

University of Redlands

InSPIRe @ Redlands

MS GIS Program Major Individual Projects

Theses, Dissertations, and Honors Projects

2015

Water Budget Analysis of Coachella Valley Aquifer

René Pretorius

University of Redlands

Follow this and additional works at: https://inspire.redlands.edu/gis_gradproj



Part of the [Geographic Information Sciences Commons](#)

Recommended Citation

Pretorius, R. (2015). *Water Budget Analysis of Coachella Valley Aquifer* (Master's thesis, University of Redlands). Retrieved from https://inspire.redlands.edu/gis_gradproj/240



This work is licensed under a [Creative Commons Attribution 4.0 License](#).

This material may be protected by copyright law (Title 17 U.S. Code).

This Thesis is brought to you for free and open access by the Theses, Dissertations, and Honors Projects at InSPIRe @ Redlands. It has been accepted for inclusion in MS GIS Program Major Individual Projects by an authorized administrator of InSPIRe @ Redlands. For more information, please contact inspire@redlands.edu.

University of Redlands

Water Budget Analysis of Coachella Valley Aquifer

A Major Individual Project submitted in partial satisfaction of the requirements
for the degree of Master of Science in Geographic Information Systems

by

René Pretorius

Ruijin Ma, Ph.D., Committee Chair

Fang Ren, Ph.D.

December 2015

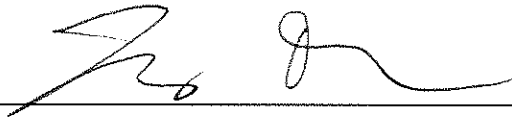
Water Budget Analysis of Coachella Valley Aquifer

Copyright © 2015

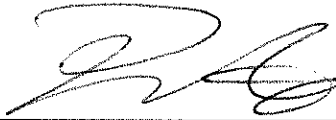
by

René Pretorius

The report of René Pretorius is approved.

A handwritten signature in black ink, appearing to be 'Fang Ren', written above a horizontal line.

Fang Ren, Ph.D.

A handwritten signature in black ink, appearing to be 'Ruijin Ma', written above a horizontal line.

Ruijin Ma, Ph.D., Committee Chair

December 2015

Acknowledgements

I want to start by thanking the Morrison Family Trust for providing the funds that made it possible for me to attend the MS GIS program. I am grateful to Mr. David Robinson, the GIS Coordinator at Coachella Valley Economic Partnership (CVEP), who conceived the project.

Dr. Ruijin Ma, my advisor, who assisted me and helped to keep me on track. I am grateful for his suggestions and guidance throughout the project. I would like to thank Dr. Fang Ren who took the time to be my second reader. Also, I am grateful to all the MS GIS program faculty, you provided assistance and guidance throughout the whole program.

My husband Rudi Pretorius, who supported me and provided unending love and encouragement throughout this program. My parents Hennie and Jeanne Smit that stood by me every step of the way and their teachings and principals guided me throughout this year. My in-laws Rudi and Erika Pretorius, who gave support and understanding to both me and Rudi this last year.

My classmates in Cohort 27 (Alex, Bryce, Pat, and Colin) and my friends from other cohorts (Neel, Shilpi, Nick), you helped me through all the tough times and good times, and made this a memorable year.

Abstract

Water Budget Analysis of Coachella Valley Aquifer

by

René Pretorius

The Coachella Valley is a desert region located in Southern California with high temperatures and low variable rainfall. It is also an oasis due to its aquifer, which is in both the Whitewater River and Salton Sea catchment regions. The water from this aquifer is used for a variety of applications by the Coachella Valley community, but it is a finite water resource and the natural recharge in this desert region is low. If the aquifer is depleted it can collapse, which would lead to infrastructure damage and affect the local economy. This study used GIS to create models and tools to assist the Coachella Valley Economic Partnership in analyzing aspects of the water budget, and what the effect of land cover change will have. These models and tools were used for estimating infiltration, change an area of interest's land cover classifications, and generated water depth surfaces. These outputs made it possible to create maps showing these results, and the results were visualized in 2-D and 3-D to find possible trends.

Table of Contents

Chapter 1 – Introduction	1
1.1 Client.....	2
1.2 Problem Statement	2
1.3 Proposed Solution	2
1.3.1 Goals and Objectives	2
1.3.2 Scope.....	3
1.3.3 Methods.....	3
1.4 Audience	3
1.5 Overview of the Rest of this Report	4
Chapter 2 – Background and Literature Review	5
2.1 Coachella Valley Aquifer	5
2.2 Rainfall Water Distribution.....	6
2.2.1 Runoff	6
2.2.2 Infiltration	7
2.2.3 Evaporation.....	7
2.3 Hydrography	7
2.3.1 Aquifer	8
2.3.2 Aquifer Discharge.....	8
2.3.3 Aquifer Recharge.....	9
2.4 Hydrological Models and GIS	10
2.5 Summary	10
Chapter 3 – Systems Analysis and Design.....	12
3.1 Problem Statement	12
3.2 Requirements Analysis	12
3.2.1 Functional Requirements	12
3.2.2 Non-Functional Requirements	13
3.3 System Design	14
3.4 Project Plan	15
3.4.1 Phases.....	15
3.4.2 Deliverables	17
3.5 Summary	17
Chapter 4 – Database Design.....	19
4.1 Conceptual Data Model	19
4.2 Logical Data Model	20
4.3 Data Sources	21
4.4 Data Scrubbing and Loading	21
4.4.1 Elevation Data Preprocessing	22
4.4.2 Soil Data Preprocessing	22
4.4.3 Land Cover Data Preprocessing.....	23
4.4.4 Wells Data Preprocessing	24
4.4.5 Hydrography Data Preprocessing	25
4.5 Summary	25

Chapter 5 – Implementation	27
5.1 Infiltration using Curve Number	27
5.1.1 Curve Number Calculation.....	28
5.1.2 Rainfall Extraction	29
5.1.3 Slope.....	30
5.1.4 Calculating β	30
5.1.5 Infiltration Calculation	31
5.2 Land Cover Change (LCC)	32
5.3 Water Level Estimation.....	34
5.3.1 Well Data Preprocessing	35
5.3.2 Water Surface Interpolation	35
5.4 Summary	37
Chapter 6 – Results and Analysis	39
6.1 Infiltration.....	39
6.1.1 Infiltration Tools.....	39
6.1.2 Infiltration Results	44
6.2 Land Cover Change Tool	47
6.3 Water Depth Surfaces.....	49
6.3.1 Water Depth Surface Tools	49
6.3.2 Water Depth Surfaces Results	51
6.3.3 3-D Visualization	55
6.4 Summary	59
Chapter 7 – Conclusions and Future Work	60
Works Cited	63
Appendix A. Python script for finding PLSS	67
Appendix B. Infiltration Python scripts and Models.....	70
Appendix C. Land Cover Change Tool.....	74
Appendix D. Unit Conversion Model and Tool	76
Appendix E. Average Water Depth Surface Tool.....	77
Appendix F. Monthly Water Depth Surfaces Tool.....	85

Table of Figures

Figure 1.1 The Study Area.....	1
Figure 2.1 Coachella Valley Aquifers	6
Figure 3.1 System Design.....	15
Figure 3.2 Project Phase	16
Figure 4.1 Conceptual Model	19
Figure 4.2 Logical Model	20
Figure 4.3 Table to Point Feature Class.....	25
Figure 5.1 Infiltration Estimation Workflow	28
Figure 5.2 Curve Number Calculation Workflow	28
Figure 5.3 Precipitation Extraction.....	29
Figure 5.4 Slope Calculation.....	30
Figure 5.5 β Coefficient	31
Figure 5.6 Land Cover Change Workflow	33
Figure 5.7 Python Script Workflow.....	33
Figure 5.8 Water Level Estimation Workflow	34
Figure 5.9 Depth to Water Adjustment Workflow	35
Figure 5.10 Water Depth Surface Workflow	36
Figure 6.1. Create Curve Number Raster User Interface.....	39
Figure 6.2 Curve Number Raster	40
Figure 6.3 Calculating Infiltration Parameters User Interface.....	41
Figure 6.4 Results from Step Two of the Infiltration Calculation	42
Figure 6.5 Calculate Final Infiltration User Interface, Variation 1 (Single Instance).....	43
Figure 6.6 Calculate Final Infiltration User Interface, Variation 2 (Workspace).....	43
Figure 6.7 Infiltration from PRISM 30 Year Precipitation Averages, January to June: a) January, b) February, c) March, d) April, e) May, f) June.	45
Figure 6.8 Infiltration from PRISM 30 Year Precipitation Averages, July to December: a) July, b) August, c) September, d) October, e) November, f) December.	46
Figure 6.9 Comparison between Infiltration (Left) and Precipitation (Right) for December, 2013 and 2014.	47
Figure 6.10 Land Cover Change Tool User Interface with Selected Parcels in Blue	48
Figure 6.11 An Example of Changing Cultivated Crops and Hay/Pastures Areas Changed to Developed, High Intensity	48
Figure 6.12 Unit Check Tool User Interface.	49
Figure 6.13 WDS Generation Tool User Interface, Variation 1 (Monthly WDS).....	50
Figure 6.14 WDS Generation Tool User Interface, Variation 2 (Average Monthly WDS)	51
Figure 6.15 Well Distribution in Three of the Four Coachella Valley Subbasins	52
Figure 6.16 WDS Results for 2012 and 2014, for January (left) and July (right).	53
Figure 6.17 Three Year Average WDS of January (left) and July (right)	54
Figure 6.18. Differences between WDS's	55
Figure 6.19 Two Side Views of all the Wells and their Measurements (1926 to 2015)	56
Figure 6.20 Closer Look at Two Wells.....	57
Figure 6.21 Close-up of Wells 04S06E18P01S and 04S06E22C01S	57

Figure 6.22 Three Wells Selected for Profile View	58
Figure 6.23 Depth to Water Profile	58
Figure B-1 Create CN Raster	70
Figure B-2 Calculate α	70
Figure B-3 Create β Raster.....	71
Figure B-4 Calculate Maximum Retention	71
Figure B-5 Calculate Initial Abstraction	72
Figure B-6 Calculate Infiltration Equation Parameters	72
Figure B-7 Infiltration Calculation for Single Instance.....	73
Figure B-8 Infiltration Calculation for Database	73
Figure D-1 Water Depth Measurement Unit Cnversion	76

List of Tables

Table 3-1	Functional Requirements	13
Table 3-2	Non-Functional Requirements.....	14
Table 4-1	Data Sources	21
Table 4-2	Hydrological Soil Groups.....	23
Table 5-1	Estimation of β Coefficient According to Land Use	31

List of Acronyms and Definitions

3-D	Three Dimensional
AOI	Area of Interest
CN	Curve Number
CVEP	Coachella Valley Economic Partnership
CVWD	Coachella Valley Water District
DEM	Digital Elevation Model
DWR	Department of Water Resources
GIS	Geographic Information Systems
GW	Groundwater
HSG	Hydrological Soil Group
HU	Hydrological Unit
I_a	Initial Abstraction
LLC	Land Cover Change
NAD	North American Datum
NHD	National Hydrology Dataset
NLCD	National Land Cover Dataset
NRCS	National Research Conservation Services
PRISM	Parameter-elevation Regressions on Independent Slopes Model Datasets
SCS	Soil Conservation Service
SSURGO	Soil Survey Geographic Database
USGS	United States Geological Survey
UTM	Universal Transverse Mercator
WGS	World Geodetic System

Chapter 1 – Introduction

The Coachella Valley is a desert region in Southern California, but also an oasis due to its aquifer. The valley extends for approximately 45 miles, averaging 15 miles wide, into Riverside County southeast from the San Bernardino Mountains and to the northern shore of the Salton Sea. The Coachella Valley study area is shown in Figure 1.1.

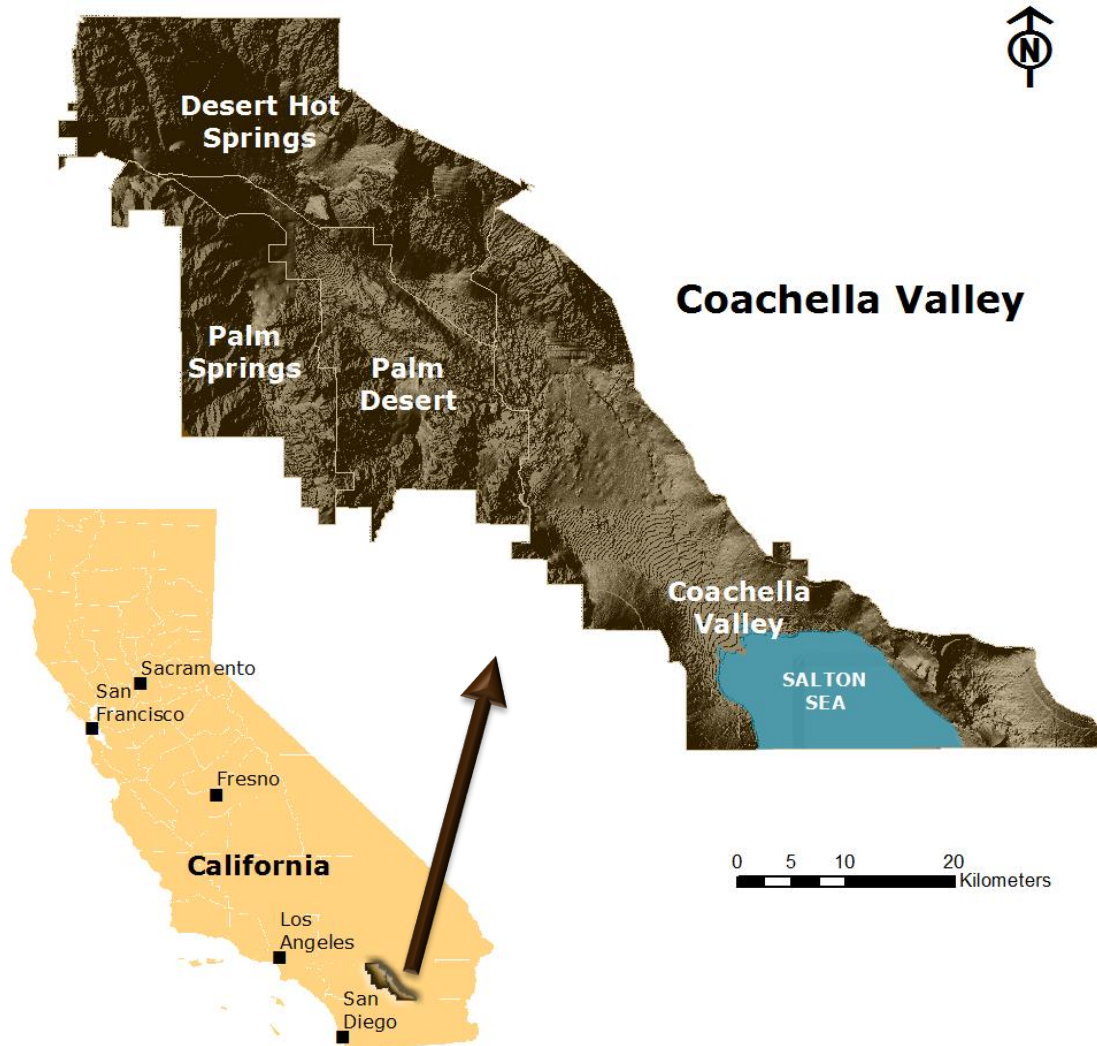


Figure 1.1 The Study Area

The valley is the northernmost extent of a vast trough that includes the Salton Sea, the Imperial Valley, and the Gulf of California. It is bounded on the west by the San Jacinto and Santa Rosa Mountains, and on the north and east by the Little San Bernardino Mountains. The aquifer is largely loose material ranging from clay to sand to gravel (unconsolidated) and is moderately to highly permeable. It is recharged both naturally

and artificially; the former through percolation of surface water, drainage of intermittent streams, and runoff from nearby mountain ranges (June, 2008).

This aquifer is the main source of all drinking and household water for the region. The majority of the golf courses in the area use the aquifer for irrigation, and it also supplies water for up to a third of the agricultural applications in the area. Other sources of water in the area are recycled and imported water from the Coachella Canal, which draws from the Colorado River (CVWD, n.d.).

The Coachella Valley Economic Partnership (CVEP) wanted to determine if tools could be created to analyze the Coachella Valley water budget – discharge and recharge of aquifer – with specific interest given to three groups impacting the aquifer and area's economy: golf courses, agriculture, and residential users. The main focus of this project was to aid the CVEP in creating a system containing these tools.

1.1 Client

The project client was CVEP, a nonprofit economic development organization devoted to attracting, retaining, and expanding business, as well as to developing a highly skilled workforce to enhance the economy and quality of life of the region. The client required desktop GIS-based tools to produce easy-to-understand water budget analysis results. The point-of-contact at CVEP was David Robinson, the GIS Coordinator.

1.2 Problem Statement

The aquifer of the Coachella Valley is a finite water source located in a desert area. Overuse of the aquifer without sufficient recharge can lead to its collapse, which would not only cause a water shortage problem and damage to property and infrastructure, but could also severely affect the local economy. CVEP does not currently have a method to analyze the aquifer discharge, recharge, and the parameters influencing it, nor do they currently have a method to predict the effect land use change would have on the water budget.

1.3 Proposed Solution

The proposed solution for this project involved the development of two geoprocessing models: a hydrological model to analyze the water budget (water use and aquifer recharge) and the parameters influencing it, and a model to predict change in the water budget should land use change. Supplementary maps to showcase some of the results from these models were created.

1.3.1 Goals and Objectives

The goal of this project was to develop a solution that would facilitate analysis of the water budget and assess the impact of land use change.

The main objectives of the project:

- Create a tool to aid in the water recharge and discharge rate analysis;
- Allow users to dynamically assess how land use change might affect the water budget.

1.3.2 Scope

This project and the associated results and models were limited to the Coachella Valley in California. The project scope included the creation of a water budget analysis and a land cover change analysis system. This system included spatial query tools to compute water discharge and recharge rate, as well as this rate's change when land use changes. The tools were developed with Python and ArcGIS Model Building for desktop use. The database was created to store the data provided to the client and the outputs created by the system.

1.3.3 Methods

The data used in this project were clipped to the Coachella Valley area, and projected to World Geodetic System Datum 1984 (WGS84) Universal Transverse Mercator (UTM) Zone 11 North. The data used consisted of well data from the Department of Water Resources, soil data from Soil Survey Geographic Database (SSURGO), geology data from United States Geological Survey (USGS) Mineral Resources, hydrography features from the National Hydrography Dataset (NHD), land cover from the National Land Cover Dataset (NLC), and rainfall from Parameter-Elevation Regressions on Independent Slopes Model Dataset (PRISM).

The project was done using Esri ArcGIS desktop 10.3, ArcGIS Model Builder, Python scripts, and Esri Arc Hydro tools 2.0. Esri Arc Hydro was used for initial processing and creation of some datasets required for later analysis. A combination of Python scripts and Model Builder models were used to create models to calculate the infiltration and estimate the groundwater level.

Model Builder application and Python scripts were used to create a land use change analysis tool. This tool allows the user draw a polygon in the study area and select the new land use for the area defined by the polygon. Using this information, the tool then calculates the new recharge rates and compares them to the original.

1.4 Audience

This project and its tools are aimed at the client, mainly the GIS Coordinator. This paper discusses how GIS was used to create tools and maps for water budget analysis, and the paper assumes familiarity with GIS, but little to no familiarity with water resources. The tools created during this project are intended for use by people familiar with GIS, while the final output maps are intended for a wider audience who may have no familiarity with GIS or water resources.

1.5 Overview of the Rest of this Report

The rest of the report consist of chapters' two to seven. Chapter Two, the literature review, looks at the different approaches and results of projects with similar problems. Chapter Three offers a description of the problem statement and the requirements analysis for completing this project. This is followed by an outline of the system design and project plan. Chapter Four details the database design, starting with the details about the conceptual data model, and going into the logical data model. This is followed by information about the data: the data sources, how the data was collected, and how the data were formatted initially. The details of the processes, tasks, and workflow to complete both functional and non-functional requirements are discussed in Chapter Five. Chapter Six contains the details about the results and analysis from the system. Chapter Seven provides conclusions and possible future expansions upon this project.

Chapter 2 – Background and Literature Review

The knowledge that water resources in arid and semi-arid region are limited drives the search for analysis and calculation of the water budgets in these areas. Hydrologic systems are influenced by many natural and human factors. In the hydrological cycle rainfall is a major component, especially for surface runoff, infiltration, and ground water recharge. Human factors relating to the discharge and recharge of ground water include the pumping of water from wells and land use.

Hydrological models were first created as stand-alone software and were not well integrated with GIS, but as the advantages of GIS became clear integration between the two fields started. Hydrological modeling is even more important in arid regions where water is scarce.

2.1 Coachella Valley Aquifer

Many arid and semi-arid areas have limited surface water resources, with the ground water being the main water source for agriculture, industrial, and domestic water supplies (Kumar, 2003). Although the Coachella Valley is a desert region, it is also an oasis due to its aquifer. This has led to numerous groups using the aquifer's water, and not all of these individuals acknowledge the recharge necessary to sustain the aquifer. The need to check usage and recharge is even more vital in light of the drought in recent years (James, 2015). Three main aquifer user groups are discussed in section 2.3.2: golf courses, agriculture, and residential.

The Coachella Valley aquifer is generally unconfined and composed of coarse-grained sediments. The San Andreas Fault acts as natural barrier to keep groundwater from migrating to other aquifers. This valley is grouped into four ground water subbasins: Indio, Mission Creek, Desert Hot Springs, and San Gorgonia. Figure 2.1 shows the four subbasins that make up the Coachella Valley aquifer, and where they lie in relation to the area's watersheds. The Colorado River is an important source for the Coachella Valley aquifer recharge. There are two Colorado River water conduits which deliver water to the Coachella Valley: the Colorado River Aqueduct and the Coachella Canal (June, 2008; Weisheit, 2008).

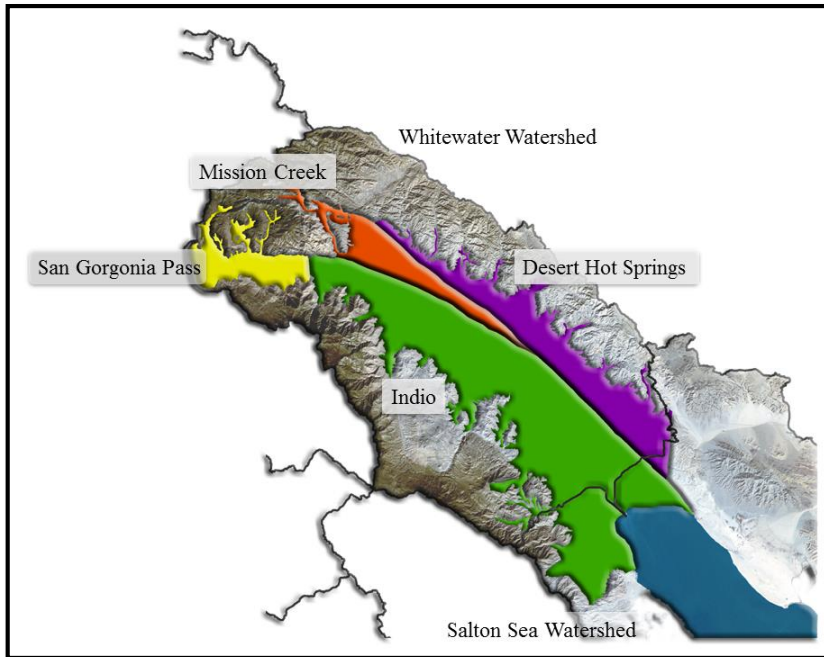


Figure 2.1 Coachella Valley Aquifers

Aquifers are geologic formations, groups of hydraulically connected formations that store and transmit significant quantities of portable ground water. They are the focal point of any hydrogeologic conceptual site model. Intergranular aquifers develop in unconsolidated sediment, composed of mixed grains varying in size and shape such as clay, silt, sand, and gravel. This is the Coachella Valley’s aquifer type. Other types of aquifers include alluvial, basin-fill, and glacial-deposit aquifers (Kresic & Mikszewski, 2013).

2.2 Rainfall Water Distribution

Rainfall in a specific area is intercepted by trees, shrubs, and other vegetation or ends up striking the ground surface to become overland flow, subsurface flow, and groundwater flow. Rainfall distribution between summer and winter varies greatly in arid regions. Generally arid regions also have low annual rainfall, and the monthly and yearly rainfall distributions vary (FAO, 1989; Hernadez et al., 2000; Hudson, n.d.).

2.2.1 Runoff

When rain hits the soil, the amount of surface water that does not directly infiltrate, evaporate, become lost to surface storage, or intercepted by plants, is turned into runoff (Poullain, 2012). Runoff from rainfall is an important variable in hydrological modelling and consists of two main elements: base flow with origins in groundwater, and surface runoff which is the accumulation of rainfall that drains to streams (Hammouri & El-Naqa, 2007).

There are two types of runoff. Hortonian flow, otherwise known as infiltration-excess runoff, occurs when rainfall intensity exceeds the rate at which water can infiltrate

the soil. Saturation-excess runoff, according to Schneiderman, occurs when rain/snowmelt encounters soils that are saturated due to a perched water table. This water table forms when the infiltration front reaches a zone of low transmission (Schneiderman et al., 2007).

2.2.2 Infiltration

Infiltration occurs when water enters the soil and is one of the most common methods by which groundwater recharge occurs in arid regions. Soil infiltration rate is the velocity at which water enters the soil. These rates are influenced by soil characteristics, vegetation, and land use practices (Schneiderman et al., 2007; USDA NRCS, 2008).

The soil characteristics include soil texture and clay mineralogy. Soil texture is a description of the pore spaces between soil particles. The size of the pore spaces determine how quickly water will move through it. Water will move more quickly through sandy soil with larger pore spaces than through clayey soil with small pores. Soils can contain different amounts and types of clay minerals. Shrinkage cracks are developed in many clayey soils when they dry. These cracks create a direct path for water to enter the soil. Clay soils with these cracks will exhibit a high infiltration capacity, but when these cracks are not present the infiltration rate is characteristically slow (USDA NRCS, 2008).

The Soil Conservation Services, now known as Natural Resource Conservation Services (NRCS), proposed a relationship for estimating cumulative infiltration. This relationship depends on the amount of precipitation (rainfall and snowmelt) received by a basin, the initial losses, and the maximum retention potential of the soil. The basis for this was the NRCS Curve Number method (Craciun & Haidu, 2011). Craciun investigated and utilized this method in a GIS methodology for indirect estimation of soil moisture. The first part of the methodology created an algorithm to estimate the infiltration and to develop a GIS module to run this algorithm.

2.2.3 Evaporation

When the NRCS Curve Number method is used to calculate the infiltration, initial losses are taken into consideration. One of the contributing factors to these losses is evapotranspiration, which can be divided up into evaporation and transpiration.

Evaporation is when a liquid water changes into vapor. This water is then removed from the source such as the soil surface, wet vegetation, pavement, or water body. Transpiration, which is the vaporization of liquid water within a plant, occurs simultaneously with evaporation. Both of these process depend on solar radiation, air temperature, relative humidity, amount of rainfall, and wind speed. The transpiration rate is also influenced by factors such as crop characteristics, cultivation methods, and environmental aspects (Zotarelli & Dukes, 2010).

2.3 Hydrography

Hydrography is the scientific study of the movement, distribution, and quality of water on Earth, including the hydrologic cycle, water resources, and environmental watershed sustainability.

Water budget is the difference between water input and water output, resulting in changes in stored water. Water budget equations can be written in terms of volumes (fixed time interval), fluxes (volume per time), and flux densities (volume per unit area of land surface per time, mm per day). The complexity of determining the water budget depends on many natural and anthropogenic factors. Climate, hydrography and hydrology, geologic and geomorphic characteristics, hydrogeologic characteristics of the surficial soils and subsurface porous media, land cover/use, presence and operation of artificial surface water reservoirs, surface/groundwater withdrawals for consumptive use and irrigation, and water waste management must all be taken into consideration (Kresic & Mikszewski, 2013).

2.3.1 Aquifer

Aquifers are classified as either confined or unconfined. A confined aquifer is below the land surface and saturated with water. This type of aquifer has layers of impermeable material above and below the aquifer. An unconfined aquifer's upper water surface – the water table – is at atmospheric pressure, and thus can rise and fall. Water seeps into it from the ground surface directly above the aquifer. Since this type of aquifer is closer to the Earth's surface compared to a confined aquifer, it is impacted by drought conditions sooner (USGS, n.d.).

