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

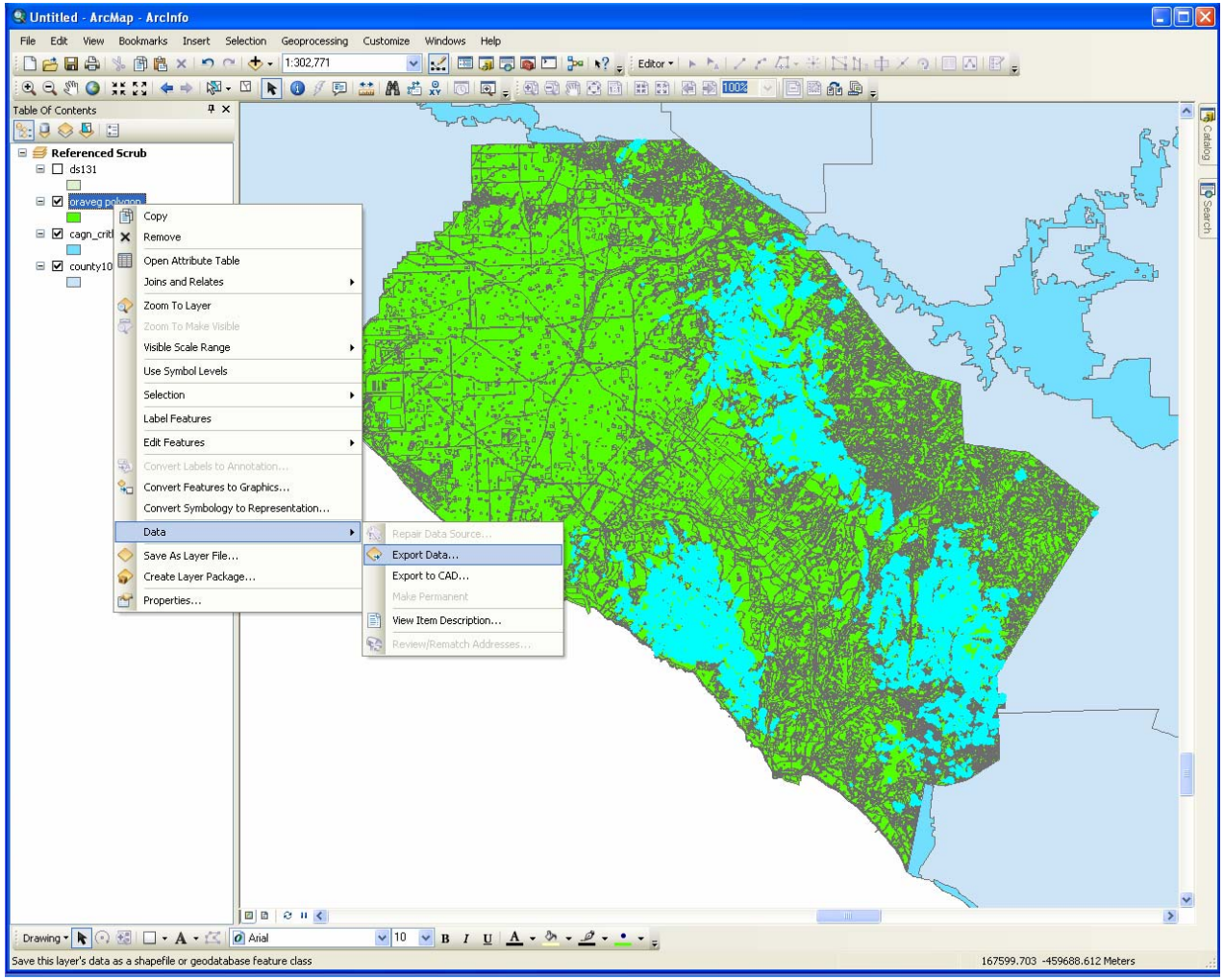
Methods:
Data preparation

- 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

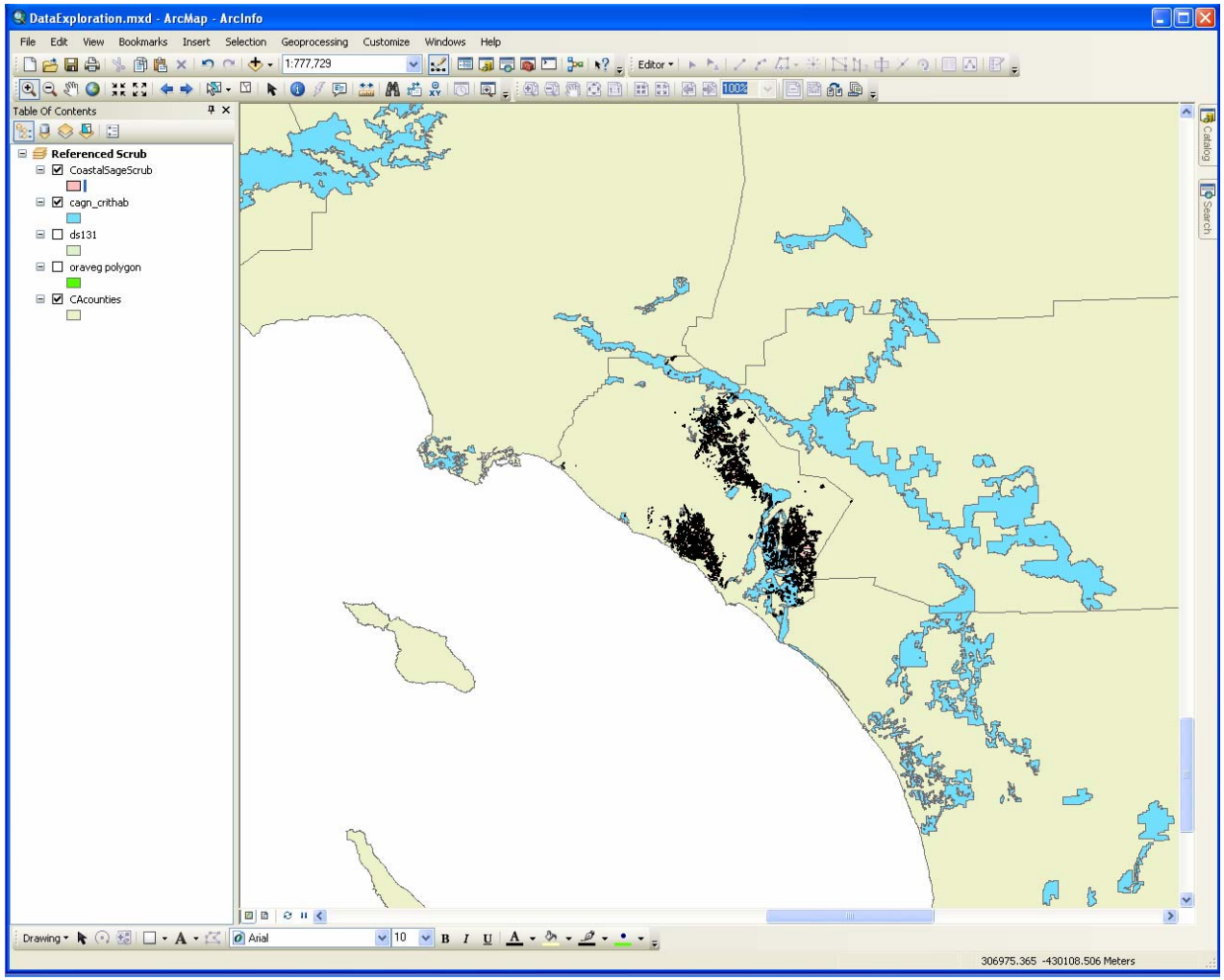
Methods:
Data preparation

- 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

try:

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 or 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)
```

Ln: 1 Col: 0

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"
    msgs = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msgs)
    arcpy.AddError(pymsg)

    print msgs
    print pymsg

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

Ln: 48 Col: 0

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)
```

Ln: 1 Col: 0

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

```
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'
```

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

```
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'
```

