 1995, *ArcView/Avenue Programmer's Reference: Class Hierarchy Quick Reference and 101 Scripts*. Santa Fe, NM: Onword Press.

Razavi, Amir H., 1995, *ArcView Developer's Guide*. Santa Fe, NM: Onword Press.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Adovasio, J.M, G.F. Fry, J.D. Gunn, and R.F. Maslowski, 1975. Prehistoric and Historic Settlement Patterns in Western Cyprus. *World Archaeology*, 6(3): 338-364.
- Baker, Katherine G., 1988. Towards an Archaeological Methodology for Expert Systems. In *Computer and Quantitative Methods in Archaeology, 1987*. British International Reports 393, International Series. Oxford, England, 229-236.
- Barlaz, Eugene, 1995. Industry Standards: Defining the GIS of Tomorrow. In *GIS/LIS '95 Annual Conference and Exposition Proceedings Volume 1*. 39-43.
- Berry, Joseph K., 1996a. What is Object-Oriented Technology Anyway? *GIS World*, 9(9): 26.
- Berry, Joseph K., 1996b. Does Anyone Object? *GIS World*, 9(11): 32.
- Binford, Lewis R., 1982. The Archaeology of Place. *Journal of Anthropological Archaeology*, 1: 5-31.
- Binford, Lewis R., 1989. *Debating Archaeology*. United Kingdom: Academic Press, Inc.
- Bossler, John, et.al., 1995. Software Trends: The Web, Ease of Use. *Geo Info Systems*. 5(8): 12-14.
- Brandt, Roel, B.J. Groenewoult, and K.L. Kavamme, 1992. An Experiment in Archaeological Site Location: Modeling in the Netherlands Using GIS Techniques. *World Archaeology*, 24(2): 268-281.
- Breckenridge, Mary B., 1983. *Age, Time, and Fertility: Applications of Exploratory Data Analysis*. New York, New York: Academic Press, Inc.
- Brown, C.Marlin, 1988. *Human-Computer Interface Design*. Sunnyvale, CA: Xerox Corporation.
- Brown, Daniel G., 1994. Predicting Vegetation Types at Treeline Using Topography and Biophysical Disturbance Variables. *Journal of Vegetation Science*, 5: 641-656.

- Charmichael, David L., 1990. GIS Predictive Modeling of Prehistoric Site Distributions in Central Montana. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. London: Taylor and Frances, 216-225.
- Christodoulou, Demetris, 1959. The Evolution of the Rural Land Use Pattern in Cyprus. *World Land Use Survey Monograph 2*, Cornwall, England: Geographical Publications Limited.
- Dahlgren, Darwin A., 1995. Object-Oriented Programming: Buzzword or Benefit? *Geo Info Systems*, 5(6): 48-51.
- Dangermond, Jack, et.al., 1996. Top 10 People of 1996: Crafting the Future of GIS. *GIS World*, 9(7): 34-42.
- Dann, Mark A. and R.W. Yerkes, 1994. Use of Geographic Information Systems for the Spatial Analysis of Frankish Settlements in the Korinthia, Greece. In *Regional Studies of the Aegean*, Lanham, MD: University Press of America, 291-315.
- Doran, Jim, 1988. Expert Systems and Archaeology: What Lies Ahead? In *Computer and Quantitative Methods in Archaeology, 1987*. British International Reports 393, International Series, Oxford, England, 237-242.
- Doxiadis, Constantinos A., 1970. Ekistics, the Science of Human Settlements. *Science*, 170(3956): 393-404.
- Ebert, James I., 1992. *Distributional Archaeology*. Albuquerque: University of New Mexico Press.
- Ebert, J.I., E.L. Camilli, and M.J. Berman, 1996. GIS in the Analysis of Distributional Archaeological Data. In *New Methods, Old Problems*, edited by Herbert D.G. Maschner, Center for Archaeological Investigations Occasional Paper No. 23, Carbondale: Southern Illinois University, 25-37.
- Eklundh, Lars and Ulrik Mårtensson, 1995. Rapid Generation of Digital Elevation Models from Topographic Maps. *International Journal of Geographic Information Systems*, 9(3): 329-340.
- Environmental Systems Research Institute, Inc., 1992. *Arc/Info Help*. Redlands, CA: Environmental Systems Research Institute, Inc.
- Fotheringham, A.S., 1992. Exploratory Spatial Data Analysis and GIS. *Environment and Planning*, 24(12): 1675-1678.