Water in unconfined aquifers is subject to losses due to plant uptake and evaporation. When a well is constructed in an unconfined aquifer, the water level will temporarily remain at the same level at which it was found, but can change later due to many factors. The surface of the zone of saturation is defined by the well water level. This surface has the same pressure everywhere as the atmospheric pressure (NGWA, n.d.).

2.3.2 Aquifer Discharge

Aquifer discharge can happen many ways, including groundwater being pumped out through wells. Well and borehole data are common datasets available for use in groundwater related projects. The permits for well construction are issued and documented by different organizations, from national to city level. Numerous agencies have databases that contain information on well use and location. This information is important for the development of groundwater management plans and groundwater simulation models (Strassberg et al., 2010).

With 123 different golf courses, Coachella Valley is a world-renowned golf destination and one of the most popular golf vacation spots in California. According to the Los Angeles Times, these golf courses use about 37 billion gallons of water annually (Stevens, 2014). Of the 123 golf courses, 53 use water from treated waste/recycled water or the Coachella Canal. The remaining 70 courses rely on wells pumping groundwater. (CVWD, n.d.; James, 2014).

Agriculture is a major industry in the valley. Some of the top crops are citrus, grapes, bell peppers, and dates. It is also a primary date production region and responsible for almost all of the dates grown in the United States (Austin, 2012; CVWD, n.d.). Regional agriculture was most successful in the lower Coachella Valley. Initially it relied on the aquifer, but this proved to be unpredictable considering that droughts would deplete it. The Coachella Canal, completed in 1984, brought more stability and reliability farming

(Patten, 2014). Currently in the fourth year of a drought, mandatory reductions in portable water have been ordered. This has resulted in farmers being forced to become more water efficient by implementing better technologies and strategies. However, many farmers have been forced to leave fields fallow and cut jobs (Marx & James, 2015).

Residents of this the Coachella Valley are accustomed to using large amounts of water, most of which comes directly from the aquifer. Besides drinking water, the residents use the water for lawns, turf-filled gated communities, and landscaping. Some of the area's largest homes are five-acre lots with green lawns, towering trees, old orange groves, and stables for horses. There has been relatively low water rates in the Coachella Valley for years, considering it is a desert region, as well as some of the highest levels of per-capita water use in California. Recently the State Water Resources Control board has included the area's main water suppliers on the list of entities required to implement 35 percent mandatory water cuts (James, 2015).

2.3.3 Aquifer Recharge

Recharge, the hydrological process where water moves downward to groundwater, is an important factor when evaluating groundwater resources but can be difficult to quantify (Kumar, 2003). Al Kuisi (2013) performed a study on groundwater recharge to develop the WetSpa model for arid regions. This model calculated the groundwater recharge as a residual term of the water balance, taking into account precipitation, evapotranspiration, surface runoff and initial losses. According to Kumar (2003) ground water recharge is a complex function of meteorological conditions, soil, vegetation, physiographic characteristics, and the properties of the geological material within the flow path. This makes quantifying the recharge a major problem in water-resource investigations. Various models have been designed to represent the actual physical process. Current methods include soil water balance method: zero flux plane method, one-dimensional soil water flow model, inverse modelling for estimation of recharge, saturated volume fluctuation method (ground water balance), and isotope techniques and solute profile techniques (Kumar, 2003).

The water balance model is a procedure to estimate the balance between the inflow and outflow of water. The soil water deficit in the standard soil water balance calculation is the volume of water required to saturate the soil and is expressed as an equivalent depth of water. The soil water balance equation can be represented as: $G_r = P - E_a + \Delta S - R_o$ where G_r is the recharge, P is the precipitation, E_a is the actual evapotranspiration, ΔS is the change in soil water storage, and R_o is the run-off (Kumar, 2003).

Ground water recharge is a major factor determining the amount of water that can be extracted from an aquifer without causing depletion (Kumar, 2003). Aquifers are recharged through both natural and non-natural methods. The Coachella Valley aquifer is naturally recharged through percolation of surface water, drainage of intermittent streams, and runoff from nearby mountain ranges. It is also artificially recharged from the Colorado River through man-made canals (June, 2008). The factors determining the amount of recharge consist of the rate and duration of rainfall, subsequent conditions at the upper boundary, antecedent soil moisture conditions, soil type, and the water table depth (Kumar, 2003).

2.4 Hydrological Models and GIS

“Geographic data models are important because they establish a common language enabling us to describe aspects of our environment systematically a consistently over large areas” (Strassberg et al., 2011, p. viii).

Considering the spatial characteristics of the parameters which control hydrologic processes, it is no surprise that GIS is now an integral part of hydrologic studies. A hydrological model can be lumped or distributed depending on whether or not the domain/basin is subdivided. Defining the watershed and the drainage network forms the basic framework for applying both lumped and distributed hydrologic models. Soil maps are another important source of hydrologic modeling parameters like soils for infiltration simulations as a function of soil properties. The runoff characteristics of the land surface are affected by land use, vegetation cover, and urbanization. This information can be derived from aerial photography or satellite imagery. In order to be useful, such land use/cover data is reclassified into parameters representative of the hydrologic processes. Some examples of this include: hydraulic roughness, surface roughness heights affecting evapotranspiration, and impervious areas that limit soil infiltration capacity. Rainfall-runoff modeling depends on the estimation of infiltration extended over large areas covering experimental plots, river basins, or regions. It is difficult to consider infiltration modeling independently from rainfall and soil moisture. Some infiltration methods include the non-linear partial differential equation (Richard’s equation), and the more simplified Green and Ampt equation (Vieux, 2004).

Arc Hydro is a set of information models and tools to assist with surface water resource analysis. It models the relationships and connections of surface water resources (features), and prepares the data for use in hydrologic models (Esri, 2013). Arc Hydro Groundwater is an expansion on the framework of Arc Hydro designed for representing groundwater datasets. It provides representation and simulation over time and space, as well as 3D hydrogeologic models (Strassberg et al., 2011).

There are other independent hydrological models such as the Groundwater (GW) Toolbox, created by the U.S. Geological Survey (USGS). This toolbox estimates base flow, surface runoff, and groundwater recharge based on streamflow data. This software allows for the retrieval of certain datasets including: hydrologic time-series data (streamflow, groundwater levels, and precipitation) from the USGS National Water Information System; and preprocessed meteorological data from the National Oceanic and Atmospheric Administration National Climatic Data Center (USGS, 2015).

2.5 Summary

This chapter introduced key concepts that were vital to complete this project. An in-depth background study was done on the Coachella Valley area, and methods on previous studies used to solve similar problems. The study area is in a desert area, with high temperature and low variable rainfall, and water being pumped from the aquifer for a variety of applications. As a result, natural recharge is low and does not match water discharge by human factors, and could lead to the aquifer collapsing. The developed system looks into the infiltration by natural methods, and water levels using well measurements. The NRCS Curve Number method is one of the most commonly used methods for calculating runoff and infiltration, due to its simplicity and widespread

applicability. An adjusted NRCS Curve Number method was used for infiltration calculations.

Chapter 3 – Systems Analysis and Design

Before a project solution could be implemented, careful planning was done to specify the requirements and the solution design. To facilitate this it was essential to understand the client's problem. This chapter begins with a brief description of the problem statement, followed by the requirements analysis, which describes the functional and non-functional requirements. The system design section explains the requirements for the software and hardware design. A project plan was created to help track project progress.

3.1 Problem Statement

The Coachella Valley aquifer is an essential part of the Coachella Valley region due to the water needs in this desert region. The vitality of the aquifer is important not only because it is a competitive asset to the region or the water it provides, but also to avoid the structural damages that may occur from its collapse when it is depleted. The CVEP wanted a method to analyze the water budget of the area and the change land cover/use on the recharge and discharge. The information from these analyses helped to determine which areas and user groups have the greatest disparity between recharge and discharge in the study area.

3.2 Requirements Analysis

Requirements are the crux of every project and determine whether a project will be a success. According to Poddar (2009) if you don't know where you going, you don't know where you might wind up. Meetings were held with David Robinson to discuss the project requirements and gain a thorough understanding of the client's needs. A requirement analysis was conducted to gain an understanding of the functions required for the project's solution, and are divided into functional and non-functional requirements.

3.2.1 Functional Requirements

Functional requirements are the capabilities that the system must provide. 0 provides an outline and a short description of the functional requirements for this project.

Table 3-1 Functional Requirements

Requirement	Description
Open, edit, export, save, or print map documents	ArcGIS desktop used to view, modify, and export maps to JPG, PDF, or PNG format.
Interactively draw area of interest	The system must allow the user to interactively draw area of interest within the study area.
Interactively select new land use/cover	The system must allow the user to select the new land use/cover for the area of interest drawn.
Calculate groundwater level surface	Use water levels from wells to create surfaces for different times.
Produce surface showing infiltration	Calculate the infiltration for study area from precipitation

The system must allow the user to open ArcGIS map documents, to view the input data and results, view pre-setup map layouts, and export or print maps with results. The system must be able to calculate runoff parameters and infiltration, and these results should be available as attribute data. The creation of recharge and groundwater surfaces for different times is a function required for the system. An interactive function is required that allows the user to outline an area of interest and assign it a new land use /cover. This result ought to be used in the tool to re-calculate the recharge and water use of the area. A geodatabase is required to store the input data as well as the initial output data from the tools and user created output data.

The project was created as an ArcGIS desktop application and used ArcGIS desktop extensions. It contained maps and a toolbox with the different models and scripts setup as tools. These maps have layout setups which display the different user groups' recharge and discharge/groundwater level change. The toolbox was installed on the local computer used for the geodatabase.

3.2.2 Non-Functional Requirements

Non-functional requirements are the considerations that focus on how well the system must perform. The types of non-functional requirements include: interfaces, usability and accessibility, integration, operational, performance, maintenance and system administration, and documentation. 0 lists the non-functional requirements and a short description for each.

Table 3-2 Non-Functional Requirements

Requirement	Description
Data format	All input and output data must be in an ArcGIS readable format.
Software	ArcGIS desktop 10.3, Spatial Analyst extension
Geoprocessing Tools	Coded using Python ArcGIS Model Builder Stored in ArcGIS toolbox
User	Familiarity with ArcGIS required to use models and little to none experience needed for output maps.
User guide documentation	Documentation on tools and input data should provide information about steps and parameters

The system was designed on a computer with 3GHz processor speed, and 8 GB of RAM. The system required ESRI ArcGIS desktop 10.3 to be installed with the Spatial Analyst extension activated.

It was required that all the input data and results should be in an ArcGIS readable format, preferably rasters, feature classes, or tables. The tool development environment had to be Python code and/or ArcGIS Model Builder, and these ought to be stored in an ArcGIS toolbox within the geodatabase. In order to use the tools, familiarity with ArcGIS is required, but little experience is needed for the pre-setup map layouts and to export or print them. Completion of the project also requires documentation on the different datasets and a guide on how to use each tool be provided.

3.3 System Design

The project was designed for a desktop platform, and did not require access to the internet. The system design is shown in Figure 3.1. A file geodatabase was created for data and tools storage, as well as data entry. This database contained vector, raster, and table data formats. All datasets obtained from the various sources required processing before being stored in the geodatabase. The different tools load different data from the geodatabase, and then store the results back in the geodatabase. ArcGIS map documents were set up with a connection to the geodatabase, template map layout showing analysis results and reference data, and the tools – scripts and models – setup in the toolbar. The map layouts can then be printed or exported to different formats: JPG, PDF, PNG, or TIFF.

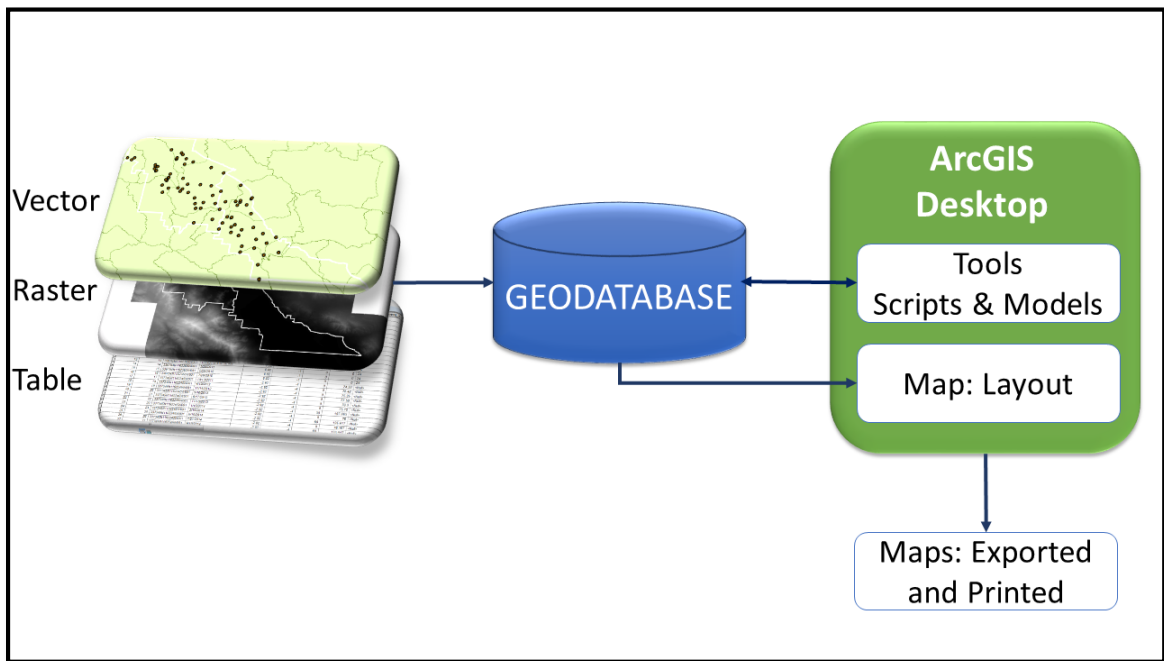


Figure 3.1 System Design

3.4 Project Plan

At the start of the project a project plan was created with associated tasks to provide a reference to initiate the project and keep it on schedule. The following section outlines the project phases and the associated tasks, the final deliverables to complete the project, and assumptions made during formulation of the models and project. During the creation of the project, modifications had to be incorporated, which are also discussed in the sections below.

3.4.1 Phases

This section discusses the three phases of the project: design, development, and deployment. Figure 3.2 shows the structure of the three phases and their associated tasks.

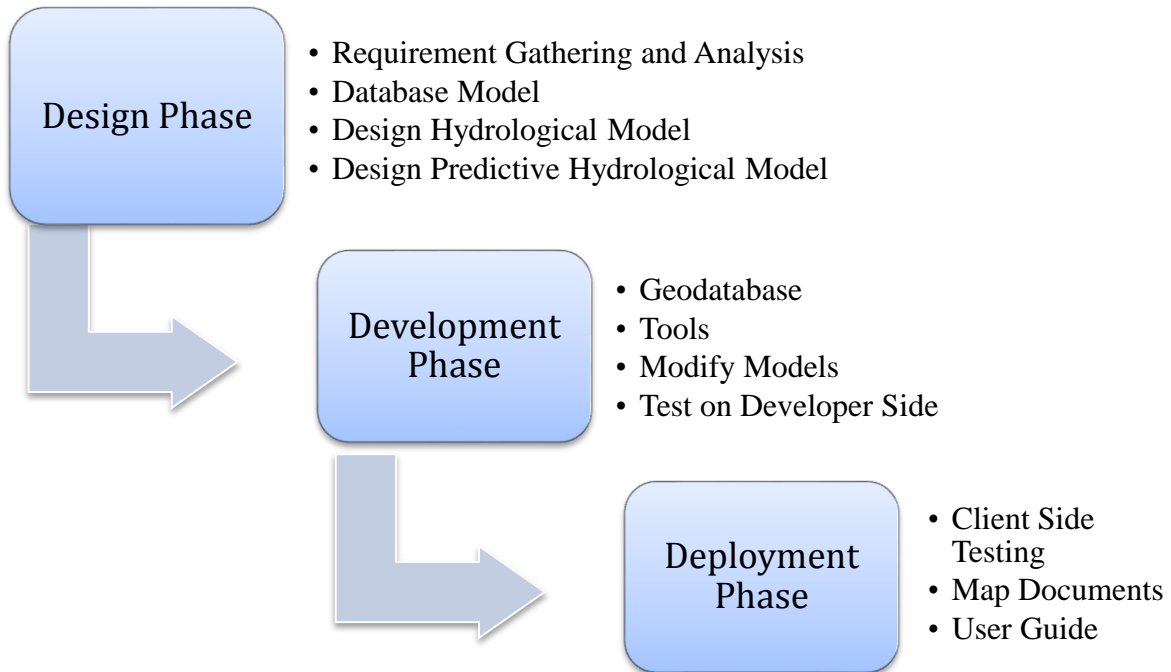


Figure 3.2 Project Phase

The design phase consisted of gathering the requirements and designing the database and the models. The requirements were specified and verified by the client during regular meetings, and used as the basis of the requirement analysis. The design of the database included identifying the datasets needed, their attributes, and how they would be stored. The Arc Hydro template database was used as an example of the hydrological data needed for the models and their various attributes. The database design also included determining which projection to use and what preprocessing needed to be done. Designing the models of this project consisted of determining the tools, functions, and equations the models would use, as well as the workflow for them. This design also expanded the dataset list and the required attributes.

Development was the second phase of the project. This included the creation of the database, construction of the required tools based on the models designed, testing each unit of the tools, and doing modifications based on test results. The creation of the database included the creation of the geodatabase for the input data and results, as well as obtaining data from different sources and preprocessing the data before entering them into the geodatabase. The tools were developed based on the designed models. The tools were created with Python scripts, ArcGIS Model Builder models, and by combining these two methods. The tools were tested in the developer environment, the results were analyzed and discussed with the client, and all modification deemed necessary were made to the models. The process of testing and modification was iterated until the client was satisfied with the results and workflow.

The final stage of the project was deployment, which consisted of final testing, final map documents, and creation of the user guide. The final testing took place on the client's side, ensuring the tools worked on the hardware and software. Map documents

were created that had the tools added to the toolbar, as well as a template layout setup that can be used to export the map to different formats (PDF, JPG, PNG, and TIFF).

Documentation was created that described the tool parameters together with a user guide on how the different tools work. Explanations of each of the datasets were also created that included their purpose, source, format, and the fields required by the models.

3.4.2 Deliverables

There were four deliverables for this project. The first deliverable was the database. In order to deliver this, the geodatabase with the input data and last set of test results were migrated to the client's system. The tools were the second deliverable, and were completed by installing the toolbox containing the tools, models, and Python scripts on the client's system. The pre-setup map documents were the third deliverable provided to the client. The last deliverable was the documentation on the datasets and the step-by-step user guide for the tools explaining the workflow and parameters.

3.5 Summary

Meetings with the client contact, Mr. David Robinson, helped to define the requirements for the project. These requirements were analyzed and a project plan was developed. The project plan consisted of three phases – design, development, and deployment – which were guides to design the different units of the project, how to develop and test them, and finally how and what should be delivered to the client.

Chapter 4 – Database Design

Database design is the process of producing a detailed model of the required database. This was a very important part of the project, since it modeled the geographic entities, their attributes, and relationships. This chapter discusses the conceptual and logical data models that were created, as well as the data sources. This is followed by a detailed explanation of the data scrubbing and preprocessing done before the data were loaded into the database and used for analysis.

4.1 Conceptual Data Model

The entities for this conceptual model included: watershed, geology, soil, precipitation, river, surface water, evaporation, runoff, basin, aquifer, subbasin, well, terrain, land use, and user group (Figure 4.1). The precipitation is the main source of water that falls in a basin forming surface water. This surface water evaporates, turns into runoff, or infiltrates through the soil of the basin into the groundwater. The runoff amount for each basin is influenced not only by precipitation, but also by soil, land use, and slope. The geology that underlies the watershed determines the type of aquifer, as well as the amount and rate water can infiltrate into the ground water of the aquifer. Water pumped from the wells drilled into the aquifer is used by different consumer groups.

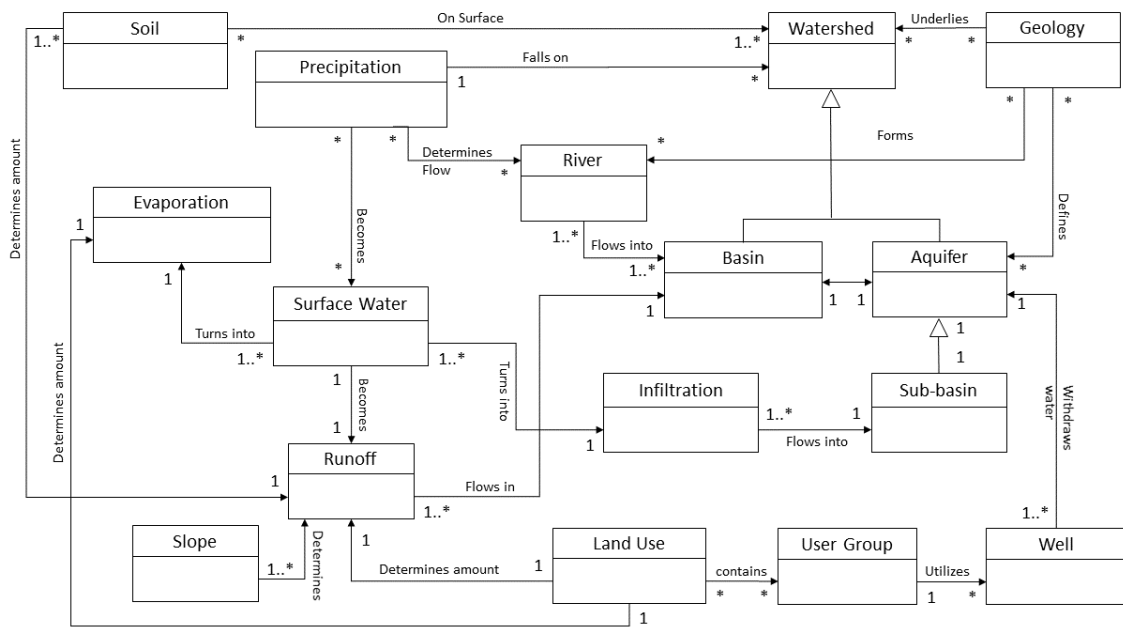


Figure 4.1 Conceptual Model

The geology that underlies the watershed consists of different rock types formed over a long time period. The geology is covered by loose soil that forms the surface of the watershed and each basin. The soil for this model was differentiated based on the four specific hydrologic soil groups: A, B, C, and D. The hydrologic soil group is an indication whether the runoff potential and infiltration are high or low. These are further

explained in section 4.4.2. The watershed contains two subclasses – aquifer and basin – with the major difference between them being that the aquifer is beneath the basin. The aquifer consists of different subbasins. The geographic entities for this project and their relationships are illustrated in Figure 4.1.

4.2 Logical Data Model

A logical data model, derived from the conceptual data model, is how data are organized or in other words how the physical creation of the geodatabase would look. The logical model schema is shown in the diagram in Figure 4.2. The Esri File Geodatabase and File Folder formats were selected for this project. Esri Arc Hydro schema was used as an initial reference for the geodatabase, and after changes were made to suite the project the initial file geodatabase (IFG) was created. This project’s focus included surface water and groundwater, and to incorporate this the initial IFG structure was modified. When Arc Hydro is used it automatically creates a feature dataset within the initial file geodatabase to store vector datasets, and a folder in the same directory to store all the rasters created from its tools. A scratch file geodatabase was created to hold the datasets that were not part of the final results. A third file geodatabase was created that held all the final results.

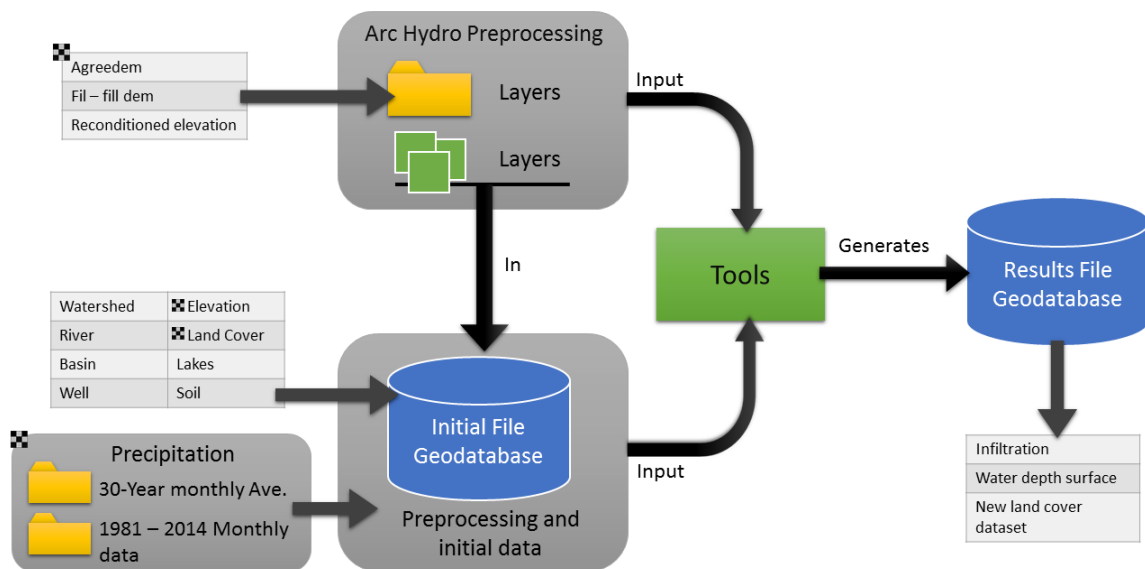


Figure 4.2 Logical Model

The initial geodatabase contained most of the data that were gathered, which were in both vector and raster formats. The precipitation data collected were initially stored in folders before being processed. This geodatabase also contained the feature dataset, named “Layers”, and feature classes that were created by Arc Hydro. The intermediate or scratch geodatabase was a temporary holding location for intermediate data created by the models and tools. The intermediate data were not meant to be long term. After all the models and tools have run successfully the intermediate data were deleted to clean the scratch geodatabase. The results geodatabase held all the final results produced by the

models and tools. These datasets would provide the client with the required information, and used in the final map documents created.

4.3 Data Sources

The client provided the Parcels dataset, while the rest of the required data had to be obtained from external sources. It was important that the data were freely available from a trustworthy source. To ensure correct and complete metadata, any data found and downloaded had to include the metadata or a link to the metadata. The datasets, where they came from, and what format they were obtained in are listed in Table 4-1.

Table 4-1 Data Sources

Dataset	Source	Format
National Elevation Dataset (NED) 10m	U.S. Department of Agriculture (USDA), Natural Resources Conservation Services, National Cartography & Geospatial Center	Raster
National Hydrography Dataset (NHD)	U.S. Geological Survey in cooperation with U.S. Environmental Protection Agency	Shapefile
Land Cover	2011 National Land Cover (NLC)	Raster
Precipitation Frequency	NOAA/NWS/Office of Hydrologic Development, Hydrometeorological Design Studies. NOAA Atlas 14 Volume 6	Raster
Precipitation Average	PRISM Climate Group at Oregon State University	Raster
Soil (SSURGO)	U.S. Department of Agriculture (USDA), Natural Resources Conservation Service	Shapefile, tables, and access database
Geology	U.S. Department of Agriculture (USDA), Service Center Agencies	Shapefile
Wells	Coachella Valley Water District (CVWD) Department of Water Resources (DWR)	Excel Spreadsheet
Parcel	County of Riverside	Feature Class
PLSS	GeoCommunicator – Bureau of Land Management’s publication site	Shapefile and xml

4.4 Data Scrubbing and Loading

Most of the original datasets downloaded had a spatial extent that covered more than just the study area, as was in different projections. Before these datasets could be loaded into the database they had to be projected to WGS 84 UTM Zone 11N and clipped. Some of the datasets required extra preprocessing to ensure all required attributes and values were

present. Other datasets had to be converted from different formats, such as Excel spreadsheets, into Esri feature classes.

