

Final Project for Introduction to GIS Programming:

Assessment of the proposed critical habitat
for the threatened coastal California gnatcatcher
(*Polioptila californica californica*)



Krysta Rogers
Geography 375
Spring 2011

Introduction:

California gnatcatcher

- Small, non-migratory songbird
- Inhabits specific vegetation community called coastal sage scrub
- Listed as threatened under the Endangered Species Act in 1993
 - Threats include:
 - Loss of habitat due to development, exotic plants, & fire
 - Reproductive loss from invasive species (e.g. brown headed cowbird)



Introduction:

Coastal sage scrub

- Plant community dominated by shrubs & other species adapted to arid conditions
 - California sage brush, sages, California buckwheat, redberry, lemonade-berry & cacti
- 90% decline from its original range
 - Threats include urban development, agriculture, & population growth
- Less than 10% of remaining habitat is protected



Introduction:

Project goals

- Use Python to evaluate the designated critical habitat for the California gnatcatcher in relation to a vegetation layer for Orange County
- Identify coastal sage scrub habitat not included within the designated critical habitat developed by USFWS

Methods:

Original files

- California gnatcatcher proposed critical habitat
 - ds131 – shapefile
- California counties
 - county100k – feature class
- Orange County vegetation
 - oraveg – feature class
 - oraveg.lut – info table
 - oraveglookup – dBase table

Methods:

Data preparation

- ArcCatalog, placed all the feature classes & shapefile into a single folder labeled “Data”
- Explored feature classes & shapefile in ArcMap
- Determine field names for queries & how vegetation is named & coded

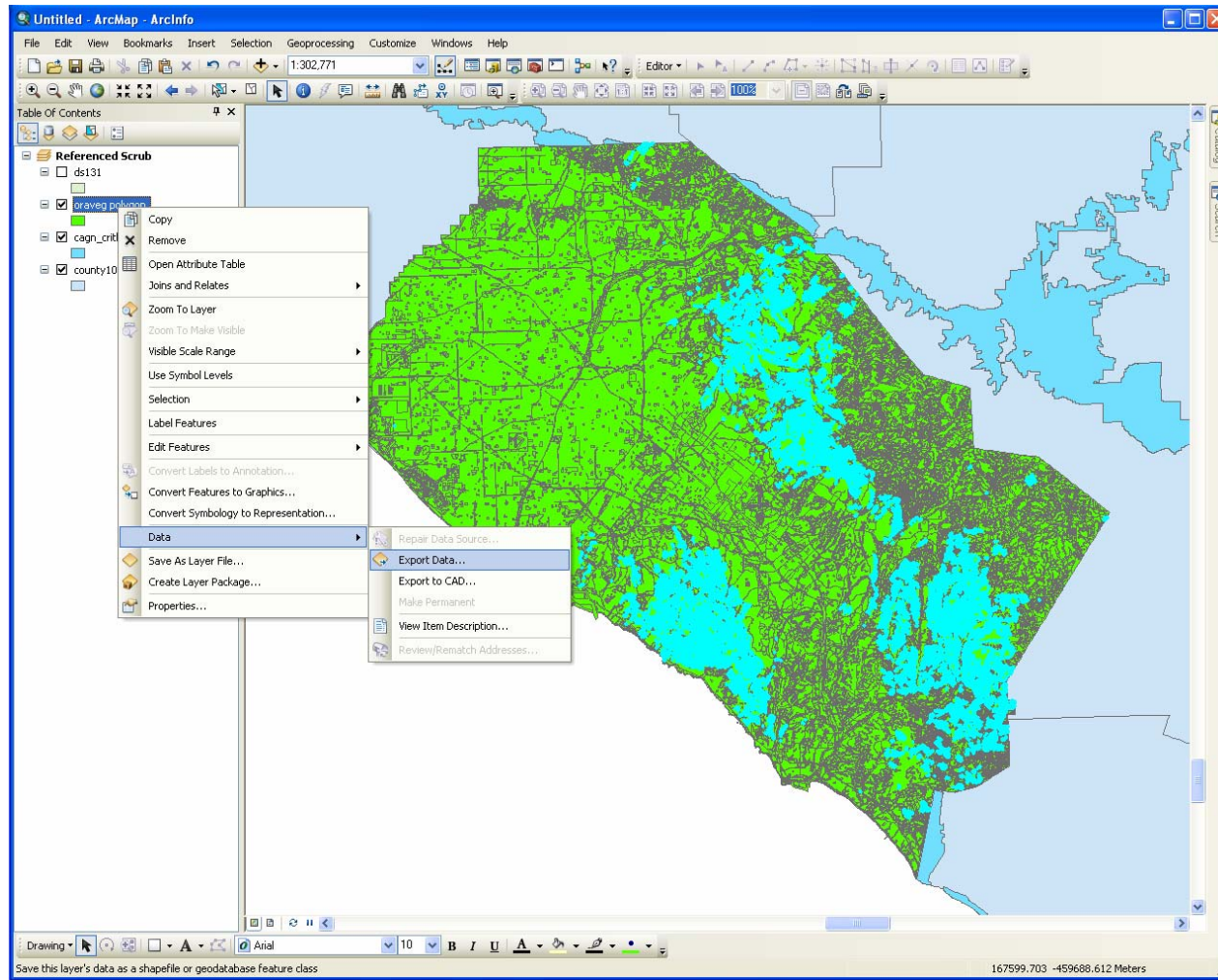
Methods:

Data preparation

- Counties layer into shapefile
 - CAcounties – shapefile
- Condensed vegetation layer into layer only including coastal sage scrub plant species
 - CoastalSageScrub – shapefile
 - SageScrubCommunities – shapefile
- Excluded “non-habitat” data from gnatcatcher critical habitat layer
 - cagn crithab – shapefile

