Map Series:
Zooplankton densities in the Sacramento River over time

Seth Strumwasser

GEOG 375

Background

- Sacramento Valley used to be an enormous **floodplain**
- Before levees, Sacramento River would flood during the winter and create huge floodplain areas and wetlands
- Very diverse and productive ecosystems
- Vital for **salmon** rearing
  - Young salmon thrive on a diet of **zooplankton** (small crustaceans and other arthropods)
- Now: floodplain converted to rice fields
- Winter discharge to river still contains abundant zooplankton

Chinook salmon

Daphnia pulex

Purpose

- Create a series of maps showing zooplankton density at various sites across multiple dates
- Create script which can be easily adapted to future data

Script overview

- Inputs
  - Csv's: a csv file for each date with zooplankton data (name format MMDDYY)
  - Shapefile of sample collection sites
  - ArcGIS Pro project
    - One pre-made map for each date (name format MMDDYY)
    - Pre-made layout for creating exports
  - Pre-made symbology layers
  - Miscellaneous background data (river, pumphouse, etc.)
- Outputs
  - 6 maps in PDF and PNG format

Tasks

- For each map…
  - Find correct csv file
  - Join csv to sites layer
  - Add layer with join to the map
  - Update symbology and layout
  - Export PDF and PNG

Script Outline

1) Imports
2) Variables
3) Try block
    1) For loop iterating over maps
        1) For loop iterating over csv's
            1) If statement for matching csv
                1) Join correct csv to sites layer
                2) Save as shapefile to make join permanent
        2) Add sites feature class with join fields
        3) Edit symbology
        4) Edit layout
        5) Export map
    2) Save project for review
4) Except block

Imports

```
print("Importing modules...")
import arcpy, sys, traceback, datetime
from os import listdir
from os.path import isfile, join
print("Modules imported.")
```

Variables

```
# variables
fd_folder = "D:\\seths_msi\\Documents\\CalTrout Mapping Projects\\Food_Density\\"
out_layer_folder = fd_folder + "output_layers\\"
out_layer = out_layer_folder + "RRSAC_sites_"
out_fc_folder = fd_folder + "output fcs\\"
# project and input layer
aprx = arcpy.mp.ArcGISProject(fd_folder + "Food_Density.aprx")
sites_layer_name = fd_folder + "RRSAC_sites.lyrx"
# csv variables
csv_loc = "D:\\seths_msi\\Documents\\CalTrout Mapping Projects\\Food_Density\\csv_s\\"
csv_list = [f for f in listdir(csv_loc) if isfile(join(csv_loc, f))]
csv_date = ''
# list of maps in project
maps = aprx.listMaps()
# layout variables
layout = aprx.listLayouts('Layout')[0]
map_frame = layout.listElements('MAPFRAME_ELEMENT')[0]
extent = map_frame.camera.getExtent()
# symbology layers
sites_symbo_lyr = fd_folder + "RRSAC_sites_symbo.lyrx"
sites_symbo_lyr_prop = fd_folder + "RRSAC_sites_symbo_proportion.lyrx"
symbo_folder = fd_folder + "symbology_layers\\"
# dictionary with discharge data
discharge_dict = {
    "022619": "7,013",
    "030619": "9,803",
    "031019": "7,217",
    "032019": "2,655",
    "032719": "2,821",
    "040319": "2,479",
    "040819": "1,787"
}
```

```
out_pdf = fd_folder + "Maps\\food_density_"
out_png = fd_folder + "Maps\\food_density_"
```

Remove extra layer

```
# remove 'Testing' map from map list
for m in maps:
    if m.name == 'Testing':
        maps.remove(m)
        print("Testing map removed.")
```

Maps loop

```
try:
    # iterate maps
    for m in maps:
        print("...")

        # set map name variable
        map_name = m.name
        print("Updating map: " + map_name)
        print("...")

        # change basemap
        print("Updating basemap...")
        m.addBasemap("Imagery")

        # add pump house layer
        RR_file = arcpy.mp.LayerFile(fd_folder + "Rough_and_Ready_symbol.lyrx")
        m.addLayer(RR_file)

        # iterate through csv files and find the one that matches the map date
        print("Joining csvs...")
        for csv in csv_list:
```