4.4.1 Elevation Data Preprocessing

The original elevation dataset had a 10m resolution and were available in 7.5 x 7.5 minute quadrangles that covered the entire United States. Only data for Riverside and San Bernadino County were downloaded. To ensure the data were applicable for the analysis, the elevation dataset had to undergo processing. First, the quadrangles that fall within the watershed for the study area were mosaicked to create a mosaic Digital Elevation Model (DEM) dataset. This mosaic DEM was converted to a raster DEM dataset by creating an empty raster dataset and loading the mosaic DEM into the empty raster dataset. This was then clipped to the Whitewater and Salton Sea watersheds. A portion of the Salton Sea watershed was used since it contains a part of the aquifer.

4.4.2 Soil Data Preprocessing

The soil data downloaded consisted of five detailed shapefiles that cover most of the study area, and a more generalized shapefile that covers the whole State of California. The downloaded data included spatial data, tabular data, and an access database. To utilize this dataset, ArcMap Soil Data Viewer from NRCS was also downloaded and installed. Before the data could be utilized by the viewer and used in this project there were certain steps recommended by the NRCS that had to be completed first. These steps included: importing the empty template access database provided, ensuring the shapefile associated with that database was in the map document, the associated database was selected by the viewer, and that the Soil Qualities and Features-Hydrologic Soil Group was mapped to the shapefile.

The soils were classified into hydrological soil groups (HSG's) to indicate infiltration and runoff potential. The four HSG groups are A, B, C, and D. The SCS soil scientist definition of each group is described in Table 4-2 (USDA NRCS, 1986).

Table 4-2 Hydrological Soil Groups

Hydrological Soil Group (HSG)	Definition	Texture
A	These soils have low runoff potential and high infiltration rates even when they are thoroughly wetted. They consist chiefly of deep, well to excessively drained sand or gravel and also have a high rate of water transmission (greater than 0.30 in/hr).	Sand, loamy sand, or sandy loam
B	These soils have moderate infiltration rates when thoroughly wetted. They consist chiefly of moderately deep to deep, moderately well to well drained soils with moderately fine to moderately coarse textures. These soils have a moderate rate of water transmission (0.15-0.30 in/hr)	Silt loam or loam
C	These soils have low infiltration rates when thoroughly wetted and they consist chiefly of soils with a layer that impedes downward movement of water and soils with moderately fine to fine texture. These soils have a low rate of water transmission (0.05-0.15 in/hr).	Sandy clay loam
D	Soils have high runoff potential. They have very low infiltration rates when thoroughly wetted. They consist chiefly of clay soils with a high swelling potential, soils with a permanent high water table, soils with a claypan or clay layer at or near the surface, and shallow soils over nearly impervious material. These soils have a very low rate of water transmission (0-0.05 in/hr).	Clay loam, silty loam, sandy clay, silty clay, or clay

Note. Adapted from USDA – NRCS Urban Hydrology for Small Watersheds Technical Release 55 (1986)

The detailed shapefiles, besides not covering the whole study area, also contained null values. All the shapefiles had to be clipped to the study area, and the detailed null features had to be intersected with the more general shapefile to obtain the missing values. The final features were then dissolved to get rid of unnecessary small features that were introduced when features were intersected.

4.4.3 Land Cover Data Preprocessing

The original raster land cover data downloaded from The National Land Cover Dataset (NLCD), encompassed California. This dataset was classified into different land cover types and symbolized accordingly. This dataset had to be cut down to the study area, which was done using the ArcGIS Extract by Mask tool. The Whitewater and a section of the Salton Sea watershed were used as the polygon mask.

4.4.4 Wells Data Preprocessing

All the well data obtained were originally in Excel format from two different sources: the Coachella Valley Water District (CVWD), and the Department of Water Resources (DWR). The Excel spreadsheet formats from the two sources were distinctly different and required different preprocessing steps. Preprocessing the data resulted in point features that contained depth measurements for different dates.

A main assumption made regarding the well dataset was that the well data obtained from the CVWD did not contain any information on the use of the wells, and the well data from DWR contained incomplete well use information. Therefore it was assumed that a well's use is the same as the use of land the well falls in.

The data the CVWD provided were in Excel spreadsheet format that contained the state well number, dates, basin name, and water depth measurement from the ground surface. These spreadsheets did not contain coordinate locations for each well, so it was necessary to extract each well coordinate from the state well number which indicates location down to a 40 acre parcel. This extraction was done by first splitting up the state well numbers into the township, range, section, quarter quarter, and well number values in the excel spreadsheet. A Python script was then created to use these values and match them to the PLSS data for Riverside. A new feature class was created from these matches containing all the above-mentioned attribute values. Since wells are point features, the new feature class was converted to points based on the center location of each parcel polygon. If there was more than one well for a 40 acre parcel, the amount of wells were created at random locations within that parcel and assigned a well number ranging from one to the amount of wells in the parcel. The fields of the final feature class containing all these points had to be adjusted to ensure they contained depth below ground surface, the well station number, the measurement, the date of the measurement, well use, and the basin name.

The water level for each well was available as feet below ground level. By estimating the locations of the wells, the accuracy of both the ground level associated to it and the groundwater surface are lowered. Since the well use was determined by the land use related to its location, this was skewed as well. If the correct locations of these wells were obtained in the future, it would increase the accuracy of the ground water surface and assigned land use.

The well data from the DWR were downloaded from the DWR water data library. The downloaded folders contained three CSV delimited files. The two files used were GST, which contains the location information for each well, and GWL, which contains the groundwater level measurements at different dates. The CSV files were put into a file geodatabase before being turned into point features. Figure 4.3 shows the model used to achieve this.

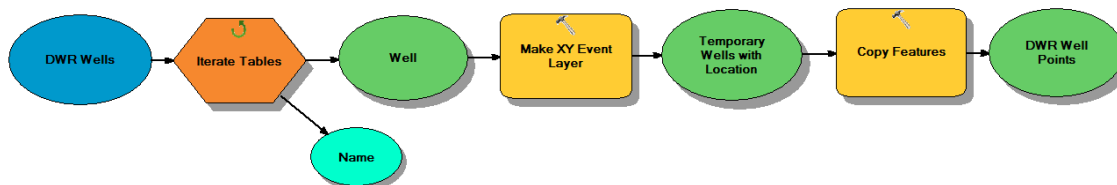


Figure 4.3 Table to Point Feature Class

After the points were created they were joined to the GWL files and point features were created with the different dates measurements. A final feature class was created with a combination of all the well locations and their respective measurements. The fields of the feature class had to be adjusted to ensure the depth below ground surface, well station number, measurement, date of the measurement, well use, and basin name fields existed and their naming were correct.

After the well data from the CVWD and DWR were combined, there were a total of 359 wells in the dataset. The dataset contained well measurement records that date back to 1926, and records as recent as 2015. Even though the well measurement dates are within such an extensive time period, the temporal coverage is not complete for each well. Thus, some well are missing data for specific months in specific years, or only contain measurements in a short time period.

4.4.5 Hydrography Data Preprocessing

The National Hydrography Data (NHD) were downloaded per county. The counties necessary to cover the watersheds for the study area were Riverside, San Bernadino, San Diego, and Imperial County. The Riverside County data contained most of the rivers and lakes required for the preprocessing of other data and analysis, but the other county data were also required to have a complete coverage. The datasets were obtained in the North American Datum of 1983 (NAD83) UTM Zone 11N, and had to be projected to WGS 84 UTM Zone 11N. The data from the different counties were combined into one line feature class and then clipped to the Whitewater and Salton Sea watersheds respectively. The data from the NHD included the watershed and basin data (HU8, HU10, and HU12), and required clipping and projecting of the data. The Whitewater watershed was used as the study area, because it covers the aquifer area. The basins within this watershed were extracted from the original feature class.

4.5 Summary

Chapter 4 discussed the different data sources, what data were needed, what the database structure looked like, and all the preprocessing that went into preparing and creating the initial data. The client provided the Parcels dataset; the other datasets were obtained from various other trustworthy sources. During the process of obtaining the data from the other sources it was verified that the metadata were present. All the data required some degree of cleaning or preprocessing, and had to be projected to WGS 84 UTM Zone 11N and clipped to the study area. The extent of preprocessing necessary for the elevation, soil,

land cover, well, and hydrograph was discussed. The assumption that the well's use relates to the land use of its location had to be made due to missing information. The inclusion of error due to estimating some of the wells was also discussed.

Chapter 5 – Implementation

The estimation of infiltration and water level required several different mathematical formulas and workflows. Current workflows in the GIS software that are simple but tedious to repeat were automated. For the implementation of the project solution, both ArcGIS Model Builder and Python scripts were utilized. Model Builder did not contain all the functionality necessary to create all the tools required. To provide the remaining functionality, Python scripts were written to create custom tools that leveraged pre-existing ArcGIS tools and Excel read/write capabilities. These scripts were created to perform specific tasks that were either used as a stand-alone tool or in a Model Builder model.

This chapter explores the GIS methods that were utilized to create tools that met the project requirements. The methods are discussed under three main headings: Infiltration Using Curve Number, Land Cover Change, and Water Level Estimation. Each section breaks down the workflow into detailed steps to explain what was used, how it was used, and why.

5.1 Infiltration using Curve Number

Infiltration is precipitation that seeps into the soil, and it is a main contributing factor for ground water recharge. The conditions above and below ground influence the amount of infiltration. The National Research Conservation Service (NRCS), previously named the Soil Conservation Service (SCS), developed the Curve Number (CN) parameter by combining the effect of soils, watershed, and land use into this single parameter (Soil Conservation Service, 1989). This project used the NRCS-CN method as the basis for calculating the infiltration. This method is based on a relationship between the amount of precipitation received, initial losses, and maximum retention potential. A workflow for estimating the infiltration was created using Model Builder. This consisted of individual tools, created from models and Python scripts, to calculate the required attributes and values. The following subsections contain detailed discussion about the GIS workflows for generating the attributes and values, as well as the formulas for estimating infiltration with the CN method. Figure 5.1 shows the workflow for the infiltration calculation model.

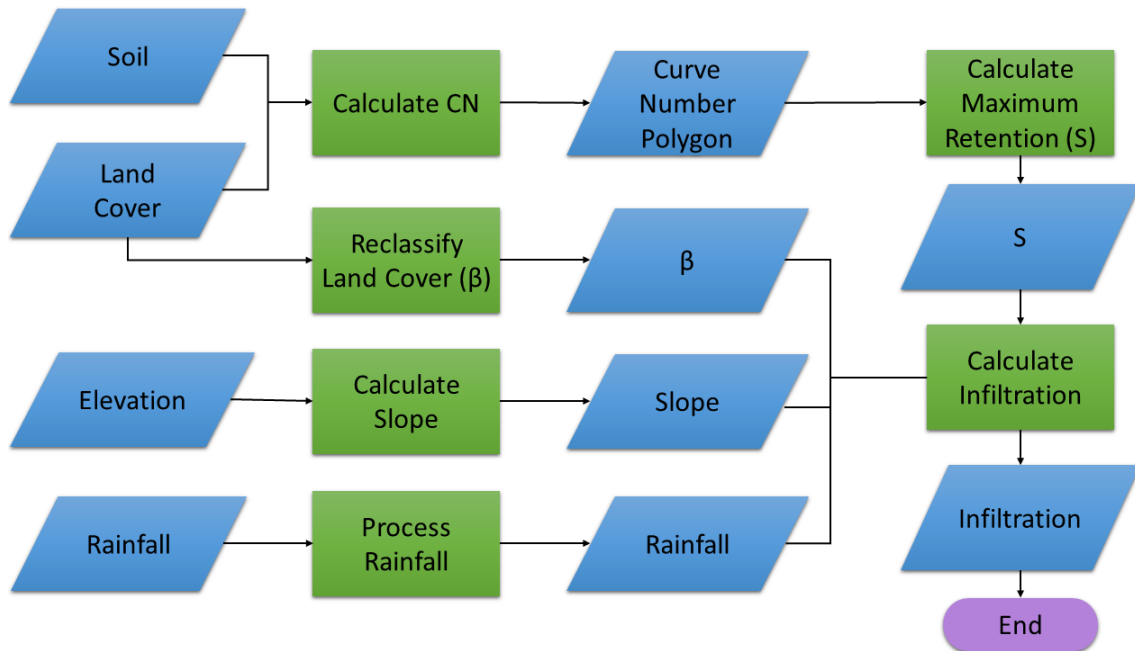


Figure 5.1 Infiltration Estimation Workflow

5.1.1 Curve Number Calculation

The CN is an empirical parameter used in hydrology for predicting runoff or infiltration from excessive rainfall. The CN values in this project ranged from 35 to 100. Lower numbers indicate a low runoff potential with higher infiltration rates; larger numbers indicate a higher runoff potential with lower infiltration rates. Figure 5.2 shows the workflow used to calculate and assign the Curve Numbers.

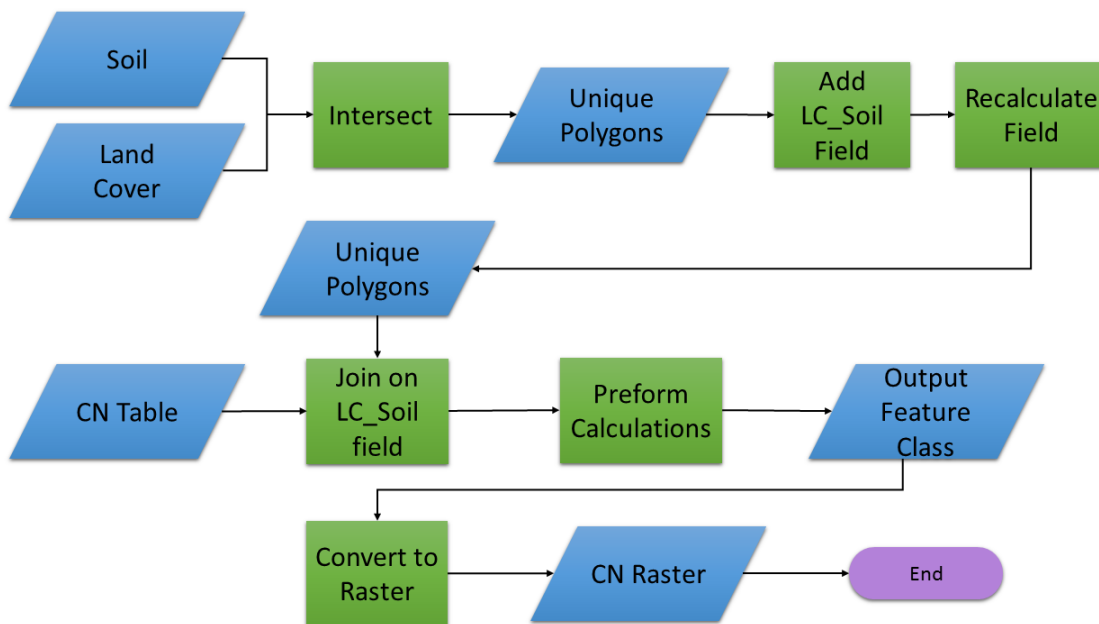


Figure 5.2 Curve Number Calculation Workflow

The calculation of the CN values required three input datasets: land cover, hydrological soil group, and CN table. For the models to run successfully, each of these datasets required the presence and calculation of certain fields. The land cover data had to be in a polygon format and contain the field “LandCover,” which specified the land cover types of the area. The hydrological soil group data had to have the field “HG” which specified the soil hydrological groups in the study area. The first step in this workflow was to use the intersect tool to combine the land cover and hydrological soil group data. Next, the “LC_Soil” field was created to store all the land cover and hydrological soil group combinations. This field was populated using the field calculator tool to join the “LandCover” and “HG” values of each record. These values were required for the join with the CN table. The CN table contained the “LC_Soil” and “CN” fields. These fields were created during the initial setup of the table. The “LC_Soil” field was populated with all possible combinations of the land cover and hydrological soil group. The “CN” field was populated with the Curve Number associated with each combination. The join field tool was used to add the “CN” field from the CN table, providing the Curve Numbers for each land cover/hydrological soil group combination to the intersected data. These polygons were then converted to rasters representing the Curve Numbers.

5.1.2 Rainfall Extraction

Rainfall was one of the main component of the infiltration calculation because it determined the time intervals and periods for which the infiltration could be calculated. The precipitation data from PRISM encompassed the entire United States. The PRISM data used had two different resolutions: the 30-year averages, 1980-2010, for each month had 800 m resolution; and the monthly data for every year from 1981-2014 had 4 km resolution. All the PRISM data had to be resample to the elevation cell size and clipped to the study area.

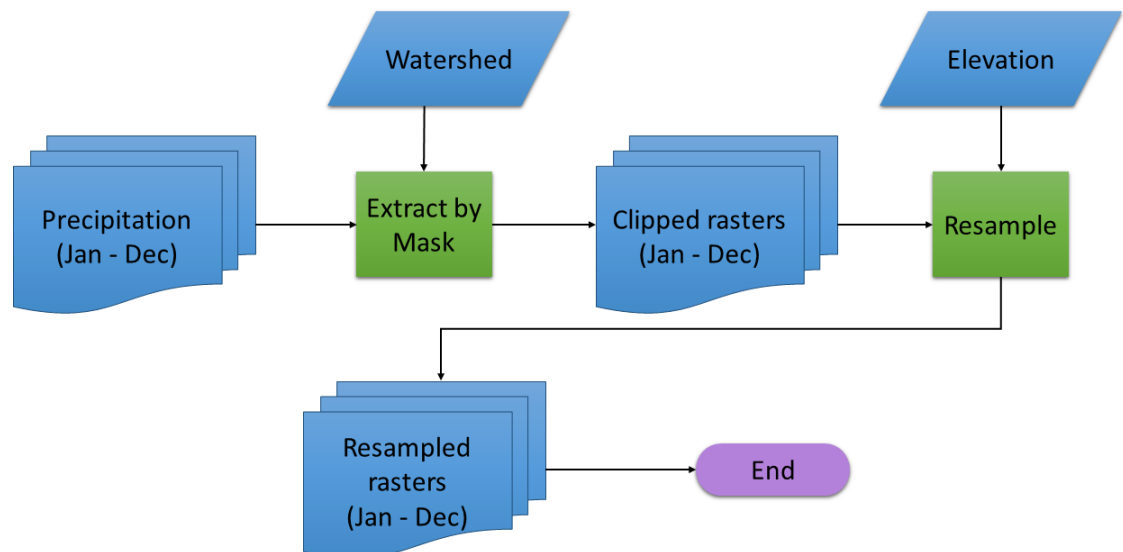


Figure 5.3 Precipitation Extraction

By using the 30-year average precipitation data an infiltration estimation was calculated for each month. Thus, the surfaces generated represents an infiltration possibility based upon the 30 years used. The 4 km precipitation data were used to calculate the infiltration that could occur in a specific month of a specific year.

5.1.3 Slope

The infiltration calculation method used in this project included slope in the formulas. The slope is required to have a unit of decimal degrees. Figure 5.4 shows the workflow followed to calculate the slope.

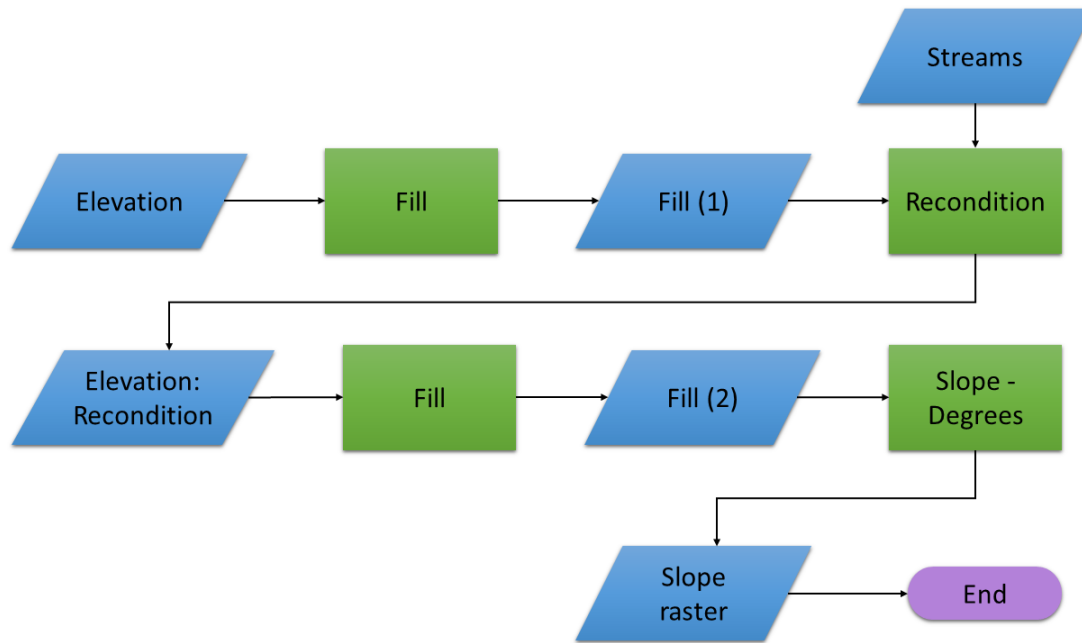


Figure 5.4 Slope Calculation

The first step was to use the Arc Hydro fill tool to remove any possible sinks in the elevation data. This filled elevation was then reconditioned to the streams of the study area. To ensure no sinks were introduced during the reconditioning, the fill tool was implemented a second time, creating the second and final filled elevation data. The final filled elevation data were used in the slope tool to generate the slope in degrees for the study area. The Arc Hydro tools are based on tools from Spatial Analyst extension. The fill tool can also be found under the Spatial Analyst Tools toolbox, but the reconditioning tool belongs to the Arc Hydro toolset.

5.1.4 Calculating β

Craciun et al. (2009) introduced the β coefficient to the calculation of infiltration. This was used to evaluate the vegetation's capacity to lag the drainage to soil for a part of the water quantity initially stored. This coefficient was created because of the idea that not all water quantities retained by a land cover type are lost and contribute to define the initial loss. A part of this initially retained water will be drained and reach the soil to contribute

to the infiltration process. Table 5-1 shows the values used for the β coefficient according to land use (Craciun, Haidu, Magyasi-Saska, & Imbrone, 2009).

Table 5-1 Estimation of β Coefficient According to Land Use

Land Use	β
Forest	0.5
Shrubbery	0.4
Grass/Pasture	0.3
Agricultural	0.2
Settlement	0.1

Note. Obtained from Craciun & Haidu (2011)

To utilize this coefficient in the final infiltration calculation, a raster with the β values was created. As shown in Figure 5.5, the land use raster (NLCD) had to be reclassified to match the values in the β coefficient table.

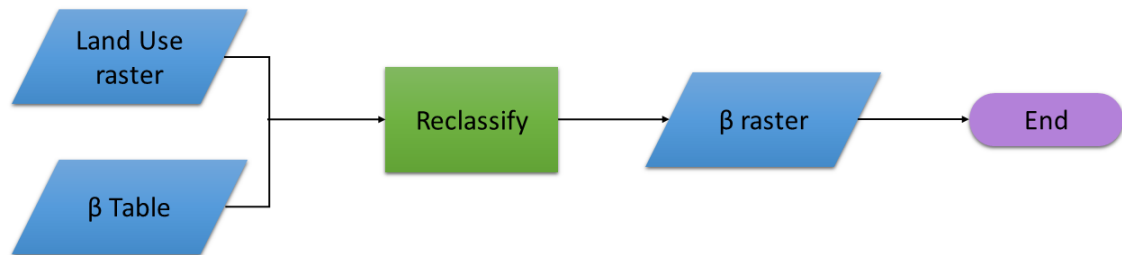


Figure 5.5 β Coefficient

5.1.5 Infiltration Calculation

Once the attributes and values of the data were calculated, and all required data were in the correct format, an algorithm based on the classical NRCS-CN method was used to calculate the infiltration. To estimate infiltration – cumulative infiltration – NRCS proposed a relationship that depends on the quantity of the precipitation, the initial losses, and maximum retention potential. Equation 5-1 shows the original form of the equation (Craciun et al., 2009; Musy & Higy, 1998).

$$F = \frac{S * (P - I_a)}{P - I_a + S} \quad (5-1)$$

where:

F = Cumulative infiltration (mm)

P = Rainfall (mm)

I_a = Initial abstraction (evaporation, retention in the canopy);

S = Maximum capacity of retention;

The NRCS proposed an empirical relationship of I_a to S, shown in Equation 5-2. In the original development the Initial Abstraction Ratio (λ) was set to a value of 0.20 (Craciun et al., 2009; Hawkins et al., 2003; Plummer & Woodward, 1998).

$$I_a = 0.2 * S \quad (5-2)$$

This equation was adjusted by Craciun et al. (2009) to take into account the β coefficient. This adjusted equation is shown in Equation 5-3.

$$I_a = 0.2 * S * \beta \quad (5-3)$$

The potential maximum retention was calculated by using Equation 5-4. This equation is used when the quantity of water is in millimeters (mm), which was appropriate since the precipitation data was in millimeters.

$$S = \frac{25400}{CN} - 254 \quad (5-4)$$

where:

S = Water retention

CN = Curve Number assigned

Some weaknesses in the original NRCS-CN model were identified, the most important problem being the failure to take slope into account. The NRCS-CN formula was adjusted, as shown in Equation 5-5, to take into account this parameter's mathematical relation to the infiltration (Craciun et al., 2009; Gajbhiye & Mishra, 2012).

$$F = \frac{S * (P - I_a)}{P - I_a + S} * (1 - I_b/100) \quad (5-5)$$

where:

I_b = Slope in degrees

The workflow developed for indirect determination of the infiltration of rain was based on integrating all the parameters defined by the NRCS method. The precipitation data used in this project were monthly data per year and monthly averages from 30 years, thus the infiltration calculated was either the monthly infiltration for a specific year or the estimated infiltration for a specific month based upon the 30-year average precipitation.

5.2 Land Cover Change (LCC)

Changes in land cover can affect the hydrological cycle, such as the infiltration within a specific area. To project influence future development would have, it was necessary to create a method to adjust the current land cover dataset with the new land cover classification for an area of interest (AOI). Molis (2012) created a geoprocessing tool to illustrate how land cover change in a user specified region would alter the recharge rates, using the island of Hawaii as a study area.

To create a tool for land cover change, a Python script was written to leverage ArcGIS tools through ArcPy. A Python script provides more control over how the tools are run, how data and records are handled, and how the outputs are generated. To perform the necessary functions, the script utilized the ArcGIS Spatial Analyst functionality through ArcPy, and the array functionality provided the NumPy Python module. NumPy provides an N-dimensional array type to work with raster data. Figure 5.6 shows the overall workflow followed to change the land cover classification.

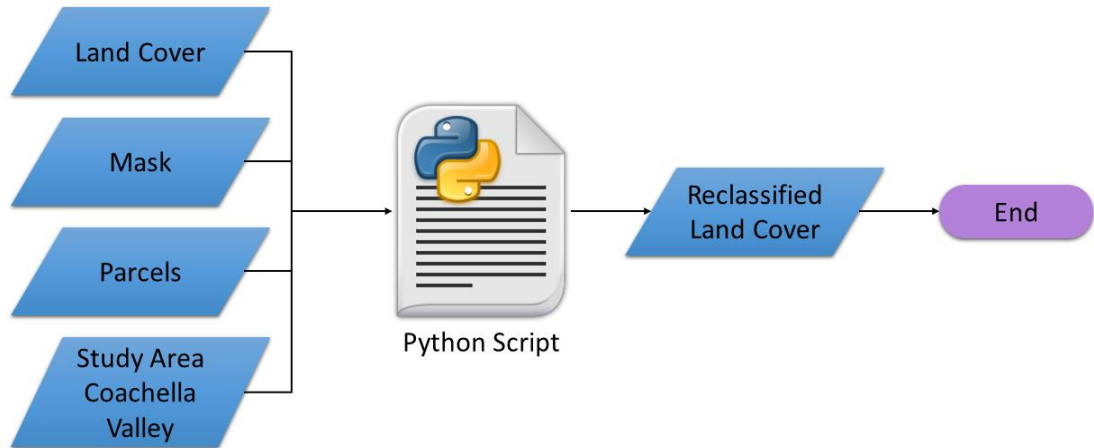


Figure 5.6 Land Cover Change Workflow

The developed Python script can be divided into seven steps, and contains two functions. The seven steps, shown in Figure 5.7 include: loading the parameter values specified by user, extracting the spatial location information of the land cover raster, loading the data into an array, extracting the area of interest (AOI) pixels, assigning new land cover classification values, saving the new land cover raster dataset, and updating the raster’s attributes. The two functions were used to find the pixel value from a land cover description, and to find the land cover description from a pixel value.