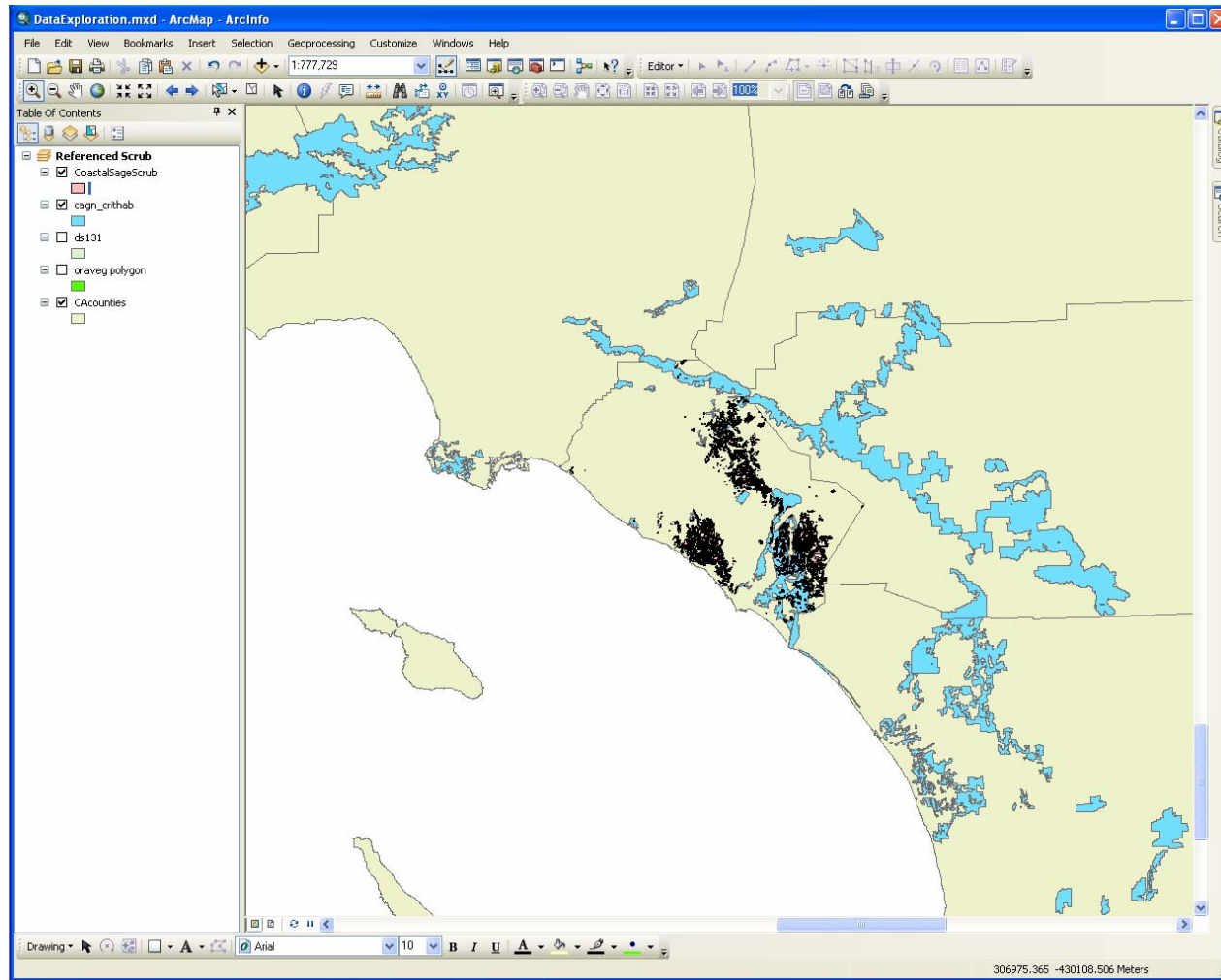
Methods:

Data preparation



Methods:

Data preparation



[Methods:

Python script development]

- Added import, arcpy, sys, traceback modules
- Identified workspace
- Added framework for try: & except: blocks
- Identified file names for each process

Methods: Python script development

import

```
KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help

# Final Project
# Created by: Krysta Rogers
# Created on: 04.01.2011
# Updated on: 05.09.2011

import arcpy, sys, traceback

arcpy.env.workspace = 'C:\\G375_Spring11\\FinalProject\\Data\\'
outpath = 'C:\\G375_Spring11\\FinalProject\\MyData\\'

# File names for selecting orange county from CA counties layer
in_counties = 'CAcounties.shp' # in county layer or "feature class"
out_county = outpath + 'Orange_County.shp' # this is the new layer for Orange county
counties_layer = 'counties' # this is the layer ("feature layer") produced from shapefile

# Clip file names for the CAGN critical habitat
infile = 'cagn_crithab.shp' # file to be clipped
clipfile = outpath + 'Orange_County.shp' # file to clip infile (orange county clips cagn habitat)
outfile = outpath + 'cagn_crithab_clip.shp' # output file, this should be clipped cagn habitat

# File names for erase analysis
# Want to look for scrub habitat not included in the cagn critical habitat
in_feature = 'SageScrubCommunities.shp' # input feature class
erase_feature = outpath + 'cagn_crithab_clip.shp' # features to be used to erase coincident features in input
out_feature_class = outpath + 'scrub_outside_crithab.shp' # name of new feature class that contains features that are outside of the erase features

# Want to join dBase table with plant community names to the code used in the vegetation layer
# Input layer has to be a feature layer not a feature class
# Will need to make a new layer before join

# File names for new layer for join
inFeature = outpath + 'scrub_outside_crithab.shp'
outsideHab = outpath + 'scrub_outside_crithab'

# File names for indexing for the join
in_table = outpath + 'scrub_outside_crithab.dbf' # table containing fields to be indexed
fields = "CTYPE" # list of fields that participate in the index, any number of fields can be specified
index_name = "INDEX" # name of new index

# File names for join
# Want to join dBase table with plant community names to the code used in the vegetation layer
in_layer = outpath + 'scrub_outside_crithab' # name of layer to which the join table will be joined
in_field = "CTYPE" # field in the input layer the join will be based on
join_table = 'oraveglookup.dbf' # table to be joined to input layer
join_field = "CTYPE" # field in table to be joined to input layer
join_type = "KEEP_COMMON" # keeps only matching records

out_layer = outpath + 'scrub_table_join' # output of Join to Layer File so join saves as permanent

try:

# Make a feature layer for Counties
print 'Make ' + counties_layer + ' feature layer.'
if arcpy.Exists(counties_layer):
    arcpy.Delete_management(counties_layer)

    arcpy.MakeFeatureLayer_management(in_counties, counties_layer)
```

try:

Methods: Python script development

except:

```
KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help

print 'Starting Erase routine'

# Parameters using variables
arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

print 'Finished Erase routine'

# Join routine to join the veg lookup table to the new layer
# This will enable me to identify the vegetation types rather than just the codes
# Need to make a feature layer from the scrub_outside_crithab shapefile made from the erase routine
# Create a feature layer from the scrub_outside_crithab shapefile

print 'Make ' + outsideHab + ' feature layer.'
if arcpy.Exists(outsideHab):
    arcpy.Delete_management(outsideHab)

arcpy.MakeFeatureLayer_management(inFeature, outsideHab)

print 'Finished Make Feature Layer routine'

# Need to add index to table

arcpy.AddIndex_management(in_table, fields, index_name, "NON_UNIQUE", "ASCENDING")

# Perform join

print 'Starting Join'

arcpy.AddJoin_management(in_layer, in_field, join_table, join_field, join_type)

print 'Finished Join routine'

# To make join permanent need to save to layer file

print 'Starting Save to Layer'

arcpy.SaveToLayerFile_management(outsideHab, out_layer)

print 'Finished saving layer'

except:

    tb = sys.exc_info()[2]
    tbinfo = traceback.format_tb(tb)[0]
    pymsg = "PYTHON ERRORS:\nTraceback Info:\n" + tbinfo + "\nError Info:\n" + str(sys.exc_type) + ": " + str(sys.exc_value) + "\n"
    msg = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msg)
    arcpy.AddError(pymsg)

print msg
print pymsg

arcpy.AddMessage(arcpy.GetMessages(1))
print arcpy.GetMessages(1)
```