Csv loop

```
for csv in csv_list:
    # create csv_date variable from file name
    csv_date = csv[0:6]
    # match the current map to the correct date
    if csv_date == map_name:
        # make feature layer from sites layer
        layer_fl = "RRSAC_sites_" + csv_date + "_fl"
        if arcpy.Exists(layer_fl):
            arcpy.Delete_management(layer_fl)
        arcpy.MakeFeatureLayer_management(sites_layer_name, layer_fl)

        # join csv to the sites layer
        print("Joining " + csv)
        arcpy.AddJoin_management(layer_fl, 'Name', csv_loc + csv, 'Site')
        print("Join success.")

        # save to feature class
        print("Saving to feature class...")
        if arcpy.Exists(out_fc_folder + "RRSAC_sites_joined_" + csv_date + ".shp"):
            arcpy.Delete_management(out_fc_folder + "RRSAC_sites_joined_" + csv_date + ".shp")
        arcpy.CopyFeatures_management(layer_fl, out_fc_folder + "RRSAC_sites_joined_" + csv_date + ".shp")
        print("Done.")

        # save generated features to output layer
        if arcpy.Exists(out_layer + csv_date + ".lyrx"):
            arcpy.Delete_management(out_layer + csv_date + ".lyrx")
        arcpy.SaveToLayerFile_management(layer_fl, out_layer + csv_date + ".lyrx")
        break
```

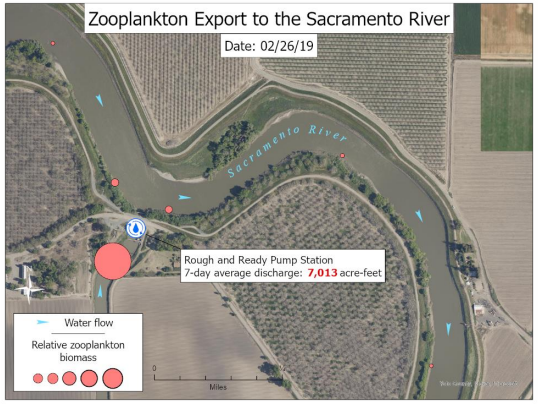
Add feature class and edit symbology

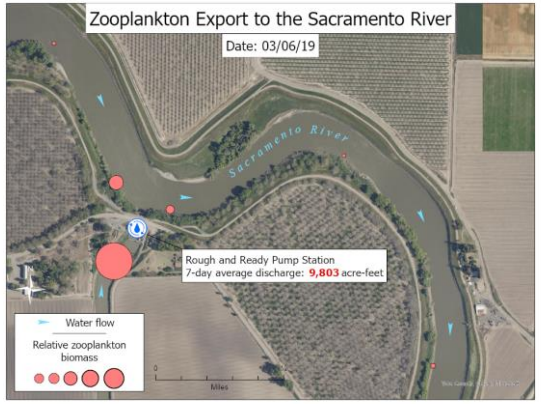
```
# add output FC to map
m.addDataFromPath(out_fc_folder + "RRSAC_sites_joined_" + csv_date + ".shp")

# apply correct symbology
print("Applying symbology...")

# get a list of layers so we can reference the complete layer
layers = m.listLayers()
for l in layers:
    print(l.name)
complete_layer = layers[0]

# apply symbology from symbology layer
# symbology_fields = [["VALUE_FIELD", "040319__11", csv_date + "__11"]]
symbo_layer = symbo_folder + "RRSAC_sites_" + csv_date + "_symbo.lyrx"
arcpy.ApplySymbologyFromLayer_management(complete_layer, symbo_layer)
```