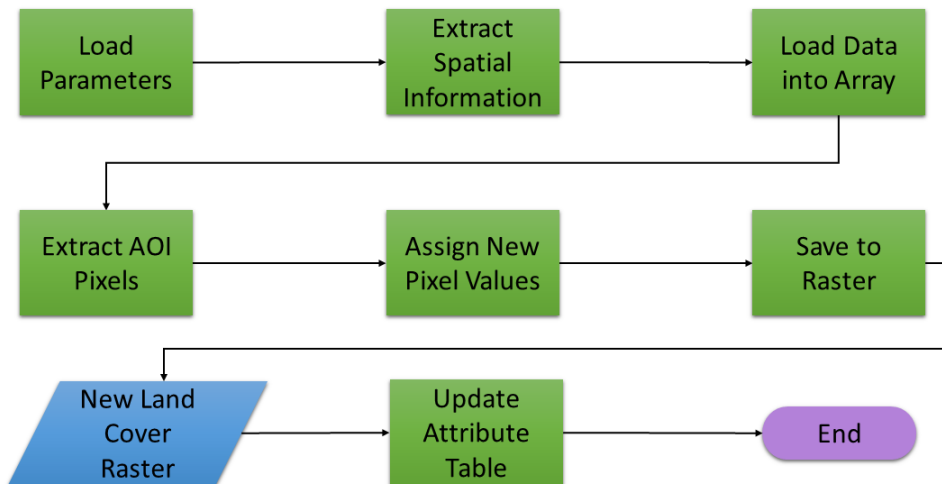


Figure 5.7 Python Script Workflow

The script starts by loading the five user specified parameters into variables. Three of the five parameters were input datasets: the original land cover, the selected Coachella Valley parcels, and the Coachella Valley study area. The remaining parameters were the new land cover raster location, and the new land cover classification selected from a dropdown list. Next, the script extracted the spatial information necessary to create the reclassified land cover dataset. This information was extracted from the metadata of the original land cover dataset specified by the user. The spatial location information was

used when the NumPy arrays were set up, the AOI was extracted, and when the new land cover dataset was saved.

Initially the script created two NumPy arrays, one for the original land cover dataset and an empty one to store the output results. Next, the AOI pixels were extracted from the original land cover dataset using the Extract by Mask function from ArcGIS. This mask consisted of the parcel polygons the user selected from the Coachella Valley Parcels. These AOI pixels were then converted to a NumPy array using the ArcGIS Raster to NumPy function, and had the same number of rows and columns as the previously created NumPy arrays.

In order to assign new values to the extracted pixels, a function was used to find the pixel values from the land cover description. This function used the ArcPy Data Access module to iterate through the data and a dictionary to find the corresponding values. Once the new values were assigned the resulting NumPy array was converted back to a raster by using the NumPy to Raster function from ArcGIS. This raster was then saved to the location specified by the user. To provide this new raster with the land cover descriptions associated with the pixel values, a second function was used. This function finds the land cover description from the pixel values. As with the first, this function used ArcPy Data Access module to iterate through the data and a data dictionary to find the corresponding values.

5.3 Water Level Estimation

The groundwater level was estimated from the well data. The well data originally contained the depth to water in feet, but this was adjusted to millimeters (mm) to correspond with the infiltration that was in mm. The 2-D groundwater levels, water depth surfaces, were created using the Spline interpolation method. Since the wells did not cover the whole Coachella Valley aquifer, some extrapolations were necessary. The depth values of the wells were positive, with higher values indicating a water level further below the ground surface. These values were adjusted to be negative values below ground surface in order for the 3-D surfaces to be generated with the correct orientation and not up-side-down. Figure 5.8 shows the workflow followed to generate the water level surfaces. The following subsections provide a detailed explanation of each step.

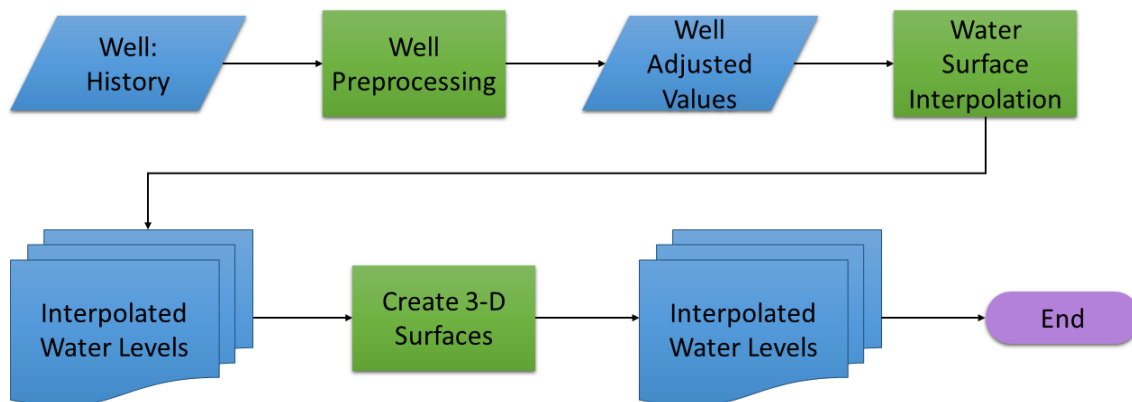


Figure 5.8 Water Level Estimation Workflow

5.3.1 Well Data Preprocessing

The water depth values of the well data required some adjustment before they could be used in the groundwater surface creation. Figure 5.9 shows the workflow used to adjust and calculate the values. The depth to water values were in feet, but meters were more appropriate since the infiltration was calculated in mm. For conversion from feet to meters the constant value of 0.3048 was used. The measurement values in the data were positive, with the larger values indicating greater depth below the ground elevation and the smaller values indicating closer levels closer to the ground. To convert the measurement values from feet to meters, a new field was added and the field calculator was used to multiply the values in feet with -0.3048. The negative of the conversion constant was used to clarify that the values are beneath the ground surface, and when they are used in a 3-D environment they will extrude downward as they should.

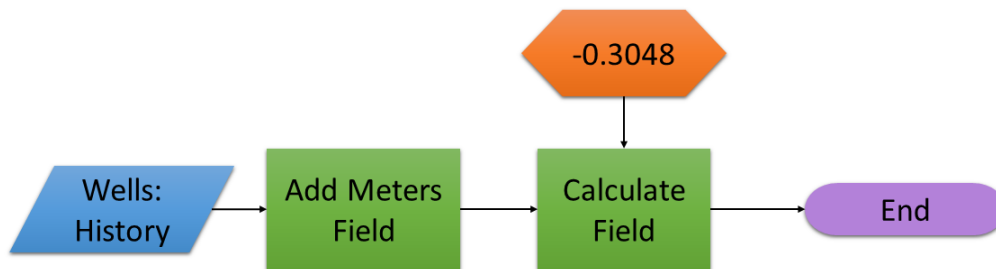


Figure 5.9 Depth to Water Adjustment Workflow

5.3.2 Water Surface Interpolation

A main component of this project was to visualize the groundwater level of the aquifer by creating surface rasters. Since the well data were not uniformly distributed and not complete for every month and every year, interpolation and extrapolation were done on averages to create the water level surfaces. These surfaces were constricted to a 1.5 Kilometer (km) buffer around three out of the four aquifer subbasins: Mission Creek, Desert Hot Springs, and Indio. A buffer was used to ensure all possible wells were included in calculations. The fourth subbasin was not used for this analysis because the number of wells located in it and the amount of measurements taken from there were too low, which resulted in too much extrapolation. Raster were extracted in the model to the boundary of the three subbasins to produce the final surfaces. The Spline with Barriers interpolation method was used to create rasters that represent the groundwater depth.

Two tools were developed to generate these surface rasters. The first tool generated monthly surface rasters for specified years, and the second tool generated surfaces for the monthly averages of a specified time period. The surfaces were monthly based to correspond to the rainfall data that were in monthly intervals. The first tool, generating monthly surfaces for a specified year, was developed as a Python script. The second tool, generating the monthly averages, was developed using ArcGIS Model Builder. Both of these tools are based on a similar workflow, which is shown by Figure 5.10.

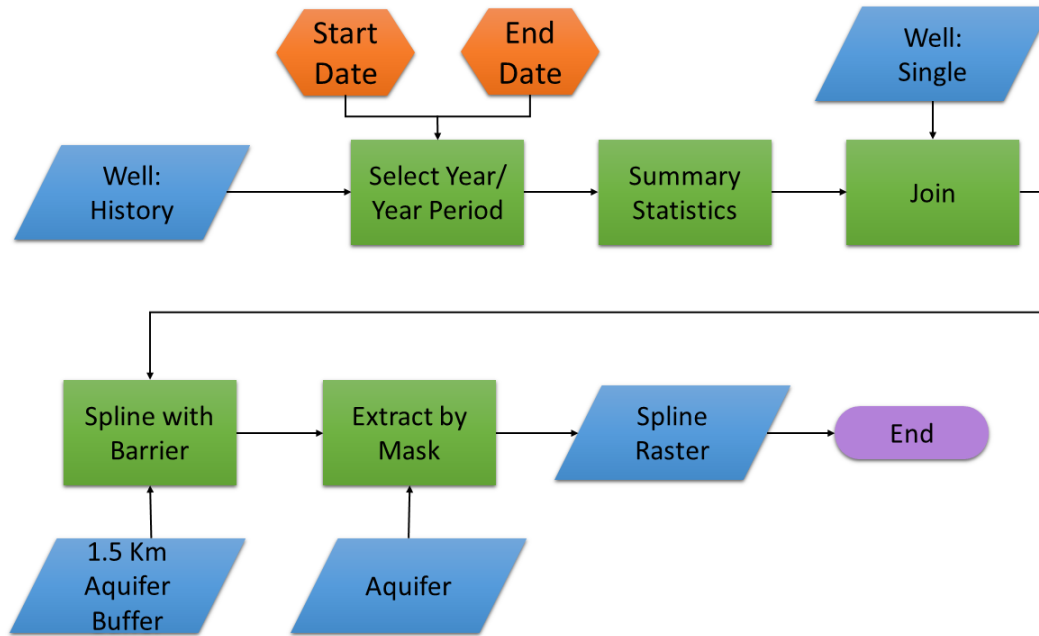


Figure 5.10 Water Depth Surface Workflow

The workflow starts with selecting the records from the well data that was measured in the specified time period, which was provided by the user. The selected records were then divided up into the different months, and all further calculations and analysis were run per month. After this division, the summary statistics tool was used to calculate the mean of the depth to water measurement for each well. This table was then joined to the well data that contained only one entry for each well. This was done to assign the mean water depth value for this instance to the well location. Next, the Spline interpolation tool was used with these values and the 1.5 km buffer around three of the aquifer subbasins. This generated raster surfaces representing the water depth, and the values were constricted to the buffer area.

The Spline interpolation method creates a smooth surface that will go through all the measurement points used, and will follow the trend set by the points in areas where there are no measurements. Even though the Spline surface goes through all the measurement points it is not restricted to the minimum and maximum values of the data. When the Spline interpolation method is used with the inconsistent data and data not covering the whole study area some problems arise. For example, when the points closest to the border have an increasing trend, then the spline surface will continue this trend and could result in very high values at the border.

These two developed tools differ from each other in that they used the start and end date provided by the user differently. The first tool generates monthly surfaces for every year from the start to end date. Thus, it runs the whole workflow for every year in the time period specified. The second tool generates an average for each month from the start to the end date. Thus, it runs the workflow once and only generates 12 rasters surfaces.

5.4 Summary

This chapter discussed how a set of tools that met the client's requirements was created by combining ArcGIS Model Builder models and Python scripts. These tools were used to calculate all the attributes and values needed to generate the final results for the infiltration, land cover change, and water level surfaces. For the infiltration calculation, the tools generated raster sets for CN, precipitation, slope, β , maximum retention, initial losses, and finally, the infiltration for each month. The tools associated with the water level estimation ensured that the depth to water values were appropriate, and used these values to calculate the water surface rasters and extract them to the specified area.

Chapter 6 – Results and Analysis

This chapter discusses the different tools developed for the project, and results generated from running them. The infiltration tools and the associated results are discussed in the first section. This is followed by an explanation of the land cover change tools, and an illustration of how this tool works. The next section explores the water depth surface tool the results generated, and a method to visualize the surfaces and the differences between them. The last section explains how the water depth surfaces and the wells were visualized in a 3-D.

6.1 Infiltration

The methodology outlined in Chapter 5 was followed to develop a set of tools that calculate the infiltration. This section explains the tools by looking at how they are used and what they require from the user. Results generated by these tools are then shown and explained.

6.1.1 Infiltration Tools

The development of the infiltration tool was divided into three steps: generate the CN raster, calculate all remaining parameters, and calculate the final infiltration by using the results from step one and two together with precipitation data.

The user interface for the tool used to generate the Curve Number raster is shown in Figure 6.1. The user specifies three input datasets and one output location. The inputs are the land cover dataset, the soil dataset, and the CN table used to assign CN values. The land cover dataset must be a raster dataset and contains the Land_Cover field. The soil datasets must be a polygon dataset and contain the HG field, and the CN table needs to contain the LC_Soil and CN fields. The output location is necessary for the resulting CN raster.

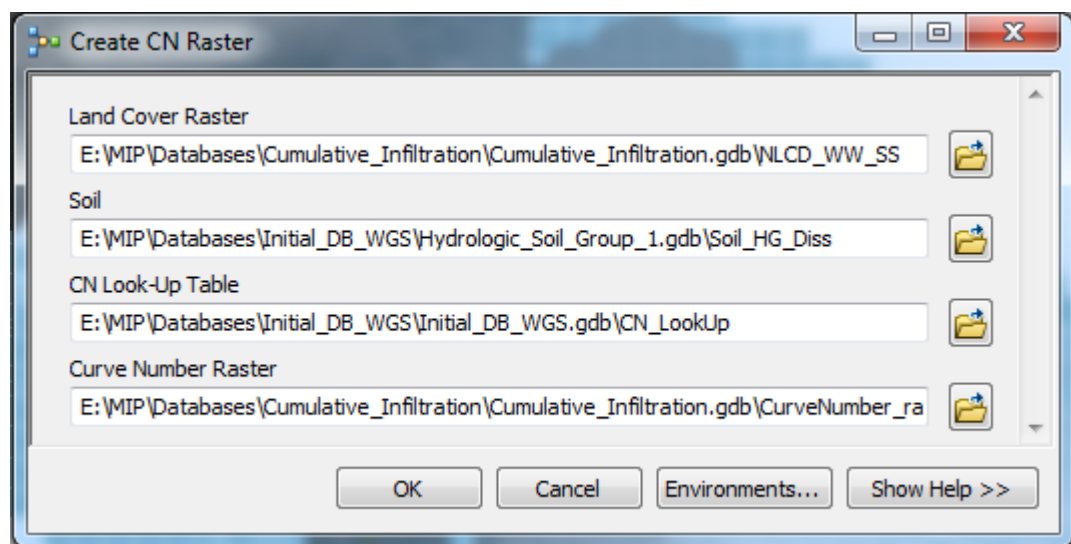


Figure 6.1. Create Curve Number Raster User Interface

Figure 6.2 shows the result of the first step. This output will have the same resolution as the land cover input data. The CN values for this raster range from 35 to 100, where low values indicate areas with higher infiltration potential and high values indicate areas with low infiltration potential. A value of 100 indicates the lowest infiltration potential. The graph in Figure 6.2 illustrates the distribution of the CN values. This graph shows that most of the CN values fall in the mid-70's, while the second and third most values fall between 35 and 60. This indicates that the region has more areas with a higher infiltration potential.

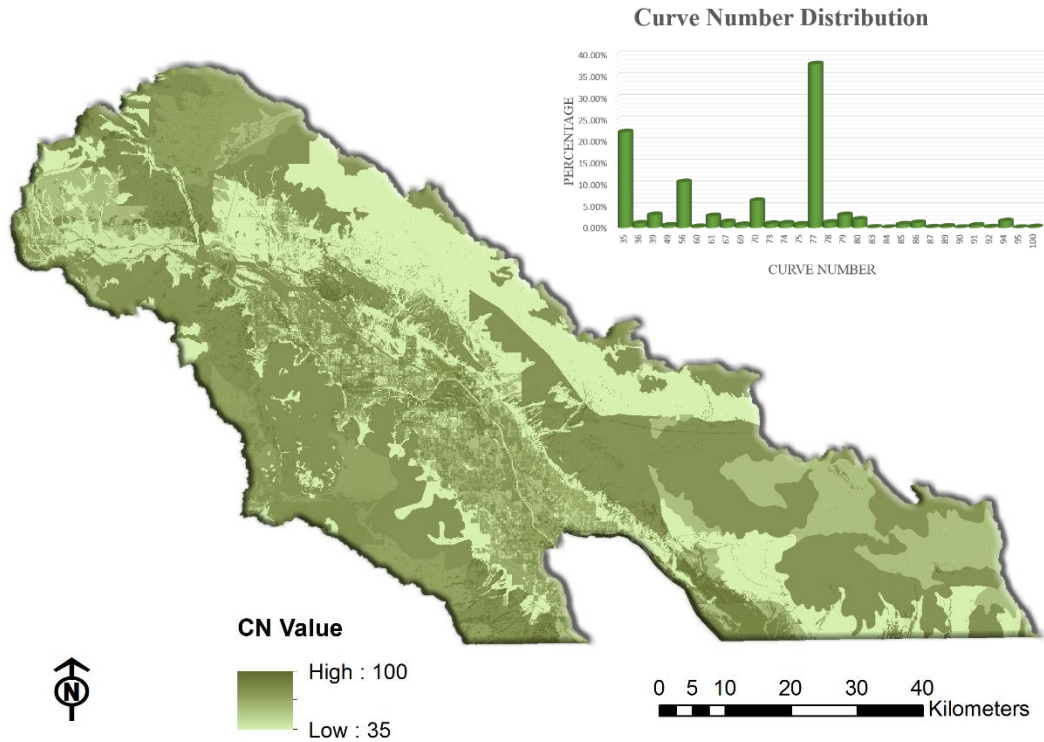


Figure 6.2 Curve Number Raster

The second step calculated the parameters of the infiltration equation: adjusted slope (α), reclassified land cover (β), maximum retention (S), and the initial abstraction (I_a). After the initial run, the first and second step tools just needed to run when a different location was taken into account or when there was a change in the land cover dataset. Figure 6.3 shows the user interface for the second step's tool.

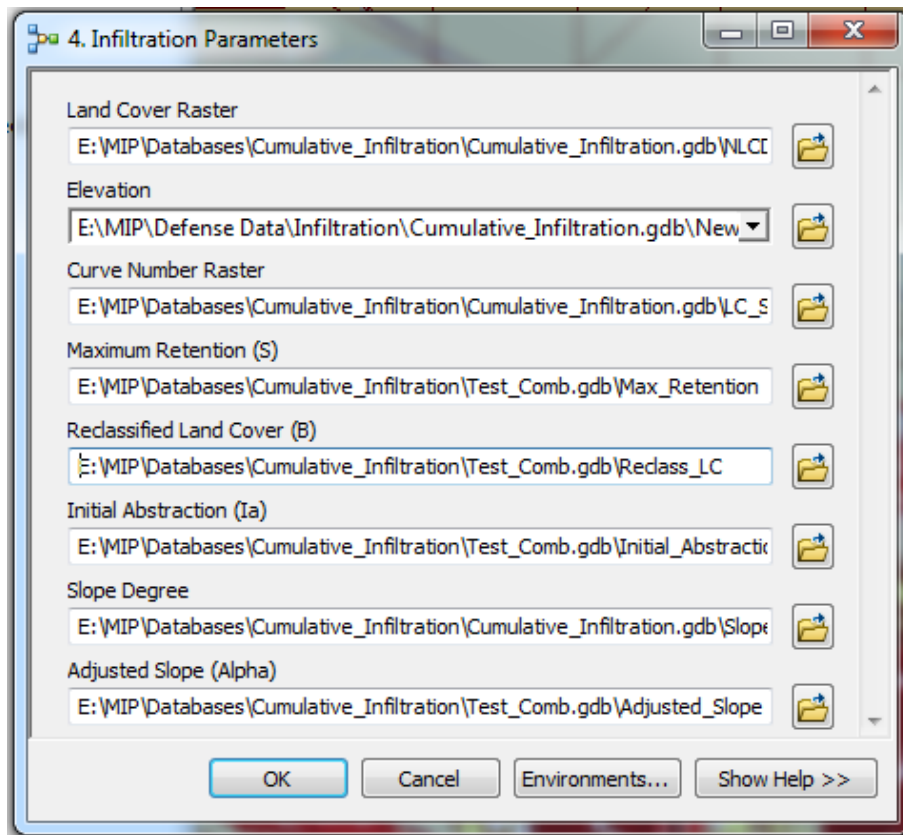


Figure 6.3 Calculating Infiltration Parameters User Interface

The required user input datasets are land cover, elevation, and the CN raster from step one. The rest are the output locations for the four parameters. The resulting rasters for the parameters are shown in Figure 6.4.

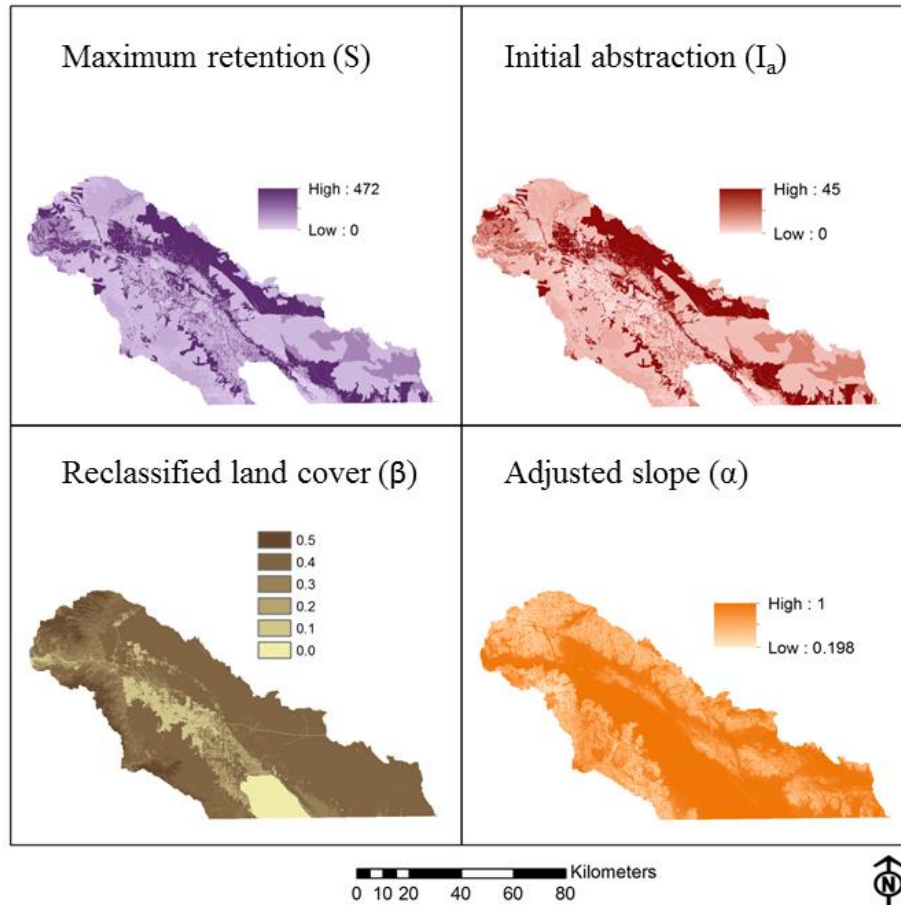


Figure 6.4 Results from Step Two of the Infiltration Calculation

There are two variations of the final step: running for a single instance, and running for every raster in a workspace that meets the name requirement. Both variations follow the same basic steps, but they are used differently.

Figure 6.5 shows the user interface to run the tool for a single instance, variation 1. The user must specify the parameters calculated in step two (Figure 6.4), except for β , and the precipitation data to be used. A location for the Scratch database (location for intermediate results) and the final infiltration calculated also need to be provided.

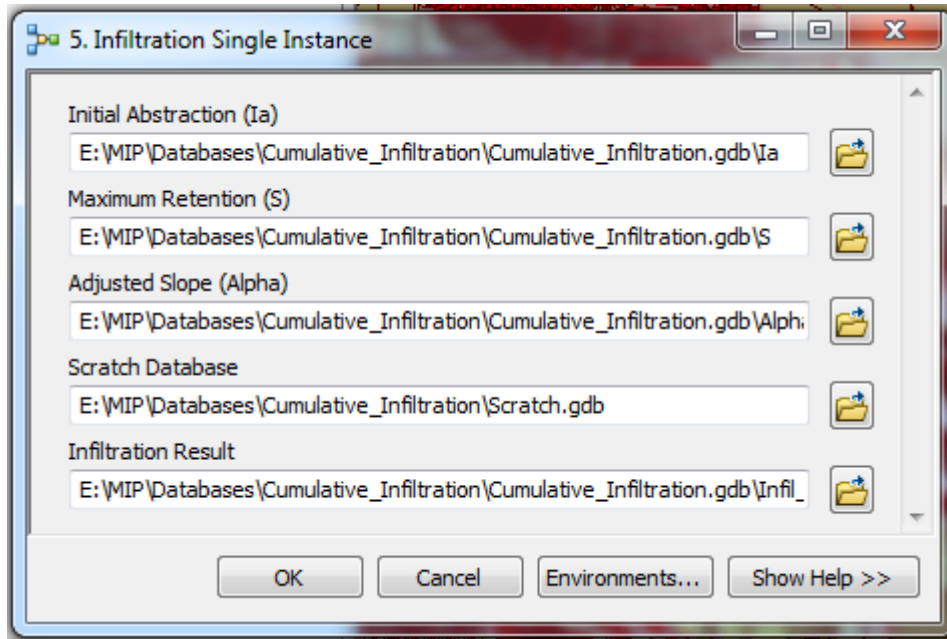


Figure 6.5 Calculate Final Infiltration User Interface, Variation 1 (Single Instance)

Figure 6.6 shows the user interface to run the tool on a database, variation 2. Here the previously calculated parameters have to be specified (Figure 6.4) except β , as well as the input workspace containing the precipitation datasets, the output database for the infiltration results, and the Scratch database. The input workspace should contain rasters with “PRISM” at the start of their name. The resulting infiltration datasets will be named based on the associated precipitation data.

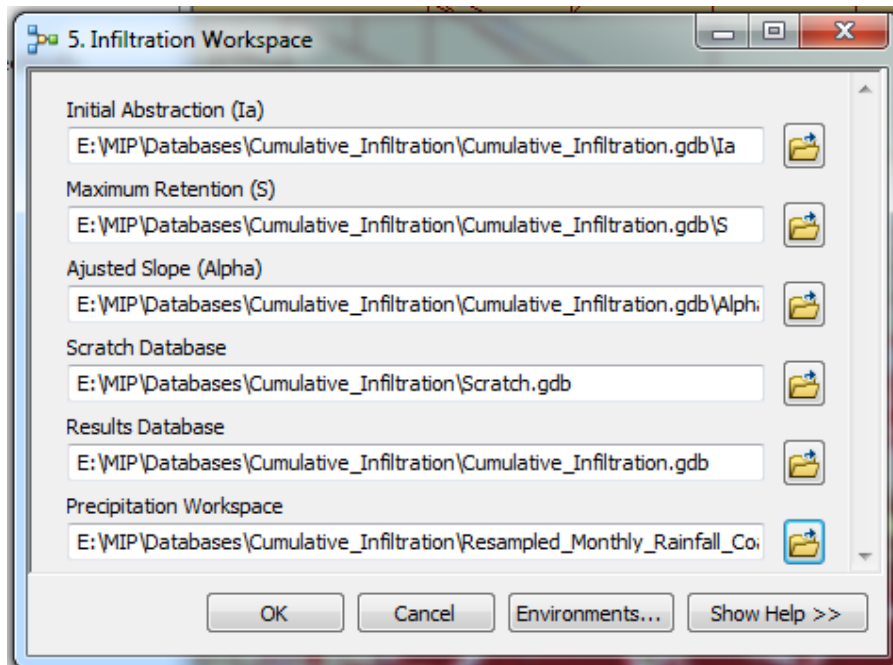


Figure 6.6 Calculate Final Infiltration User Interface, Variation 2 (Workspace)

6.1.2 Infiltration Results

The infiltration was first estimated for each month using the 30 year averages (1980 – 2010) from the 800m PRISM data. Figure 6.7 shows the results from January to June, and Figure 6.8 shows the results from July to December. These results show a trend of higher infiltration during the winter months compared to the summer months. The western mountainside and the north of the Coachella Valley show a higher infiltration trend in both summer and winter months.