- Hadjiraftis, Andreas, 1996. Department of Lands and Surveys, Nicosia, Cyprus. Personal Communication.
- Hadjisavvas, S., 1977, The Archaeological Survey of Paphos, A Preliminary Report. *Dept. of Antiquities Annual Report, 1977*, 222-231.
- Harris, T. and G. Lock, 1995. Toward an Evaluation of GIS in European Archaeology: the Past, Present and Future of Theory and Applications. In *Archaeology and Geographical Information Systems*, edited by Gary Lock and Zoran Stancic. London: Taylor and Frances, 349-366.
- Hasenstab, R.J. and B. Resnick, 1990. GIS in Historical Predictive Modeling: The Fort Drum Project. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. Bristol, Pennsylvania: Taylor and Frances, 284-306.
- Herscher, Ellen, 1995. Archaeology in Cyprus. *American Journal of Archaeology*, 99: 257-294.
- Hopkins, Lewis D., 1977. Methods for Generating Land Suitability Maps: A Comparative Evaluation. *Journal of the American Institute of Planners*, 43(4): 386-400.
- Hutchinson, M.F., 1989. A New Procedure for Gridding Elevation and Stream Line Data with Automatic Removal of Spurious Pits. *Journal of Hydrology*, 106: 211-232.
- Jacobson, Esther, 1994. Patterns on the Steppe: Applying GIS to the Archaeology of the Altay Mountains. *Geo Info Systems*, 4(3): 33-45.
- Johnson, Gregory A., 1977. Aspects of Regional Analysis in Archaeology. *Annual Review of Anthropology*, 6: 479-508.
- Knapp, Bernard, 1995. The Sydney Cyprus Survey Project (SCSP)-1995 Season. *Old World Archaeology Newsletter*, 9(1): 10-18.
- Knapp, Bernard, 1996. Department of Archaeology, University of Glasgow, Scotland. Personal Communication.
- Kohler, Timothy A. and S.C. Parker, 1986. Predictive Models for Archaeological Resource Location. In *Advances in Archaeological Method and Theory*, Vol. 9, edited by M.B. Schiffer. New York: Academic Press, 397-452.
- Kopp, Steve, 1997. Product Specialist, Spatial Analyst Team, Environmental Systems Research Institute, Inc. Personal Communication.

- Kvamme, Kenneth L., 1990. One-Sample Tests in Regional Archaeological Analysis: New Possibilities Through Computer Technology. *American Antiquity*, 55(2): 367-381.
- Kvamme, Kenneth L., 1995. A View from Across the Water: The North American Experience in Archaeological GIS. In *Archaeology and Geographical Information Systems*, edited by Gary Lock and Zoran Stancic. London: Taylor and Frances, 1-14.
- Lee, Y.C. and F.L. Chin, 1995. An Iconic Query Language for Topological Relationships in GIS. *International Journal of Geographical Information Systems*, 9(1): 25-46.
- Lock, Gary R. and Trevor M. Harris, 1991. Integrating Spatial Information in Computerized Sites and Monuments Records: Meeting Archaeological Requirements in the 1990's. In *Computer and Quantitative Methods in Archaeology, 1990*. British International Reports 565, International Series. Oxford, England, 165-173.
- Lock, Gary R. and Trevor M. Harris, 1992. Visualizing Spatial Data: The Importance of Geographic Information Systems. In *Archaeology and the Information Age: A Global Perspective*, edited by Paul Reilly. London: Routledge, 81-96.
- Makarovic, B., 1977. Composite Sampling for Digital Terrain Models. *ITC Journal*, (3): 406-431.
- Marble, Duane F., 1990. The Potential Methodological Impact of Geographic Information Systems on the Social Sciences. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. Bristol, Pennsylvania: Taylor and Frances, 9-21.
- Marble, Duane F., 1996. *The Human Effort Involved in Movement Over Natural Terrain*. Part I of the Final Report Submitted Under National Park Service Contract 6115-4-8031.
- Miller, P., 1995. How to Look Good and Influence People: Thoughts on the Design and Interpolation of an Archaeological GIS. In *Archaeology and Geographical Information Systems*, edited by Gary Lock and Zoran Stancic. London: Taylor and Frances, 349-366.
- Obermeyer, Nancy J., 1995. The Hidden GIS Technocracy. *Cartography and Geographic Information Systems*, 22(1): 78-83.