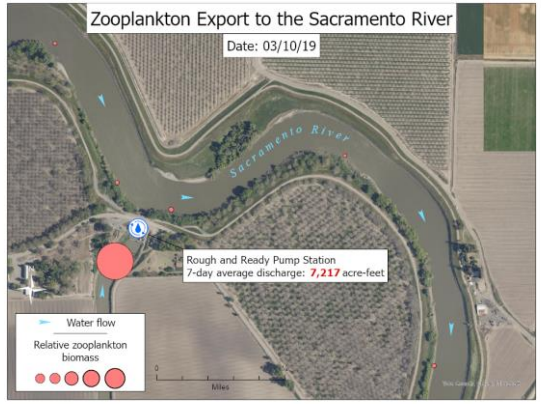
Modify layout and export

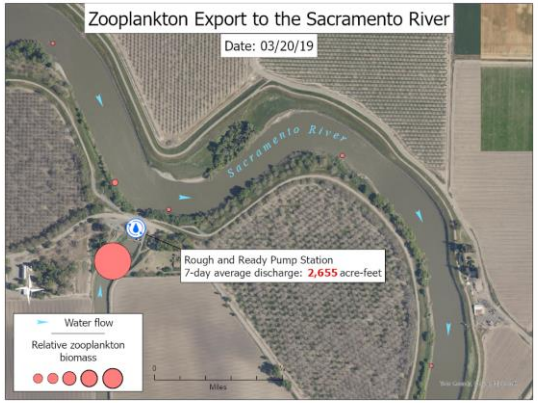
```
# modify layout (keep this inside of map iteration)
#
# assigning correct map to map frame
print("Assigning correct map to map frame...")
map_frame.map = m

# update map elements
print("Updating map elements...")
elements = layout.listElements()
for e in elements:
    if e.name == 'title':
        e.text = 'Date: ' + map_name[0:2] + "/" + map_name[2:4] + "/" + map_name[4:6]
    if e.name == 'discharge':
        e.text = str(discharge_dict[csv_date])

# set map frame extent
print("Setting map frame extent...")
map_frame.camera.setExtent(extent)

# export map (PDF)
print("Exporting to PDF...")
if arcpy.Exists(out_pdf + map_name + ".pdf"):
    arcpy.Delete_management(out_pdf + map_name + ".pdf")
layout.exportToPDF(out_pdf + map_name + ".pdf")

# export map (PNG)
if arcpy.Exists(out_png + map_name + ".png"):
    arcpy.Delete_management(out_png + map_name + ".png")
layout.exportToPNG(out_png + map_name + ".png")
```

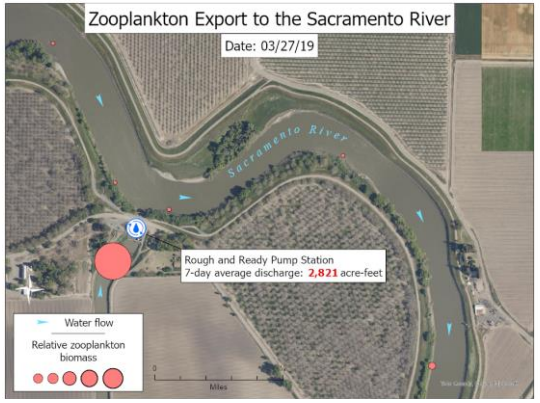
Output

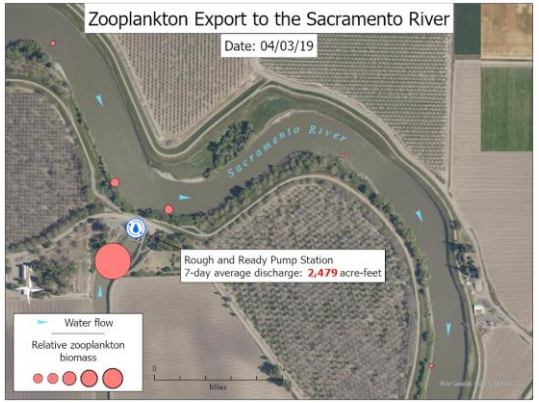

Zooplankton Export to the Sacramento River
Date: 02/26/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 7,013 acre-feet
Water flow
Relative zooplankton biomass
0
Miles

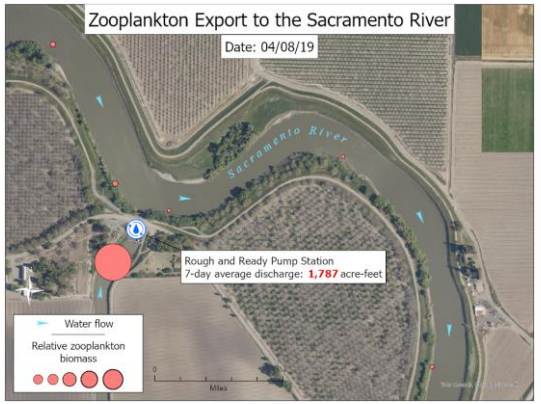

Zooplankton Export to the Sacramento River
Date: 03/06/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 9,803 acre-feet
Water flow
Relative zooplankton biomass
0
Miles