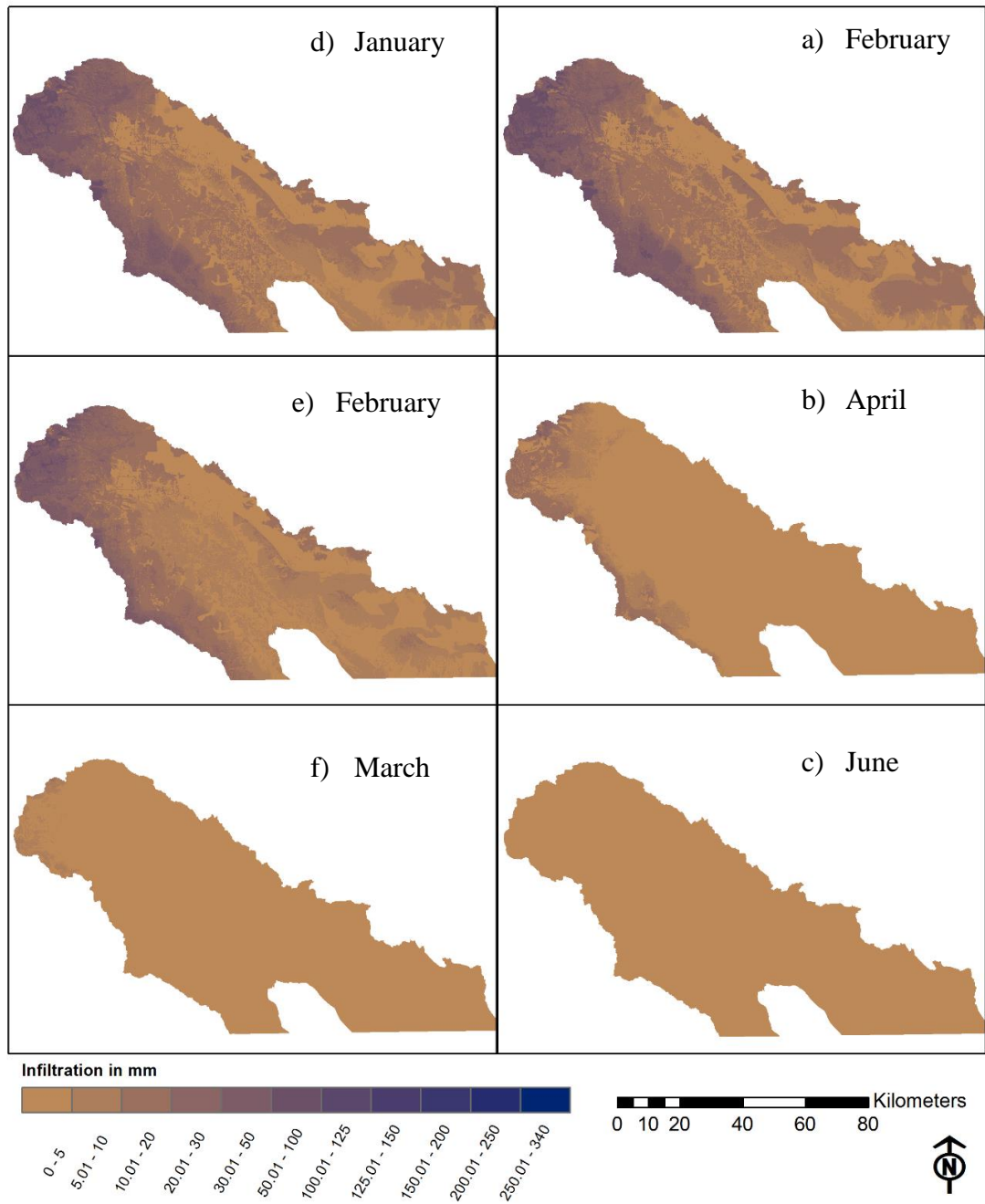


Figure 6.7 Infiltration from PRISM 30 Year Precipitation Averages, January to June: a) January, b) February, c) March, d) April, e) May, f) June.

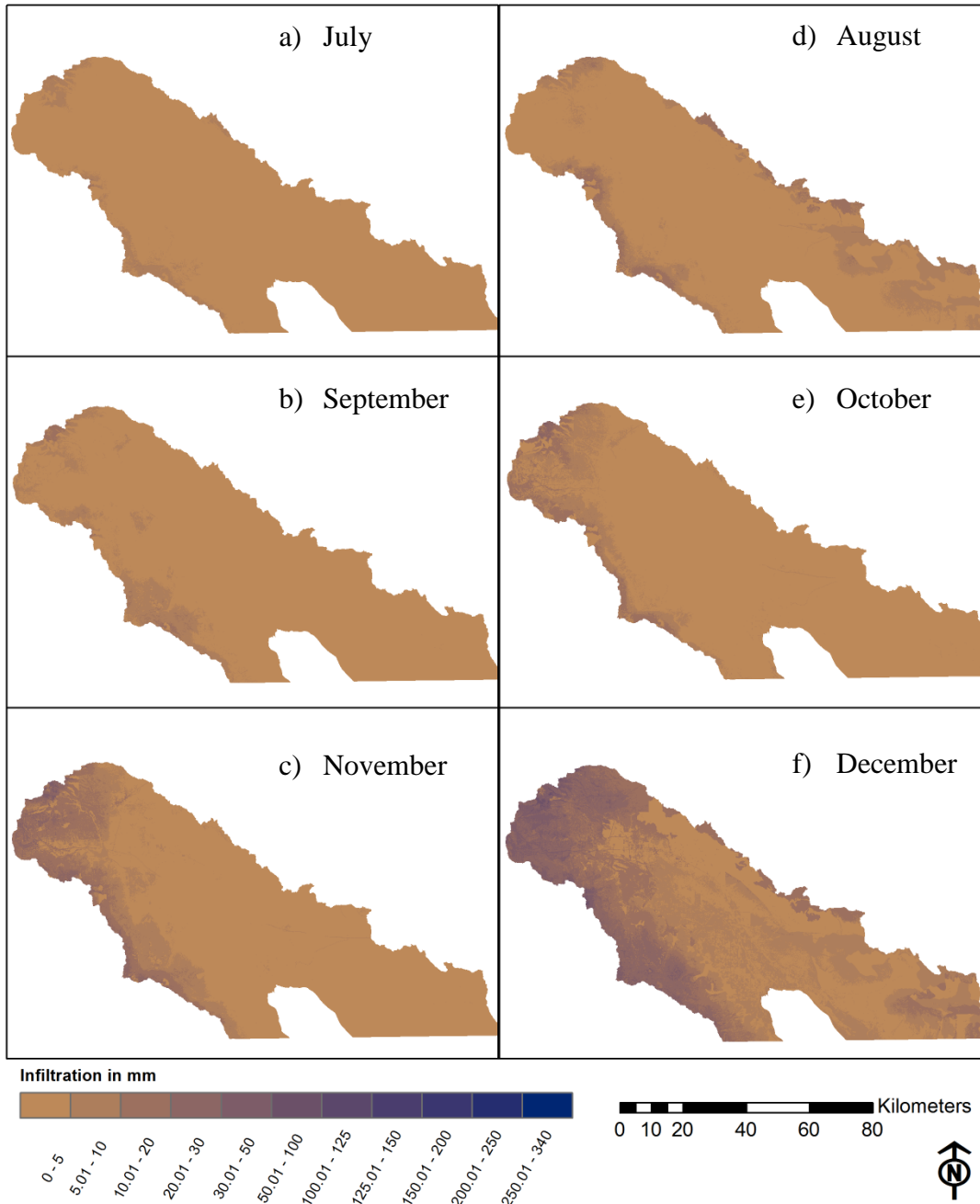


Figure 6.8 Infiltration from PRISM 30 Year Precipitation Averages, July to December: a) July, b) August, c) September, d) October, e) November, f) December.

For shorter time periods, the user can use the 4 km PRISM data to calculate infiltration. The 4 km data can be used to calculate infiltration for each month between 1981 and 2014. For an example the infiltration was calculated each month in 2013 and 2014. Figure 6.9 shows a comparison between the infiltration and the precipitation in December of these two years.

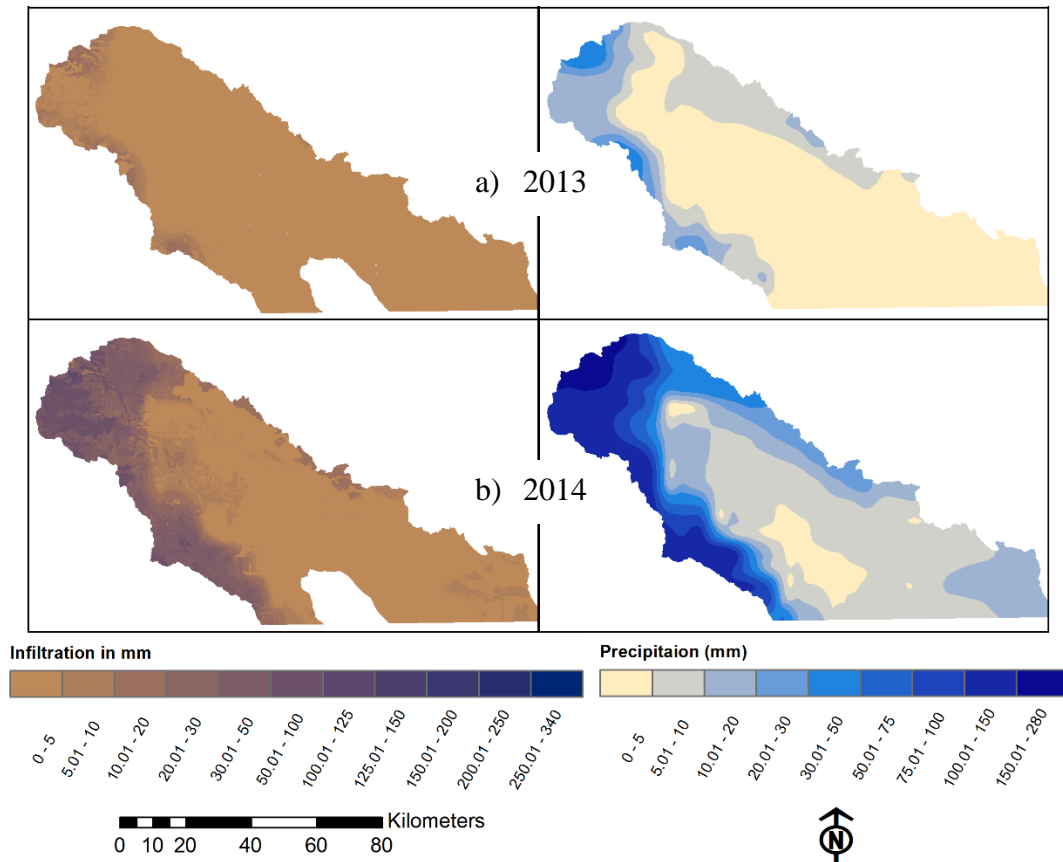


Figure 6.9 Comparison between Infiltration (Left) and Precipitation (Right) for December, 2013 and 2014.

The comparison in Figure 6.9 illustrates that the infiltration for a specific time period is related to the precipitation. Thus, the less precipitation that occurs the less infiltration can occur.

6.2 Land Cover Change Tool

A potential use for these sets of tools is to show the effects that a change in the land cover have on the infiltration. The land cover change (LCC) tool developed for the project provided the ability to change the land cover classification of a selected area to a land cover classification specified by the user. The land cover classifications available in the tool are limited to those of the current study area, but can be expanded for use in other regions. Figure 6.10 shows the user interface of the LCC tool and an area of interest selected. The user provides the original land cover data that will be changed. Next the user specifies the CV_Parcels layer from the dropdown list and select parcels from this dataset. This indicates the area of interest where the land cover values will be reassigned. The user will be prompted to provide a name and location for the new land cover dataset. A dropdown list of possible land cover classification is provided. The user selects one of these land cover description to assign to the area of interest and provides the location of the study area polygon. In this example the Whitewater River catchment and a part of the Salton Sea Catchment were used.

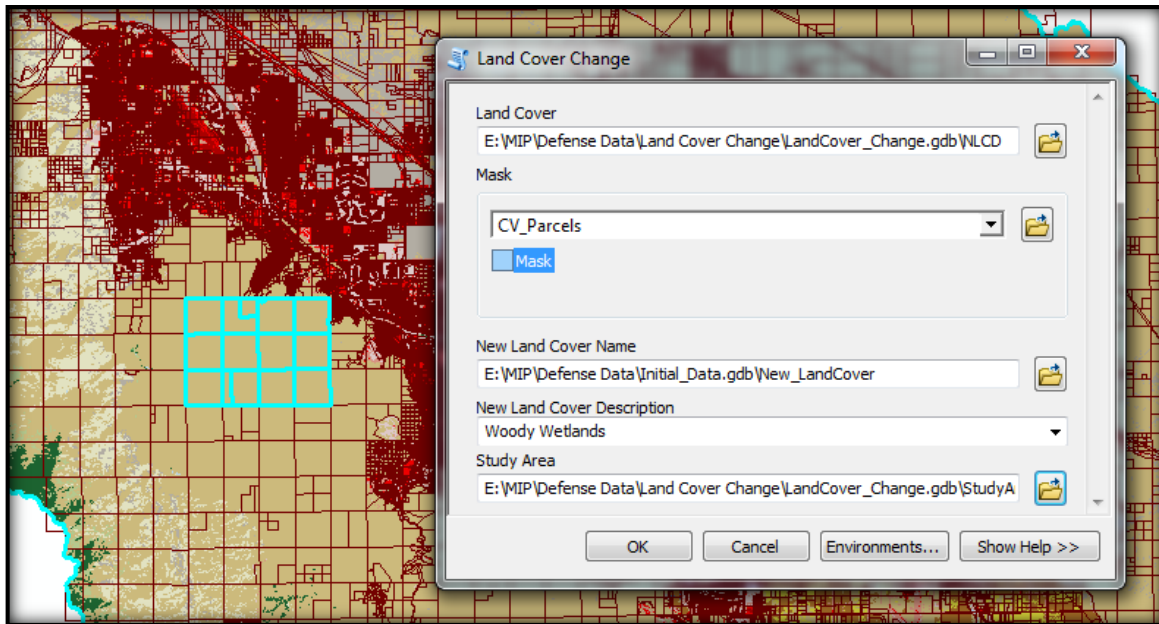


Figure 6.10 Land Cover Change Tool User Interface with Selected Parcels in Blue

The following are some examples of how these tools would work. The setup of the tool and the results when an area is selected are shown in Figure 6.10. Figure 6.11 illustrates how a few Cultivated Crops and Hay/Pastures areas are changed to a Developed, High Intensity area. The light blue lines indicate the parcels that were selected as the area of interest to change the land cover classification. When comparing the left and right images in Figure 6.11, it is clear that the land cover classification has been changed. The light brown/yellow indicating Cultivated Crops/Hay Pasture was changed to dark red, indicating Developed, High Intensity land cover.

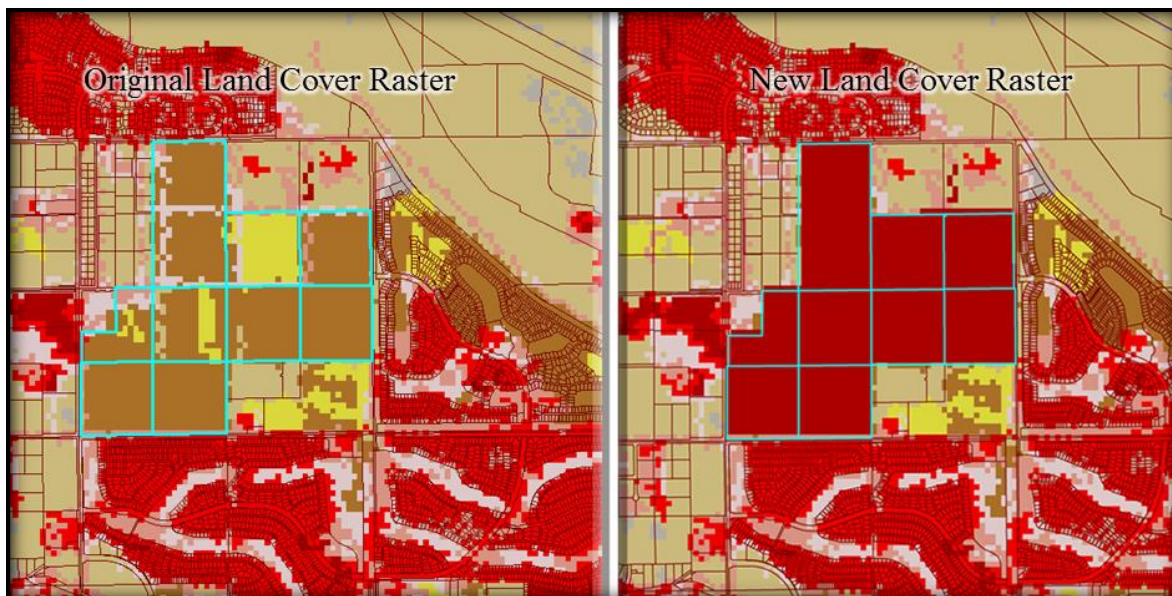


Figure 6.11 An Example of Changing Cultivated Crops and Hay/Pastures Areas Changed to Developed, High Intensity

After the LCC tool is used to change the land cover classification of an area of interest, the infiltration tools are run again to see what the new infiltration will be. All steps of the infiltration calculation must be run.

6.3 Water Depth Surfaces

The methodology described in Chapter 5 was used to develop tools to generate water depth surfaces (WDS). These were developed by using ArcGIS Model Builder and Python scripts. They generate results on a monthly basis for different years, and monthly averages for a time period. These results are compared to gain a better understanding of the aquifer and explain possible problems associated with the current data.

6.3.1 Water Depth Surface Tools

The water depth surface (WDS) tools were divided up into two parts: ensure the measurements are in the required units; for the second part the main tool generated the water depth surfaces using the spline interpolation method. Figure 6.12 shows the user interface for the first part of the tool. This tool requires the user to only select the well point features with the measurement information. One field is then added to the dataset.

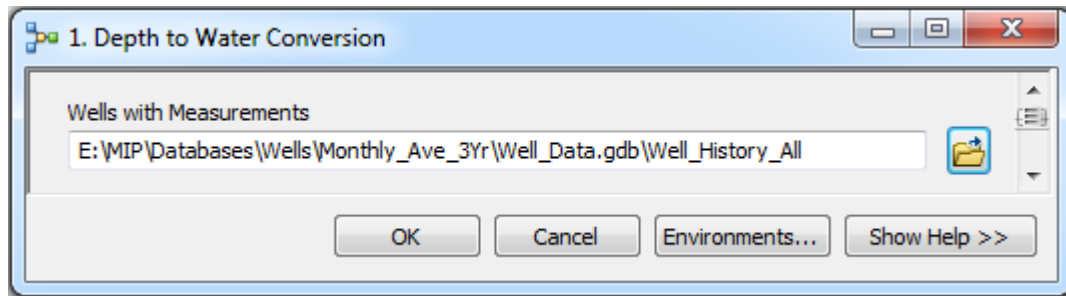


Figure 6.12 Unit Check Tool User Interface.

The second part of the tool generated the WDS. There are two variations of this second part: one to generate monthly WDS for each year in a specified time period, and one variation that generates average monthly WDS for time period specified. These two variations are based on the same back-end methodology.

Figure 6.13 shows the user interface to generate monthly WDS for each year. This requires the user to specify the well dataset with all measurements and the well location dataset. Next, the user needs to insert a start and end year, thus the process will be run for every year from beginning to end. The user must specify the cell size of the raster surface to be generated. A smaller cell sizes will cause the tool to take longer to run. The dataset containing the three aquifer subbasins needs to be specified by the user, as well as the location for the intermediate results (Scratch Database) and for the final results (Results Database).

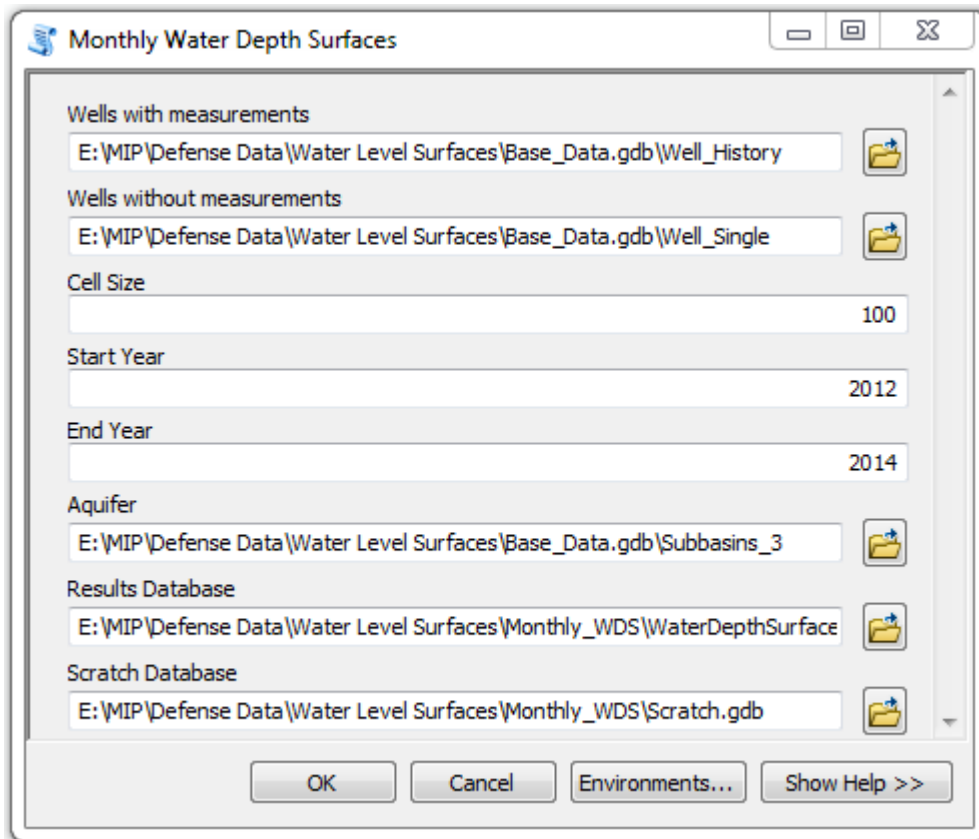


Figure 6.13 WDS Generation Tool User Interface, Variation 1 (Monthly WDS)

Figure 6.14 shows the user interface used to generate monthly averages for the specified time period. This tool required the wells dataset with the measurement and location information, and the cell size of the surfaces to be generated. The user must specify the start and end date, which designates the period to average for each month. It also requires workspace locations to be set for the intermediate results (Scratch Database), and the final spline surfaces that were generated and extracted to the study area (Results Database). Lastly it requires the dataset with the three aquifer subbasins.

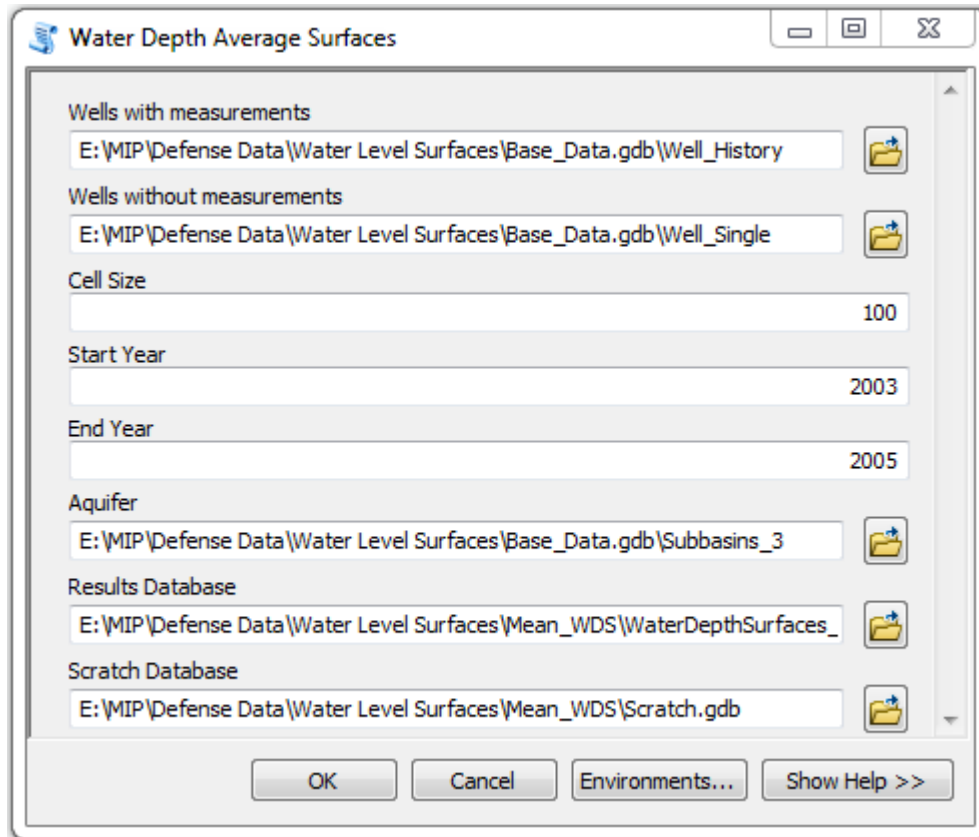


Figure 6.14 WDS Generation Tool User Interface, Variation 2 (Average Monthly WDS)

6.3.2 Water Depth Surfaces Results

Both the variations of the WDS generation tool was run to obtain results to compare and analyze. The amount of wells and the measurements from each well used when the tool is run was also investigated, since it was established that the temporal data coverage of the wells are not complete. Figure 6.15 shows the wells and how they are distributed across the three subbasins: Desert Hot Springs, Indio, and Mission Creek.

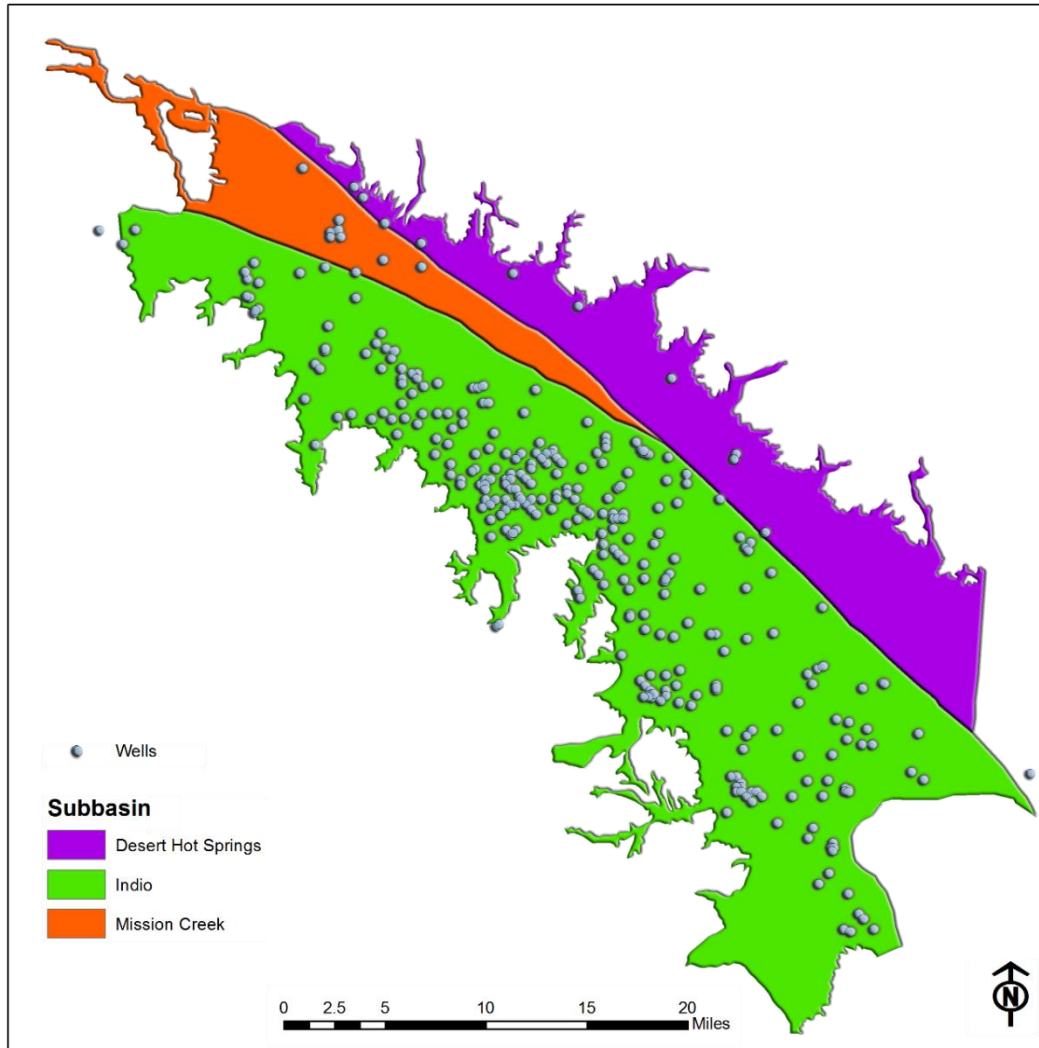


Figure 6.15 Well Distribution in Three of the Four Coachella Valley Subbasins

Monthly results were generated for 2012 and 2014, and Figure 6.16 shows January and July for these two years with the wells used to generate these surfaces. When each of these four surfaces were generated, the amount of measurement records selected by the tool differed. This has to be kept in mind when the user analyzes these images to see changes in the surfaces and find possible trends, since this problem with the temporal data coverage influences the accuracy and consistency of the surfaces.