Methods: Python script development

File names

```
KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help

# Final Project
# Created by: Krysta Rogers
# Created on: 04.01.2011
# Updated on: 05.09.2011

import arcpy, sys, traceback

arcpy.env.workspace = 'C:\\G375_Spring11\\FinalProject\\Data\\'
outpath = 'C:\\G375_Spring11\\FinalProject\\MyData\\'

# File names for selecting orange county from CA counties layer
in_counties = 'CAcounties.shp' # in county layer or "feature class"
out_county = outpath + 'Orange_County.shp' # this is the new layer for Orange county
counties_layer = 'counties' # this is the layer ("feature layer") produced from shapefile

# Clip file names for the CAGN critical habitat
infile = 'cagn_crithab.shp' # file to be clipped
clipfile = outpath + 'Orange_County.shp' # file to clip infile (orange county clips cagn habitat)
outfile = outpath + 'cagn_crithab_clip.shp' # output file, this should be clipped cagn habitat

# File names for erase analysis
# Want to look for scrub habitat not included in the cagn critical habitat
in_feature = 'SageScrubCommunities.shp' # input feature class
erase_feature = outpath + 'cagn_crithab_clip.shp' # features to be used to erase coincident features in input
out_feature_class = outpath + 'scrub_outside_crithab.shp' # name of new feature class that contains features that are outside of the erase features

# Want to join dBase table with plant community names to the code used in the vegetation layer
# Input layer has to be a feature layer not a feature class
# Will need to make a new layer before join

# File names for new layer for join
inFeature = outpath + 'scrub_outside_crithab.shp'
outsideHab = outpath + 'scrub_outside_crithab'

# File names for indexing for the join
in_table = outpath + 'scrub_outside_crithab.dbf' # table containing fields to be indexed
fields = "CTYPE" # list of fields that participate in the index, any number of fields can be specified
index_name = "INDEX" # name of new index

# File names for join
# Want to join dBase table with plant community names to the code used in the vegetation layer
in_layer = outpath + 'scrub_outside_crithab' # name of layer to which the join table will be joined
in_field = "CTYPE" # field in the input layer the join will be based on
join_table = 'oraveglookup.dbf' # table to be joined to input layer
join_field = "CTYPE" # field in table to be joined to input layer
join_type = "KEEP_COMMON" # keeps only matching records

out_layer = outpath + 'scrub_table_join' # output of Save to Layer File so join saves as permanent

try:

# Make a feature layer for Counties
print 'Make ' + counties_layer + ' feature layer.'
if arcpy.Exists(counties_layer):
    arcpy.Delete_management(counties_layer)

    arcpy.MakeFeatureLayer_management(in_counties, counties_layer)
```

Methods:

Python script development

- Developed script
 - Make Feature Layer for counties
 - Query Orange County
 - Clip for critical habitat by Orange County
 - Erase analysis to evaluate habitat outside of critical habitat
 - Add Join to add lookup table to vegetation layer
 - Index
 - Save to Layer

Methods:

Python script development

■ Make Feature Layer for counties

Make Feature Layer,
Query, &
Select by Attribute

```
KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help
# Create a feature layer for counties
in_field = "CTYPE" # field in the input layer the join will be based on
join_table = "orveglookup.dbf" # table to be joined to input layer
join_field = "CTYPE" # field in table to be joined to input layer
join_type = "KEEP_COMMON" # keeps only matching records

out_layer = outpath + 'scrub_table_join' # output of Save to Layer File so join saves as permanent

try:
    # Make a feature layer for Counties
    print 'Make ' + counties_layer + ' feature layer.'
    if arcpy.Exists(counties_layer):
        arcpy.Delete_management(counties_layer)

    arcpy.MakeFeatureLayer_management(in_counties, counties_layer)

    # Create a query statement to select Orange County (FEAT_NAME = Orange County)
    # Create query statement to select Orange County
    query = '"FEAT_NAME" = \'Orange County\''

    # Select by attributes using the query
    # Select features by attribute using query

    print 'Querying counties'
    arcpy.SelectLayerByAttribute_management(counties_layer, 'NEW_SELECTION', query)

    # Clip for cagn critical habitat in orange county
    # check to see if feature class already exists
    # if it does, delete it
    if arcpy.Exists(outfile):
        arcpy.Delete_management(outfile)

    print 'Starting Clip routine'

    # Parameters using variables
    arcpy.Clip_analysis(infile, clipfile, outfile)

    print 'Finished Clip routine'

    # Erase analysis to determine the habitat that occurs outside of cagn critical habitat
    # that should be assessed to determine if it is decent habitat.

    # check to see if feature class already exists
    # if it does, delete it
    if arcpy.Exists(out_feature_class):
        arcpy.Delete_management(out_feature_class)

    print 'Starting Erase routine'

    # Parameters using variables
    arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

    print 'Finished Erase routine'

Ln: 87 Col: 0
```

Methods:

Python script development

- Clip for critical habitat by Orange County

Clip

```
KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help
# Join two layers to create a new layer. # Make a layer to which the join table will be joined
in_field = "CTYPE" # field in the input layer the join will be based on
join_table = 'oraveglookup.dbf' # table to be joined to input layer
join_field = "CTYPE" # field in table to be joined to input layer
join_type = "KEEP_COMMON" # keeps only matching records

out_layer = outpath + 'scrub_table_join' # output of Save to Layer File so join saves as permanent

try:

# Make a feature layer for Counties
print 'Make ' + counties_layer + ' feature layer.'
if arcpy.Exists(counties_layer):
    arcpy.Delete_management(counties_layer)

    arcpy.MakeFeatureLayer_management(in_counties, counties_layer)

# Create a query statement to select Orange County (FEAT_NAME = Orange County)
# Create query statement to select Orange County
query = '"FEAT_NAME" = \'Orange County\''

# Select by attributes using the query
# Select features by attribute using query

print 'Querying counties'
arcpy.SelectLayerByAttribute_management(counties_layer, 'NEW_SELECTION', query)

# Clip for cagn critical habitat in orange county

# check to see if feature class already exists
# if it does, delete it
if arcpy.Exists(outfile):
    arcpy.Delete_management(outfile)

print 'Starting Clip routine'

# Parameters using variables
arcpy.Clip_analysis(infile, clipfile, outfile)

print 'Finished Clip routine'

# Erase analysis to determine the habitat that occurs outside of cagn critical habitat
# that should be assessed to determine if it is decent habitat.

# check to see if feature class already exists
# if it does, delete it

if arcpy.Exists(out_feature_class):
    arcpy.Delete_management(out_feature_class)

print 'Starting Erase routine'

# Parameters using variables
arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

print 'Finished Erase routine'
```


Methods:

Python script development

- Erase analysis to evaluate habitat outside of critical habitat

Erase

```
KRogers_FinalProject2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProject2.py
File Edit Format Run Options Windows Help
arcpy.SelectLayerByAttribute_management(counties_layer, 'NEW_SELECTION', query)

# Clip for eagn critical habitat in orange county

# check to see if feature class already exists
# if it does, delete it
if arcpy.Exists(outfile):
    arcpy.Delete_management(outfile)

print 'Starting Clip routine'

# Parameters using variables
arcpy.Clip_analysis(infile, clipfile, outfile)

print 'Finished Clip routine'

# Erase analysis to determine the habitat that occurs outside of eagn critical habitat
# that should be assessed to determine if it is decent habitat.

# check to see if feature class already exists
# if it does, delete it
if arcpy.Exists(out_feature_class):
    arcpy.Delete_management(out_feature_class)

print 'Starting Erase routine'

# Parameters using variables
arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

print 'Finished Erase routine'

# Join routine to join the veg lookup table to the new layer
# This will enable me to identify the vegetation types rather than just the codes
# Need to make a feature layer from the scrub_outside_crithab shapefile made from the erase routine

# Create a feature layer from the scrub_outside_crithab shapefile

print 'Make ' + outsideHab + ' feature layer.'
if arcpy.Exists(outsideHab):
    arcpy.Delete_management(outsideHab)

arcpy.MakeFeatureLayer_management(inFeature, outsideHab)

print 'Finished Make Feature Layer routine'

# Need to add index to table

arcpy.AddIndex_management(in_table, fields, index_name, "NON_UNIQUE", "ASCENDING")

# Perform join

print 'Starting Join'
arcpy.AddJoin_management(in_layer, in_field, join_table, join_field, join_type)

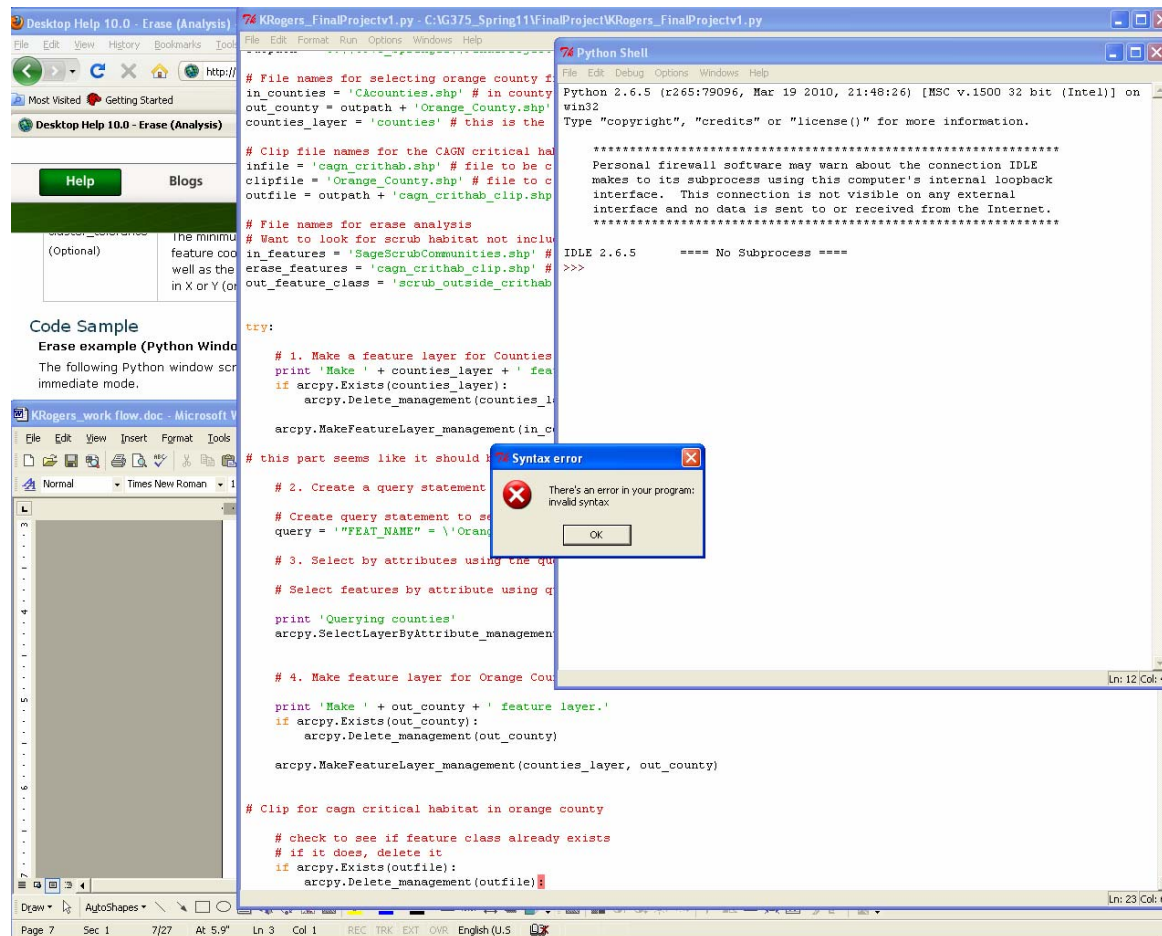
print 'Finished Join routine'

# To make join permanent need to save to layer file
```

Methods:

Python script development

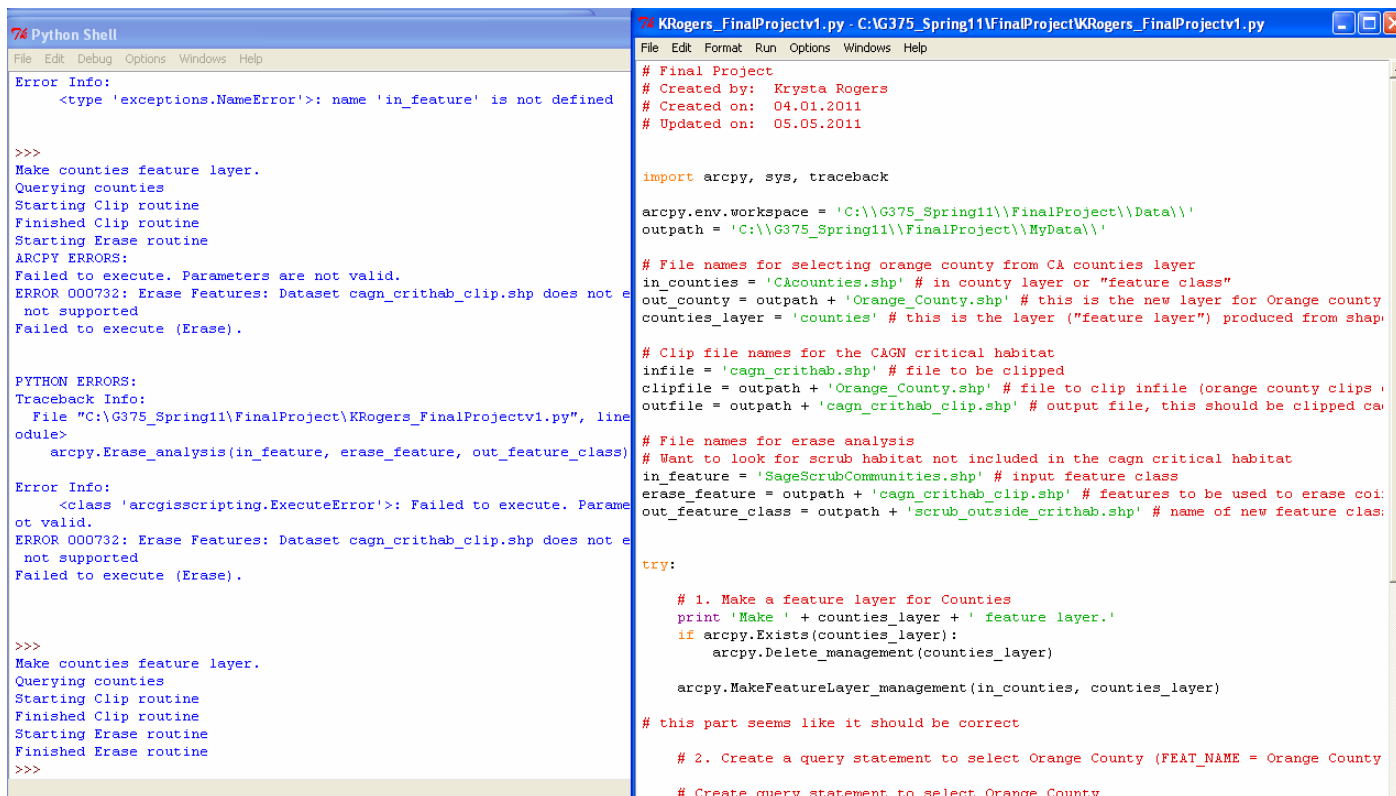
- Checked script for errors



Methods:

Python script development

- Fixed the errors & re-ran the code



```
Python Shell
File Edit Debug Options Windows Help

Error Info:
  <type 'exceptions.NameError': name 'in_feature' is not defined

>>>
Make counties feature layer.
Querying counties
Starting Clip routine
Finished Clip routine
Starting Erase routine
ARCPY ERRORS:
Failed to execute. Parameters are not valid.
ERROR 000732: Erase Features: Dataset cagn_crithab_clip.shp does not exist
not supported
Failed to execute (Erase).

PYTHON ERRORS:
Traceback Info:
  File "C:\G375_Spring11\FinalProject\KRogers_FinalProjectv1.py", line 10, in <module>
    arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)
  <class 'arcpy.ExecuteError': Failed to execute. Parameters are not valid.
ERROR 000732: Erase Features: Dataset cagn_crithab_clip.shp does not exist
not supported
Failed to execute (Erase).

>>>
Make counties feature layer.
Querying counties
Starting Clip routine
Finished Clip routine
Starting Erase routine
Finished Erase routine
>>>
```

```
KRogers_FinalProjectv1.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv1.py
File Edit Format Run Options Windows Help

# Final Project
# Created by: Krysta Rogers
# Created on: 04.01.2011
# Updated on: 05.05.2011

import arcpy, sys, traceback

arcpy.env.workspace = 'C:\\G375_Spring11\\FinalProject\\Data\\'
outpath = 'C:\\G375_Spring11\\FinalProject\\MyData\\'

# File names for selecting orange county from CA counties layer
in_counties = 'CAcounties.shp' # in county layer or "feature class"
out_county = outpath + 'Orange_County.shp' # this is the new layer for Orange county
counties_layer = 'counties' # this is the layer ("feature layer") produced from shap

# Clip file names for the C&G critical habitat
infile = 'cagn_crithab.shp' # file to be clipped
clipfile = outpath + 'Orange_County.shp' # file to clip infile (orange county clips)
outfile = outpath + 'cagn_crithab_clip.shp' # output file, this should be clipped ca

# File names for erase analysis
# Want to look for scrub habitat not included in the cagn critical habitat
in_feature = 'SageScrubCommunities.shp' # input feature class
erase_feature = outpath + 'cagn_crithab_clip.shp' # features to be used to erase coi
out_feature_class = outpath + 'scrub_outside_crithab.shp' # name of new feature clas

try:

    # 1. Make a feature layer for Counties
    print 'Make ' + counties_layer + ' feature layer.'
    if arcpy.Exists(counties_layer):
        arcpy.Delete_management(counties_layer)

    arcpy.MakeFeatureLayer_management(in_counties, counties_layer)

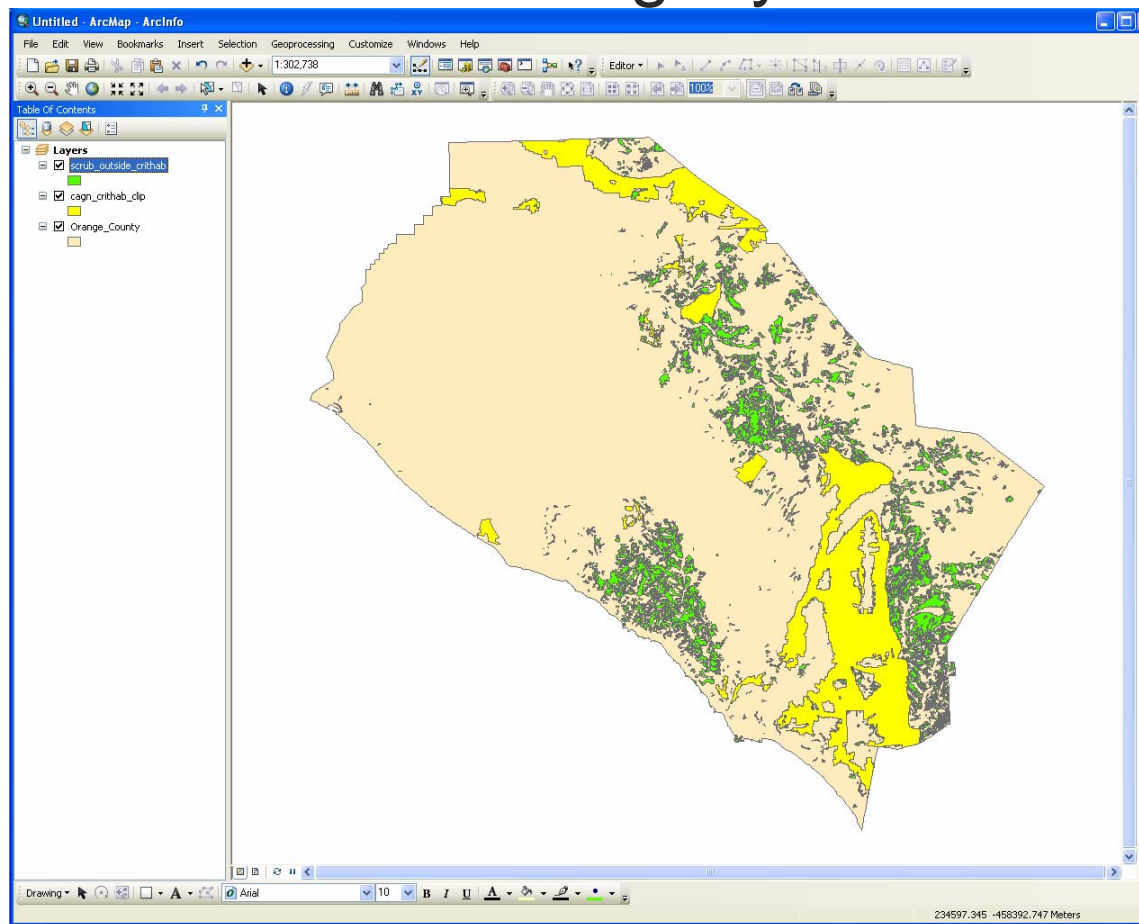
# this part seems like it should be correct

# 2. Create a query statement to select Orange County (FEAT_NAME = Orange County)
# Create query statement to select Orange County
```