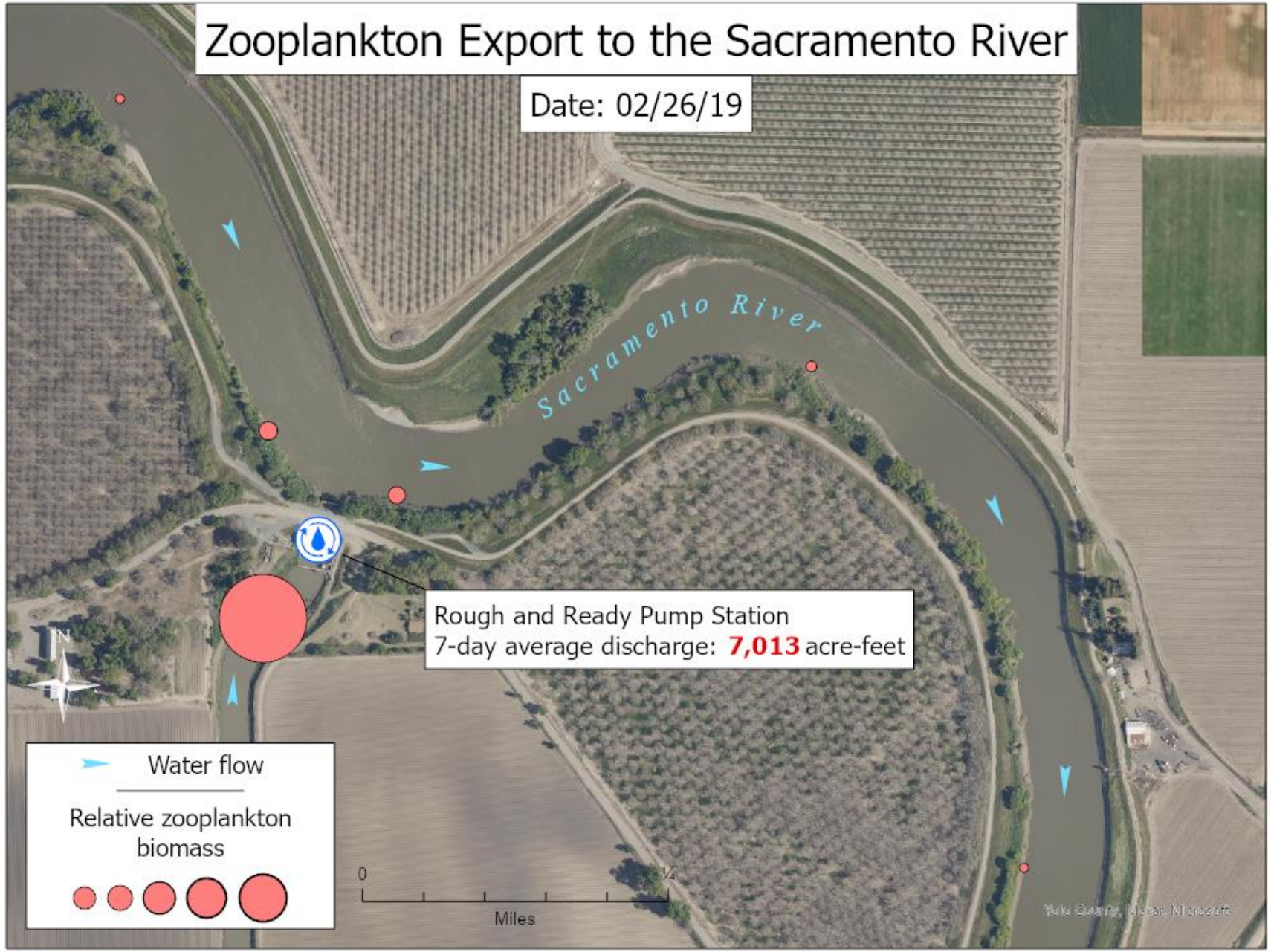

Zooplankton Export to the Sacramento River
Date: 03/10/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 7,217 acre-feet
Water flow
Relative zooplankton biomass
0
Miles


Zooplankton Export to the Sacramento River
Date: 03/20/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 2,655 acre-feet
Water flow
Relative zooplankton biomass
0
Miles


Zooplankton Export to the Sacramento River
Date: 03/27/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 2,821 acre-feet
Water flow
Relative zooplankton biomass
0
Miles


Zooplankton Export to the Sacramento River
Date: 04/03/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 2,479 acre-feet
Water flow
Relative zooplankton biomass
0
Miles


Zooplankton Export to the Sacramento River
Date: 04/08/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 1,787 acre-feet
Water flow
Relative zooplankton biomass
0
Miles


Zooplankton Export to the Sacramento River
Date: 02/26/19
Sacramento River
Rough and Ready Pump Station
7-day average discharge: 7,013 acre-feet
N
Water flow
Relative zooplankton biomass
0
¼
Miles

Issues

- Symbology
  - No routines to edit the symbology directly (proportional symbols)
  - Used ApplySymbologyFromLayer_management routine
  - Very finicky – input layer and symbology layer must match in format and fields
  - Had to use the output feature classes to create symbology layers so that they were identical
  - Need unique symbology layer for each date because of data ranges
  - Initially wanted to do pie charts showing breakdown of types of zooplankton – difficult to make sure field names match
- Using correct objects – feature layers, layers, layer files
  - Created feature layer of sampling sites, saved to feature class, added back using addDataFromPath (creates feature layer)

Questions?