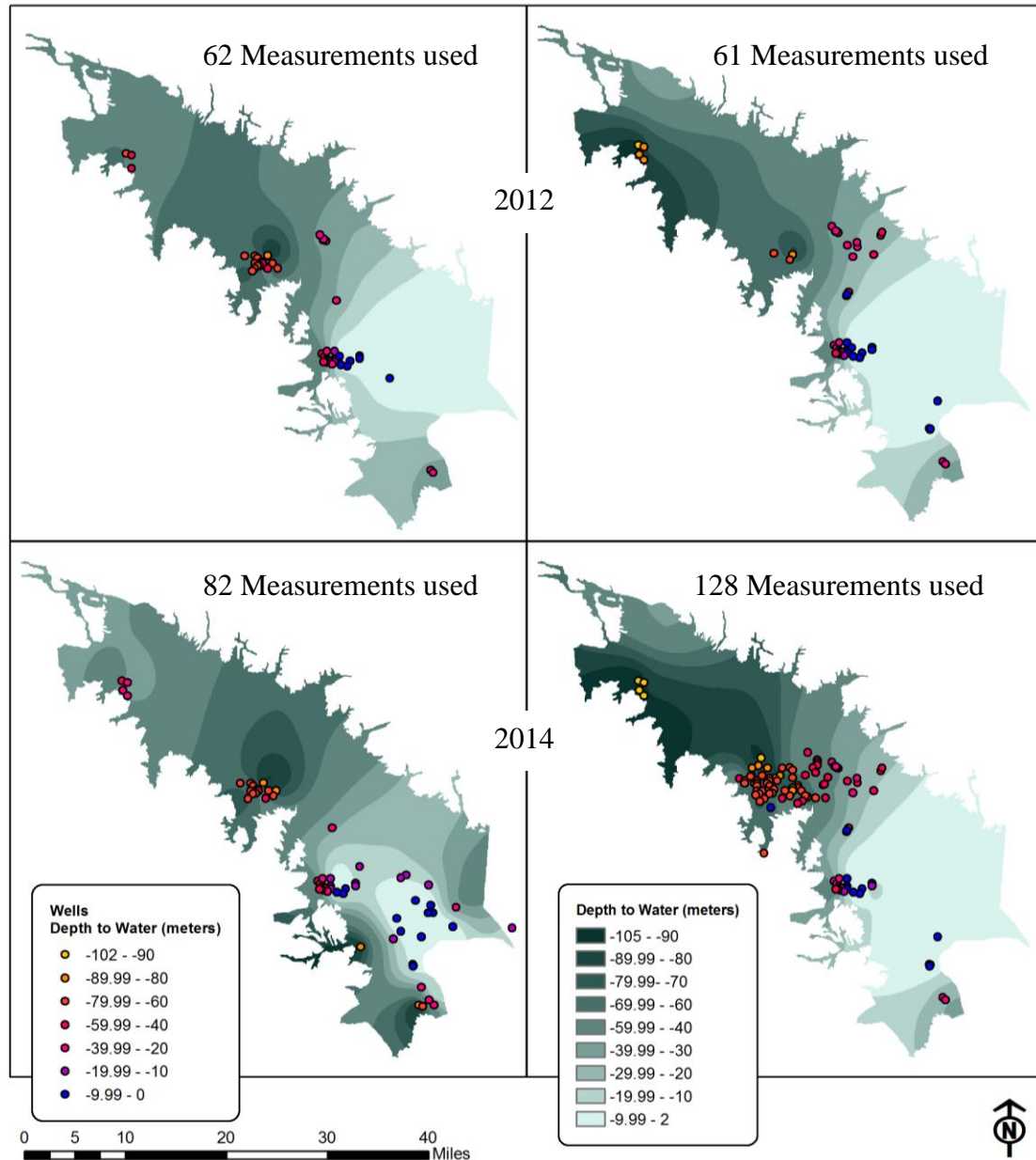


Figure 6.16 WDS Results for 2012 and 2014, for January (left) and July (right).

The average monthly WDS for a specified time period were then generated by running the Water Depth Average Surfaces tool, which is variation 2 of the WDS generation tool. Two sets of three year average surfaces were generated as examples: 2003 to 2005, and 2012 to 2014. The average January and July surfaces for both of these three year periods are shown by Figure 6.17, together with the wells used to generate the surfaces. By looking at these figures it is clear that the number of wells used to generate each of these average surfaces differ, which is due to the incomplete temporal coverage. To obtain the average value for each well that has measurements within this time period, the summed measurements were divided by the number of records used for that well.

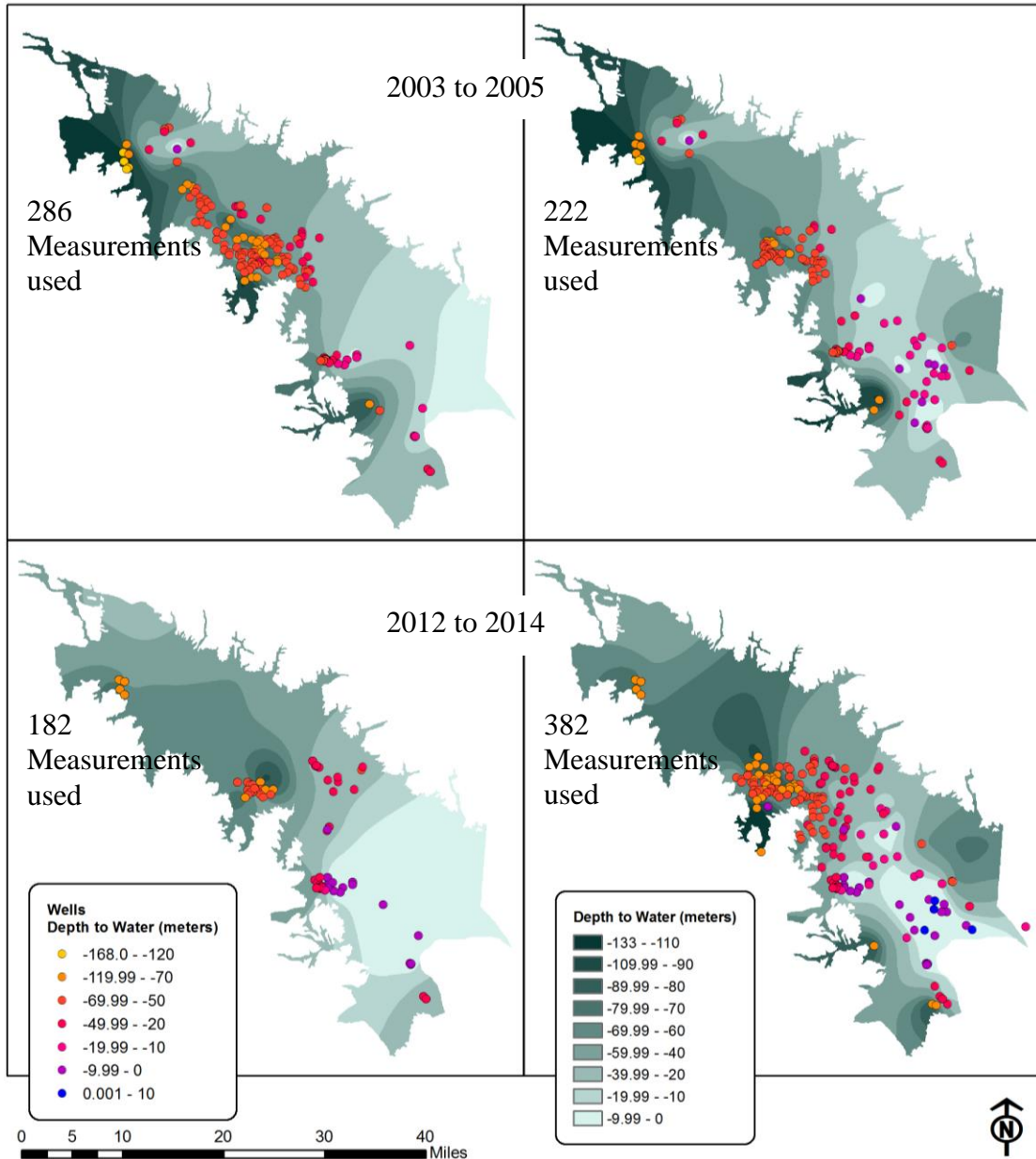


Figure 6.17 Three Year Average WDS of January (left) and July (right)

The surfaces generated by the WDS tools can be used to gain a better understanding of the WDS and how it changes over time. One method to utilize these surfaces is to calculate the difference between two surfaces to see the change over time. To illustrate this two years were considered: 2012, and 2014. The differences between two winter months, a winter and summer month, and start and end of the year were generated. Figure 6.18 shows the results obtained for the difference between WDS's. These differences can then be examined by the user to find possible trends.

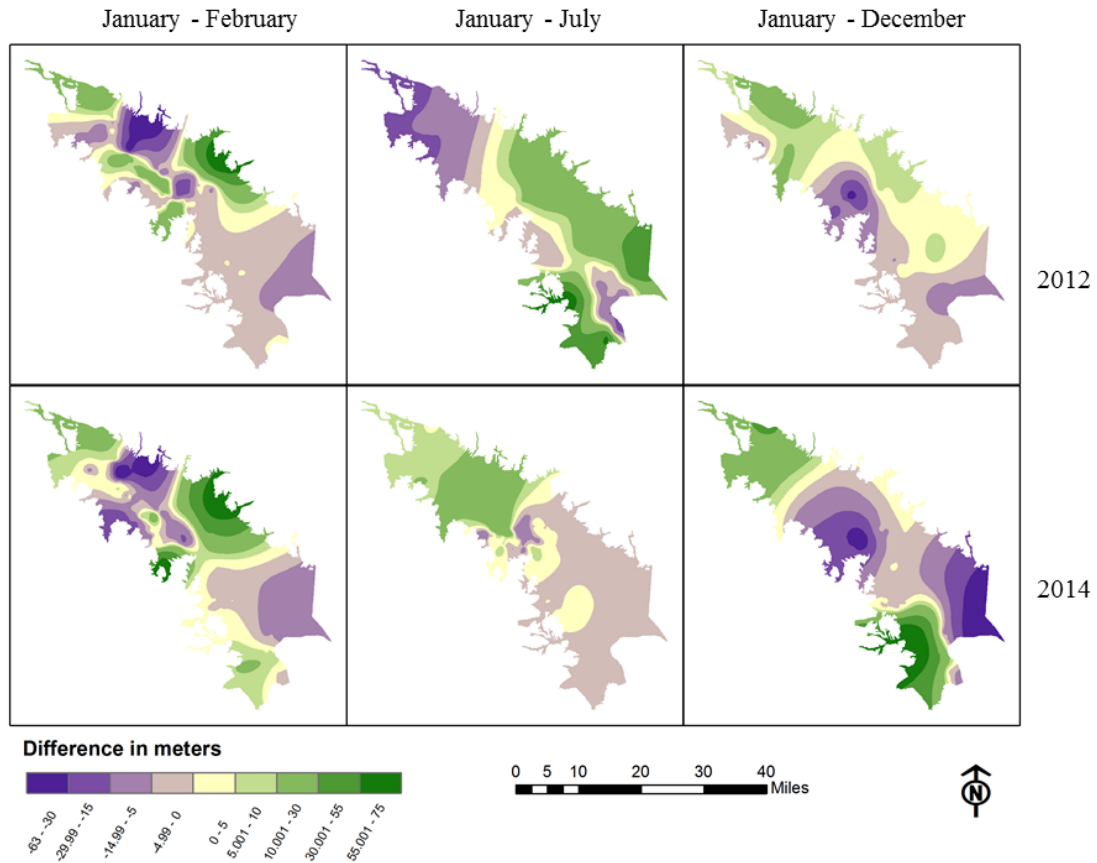


Figure 6.18. Differences between WDS's

In Figure 6.18 the purple colors indicate where January's water depth was greater, and the green colors indicate where the compared month (February, July, and December) showed deeper levels. The user can visually compare these images to find trend over years or between months. For example, the difference between January and February of 2012 and 2014 is similar, while the difference between January and December in these two years are very different.

6.3.3 3-D Visualization

To help gain a better understanding of the water depth level and how it changes over time a 3-D environment was used. The raster surfaces generated and the well measurements were visualized in a 3-D setting, which required ArcScene. A profile view was also created of the wells and their measurements to illustrate the rise and fall of the water depth.

When the surface rasters generated were pulled into the 3-D environment, flat surfaces were created, due to the difference of the vertical values being too small. These flat surfaces were not useful for visualizing the water depth level or to illustrate the differences between them. To obtain better 3-D visualization the surface values had to be adjusted. To adjust these values they were firstly extruded by a 100. This kept the value

ratio the same, while producing surfaces that better represented the differences in water levels.

The well data were also pulled into the 3-D setting to visualize how the wells and their measurements were distributed; not only on but beneath the ground surface. They were converted to a point feature class that contains z-values (elevation) by using the ArcGIS Feature to 3-D by Attribute tool. Next, they had to undergo the same adjustment as the surfaces to see the difference between the measurements. Figure 6.19 shows a side view, as seen from the east. The dark blue points are deeper beneath the ground surface than the yellow points. Looking at this side view, the middle of the aquifer contains more wells than the Salton Sea area (left of image) or the northern area (right of image).

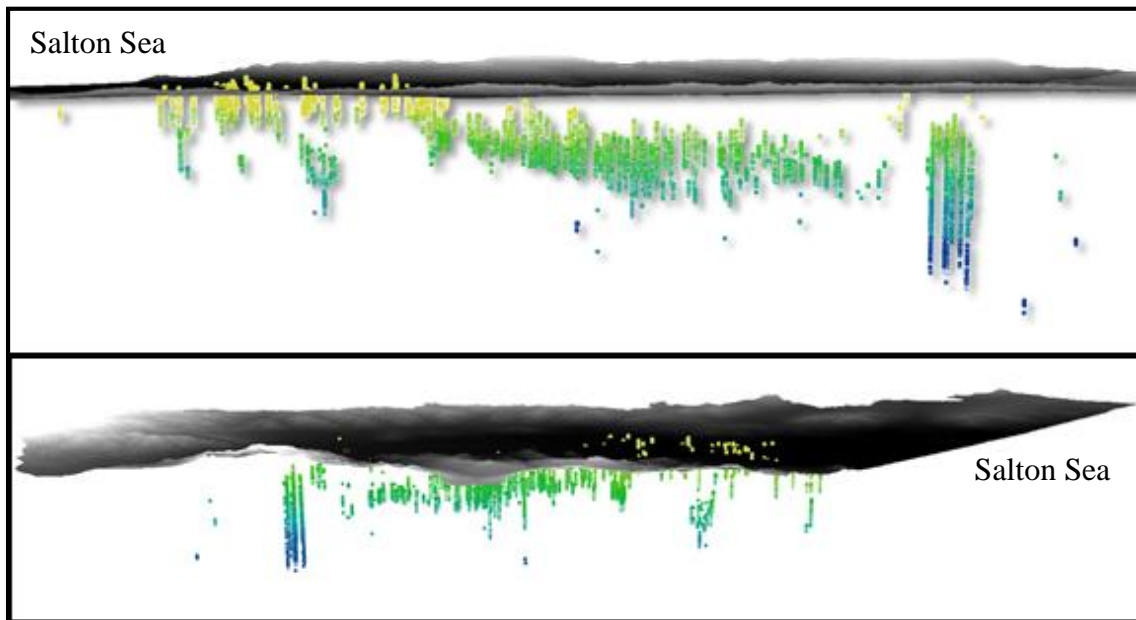


Figure 6.19 Two Side Views of all the Wells and their Measurements (1926 to 2015)

Taking a closer look at specific wells can help determine if an area's water level is stable or unstable. It can also give an indication of the relationship between the recharge and the discharge. Since when the recharge balances out the discharge the water depth should stay the same, and would be seen as stable. Figure 6.20 shows the location of the ground surface of well number 04S06E18P01S and 04S06E22C01S, and their measurements classified according to the year it was measured.

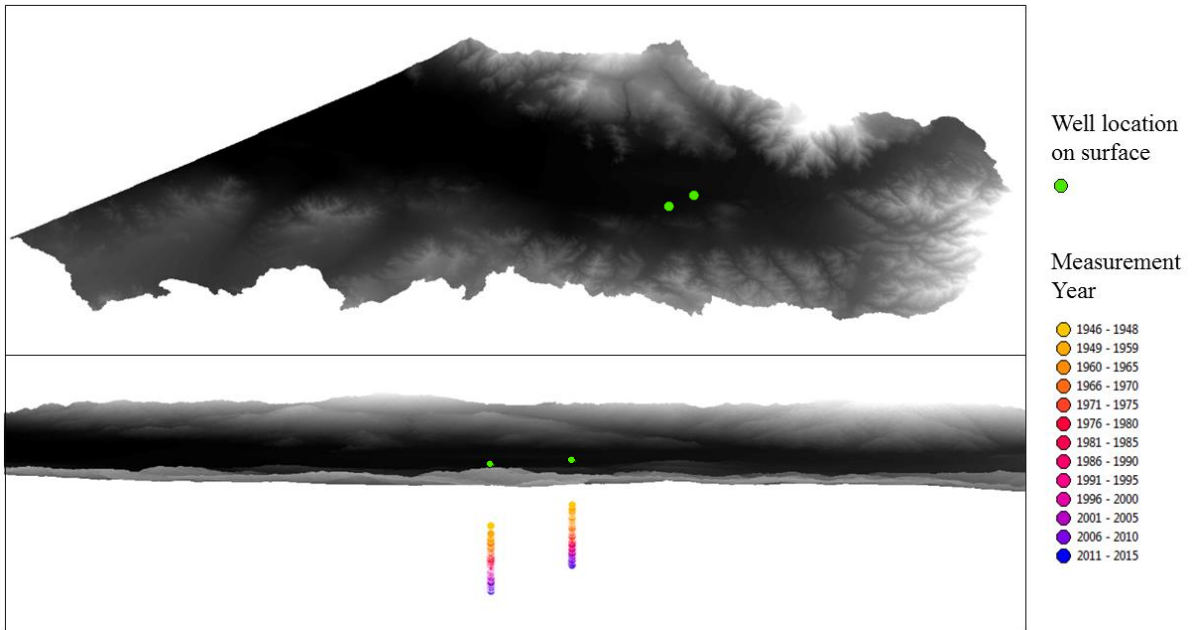


Figure 6.20 Closer Look at Two Wells

For these two selected wells the water depth has not been stable, as the measurements taken in earlier years are closer to the ground surface than the more recent measurements. Figure 6.21 shows a zoomed-in image of the wells in Figure 6.20. This image shows the trend of these wells water depth getting deeper clearly.

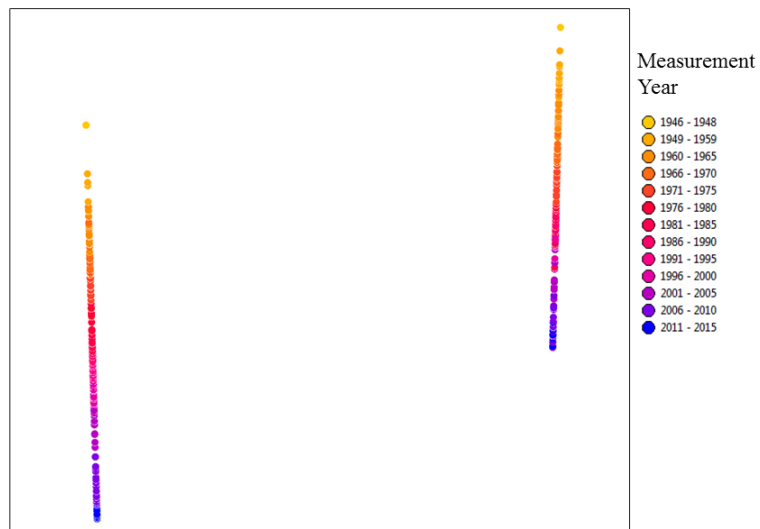


Figure 6.21 Close-up of Wells 04S06E18P01S and 04S06E22C01S

Lastly, a profile view of the wells and their measurements were created. This helped to understand the rise and fall of the water for a specific well, or a group of wells. To illustrate this three wells were selected, shown by Figure 6.22.

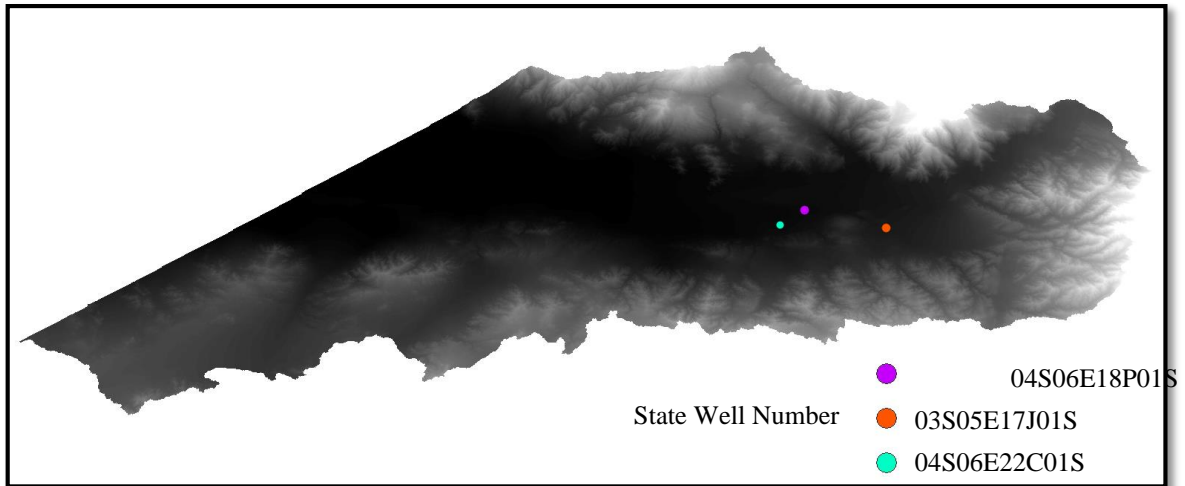


Figure 6.22 Three Wells Selected for Profile View

Figure 6.23 shows the profile view of these three wells: 03S05E17J01S, 04S06E18P01S, and 04S06E22C01S.

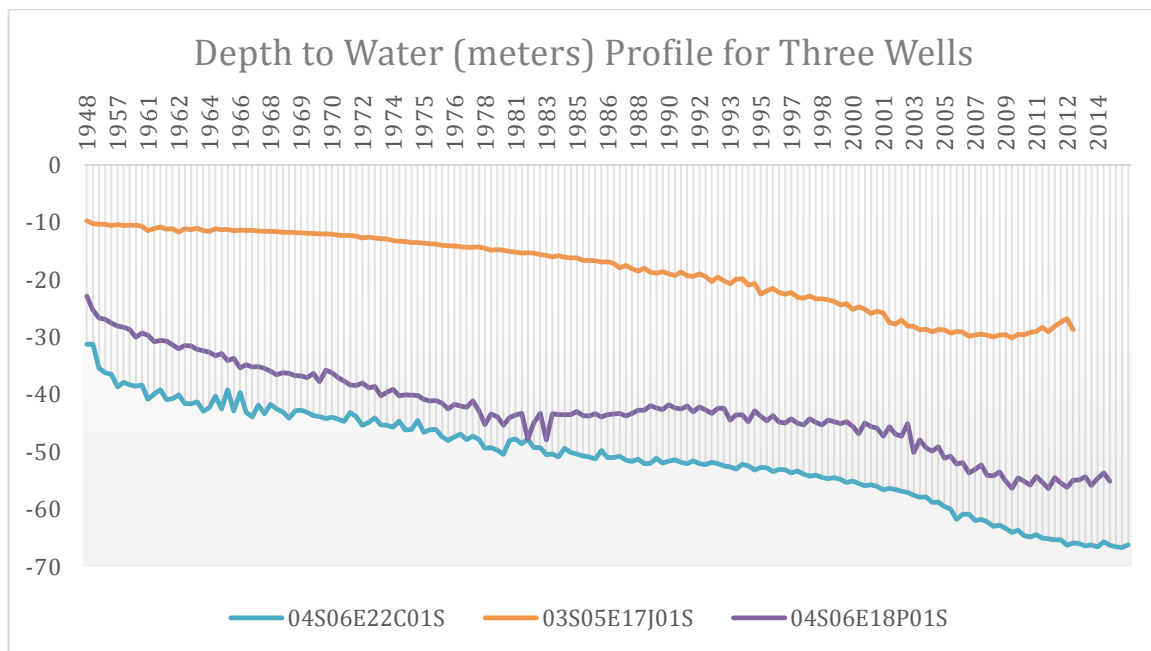


Figure 6.23 Depth to Water Profile

All three of these wells show an overall increase in the depth to water, thus the water level is getting deeper below the ground surface. Wells 03S05E17J01S and 04S06E18P01S started to show a decrease in the water depth, water level increasing and getting closer to ground surface, during their most recent measurements. This could be an indication of better recharge in the area.

6.4 Summary

The tools developed and what their results could be were discussed in this chapter. The user can calculate the infiltration for different precipitation data, and examples were used to show what the results would look like. The land use of specified areas within the study area can be changed when the user interactively selects an area of interest and new land cover classification. The infiltration calculation can be run on this new land cover to see what the change in infiltration would be. The last tool discussed generates water depth surfaces, which was generated from the well measurements. Methods of how these surfaces could be used by the user were discussed, and included looking at the differences between surfaces and visualizing the results in 3-D.

Chapter 7 – Conclusions and Future Work

The Coachella Valley is an oasis in a desert region, due to its aquifer. This aquifer is a main source of water for the region. As long as the aquifer can still provide water, the region will continue to grow and develop. Observation over the years show that in some areas the depth of water table is getting lower. When this aquifer is depleted it will collapse and result in structural and infrastructure damage. The CVEP needed a method to analyze the Coachella Valley water budget, to help with future development decisions. The GIS tools developed in this project provide a way to easily analyze and visualize aspects of the water budget.

The tools were developed using Python scripts and ArcGIS Model Builder. Python was used for data preprocessing and cleanup. The adjusted Curve Number infiltration method was integrated with ArcGIS through a custom set of tools. This toolset consisted of the three parts to calculate the infiltration: calculating the CN raster, calculating the equation parameters, and calculating the final infiltration. The water depth surfaces were generated using a different set of custom which consisted of two parts: converting measurement units, and generating WDS using the spline interpolation method. Both toolsets were created using ArcGIS Model Builder and Python scripts. The Land Cover Change (LCC) tool was created using a Python script, allowing the user to change the land use classification for selected parcels and a selected land use classification.