- Pereira, J.M.C. and R.M. Itami, 1991. GIS-Based Habitat Modeling Using Logistic Multiple Regression: A Study of the Mt. Graham Squirrel. *Photogrammetric Engineering and Remote Sensing*, 57(11): 1475-1486.
- Pinson, Lewis J. and R.S. Wiener, 1990. *Applications of Object-Oriented Programming*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- Ravden, Susannah and G. Johnson, 1989. *Evaluating Usability of Human-Computer Interfaces: A Practical Method*. Chichester, England: Ellis Horwood Limited.
- Razavi, Amir H., 1995. *ArcView Developer's Guide*. Santa Fe: OnWord Press.
- Savage, Stephen H., 1990. GIS in Archaeological Research. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. Bristol, Pennsylvania: Taylor and Frances, 22-32.
- Shields, Deborah J., S.W. Todd, and D.D. Brown, 1996, Determining Match Probabilities Between Two Mineral Location Databases Using Logistic Regression. In *GIS/LIS '96 Annual Conference and Exposition Proceedings*. 421-434.
- Shermer, Shirley J. and J.A. Tiffany, 1985. Environmental Variables as Factors in Site Location: An Example from the Upper Midwest. *Midcontinental Journal of Archaeology*, 10(2): 215-240.
- Symeonoglou, S., 1972. Archaeological Survey in the Area of Phlamoudhi, Cyprus. *Dept. of Antiquities Annual Report, 1972*, 187-198.
- Thomas, David Hurst, 1989. *Archaeology*. Fort Worth, Texas: Holt, Rinehart, and Winston, Inc.
- Tobler, Waldo, 1988. Resolution, Resampling and All That. In *Building Databases for Global Science*, edited by H. Mounsey and R.F. Tomlinson. London: Taylor and Francis, 129-137.
- Toumazou, M.K., 1996. Davidson College, Davidson, North Carolina. Personal Communication.
- Toumazou, M.K., R.W. Yerkes, and P.N. Kardulias, 1995. *Athienou Archaeological Project: Interim Report of Investigations in the Malloura Valley 1990-1995*.
- van Leusen P.M., 1995. GIS and Archaeological Resource Management: A European Agenda. In *Archaeology and Geographical Information Systems*, edited by Gary Lock and Zoran Stancic. London: Taylor and Frances, 27-42.

- van Leusen, P.M., 1996. Locational Modeling in Dutch Archaeology. In *New Methods, Old Problems*, edited by Herbert D.G. Maschner, Center for Archaeological Investigations Occasional Paper No. 23, Carbondale: Southern Illinois University, 177-197.
- Warren, Robert E., 1990a. Predictive Modeling in Archaeology: A Primer. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. London: Taylor and Frances, 90-111.
- Warren, Robert E., 1990b. Predictive Modeling of Archaeological Site Location: A Case Study in the Midwest. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. London: Taylor and Frances, 201-215.
- Willey, G.R., 1953. Prehistoric Settlement Patterns in the Virú Valley, Peru. *Bureau of American Ethnology Bulletin Number 155*. Washington DC: Bureau of American Ethnology.
- Williams, I., W.F. Limp, and F.L. Briuer, 1990. GIS and Exploratory Data Analysis. In *Interpreting Space: GIS and Archaeology*, edited by K.M.S. Allen, S.W. Green, and E.B.W. Zubrow. London: Taylor and Frances, 239-273.
- Williams, Matthew P., Posted Date: unknown. GIS in Archaeology. *World Wide Web*, (<http://www.utexas.edu/depts/grg/gcraft/notes/gisapps/gisarch2.html>), Download Date: May 28, 1997.
- Wilson, J.D., 1996. CAD/GIS Conversion: Creating the Next Generation GIS. *GIS World*. 9(5): 36-39.
- Wood, John J., 1978. Optimal Location in Settlement Space: A Model for Describing Locational Strategies. *American Antiquity*. 43(2): 258-270.
- Worboys, Michael F., 1994. Object-Oriented Approaches to Geo-Referenced Information. *International Journal of Geographical Information Systems*, 8(4): 385-399.

MICHIGAN STATE UNIV. LIBRARIES



31293020581876