Ln: 87 Col: 0

Methods:
Python script development

- Erase analysis to evaluate habitat outside of critical habitat

Erase

KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py

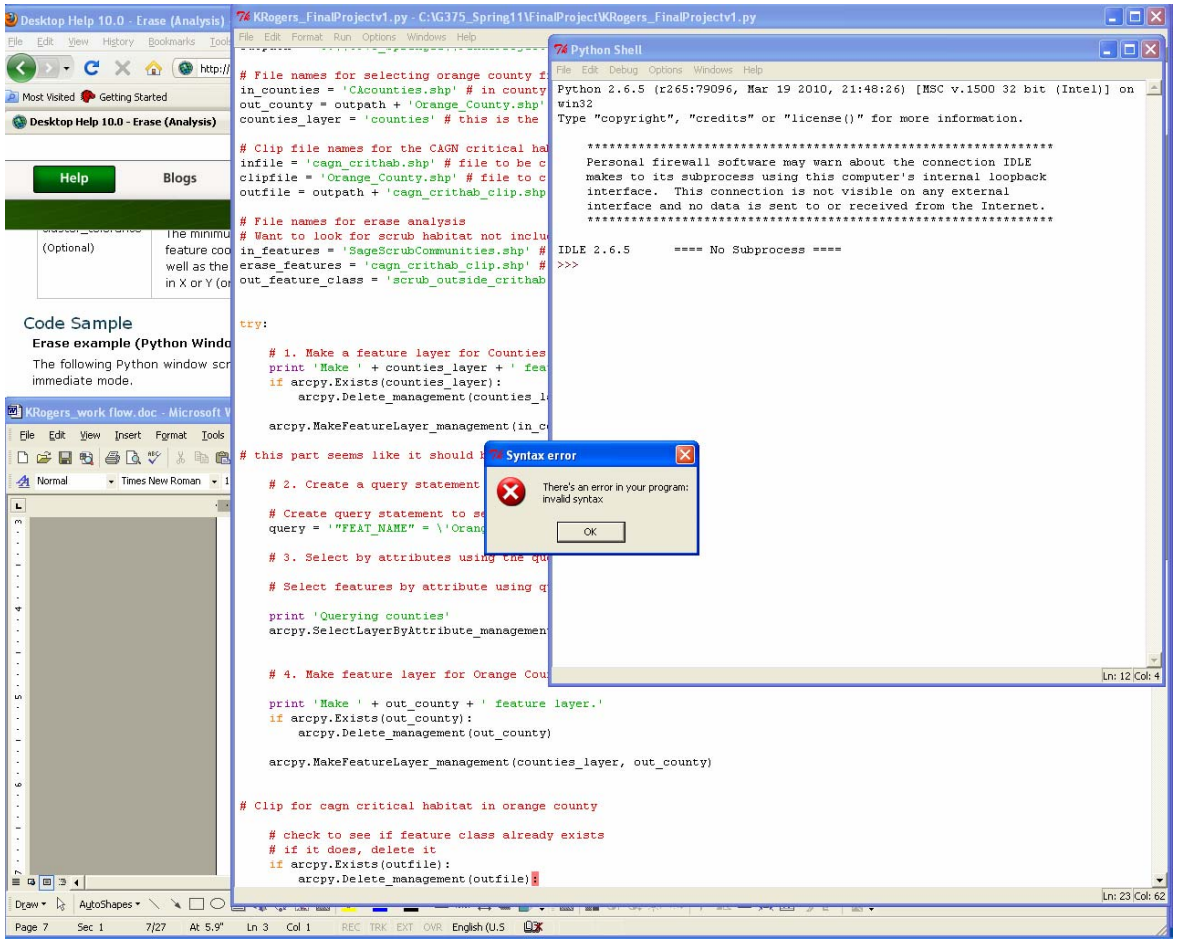
File Edit Format Run Options Windows Help

```
    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'

# 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
```