The developed system included an initial ArcGIS file geodatabase, which housed most of the data required by the different tools, such as elevation, land cover, CN table, and soil datasets. The 800 m and 4 km PRISM precipitation data were stored in their own geodatabases, and each also had a geodatabase to store the clipped and resampled rasters. Empty file geodatabases were provided to store the results from the infiltration, WDS, and LCC toolsets, to better organize results.

In calculating infiltration a standard value was used for the initial loss. In the future a study could be done to adjust this value to more accurately represent the Coachella Valley and make the infiltration estimation more accurate. The two sets of precipitation data used in this project had coarse temporal or spatial resolution. Using precipitation data with improved temporal and spatial resolution, better comparisons could be made in the future between the water levels, infiltration, and recharge to better understand the water budget. Future studies could look into using different kind of precipitation data from different sources, such as data from NOAA, to obtain better resolution and to examine and compare the results from the different precipitation data.

It was not possible to calculate the total recharge of the aquifer in this project, because some required data were not available. Future studies could use the water budget equation to calculate the total recharge of the aquifer. This equation required the recharge by human factor, the site location, and amount. Using the Curve Number method, the runoff could also be calculated and used in the water budget equation with higher accuracy with improved precipitation data. Other future studies could also investigate the hydraulic conductivity, formation porosity, saturated thickness, and formation transmissivity to calculate the Darcy's Flow of the infiltrated water (groundwater flow). This would provide information on the amount of groundwater from infiltration each subbasin will receive. This information can then be used in the water budget equation to improve its accuracy.

Currently well water levels are not measured on a consistent basis, future studies could look into how to improve the data collection, especially more consistent measurement dates, of these wells to generate a more accurate interpretation of the water depth surfaces. The well water usage is also not tracked or updated after the initial permit was issued. The data used in this study had to rely on incomplete well water use data. Future studies could use updated well use data or find a method to update this usage information, which will provide a better look at which user groups use more of the aquifer water.

In conclusion, the developed tools fulfilled the functional and non-functional requirements set by the client. The system provides the user with surfaces for infiltration and water levels, as well as maps that help with informed decision making for future developments.

Works Cited

- Al Kuisi, M., & El-Naqa, A. (2013). GIS based spatial groundwater recharge estimation in the Jafr Basin, Jordan - application of wetpass models for arid regions. *Revista Mexicana de Ciencias Geologicas*, 30(1), 96–109.
- Austin, C. (2012). Agriculture in California's Coachella Valley Maven's Photoblog. Retrieved March 17, 2015, from <http://mavensphotoblog.com/2012/06/04/agriculture-in-californias-coachella-valley/>
- Craciun, A. I., & Haidu, I. (2011). *Indirect estimation of soil moisture, using GIS, for modelling the high floods generated by rain. Applications in Apuseni Mountains (Doctoral dissertation)*. "Babeş-Bolyai" University.
- Craciun, A. I., Haidu, I., Magyasi-Saska, Z., & Imbrone, A. I. (2009). Estimation of Runoff Coefficient According to Soil Moisture Using GIS Techniques. *Geographia Technica*, 2, 1–10.
- CVWD. (n.d.). Water and the Coachella Valley - Coachella Valley Water District. Retrieved March 17, 2015, from <http://www.cvwd.org/about/waterandcv.php#where>
- Esri. (2013). Arc Hydro: GIS for Water Resources. Retrieved March 31, 2015, from <http://www.esri.com/library/fliers/pdfs/archydro.pdf>
- FAO. (1989). The arid environments. In *Arid zone forestry: A guide for field technicians*. Food and Agriculture Organization of the United Nations. Retrieved from <http://www.fao.org/docrep/t0122e/t0122e03.htm>
- Gajbhiye, S., & Mishra, S. K. (2012). Application of NRSC-SCS Curve Number Model in Runoff Estimation Using RS & GIS. *IEEE-International Conference On Advances In Engineering, Science and Management (ICAESM)*, 346–352.
- Hammouri, N., & El-Naqa, A. (2007). Hydrological modeling of ungauged wadis in arid environments using GIS: A case study of Wadi Madoneh in Jordan. *Revista Mexicana de Ciencias Geologicas*, 24(2), 185–196.
- Hawkins, R. H., Jiang, R., Woodward, D. E., Hjelmfelt, A. T., Van Mullem, J. E., & Quan, Q. (2003). Runoff Curve Number Method: Examination of the Initial Abstraction Ratio. In *World Water & Environmental Resources Congress 2003* (pp. 1–10). American Society of Civil Engineers. [http://doi.org/10.1061/40685\(2003\)308](http://doi.org/10.1061/40685(2003)308)
- Hernandez, M., Miller, S. N., Goodrich, D. C., Goff, B. F., Kepner, W. F., Edmonds, C. M., & Bruce Jones, K. (2000). Modeling Runoff Response to Land Cover and Rainfall Spatial Variability in Semi-Arid Watersheds. *Environmental Monitoring and Assessment*, 64, 285–298.
- Hudson, N. W. (n.d.). Soil and water conservation in semi-arid areas. Retrieved March 31, 2015, from http://www.fao.org/docrep/t0321e/t0321e-11.htm#P1259_112543
- James, I. (2014). Despite drought, steady groundwater pumping. Retrieved November 8, 2015, from <http://www.desertsun.com/story/news/environment/2014/12/04/coachella-valley->

water-consumption-holds-steady/19877737/

- James, I. (2015). Desert oasis collides with drought. Retrieved November 8, 2015, from <http://www.desertsun.com/longform/news/environment/2015/04/17/desert-oasis-collides-drought/25942225/>
- June, S. (2008). Coachella Valley Basin. Retrieved March 17, 2015, from <http://academic.emporia.edu/schulmem/hydro/TERMPROJECTS/2008/June/sj571.html>
- Kresic, N., & Mikszewski, A. (2013). *Hydrological Conceptual Site Models*. Boca Raton: CRC Press.
- Kumar, C. (2003). Estimation of ground water recharge using soil moisture balance approach. *Journal of Soil and Water Conservation, Soil ...*, 247667, -. Retrieved from http://www.researchgate.net/publication/215973823_Estimation_of_Ground_Water_Recharge_Using_Soil_Moisture_Balance_Approach/file/f4d6a99d527a0c5d1d780d430aadc7d1.pdf
- Marx, J., & James, I. (2015). Farm water use comes under scrutiny. Retrieved November 8, 2015, from <http://www.desertsun.com/longform/news/environment/2015/04/20/farm-water-use-comes-scrutiny/26076211/>
- Molis, A. J. (2012). *Modeling Hydrologic Impacts of Land Cover Changes to the Keauhou Aquifer (Master's thesis)*. University of Redlands. Retrieved from http://inspire.redlands.edu/gis_gradproj/157
- Musy, A., & Higy, C. (1998). *Hydrologie Appliquée*. Bucarest: H*G*A.
- NGWA. (n.d.). Unconfined or water table aquifers - National Groundwater Association. Retrieved September 24, 2015, from <http://www.ngwa.org/fundamentals/hydrology/pages/unconfined-or-water-table-aquifers.aspx>
- Patten, C. (2014). Farms to Tables Around the World - Coachella Valley takes an eco-friendly approach to growing and distributing food. *Palm Springs Life*. Retrieved from <http://www.palmspringslife.com/Palm-Springs-Life/Coachella-Valley-Vision/September-2013/Farms-to-Tables-Around-the-World/>
- Plummer, A., & Woodward, D. (1998). The origin and derivation of Ia/S in the runoff curve number system. *International Water Resources Engineering Conference*, 1260–1265.
- Poddar, V. (2009). *A Project Manager's Perspective: The Importance of Early and Thorough Requirements Capture*. Retrieved from <https://books.google.com/books?id=2ECCrgEACAAJ>
- Poullain, J. (2012). Estimating Storm Water Runoff. In *Florida Erosion and Sediment Control Inspector's Manual*. PDH Online. Retrieved from [http://www.pdhonline.org/courses/h119/stormwater runoff.pdf](http://www.pdhonline.org/courses/h119/stormwater%20runoff.pdf)

- Schneiderman, E. M., Steenhuis, T. S., Thongs, D. J., Easton, Z. M., Zion, M. S., Neal, A. L., ... Todd Walter, M. (2007). Incorporating variable source area hydrology into a curve-number-based watershed model. *Hydrological Processes*, 21(25), 3420–3430. <http://doi.org/10.1002/hyp.6556>
- Soil Conservation Service. (1989). *Engineering Hydrology Training Series Module 104 Study Guide*.
- Stevens, M. (2014). With flush aquifer, Coachella Valley golf courses slow to conserve. Retrieved March 17, 2015, from <http://www.latimes.com/local/california/la-me-adv-golf-water-20141221-story.html#page=1>
- Strassberg, G., Jones, N. L., & Lemon, A. (2010). Arc Hydro Groundwater Data Model and Tools : Overview and Use Cases. *AQUA Mundi*, 1(2), 101–114. <http://doi.org/10.4409/Am-018-10-0014>
- Strassberg, G., Jones, N. L., & Maidment, D. R. (2011). *Arc Hydro Ground Water: GIS for Hydrogeology* (1st ed.). Redlands: Esri Press.
- USDA NRCS. (1986). *Urban Hydrology for Small Watersheds, Technical Release 55* (2nd ed.). USDA-NRCS, Conservation Engineering Division.
- USDA NRCS. (2008). Soil Quality Indicators: Infiltration. Retrieved from http://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053289.pdf
- USGS. (n.d.). USGS FAQs - Groundwater and Aquifers - What is the difference between a confined and a water-table (unconfined) aquifer? Retrieved September 24, 2015, from <http://www.usgs.gov/faq/categories/9812/2776>
- USGS. (2015). USGS Release: Groundwater Toolbox. Retrieved March 31, 2015, from <http://www.usgs.gov/newsroom/article.asp?ID=4105&from=rss#.VRsL5vnF9G2>
- Vieux, B. E. (2004). *Distributed Hydrologic Models Using GIS*. Norwell: Kluwer Academic Publishers.
- Weisheit, J. (2008). On The Colorado - Articles. Retrieved September 24, 2015, from <http://www.onthecolorado.com/articles.cfm?mode=detail&id=1225829163240>
- Zotarelli, L., & Dukes, M. (2010). Step by step calculation of the Penman-Monteith Evapotranspiration (FAO-56 Method). *Institute of Food and Agricultural Sciences*, 1–10. Retrieved from <https://edis.ifas.ufl.edu/pdf/FILES/AE/AE45900.pdf>

Appendix A. Python script for finding PLSS

```
import arcpy
import os
import sys
import xlrd
import xlwt

# Function to find the letter associated with the 40 acre parcel
def Find_40Acres_letter(qqsection):
    QQ = {'A': "NENE", 'B': "NWNE", 'C': "NENW", 'D': "NWNW", 'E': "SWNW", 'F': "SENW",
          'G': "SWNE", 'H': "SENE", 'J': "NESE", 'K': "NWSE", 'L': "NESW", 'M': "NWSW", 'N': "SWSW",
          'P': "SESW", 'Q': "SWSE", 'R': "SESE"}
    return QQ.get(qqsection, "nothing")

arcpy.enc.workspace = arcoy.GetParameterText(0)

# Assign variables
QQ_fc = 'LADESC'
QQ_fc_Alt = 'ALT_LADESC'
QQ_fc_Fields = ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey']

# Open the excel spreadsheets to use
wells_13_15 = xlrd.open_workbook(r'C:\MS GIS\MIP\CVWD\Well Levels History 2013-2015 (4-9-15)_mod.xls')
wells_13_15_sheet = wells_13_15.sheet_by_name('Sheet1')

wells_13 = xlrd.open_workbook(r'C:\MS GIS\MIP\CVWD\Copy of Well Levels History_(5-22-13)_mod.xls')
wells_13_sheet1 = wells_13.sheet_by_name('Sheet1')
wells_13_sheet2 = wells_13.sheet_by_name('Sheet2')

# Set-up of rows and columns
start_row = 2
ncols = wells_13_15_sheet.ncols
nrows = wells_13_15_sheet.nrows

ncols_1 = wells_13_sheet1.ncols
ncols_2 = wells_13_sheet2.ncols
nrows_1 = wells_13_sheet1.nrows
nrows_2 = wells_13_sheet2.nrows

row_values = []
state_well_found = []
lndkey_sect_qq = ''

# Loop through rows in excel spreadsheet
for a in range(1, nrows_1):
    # Assign values from spreadsheet to variables
    state_well_nr = wells_13_sheet1.cell_value(a,1)
    township = wells_13_sheet1.cell_value(a, 4)
    rangeE = wells_13_sheet1.cell_value(a, 6)
    section = wells_13_sheet1.cell_value(a, 8)
    QQ_letter = wells_13_sheet1.cell_value(a, 9)
    well_nr = wells_13_sheet1.cell_value(a,10)

    QQ_converted = Find_40Acres_letter(QQ_letter)
    if state_well_nr in state_well_found:
        continue

# Loop through feature class to find corresponding township, range, and section to find correct parcel.
# Done for both LADESC and ALT_LADESC
# Use Search cursor to iterate through PLSS feature class
with arcpy.da.SearchCursor(QQ_fc_Alt, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey'],
```

```

'Shape@']) as sc1_alt:
    for scrow_alt in sc1_alt:
        township_fc_alt = scrow_alt[3][6:8]
        range_fc_alt = scrow_alt[3][11:13]
        section_fc_alt = scrow_alt[1][1:]

        # If section, township, range, and quarter quarter values from excel row
        matches that of PLSS
        if section_fc_alt == section and township_fc_alt == township and range_fc_alt
        == rangeE and scrow_alt[2] == QQ_converted:
            lndkey_sect_qq = scrow_alt[3] + '_' + section_fc_alt + '_' + QQ_converted
            row_values.append((section_fc_alt, scrow_alt[2], scrow_alt[3],
            state_well_nr, scrow_alt[4], lndkey_sect_qq, 'excel_13_Sheet1'))
            state_well_found.append(state_well_nr)

        with arcpy.da.SearchCursor(QQ_fc, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey',
        'Shape@']) as sc1:
            for scrow in sc1:
                township_fc = scrow[3][6:8]
                range_fc = scrow[3][11:13]
                section_fc = scrow[1][1:]

                # If section, township, range, and quarter quarter values from excel row
                matches that of PLSS
                if section_fc == section and township_fc == township and range_fc == rangeE
                and scrow[2] == QQ_converted:
                    lndkey_sect_qq = scrow[3] + '_' + section_fc + '_' + QQ_converted
                    row_values.append((section_fc, scrow[2], scrow[3], state_well_nr,
                    scrow[4], lndkey_sect_qq, 'excel_13_Sheet1'))
                    state_well_found.append(state_well_nr)

print 'Done with excel workbook -2013 sheet1'

# First excel workbook, sheet 2
# Same process as above
for b in range(1, nrows_2):
    # Assign values from spreadsheet to variables
    state_well_nr = wells_13_sheet2.cell_value(b,1)
    township = wells_13_sheet2.cell_value(b, 4)
    rangeE = wells_13_sheet2.cell_value(b, 6)
    section = wells_13_sheet2.cell_value(b, 8)
    QQ_letter = wells_13_sheet2.cell_value(b, 9)
    well_nr = wells_13_sheet2.cell_value(b,10)

    QQ_converted = Find_40Acres_letter(QQ_letter)

    if state_well_nr in state_well_found:
        continue

    with arcpy.da.SearchCursor(QQ_fc_Alt, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey',
    'Shape@']) as sc2_alt:
        for scrow_alt in sc2_alt:
            township_fc_alt = scrow_alt[3][6:8]
            range_fc_alt = scrow_alt[3][11:13]
            section_fc_alt = scrow_alt[1][1:]

            # If section, township, range, and quarter quarter values from excel row
            matches that of PLSS
            if section_fc_alt == section and township_fc_alt == township and range_fc_alt
            == rangeE and scrow_alt[2] == QQ_converted:
                lndkey_sect_qq = scrow_alt[3] + '_' + section_fc_alt + '_' + QQ_converted
                row_values.append((section_fc_alt, scrow_alt[2], scrow_alt[3],
                state_well_nr, scrow_alt[4], lndkey_sect_qq, 'excel_13_Sheet2'))
                state_well_found.append(state_well_nr)

        with arcpy.da.SearchCursor(QQ_fc, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey',
        'Shape@']) as sc2:
            for scrow in sc2:
                township_fc = scrow[3][6:8]
                range_fc = scrow[3][11:13]

```

```

        section_fc = scrow[1][1:]

        # If section, township, range, and quarter quarter values from excel row
        matches that of PLSS
        if section_fc == section and township_fc == township and range_fc == rangeE
and scrow[2] == QQ_converted:
            lndkey_sect_qq = scrow[3] + '_' + section_fc + '_' + QQ_converted
            row_values.append((section_fc, scrow[2], scrow[3], state_well_nr,
scrow[4], lndkey_sect_qq, 'excel_13_Sheet2'))
            state_well_found.append(state_well_nr)

# Second excel workbook
# Same process as above
for i in range(1, nrows):
    # Assign values from spreadsheet to variables
    state_well_nr = wells_13_15_sheet.cell_value(i,1)
    township = wells_13_15_sheet.cell_value(i, 4)
    rangeE = wells_13_15_sheet.cell_value(i, 6)
    section = wells_13_15_sheet.cell_value(i, 8)
    QQ_letter = wells_13_15_sheet.cell_value(i, 9)
    well_nr = wells_13_15_sheet.cell_value(i,10)

    QQ_converted = Find_40Acres_letter(QQ_letter)

    if state_well_nr in state_well_found:
        continue

    with arcpy.da.SearchCursor(QQ_fc_Alt, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey',
'Shape@']) as sc_alt:
        for scrow_alt in sc_alt:
            township_fc_alt = scrow_alt[3][6:8]
            range_fc_alt = scrow_alt[3][11:13]
            section_fc_alt = scrow_alt[1][1:]

            # If section, township, range, and quarter quarter values from excel row
            matches that of PLSS
            if section_fc_alt == section and township_fc_alt == township and range_fc_alt
== rangeE and scrow_alt[2] == QQ_converted:
                lndkey_sect_qq = scrow_alt[3] + '_' + section_fc_alt + '_' + QQ_converted
                row_values.append((section_fc_alt, scrow_alt[2], scrow_alt[3],
state_well_nr, scrow_alt[4], lndkey_sect_qq, 'excel_15_Sheet1'))
                state_well_found.append(state_well_nr)

            with arcpy.da.SearchCursor(QQ_fc, ['OBJECTID_1', 'sectn', 'qqsection', 'lndkey',
'Shape@']) as sc:
                for scrow in sc:
                    township_fc = scrow[3][6:8]
                    range_fc = scrow[3][11:13]
                    section_fc = scrow[1][1:]

                    # If section, township, range, and quarter quarter values from excel row
                    matches that of PLSS
                    if section_fc == section and township_fc == township and range_fc == rangeE
and scrow[2] == QQ_converted:
                        lndkey_sect_qq = scrow[3] + '_' + section_fc + '_' + QQ_converted
                        row_values.append((section_fc, scrow[2], scrow[3], state_well_nr,
scrow[4], lndkey_sect_qq, 'excel_15_Sheet1'))
                        state_well_found.append(state_well_nr)

# Insert values collected above into empty feature class.
insert = arcpy.da.InsertCursor('Redo_1', ['sectn', 'qqsection', 'lndkey', 'State_Well_Nr',
'SHAPE@', 'lndkey_sect_qq', 'Sheet'])
for row in row_values:
    insert.insertRow(row)
del insert
del sc