Methods:

Preliminary results in ArcMap

- Looked at the resulting layers in ArcMap



Methods:

Python script development

- Add Join to add lookup table to vegetation layer

Add Join

```
KHogers_FinalProjectv2.py - C:\G3\5_Spring11\FinalProject\KHogers_FinalProjectv2.py
File Edit Format Run Options Windows Help

arcpy.Delete_management(out_feature_class)

print 'Starting Erase routine'

# Parameters using variables
arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

print 'Finished Erase routine'

# Join routine to join the veg lookup table to the new layer
# This will enable me to identify the vegetation types rather than just the codes
# Need to make a feature layer from the scrub_outside_crithab shapefile made from the erase routine

# Create a feature layer from the scrub_outside_crithab shapefile
print 'Make ' + outsideHab + ' feature layer.'
if arcpy.Exists(outsideHab):
    arcpy.Delete_management(outsideHab)

arcpy.MakeFeatureLayer_management(in_feature, outsideHab)

print 'Finished Make Feature Layer routine'

# Need to add index to table
arcpy.AddIndex_management(in_table, fields, index_name, "NON_UNIQUE", "ASCENDING")

# Perform join
print 'Starting Join'
arcpy.AddJoin_management(in_layer, in_field, join_table, join_field, join_type)

print 'Finished Join routine'

# To make join permanent need to save to layer file
print 'Starting Save to Layer'
arcpy.SaveToLayerFile_management(outsideHab, out_layer)

print 'Finished moving layer'

except:

    tb = sys.exc_info()[2]
    tbinfo = traceback.format_tb(tb)[0]
    pymsg = "PYTHON ERRORS:\nTraceback info:\n" + tbinfo + "\nError info:\n" + str(sys.exc_type) + ": " + str(sys.exc_value) + "\n"
    msg = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msg)
    arcpy.AddError(pymsg)

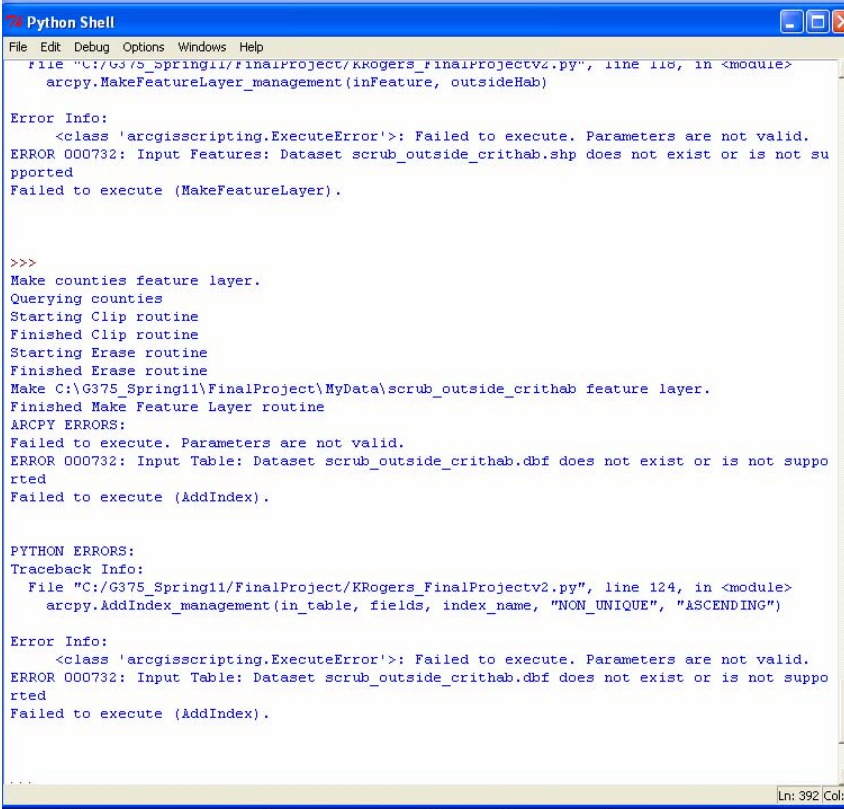
    print msg
    print pymsg

arcpy.AddMessage(arcpy.GetMessages(1))
print arcpy.GetMessages(1)
```

Methods:

Python script development

- Ran script again, error – need to add an index before performing the Add Join



```
Python Shell
File Edit Debug Options Windows Help
File "C:/G375_Spring11/FinalProject/KRogers_FinalProjectv2.py", line 118, in <module>
    arcpy.MakeFeatureLayer_management(inFeature, outsideHab)

Error Info:
<class 'arcgisscripting.ExecuteError'>: Failed to execute. Parameters are not valid.
ERROR 000732: Input Features: Dataset scrub_outside_crithab.shp does not exist or is not supported
Failed to execute (MakeFeatureLayer).

>>>
Make counties feature layer.
Querying counties
Starting Clip routine
Finished Clip routine
Starting Erase routine
Finished Erase routine
Make C:\G375_Spring11\FinalProject\MyData\scrub_outside_crithab feature layer.
Finished Make Feature Layer routine
ARCPY ERRORS:
Failed to execute. Parameters are not valid.
ERROR 000732: Input Table: Dataset scrub_outside_crithab.dbf does not exist or is not supported
Failed to execute (AddIndex).

PYTHON ERRORS:
Traceback Info:
File "C:/G375_Spring11/FinalProject/KRogers_FinalProjectv2.py", line 124, in <module>
    arcpy.AddIndex_management(in_table, fields, index_name, "NON_UNIQUE", "ASCENDING")

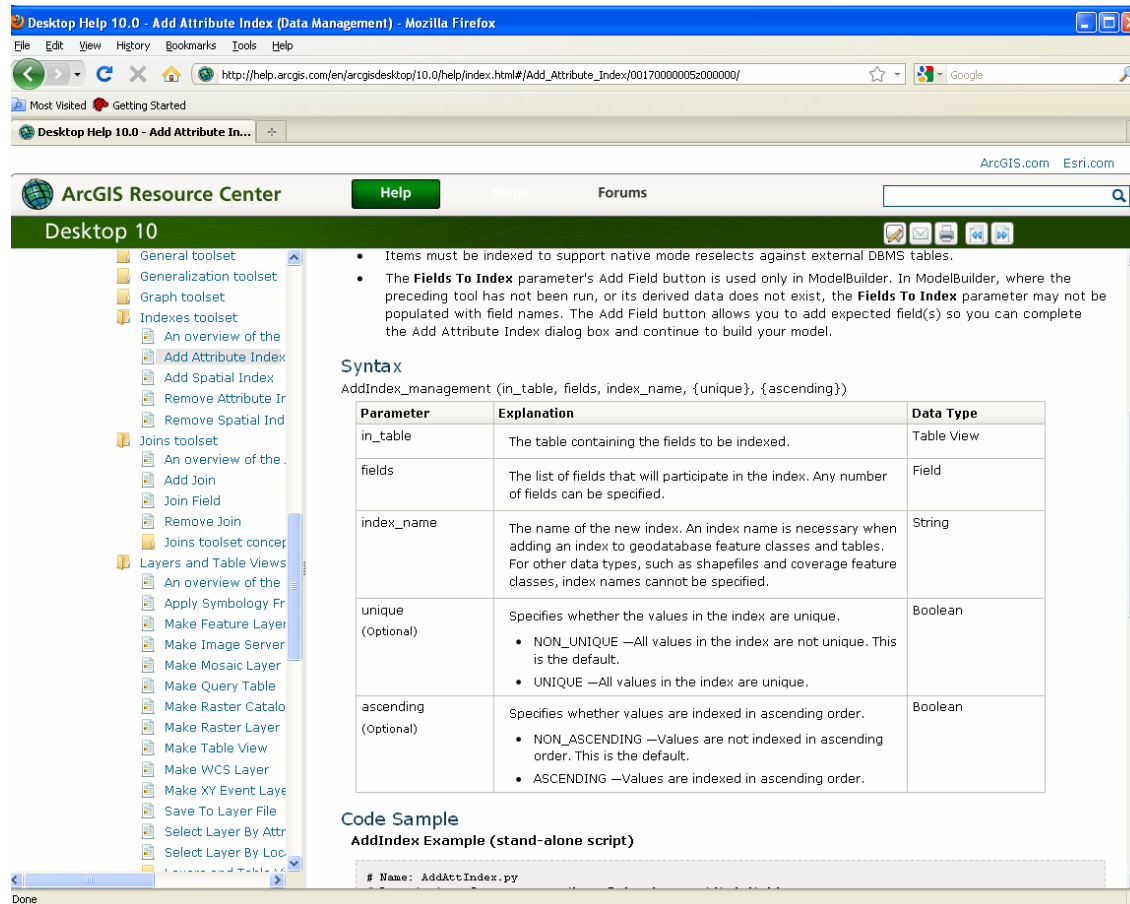
Error Info:
<class 'arcgisscripting.ExecuteError'>: Failed to execute. Parameters are not valid.
ERROR 000732: Input Table: Dataset scrub_outside_crithab.dbf does not exist or is not supported
Failed to execute (AddIndex).

Ln: 392/Col: 4
```

Methods:

Python script development

- Researched the Add Index routine



The screenshot shows the ArcGIS Desktop 10.0 Help window in Mozilla Firefox. The browser address bar shows the URL: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Add_Attribute_Index/00170000052000000/. The page title is "Desktop Help 10.0 - Add Attribute Index (Data Management)". The left sidebar shows the "Indexes toolset" expanded, with "Add Attribute Index" selected. The main content area displays the following information:

- Items must be indexed to support native mode reselects against external DBMS tables.
- The **Fields To Index** parameter's Add Field button is used only in ModelBuilder. In ModelBuilder, where the preceding tool has not been run, or its derived data does not exist, the **Fields To Index** parameter may not be populated with field names. The Add Field button allows you to add expected field(s) so you can complete the Add Attribute Index dialog box and continue to build your model.

Syntax
AddIndex_management (in_table, fields, index_name, {unique}, {ascending})

Parameter	Explanation	Data Type
in_table	The table containing the fields to be indexed.	Table View
fields	The list of fields that will participate in the index. Any number of fields can be specified.	Field
index_name	The name of the new index. An index name is necessary when adding an index to geodatabase feature classes and tables. For other data types, such as shapefiles and coverage feature classes, index names cannot be specified.	String
unique (Optional)	Specifies whether the values in the index are unique. <ul style="list-style-type: none">NON_UNIQUE —All values in the index are not unique. This is the default.UNIQUE —All values in the index are unique.	Boolean
ascending (Optional)	Specifies whether values are indexed in ascending order. <ul style="list-style-type: none">NON_ASCENDING —Values are not indexed in ascending order. This is the default.ASCENDING —Values are indexed in ascending order.	Boolean

Code Sample
AddIndex Example (stand-alone script)

```
# Name: AddAttIndex.py
```

Methods: Python script development

- Add Index, Add Join, & Save To Layer to make join permanent

Add Index,
Add Join,
& Save To Layer

```
KRogers_FinalProjectv2.py - C:/G375_Spring11/FinalProject/KRogers_FinalProjectv2.py
File Edit Format Run Options Windows Help
arcpy.Delete_management(out_feature_class)

print 'Starting Erase routine'

# Parameters using variables
arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

print 'Finished Erase routine'

# Join routine to join the veg lookup table to the new layer
# This will enable me to identify the vegetation types rather than just the codes
# Need to make a feature layer from the scrub_outside_crithab shapefile made from the erase routine

# Create a feature layer from the scrub_outside_crithab shapefile

print 'Make ' + outsideHab + ' feature layer.'
if arcpy.Exists(outsideHab):
    arcpy.Delete_management(outsideHab)

arcpy.MakeFeatureLayer_management(inFeature, outsideHab)

print 'Finished Make Feature Layer routine'

# Need to add index to table

arcpy.AddIndex_management(in_table, fields, index_name, "NON_UNIQUE", "ASCENDING")

print 'Starting Join'

arcpy.AddJoin_management(in_layer, in_field, join_table, join_field, join_type)

print 'Finished Join routine'

# To make join permanent need to save to layer file

print 'Starting Save to Layer'

arcpy.SaveToLayerFile_management(outsideHab, out_layer)

print 'Finished saving layer'

except:

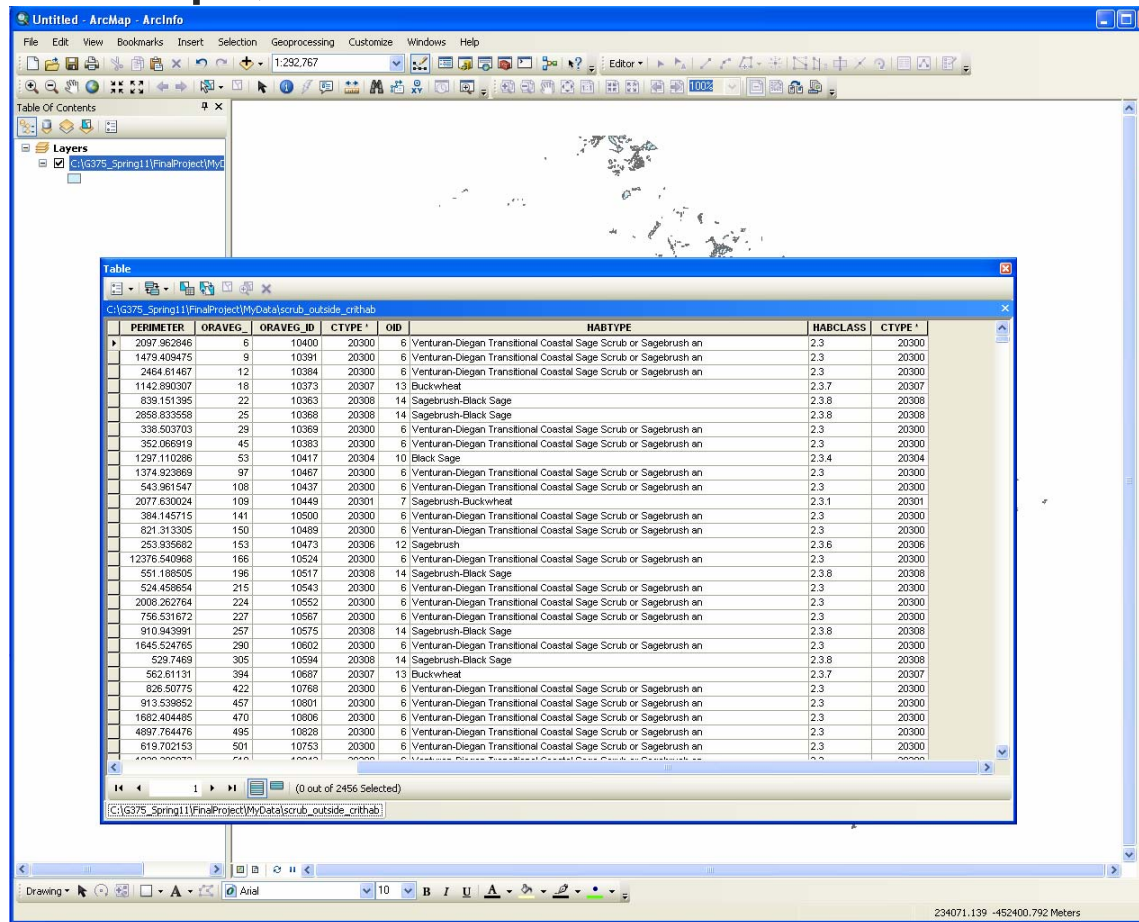
    tb = sys.exc_info()[2]
    tbinfo = traceback.format_tb(tb)[0]
    pymsg = "PYTHON ERRORS:\nTraceback Info:\n" + tbinfo + "\nError Info:\n" + str(sys.exc_type) + ": " + str(sys.exc_value) + "\n"
    msg = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msgs)
    arcpy.AddError(msgs)
```