Ln: 100 Col: 0

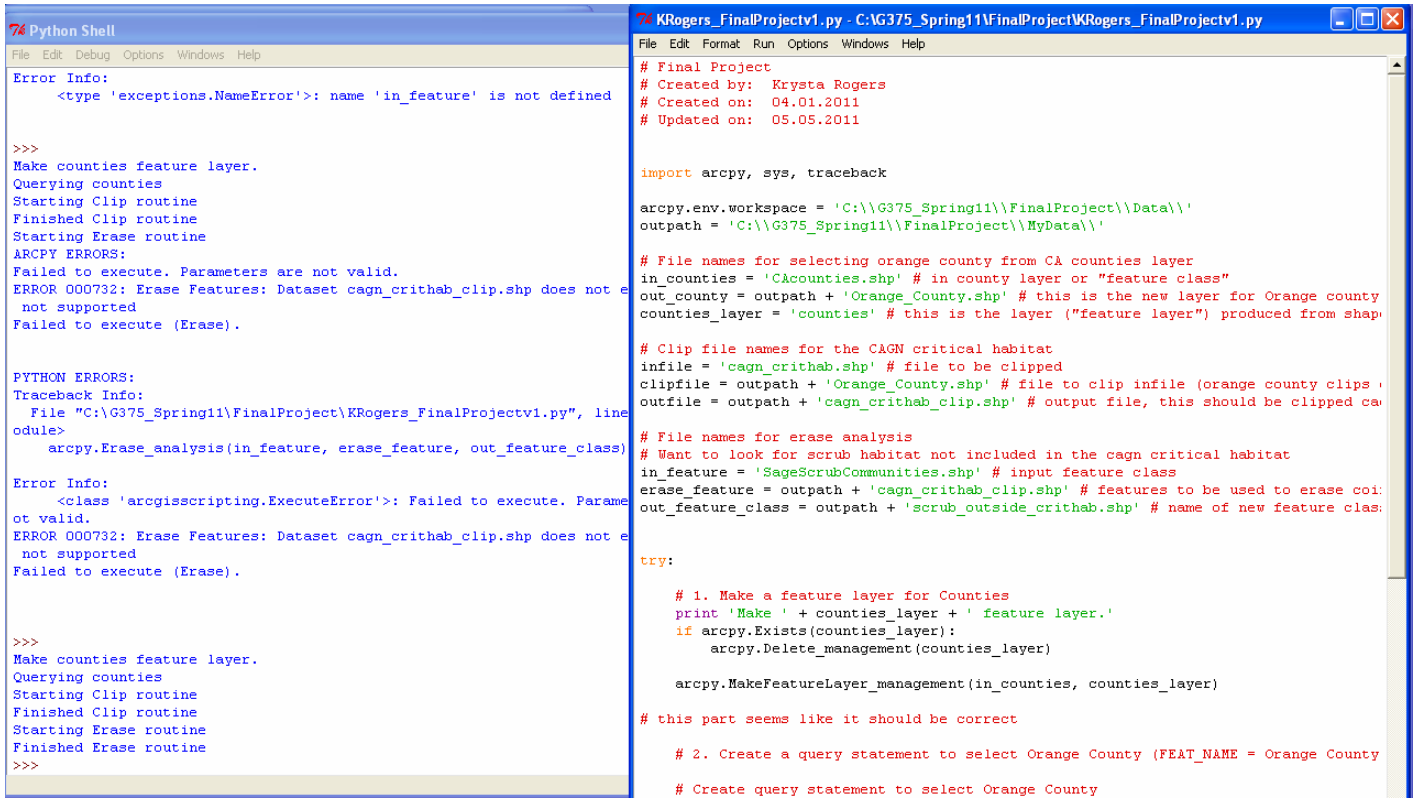
Methods:
Python script development

- 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 e
 not supported
Failed to execute (Erase).


PYTHON ERRORS:
Traceback Info:
  File "C:\G375_Spring11\FinalProject\KRogers_FinalProjectv1.py", line
odule>
    arcpy.Erase_analysis(in_feature, erase_feature, out_feature_class)

Error Info:
     <class 'arcgisscripting.ExecuteError'>: Failed to execute. Parame
ot valid.
ERROR 000732: Erase Features: Dataset cagn_crithab_clip.shp does not e
 not supported
Failed to execute (Erase).