```

Appendix B. Infiltration Python scripts and Models

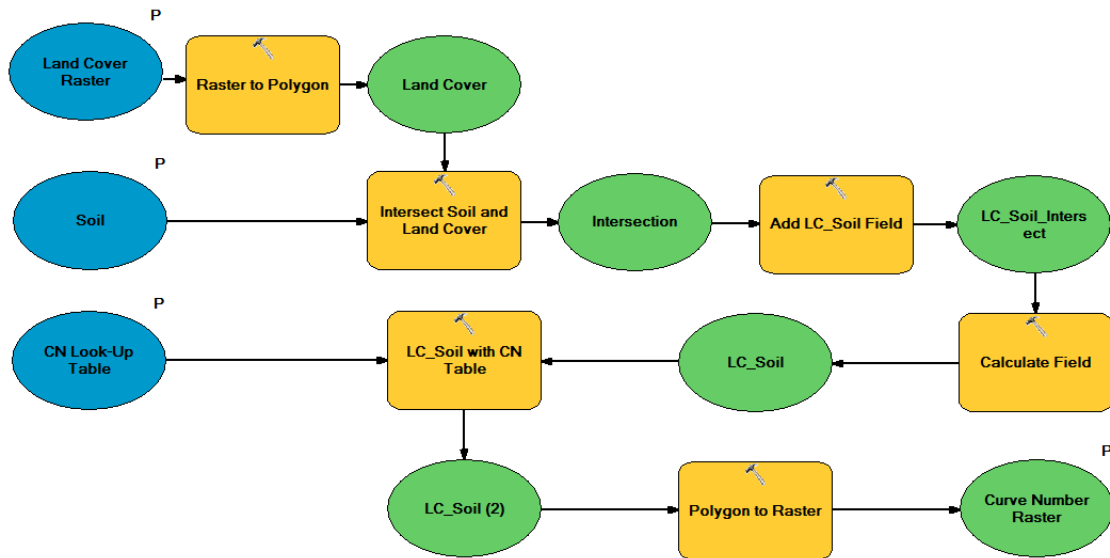


Figure B-1 Create CN Raster

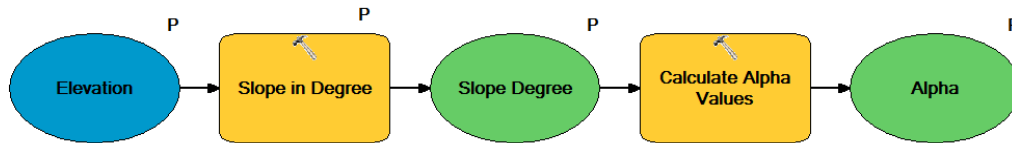


Figure B-2 Calculate α

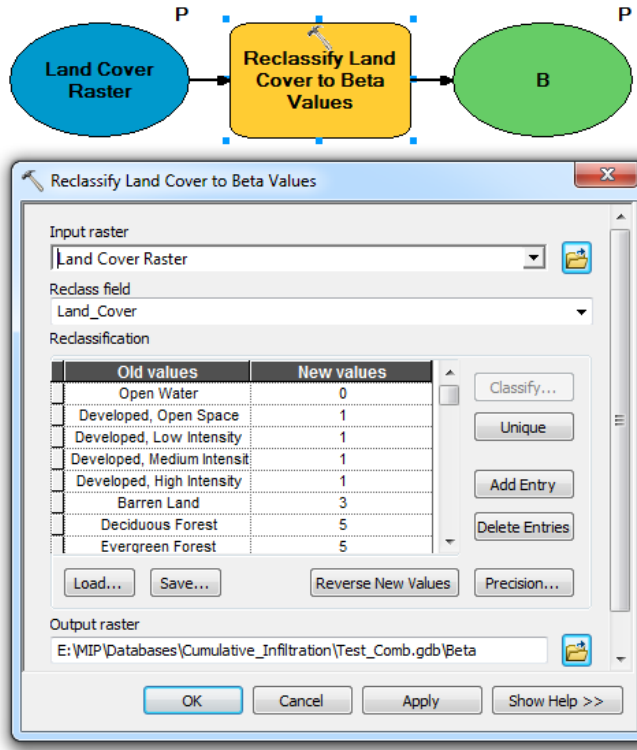


Figure B-3 Create β Raster

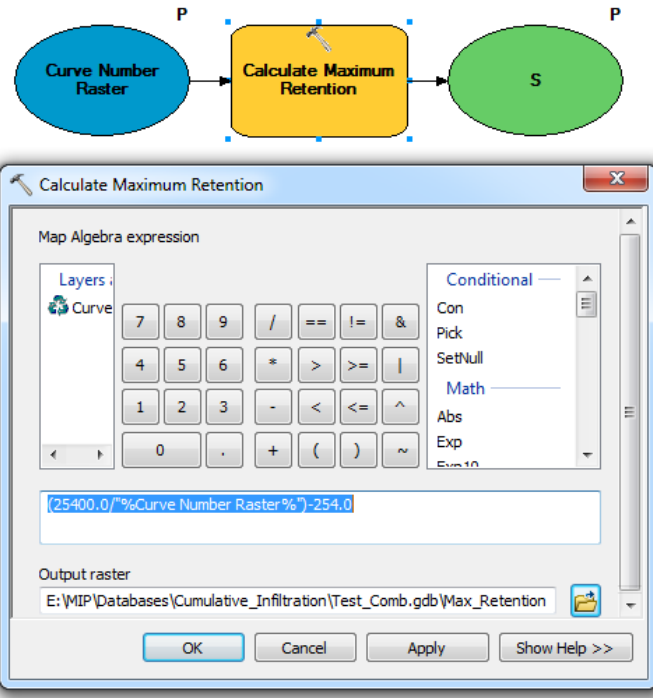


Figure B-4 Calculate Maximum Retention

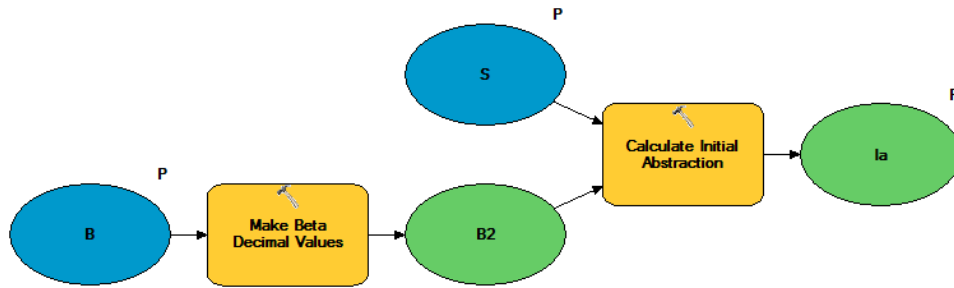


Figure B-5 Calculate Initial Abstraction

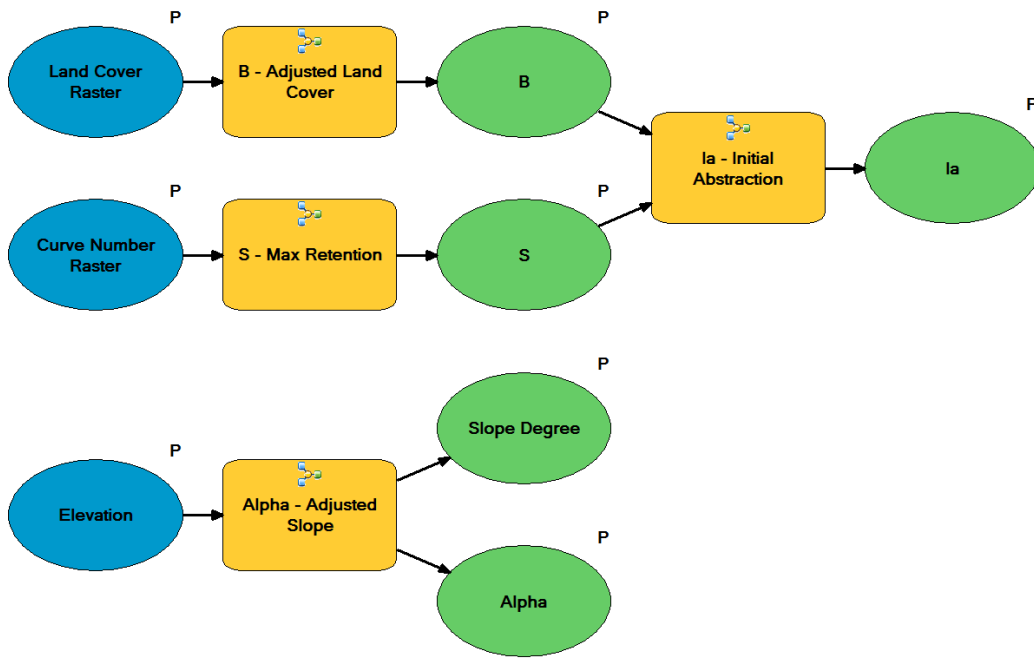


Figure B-6 Calculate Infiltration Equation Parameters

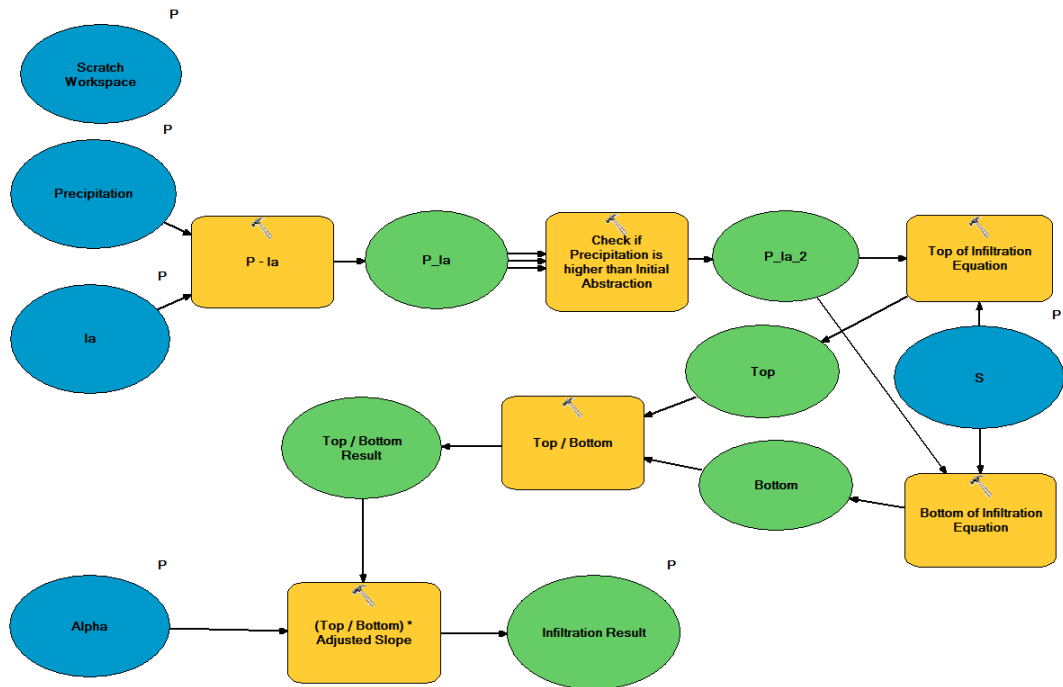


Figure B-7 Infiltration Calculation for Single Instance

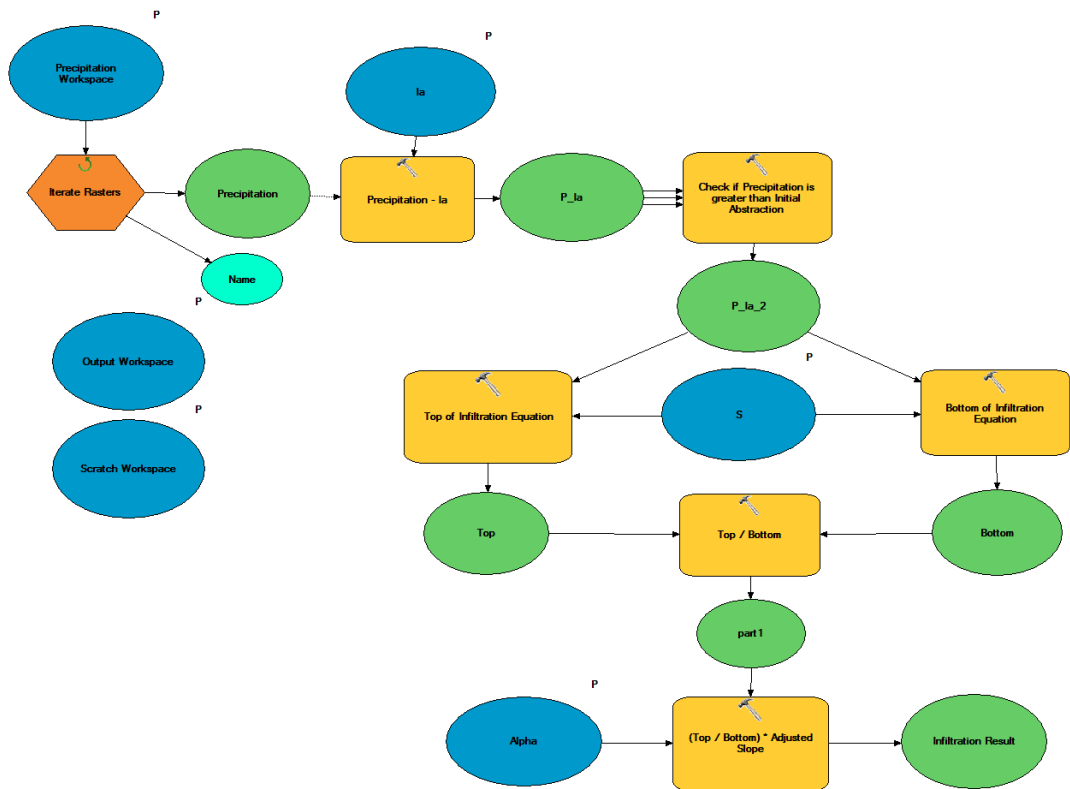


Figure B-8 Infiltration Calculation for Database

Appendix C. Land Cover Change Tool

```
import arcpy, sys, numpy, math
arcpy.CheckOutExtension('Spatial')
from arcpy.sa import *
from osgeo import gdal, gdalconst

arcpy.env.overwriteOutput = True
##### Setup Parameters #####
# There are seven parameters setup.

nlcd_raster = arcpy.GetParameterAsText(0)
mask_fc = arcpy.GetParameterAsText(1)
outName = arcpy.GetParameterAsText(2)
new_LC = arcpy.GetParameterAsText(3)
Study_Area = arcpy.GetParameterAsText(4)

##### Land Cover Metadata #####
# Load Land Cover input as a Raster
# Extract metadata information
nlcd = Raster(nlcd_raster)

nlcd_meta = arcpy.Describe(nlcd)
sr = nlcd_meta.SpatialReference
ext = nlcd_meta.Extent

col =nlcd.width
row = nlcd.height
lower_left = nlcd_meta.Extent.lowerLeft

arcpy.AddMessage('Metadata extracted')

##### NumPy Array setup #####
# The Land Cover raster is converted to a NumPy array
# A numpy array filled with zeros is created, with same amount of columns and rows
imgData = arcpy.RasterToNumPyArray(nlcd)
imgData_zeros = numpy.zeros((row,col), dtype=numpy.int32)

arcpy.AddMessage('Arrays setup')

##### Extract Area of Interes #####
# After the area of interest is extracted it is converted to a NumPy array
mask = ExtractByMask(nlcd, mask_fc).save('Mask_extract')
lc = numpy.matrix(arcpy.RasterToNumPyArray("Mask_extract",lower_left, col, row, 0),
dtype=numpy.int32)

arcpy.AddMessage('Area of interest extracted')

##### Functions #####
# Function to find the value associated with the land cover description
def lc_value(LandCover, description):
    lc_ass = {}
    with arcpy.da.SearchCursor(LandCover, ['Land_Cover', 'Value']) as sc:
        for lc in sc:
            lc_ass[lc[0]] = lc[1]
    return lc_ass.get(description, "nothing")
```

```

# Function to find land cover description associated with the values
def update_description(LandCover, value):
    lc_dict = {}
    with arcpy.da.SearchCursor(LandCover, ['Value','Land_Cover']) as sc:
        for row in sc:
            lc_dict[row[0]] = row[1]
    return lc_dict.get(value, "nothing")

##### Assign new values #####

arcpy.AddMessage('Assinging new land cover values')

new_LC_value = lc_value(nlcd_raster, new_LC)

# Iterate through the columns and rows area of interest NumPy array
for i in range(0,row):
    for j in range(0, col):
        if lc[i,j] >0:
            imgData_zeros[i,j] = new_LC_value
        else:
            imgData_zeros[i,j] = imgData[i,j]

arcpy.AddMessage('Finished assinging new land cover values')

##### Save New Land Cover raster #####

arcpy.AddMessage('Save new land cover raster')

# Setup output coordinate system
arcpy.env.outputCoordinateSystem = sr
# Convert array back to a raster and extract to Coachella Valley area
outRaster = arcpy.NumPyArrayToRaster(imgData_zeros,lower_left,30,30)
extract = ExtractByMask(outRaster, Study_Area).save(outName)

arcpy.AddMessage('Build attribute table')

# Build attribute table for new raster
arcpy.BuildRasterAttributeTable_management(outName, "Overwrite")

# Add new field to hold land cover description
arcpy.AddField_management(in_table=outName, field_name="Land_Cover", field_type="TEXT",
field_precision="", field_scale="", field_length="300", field_alias="",
field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")

arcpy.AddMessage('Fill attribute table with land cover descriptions')

# Update land cover description values using update_description function
with arcpy.da.UpdateCursor(outName, ["Value", "Land_Cover"]) as uc:
    for uc_row in uc:
        uc_row[1] = update_description(nlcd_raster, uc_row[0])
        uc.updateRow(uc_row)

```

Appendix D. Unit Conversion Model and Tool

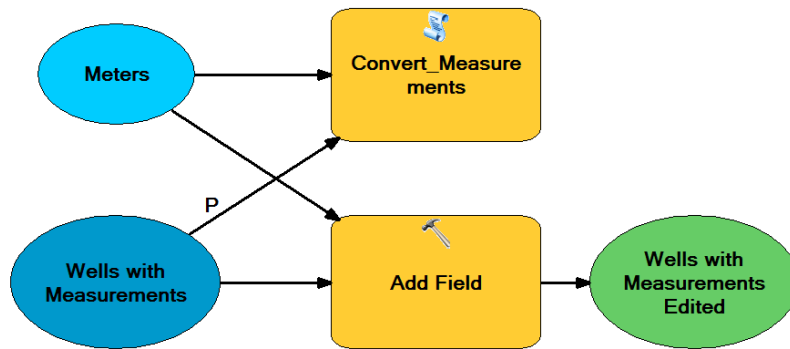


Figure D-1 Water Depth Measurement Unit Conversion

Update measurement unit script:

```
import arcpy

# Load parameters
fc = arcpy.GetParameterAsText(0)
depth_meters = arcpy.GetParameterAsText(1)

# Iterate through every record to update the new field
# The updated values are a conversion from feet to meters, by -0.3048
with arcpy.da.UpdateCursor(fc, ["DEPTH_TO_WATER_FT", depth_meters]) as uc:
    for row1 in uc:
        row1[1] = row1[0]*-0.3048
        uc.updateRow(row1)
```

Appendix E. Average Water Depth Surface Tool

```
# Import arcpy module
import arcpy
import os, sys
from arcpy import gp

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")
arcpy.env.overwriteOutput = True

# Script arguments
Well_History = arcpy.GetParameterAsText(0)
Wells = arcpy.GetParameterAsText(1)
Cell_Size = arcpy.GetParameterAsText(2)
start_Date = int(arcpy.GetParameterAsText(3))
end_Date = int(arcpy.GetParameterAsText(4))
Aquifer = arcpy.GetParameterAsText(5)
Results_Workspace = arcpy.GetParameterAsText(6)
Scratch_Workspace = arcpy.GetParameterAsText(7)

# Local variables:
# Original Single Wells
period = str(start_Date) + '_' + str(end_Date)
All_Year_ = os.path.join(Results_Workspace, 'All_' + period)

# January variables:
Jan_Statistics = os.path.join(Scratch_Workspace, 'Jan_Statistics')
Jan_Spline = os.path.join(Scratch_Workspace, 'Jan_' + period)
Jan_Extract = os.path.join(Results_Workspace, 'Jan_' + period+ '_Average')

# February variables:
Feb_Statistics = os.path.join(Scratch_Workspace, 'Feb_Statistics')
Feb_Spline = os.path.join(Scratch_Workspace, 'Feb_' + period)
Feb_Extract = os.path.join(Results_Workspace, 'Feb_' + period+ '_Average')

# March variables:
Mar_Statistics = os.path.join(Scratch_Workspace, 'Mar_Statistics')
Mar_Spline = os.path.join(Scratch_Workspace, 'Mar_' + period)
Mar_Extract = os.path.join(Results_Workspace, 'Mar_' + period+ '_Average')

# April variables:
Apr_Statistics = os.path.join(Scratch_Workspace, 'Apr_Statistics')
Apr_Spline = os.path.join(Scratch_Workspace, 'Apr_' + period)
Apr_Extract = os.path.join(Results_Workspace, 'Apr_' + period+ '_Average')

# May variables:
May_Statistics = os.path.join(Scratch_Workspace, 'May_Statistics')
May_Spline = os.path.join(Scratch_Workspace, 'May_' + period)
May_Extract = os.path.join(Results_Workspace, 'May_' + period+ '_Average')

# June variables:
Jun_Statistics = os.path.join(Scratch_Workspace, 'Jun_Statistics')
Jun_Spline = os.path.join(Scratch_Workspace, 'Jun_' + period)
Jun_Extract = os.path.join(Results_Workspace, 'Jun_' + period+ '_Average')

# July variables:
```

```

Jul_Statistics = os.path.join(Scratch_Workspace, 'Jul_Statistics')
Jul_Spline = os.path.join(Scratch_Workspace, 'Jul_' + period)
Jul_Extract = os.path.join(Results_Workspace, 'Jul_' + period+ '_Average')

# August variables:
Aug_Statistics = os.path.join(Scratch_Workspace, 'Aug_Statistics')
Aug_Spline = os.path.join(Scratch_Workspace, 'Aug_' + period)
Aug_Extract = os.path.join(Results_Workspace, 'Aug_' + period+ '_Average')

# September variables:
Sep_Statistics = os.path.join(Scratch_Workspace, 'Sep_Statistics')
Sep_Spline = os.path.join(Scratch_Workspace, 'Sep_' + period)
Sep_Extract = os.path.join(Results_Workspace, 'Sep_' + period+ '_Average')

# October variables:
Oct_Statistics = os.path.join(Scratch_Workspace, 'Oct_Statistics')
Oct_Spline = os.path.join(Scratch_Workspace, 'Oct_' + period)
Oct_Extract = os.path.join(Results_Workspace, 'Oct_' + period + '_Average')

# November variables:
Nov_Statistics = os.path.join(Scratch_Workspace, 'Nov_Statistics')
Nov_Spline = os.path.join(Scratch_Workspace, 'Nov_' + period)
Nov_Extract = os.path.join(Results_Workspace, 'Nov_' + period+ '_Average')

# December variables:
Dec_Statistics = os.path.join(Scratch_Workspace, 'Dec_Statistics')
Dec_Spline = os.path.join(Scratch_Workspace, 'Dec_' + period)
Dec_Extract = os.path.join(Results_Workspace, 'Dec_' + period+ '_Average')

##### Select Period #####
# Process: Select
arcpy.Select_analysis(Well_History, All_Year_, "YEAR >="+str(start_Date) + " AND YEAR
<=" + str(end_Date))

##### Make Feature Layers #####

# Process: Make Feature Layer for well data with single records
Wells_Layer = arcpy.MakeFeatureLayer_management(Wells, "Wells_Single_Layer", "", "",
"OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;SUBBASIN SUBBASIN VISIBLE NONE;ST
)

# Process: Make Feature Layer for well data with measurements
All_Year_Layer = arcpy.MakeFeatureLayer_management(All_Year_, "Wells_All_Layer", "", "",
"OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;SUBBASIN SUBBASIN VISIBLE NONE;ST
)

# Create buffer around the aquifer sub-basins
# One buffer will be 1.5 km and the second will be 2 pixels smaller than the first

```

```

Spline_Buffer = arcpy.Buffer_analysis(in_features=Aquifer,
out_feature_class='in_memory/Spline_Buffer', buffer_distance_or_field="1.5 Kilometers",
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field="",
method="PLANAR")
extract_Value = int(Cell_Size)*-2
buff_distance = str(extract_Value) + ' Kilometers'
Extract_Buffer = arcpy.Buffer_analysis(in_features=Spline_Buffer,
out_feature_class='in_memory/Extract_Buffer',buffer_distance_or_field= extract_Value,
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field="",
method="PLANAR")

print 'Temporary feature layers created'

##### January #####
# Process: Select Layer By Attribute January
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =1")

# Process: Summary Statistics for January
arcpy.Statistics_analysis(All_Year_Layer, Jan_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jan_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jan_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jan_Spline, Extract_Buffer, Jan_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with January')

##### February #####
# Process: Select Layer By Attribute Febraury
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =2")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Feb_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Feb_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

```



```

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Feb_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Feb_Spline, Extract_Buffer, Feb_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with February')

##### March #####
# Process: Select Layer By Attribute March
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =3")

# Process: Summary Statistics March
arcpy.Statistics_analysis(All_Year_Layer, Mar_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Mar_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Mar_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Mar_Spline, Extract_Buffer, Mar_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with March')

##### April #####
# Process: Select Layer By Attribute to find April
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =4")

# Process: Summary Statistics April
arcpy.Statistics_analysis(All_Year_Layer, Apr_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Apr_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Apr_Spline, "0")

```

```

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Apr_Spline, Extract_Buffer, Apr_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with April')

##### May #####
# Process: Select Layer By Attribute to find May
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =5")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, May_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=May_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, May_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(May_Spline, Extract_Buffer, May_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with May')

##### June #####
# Process: Select Layer By Attribute to find June
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =6")

# Process: Summary Statistics for June
arcpy.Statistics_analysis(All_Year_Layer, Jun_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jun_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jun_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jun_Spline, Extract_Buffer, Jun_Extract)

```

```

arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with June')

##### July #####
# Process: Select Layer By Attribute to find July
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =7")

# Process: Summary Statistics for July
arcpy.Statistics_analysis(All_Year_Layer, Jul_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jul_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jul_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jul_Spline, Extract_Buffer, Jul_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with July')

##### August #####
# Process: Select Layer By Attribute to find August
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =8")

# Process: Summary Statistics for August
arcpy.Statistics_analysis(All_Year_Layer, Aug_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Aug_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Aug_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Aug_Spline, Extract_Buffer, Aug_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with August')

```

```

##### September #####
# Process: Select Layer By Attribute to find September
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =9")

# Process: Summary Statistics for September
arcpy.Statistics_analysis(All_Year_Layer, Sep_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Sep_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Sep_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Sep_Spline, Extract_Buffer, Sep_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with September')

##### October #####
# Process: Select Layer By Attribute to find October
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =10")

# Process: Summary Statistics for October
arcpy.Statistics_analysis(All_Year_Layer, Oct_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Oct_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Oct_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Oct_Spline, Extract_Buffer, Oct_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with October')

##### November #####
# Process: Select Layer By Attribute to find November

```

```

arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =11")

# Process: Summary Statistics for November
arcpy.Statistics_analysis(All_Year_Layer, Nov_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Nov_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Nov_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Nov_Spline, Extract_Buffer, Nov_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with November')

##### December #####
# Process: Select Layer By Attribute to find December
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =12")

# Process: Summary Statistics for December
arcpy.Statistics_analysis(All_Year_Layer, Dec_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join - join mean depth field to temporary wells
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Dec_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Dec_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Dec_Spline, Extract_Buffer, Dec_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with December')
arcpy.AddMessage('Done with period: ' + str(period))

# Cleaning features classes from in_memory
arcpy.DeleteFeatures_management(Spline_Buffer)
arcpy.DeleteFeatures_management(Extract_Buffer)

```

Appendix F. Monthly Water Depth Surfaces Tool

```
# Import arcpy module
import arcpy
import os, sys
from arcpy import gp

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")
arcpy.env.overwriteOutput = True

# Script arguments
Well_History = arcpy.GetParameterAsText(0)
Wells = arcpy.GetParameterAsText(1)
Cell_Size = arcpy.GetParameterAsText(2)
start_Date = int(arcpy.GetParameterAsText(3))
end_Date = int(arcpy.GetParameterAsText(4))
Aquifer = arcpy.GetParameterAsText(5)
Results_Workspace = arcpy.GetParameterAsText(6)
Scratch_Workspace = arcpy.GetParameterAsText(7)

for Year in range(start_Date, end_Date+1):
    arcpy.AddMessage('Start with year: ' + str(Year))
    # Local variables:
    # Original Single Wells
    All_Year_ = os.path.join(Results_Workspace, 'All_' + str(Year))

    # January variables:
    Jan_Statistics = os.path.join(Scratch_Workspace, 'Jan_Statistics')
    Jan_Spline = os.path.join(Scratch_Workspace, 'Jan_' + str(Year))
    Jan_Extract = os.path.join(Results_Workspace, 'Jan_' + str(Year) + '_Result')

    # February variables:
    Feb_Statistics = os.path.join(Scratch_Workspace, 'Feb_Statistics')
    Feb_Spline = os.path.join(Scratch_Workspace, 'Feb_' + str(Year))
    Feb_Extract = os.path.join(Results_Workspace, 'Feb_' + str(Year) + '_Result')

    # March variables:
    Mar_Statistics = os.path.join(Scratch_Workspace, 'Mar_Statistics')
    Mar_Spline = os.path.join(Scratch_Workspace, 'Mar_' + str(Year))
    Mar_Extract = os.path.join(Results_Workspace, 'Mar_' + str(Year) + '_Result')

    # April variables:
    Apr_Statistics = os.path.join(Scratch_Workspace, 'Apr_Statistics')
    Apr_Spline = os.path.join(Scratch_Workspace, 'Apr_' + str(Year))
    Apr_Extract = os.path.join(Results_Workspace, 'Apr_' + str(Year) + '_Result')

    # May variables:
    May_Statistics = os.path.join(Scratch_Workspace, 'May_Statistics')
    May_Spline = os.path.join(Scratch_Workspace, 'May_' + str(Year))
    May_Extract = os.path.join(Results_Workspace, 'May_' + str(Year) + '_Result')

    # June variables:
    Jun_Statistics = os.path.join(Scratch_Workspace, 'Jun_Statistics')
    Jun_Spline = os.path.join(Scratch_Workspace, 'Jun_' + str(Year))
    Jun_Extract = os.path.join(Results_Workspace, 'Jun_' + str(Year) + '_Result')
```

```

# July variables:
Jul_Statistics = os.path.join(Scratch_Workspace, 'Jul_Statistics')
Jul_Spline = os.path.join(Scratch_Workspace, 'Jul_' + str(Year))
Jul_Extract = os.path.join(Results_Workspace, 'Jul_' + str(Year)+ '_Result')

# August variables:
Aug_Statistics = os.path.join(Scratch_Workspace, 'Aug_Statistics')
Aug_Spline = os.path.join(Scratch_Workspace, 'Aug_' + str(Year))
Aug_Extract = os.path.join(Results_Workspace, 'Aug_' + str(Year)+ '_Result')

# September variables:
Sep_Statistics = os.path.join(Scratch_Workspace, 'Sep_Statistics')
Sep_Spline = os.path.join(Scratch_Workspace, 'Sep_' + str(Year))
Sep_Extract = os.path.join(Results_Workspace, 'Sep_' + str(Year)+ '_Result')

# October variables:
Oct_Statistics = os.path.join(Scratch_Workspace, 'Oct_Statistics')
Oct_Spline = os.path.join(Scratch_Workspace, 'Oct_' + str(Year))
Oct_Extract = os.path.join(Results_Workspace, 'Oct_' + str(Year)+ '_Result')

# November variables:
Nov_Statistics = os.path.join(Scratch_Workspace, 'Nov_Statistics')
Nov_Spline = os.path.join(Scratch_Workspace, 'Nov_' + str(Year))
Nov_Extract = os.path.join(Results_Workspace, 'Nov_' + str(Year)+ '_Result')

# December variables:
Dec_Statistics = os.path.join(Scratch_Workspace, 'Dec_Statistics')
Dec_Spline = os.path.join(Scratch_Workspace, 'Dec_' + str(Year))
Dec_Extract = os.path.join(Results_Workspace, 'Dec_' + str(Year)+ '_Result')

##### Select Year #####
# Process: Select
arcpy.Select_analysis(Well_History, All_Year_, "YEAR =" + str(Year))

##### Make Feature Layers #####

# Process: Make Feature Layer for well data with single records
Wells_Layer = arcpy.MakeFeatureLayer_management(Wells, "Wells_Single_Layer", "", "",
"OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;SUBBASIN SUBBASIN VISIBLE NONE;ST
)

# Process: Make Feature Layer for well data with measurements
All_Year_Layer = arcpy.MakeFeatureLayer_management(All_Year_, "Wells_All_Layer", "",
"",
NONE;STATE_WELL_NR STATE_WELL_NR VISIBLE NONE;LATITUDE LATITUDE VISIBLE NONE;LONGITUDE
LONGITUDE VISIBLE NONE;DEPTH_TO_WATER_FT DEPTH_TO_WATER_FT VISIBLE NONE;MEASUREMENT_DATE
MEASUREMENT_DATE VISIBLE NONE;YEAR YEAR VISIBLE NONE;MONTH MONTH VISIBLE
NONE;Depth_To_Water_Meters Depth_To_Water_Meters VISIBLE NONE;Depth_To_Water_Neg
Depth_To_Water_Neg VISIBLE NONE")

# Create buffer around the aquifer sub-basins
# One buffer will be 1.5 km and the second will be 2 pixels smaller than the first
Spline_Buffer = arcpy.Buffer_analysis(in_features=Aquifer,
out_feature_class='in_memory/Spline_Buffer', buffer_distance_or_field="1.5 Kilometers",
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field="",
method="PLANAR")

```

```

extract_Value = int(Cell_Size)*-2
buff_distance = str(extract_Value) + ' Kilometers'
Extract_Buffer = arcpy.Buffer_analysis(in_features=Spline_Buffer,
out_feature_class='in_memory/Extract_Buffer',buffer_distance_or_field= extract_Value,
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field="",
method="PLANAR")

print 'Temporary feature layers created'

##### January #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =1")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Jan_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jan_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jan_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jan_Spline, Extract_Buffer, Jan_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with January')

##### February #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =2")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Feb_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Feb_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Feb_Spline, "0")

```



```

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Feb_Spline, Extract_Buffer, Feb_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with February')

##### March #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =3")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Mar_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Mar_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Mar_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Mar_Spline, Extract_Buffer, Mar_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with March')

##### April #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =4")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Apr_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Apr_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Apr_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Apr_Spline, Extract_Buffer, Apr_Extract)
arcpy.gp.Delete_management(temp_Wells)

```

```

arcpy.AddMessage('Done with April')

##### May #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =5")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, May_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=May_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, May_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(May_Spline, Extract_Buffer, May_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with May')

##### June #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =6")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Jun_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jun_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jun_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jun_Spline, Extract_Buffer, Jun_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with June')

```

```

##### July #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =7")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Jul_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Jul_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Jul_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Jul_Spline, Extract_Buffer, Jul_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with July')

##### August #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =8")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Aug_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Aug_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Aug_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Aug_Spline, Extract_Buffer, Aug_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with August')

##### September #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =9")

```

```

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Sep_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Sep_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Sep_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Sep_Spline, Extract_Buffer, Sep_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with September')

##### October #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =10")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Oct_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Oct_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Oct_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Oct_Spline, Extract_Buffer, Oct_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with October')

##### November #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =11")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Nov_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

```

```

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Nov_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Nov_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Nov_Spline, Extract_Buffer, Nov_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with November')

##### December #####
# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(All_Year_Layer, "NEW_SELECTION", "MONTH =12")

# Process: Summary Statistics
arcpy.Statistics_analysis(All_Year_Layer, Dec_Statistics, "Depth_To_Water_Neg MEAN",
"STATE_WELL_NR")

# Process: Copy Features
# Create temporary wells dataset to which the mean depth to water field will be added.
temp_Wells = arcpy.CopyFeatures_management(Wells, 'in_memory/Wells_Single_Join')

# Process: Add Join
arcpy.JoinField_management(in_data=temp_Wells, in_field="STATE_WELL_NR",
join_table=Dec_Statistics, join_field="STATE_WELL_NR", fields="MEAN_Depth_To_Water_Neg")

# Process: Spline with Barriers
arcpy.SplineWithBarriers_3d(temp_Wells, "MEAN_Depth_To_Water_Neg", Spline_Buffer,
Cell_Size, Dec_Spline, "0")

# Process: Extract by Mask
arcpy.gp.ExtractByMask_sa(Dec_Spline, Extract_Buffer, Dec_Extract)
arcpy.gp.Delete_management(temp_Wells)

arcpy.AddMessage('Done with December')
arcpy.AddMessage('Done with year: ' + str(Year))

# Cleaning features classes from in_memory
arcpy.DeleteFeatures_management(Spline_Buffer)
arcpy.DeleteFeatures_management(Extract_Buffer)

```