Methods:

Python script development

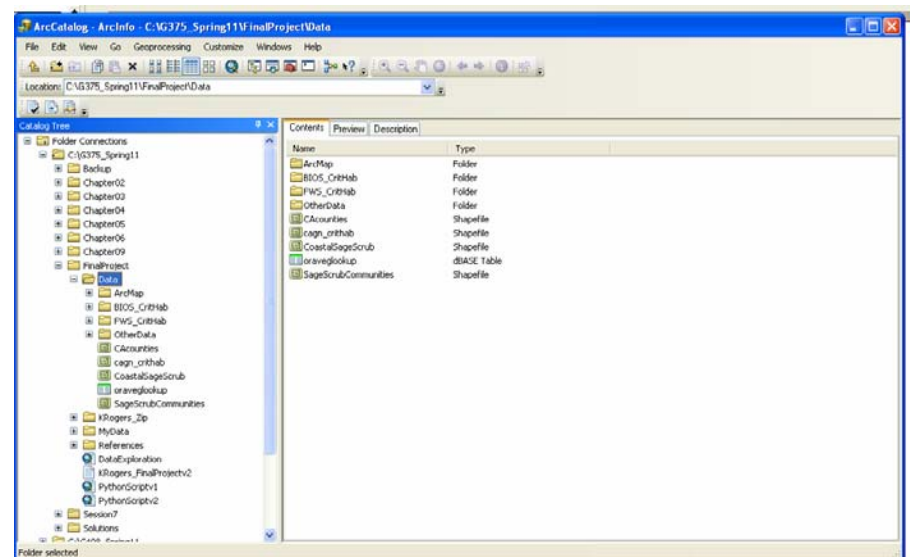
- Ran script, no errors!



Results:

ArcMap

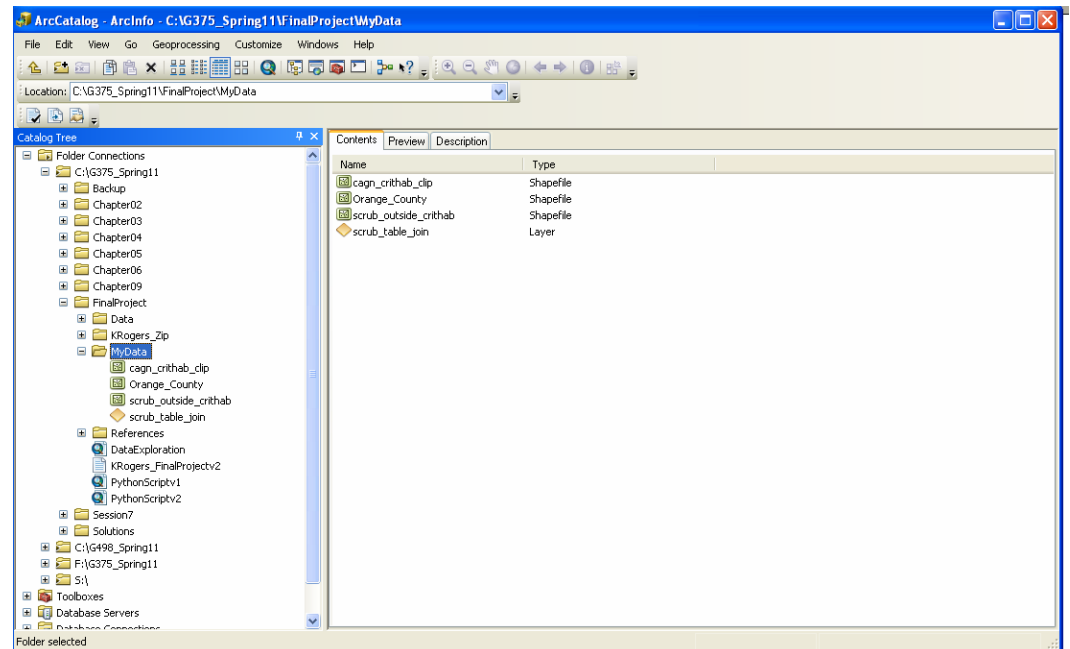
- Files produced in ArcMap for use in Python
 - CACounties
 - cagn_crithab
 - CoastalSageScrub
 - SageScrubCommunities



Results:

Python

- Files produced in Python
 - cagn_crithab_clip.shp
 - Orange_County.shp
 - scrub_outside_crithab.shp
 - scrub_table_join



Discussion:

Lessons learned

- Tackled this project with a step-by-step approach
- Parceled out input/output file names for each process
- Utilized help files & class materials for reference
- Look for syntax errors, misspellings, & incorrect path names

Discussion:

Critical habitat

- Coastal sage scrub habitat does exist outside the designated critical habitat for the California gnatcatcher
- Warrants monitoring to determine if habitat might be utilized by gnatcatchers in the future



Peter Gallagher