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

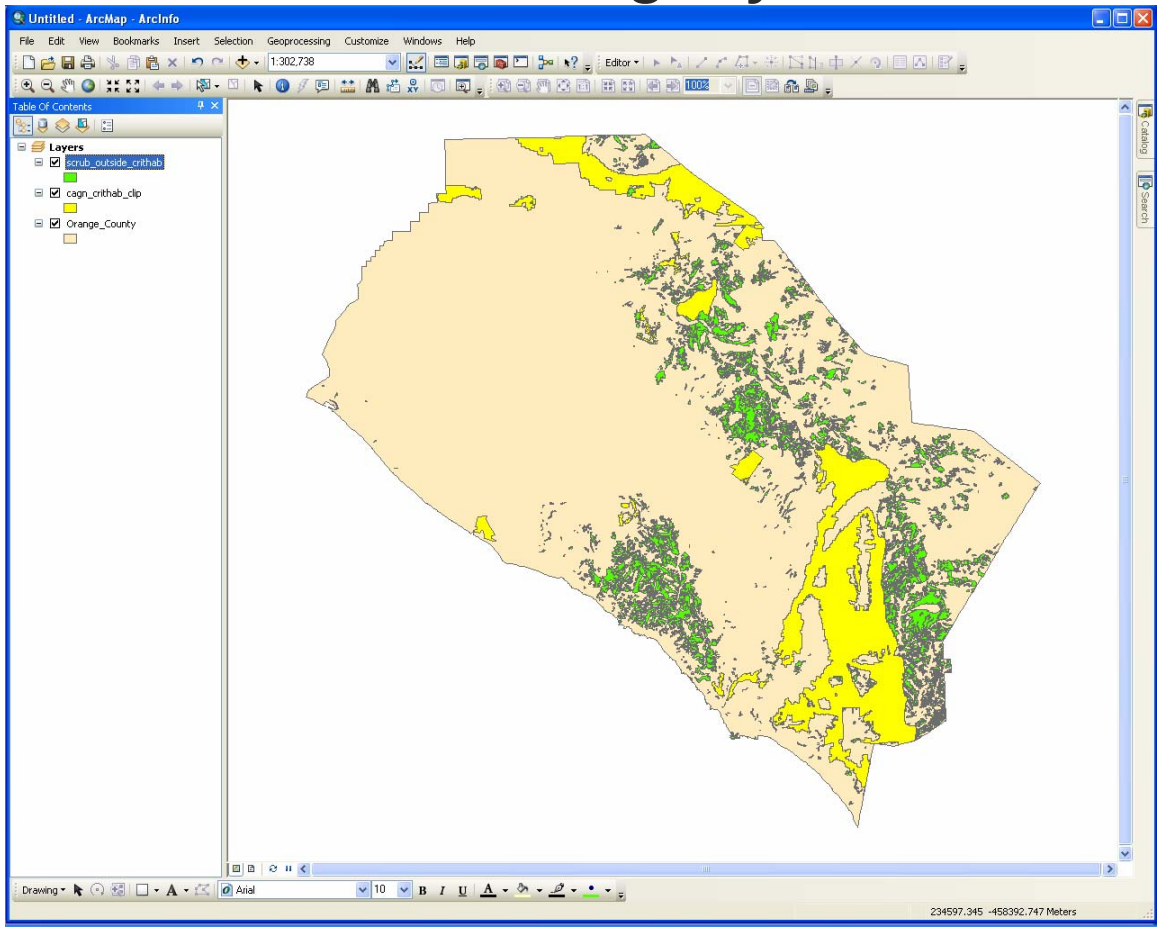
KRogers_FinalProjectv1.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv1.py

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 CAGN 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

KRogers_FinalProjectv2.py - C:\G375_Spring11\FinalProject\KRogers_FinalProjectv2.py

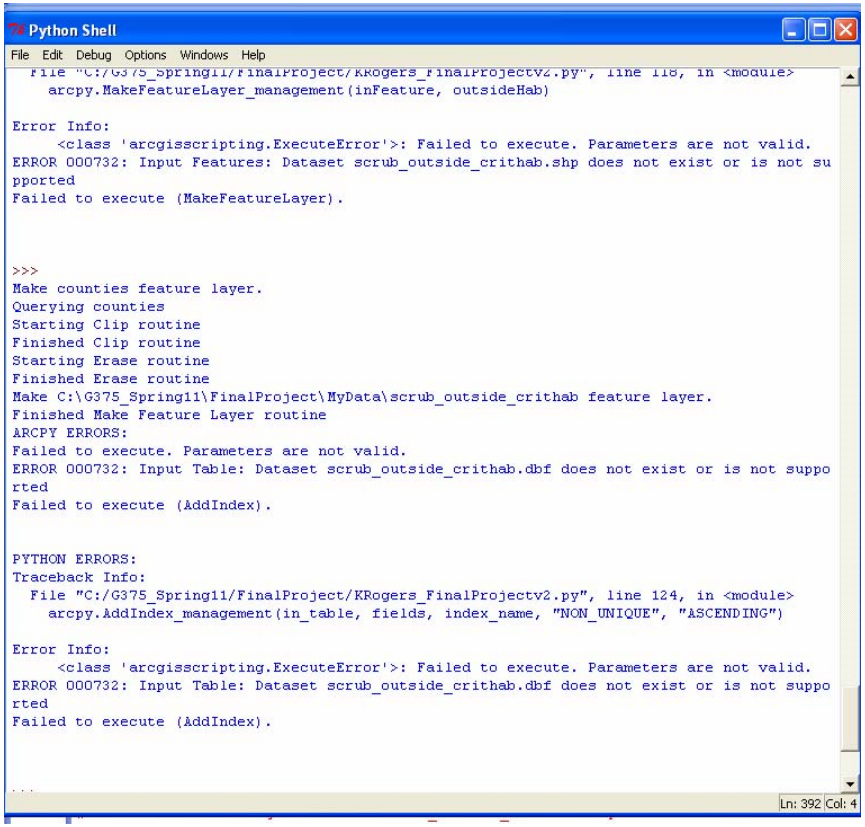
File Edit Format Run Options Windows Help

```
        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")

# 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"
    msgs = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msgs)
    arcpy.AddError(pymsg)

    print msgs
    print pymsg

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

Ln: 124 Col: 0

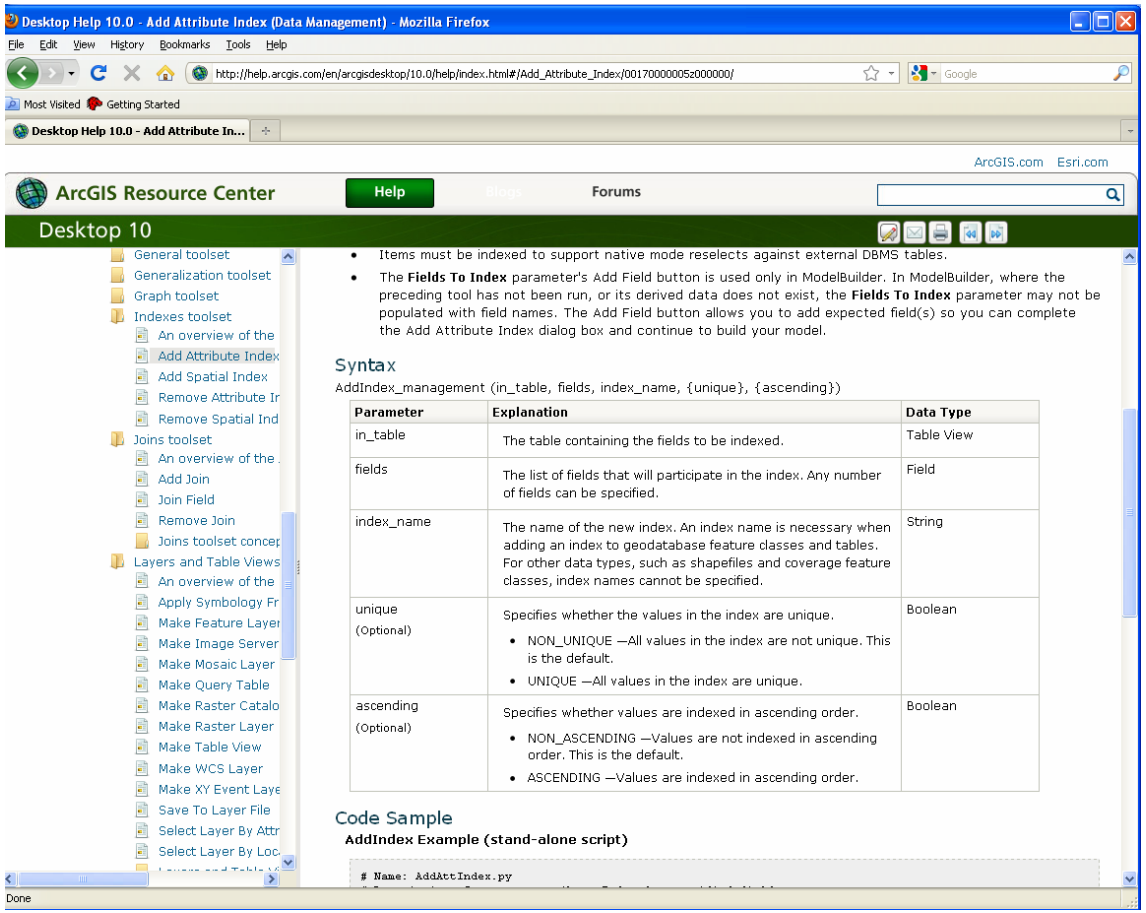
Methods:
Python script development

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



Methods:
Python script development

- Researched the Add Index routine



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

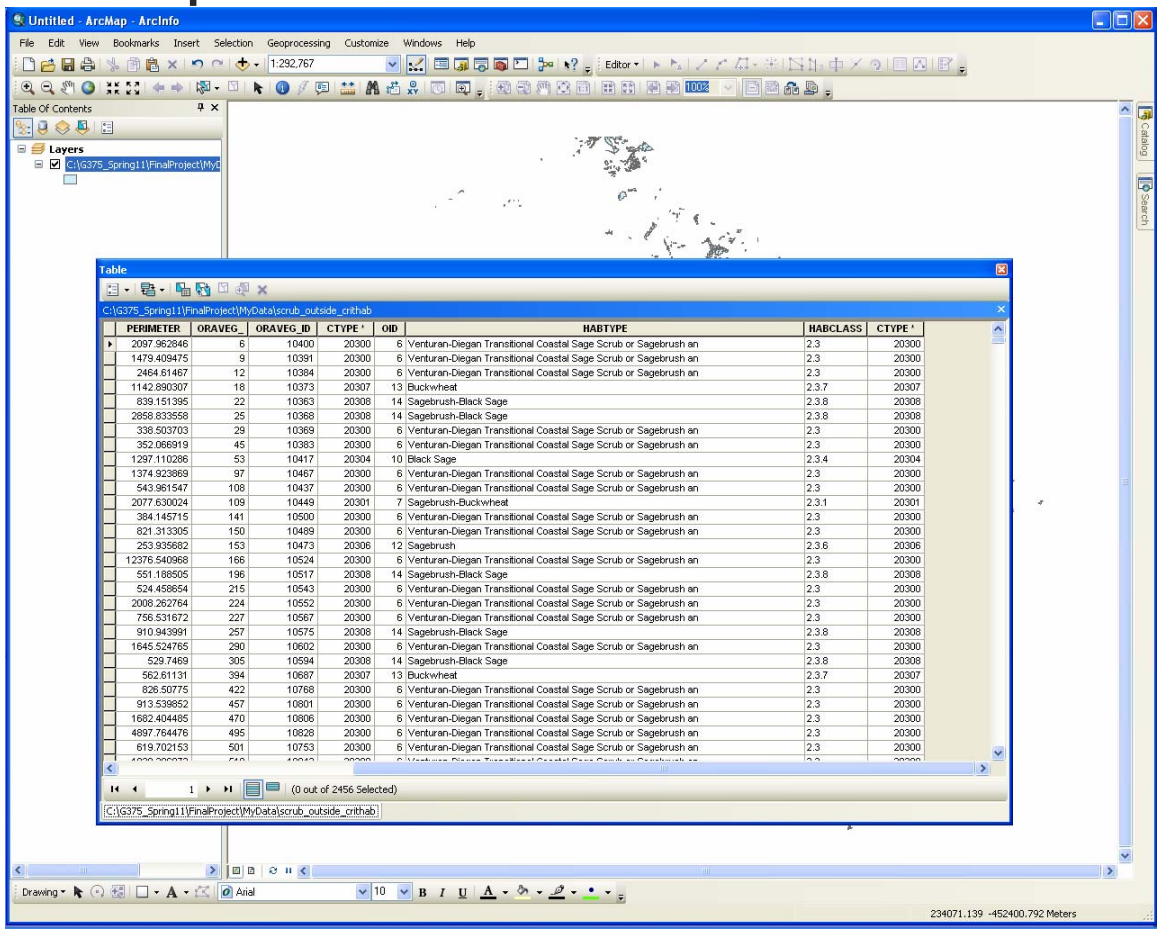
File Edit Format Run Options Windows Help

```
    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"
    msgs = "ARCPY ERRORS:\n" + arcpy.GetMessages(2) + "\n"

    arcpy.AddError(msgs)
```

Ln: 124 Col: 86

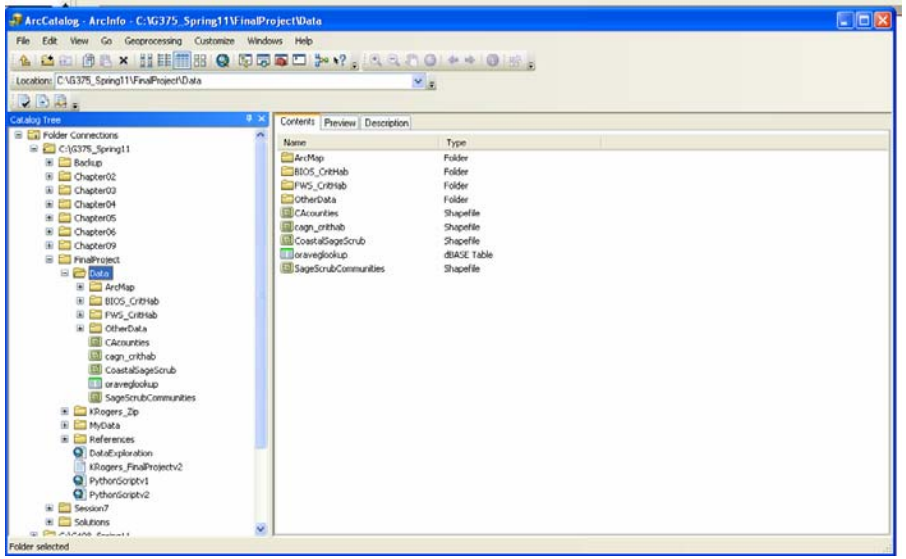
Methods:
Python script development

- Ran script, no errors!



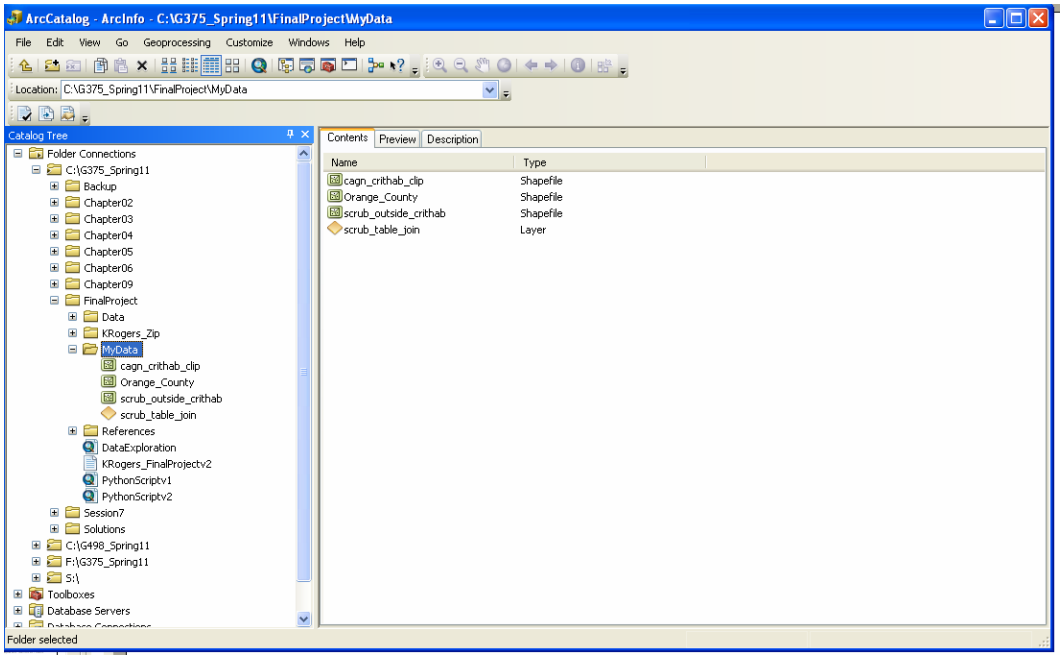
Results:
ArcMap

- 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