GEOG132_nb1_watershed_loading

February 2, 2022

1 Geography 132 – Assignment #1

Due Feb. 2, 2022, 9am >### Instructions > > Your assignment is to conduct a simple analysis of loading from the Goleta Slough watershed to the nearby coastal ocean. > > #### The Situation > > You work for the California Department of Fish & Game, which is responsible for managing the https://wildlife.ca.gov/Lands/Places-to-Visit/Goleta-Slough-ER. DFG is planning to conduct an assessment of contaminated sediments in the Slough in order to understand whether these sediments pose a risk to recreational fishing that occurs at the Goleta Pier. You have been asked by your supervisor to make a preliminary estimate of loading rates from the Slough to the surrounding coastal ocean. Such an exercise is useful for a number of reasons, including determining the necessary sensitivity level of analytic tests in advance and providing a general check for results. > >For your assignment, you will be asked to produce a Jupyter Notebook that incorporates the following elements. >1. A short (two or three paragraph) introduction to chemical contaminants and their connection to sediments. This summary should synthesize material from lecture as well as the reading material from GauchoSpace. >2. A map of the study location. >3. A (hypothetical) estimate of contaminant loading rates from the slough to the coastal ocean. You will be asked to make this estimate two ways. First, you must create a table showing the unit conversions necessary to obtain your answer. Second, you must calculate your answer using Python by defining variables for each of the terms in your calculation and then computing and displaying the answer.

1.1 1. Chemical Contaminants Background

> (30%) Synthesizing the material from lecture and the reading assigned in Segar (see GauchoSpace), describe in general the problems associated with chemical contaminants. This synthesis need not be exhaustive, but you should cover the general classes of contaminants and how/why they are considered contaminants. You will be graded on the quality and effort of your descriptions

- Probelms with chemical contaminants is that many of them are man made and spread to people and make them ill and spread to waters contaminating water supply and ecosystems. DDT is known as Dichlorodiphenyltrichloroethane which was man made and used as a agricultural pesticide which had chemical effects on liver, kidneys, and nervous system. Another example is the amount of mercury in the San Francisco Bay which impacts the marine environment and this was caused by gold minning during the gold rush. Lastly PCB's were also man made and known as Polychlorinated Biphenyls which cause cancer and is a organic chemical, all the chemicals I talked about have very harmful effects on the human body and many get sick from eating seafood. Coastal areas are most at risk when coming in contact with chemical contaminants.

1.2 2. A map of the study location

(35%) Make a plot that includes * the Goleta Bay shoreline * a shaded DEM showing the watershed that drains into Goleta Bay; be sure to include a labeled colorbar * a symbol denoting the location of the Goleta Pier

You should show all code necessary to produce this plot. You do not need to submit the data files used to generate the plot

[1]:
```
import sys
!{sys.executable} -m pip install pysheds
```

```
Collecting pysheds
  Using cached pysheds-0.3.2-py3-none-any.whl (87 kB)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from pysheds) (1.21.5)
Collecting numba
  Using cached
numba-0.55.1-1-cp37-cp37m-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (3.3
MB)
Requirement already satisfied: pandas in /opt/conda/lib/python3.7/site-packages
(from pysheds) (1.3.5)
Collecting scikit-image
  Using cached
scikit_image-0.19.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
(13.3 MB)
Collecting affine
  Using cached affine-2.3.0-py2.py3-none-any.whl (15 kB)
Collecting rasterio>=1
  Using cached rasterio-1.2.10-cp37-cp37m-manylinux1_x86_64.whl (19.3 MB)
Collecting geojson
  Using cached geojson-2.5.0-py2.py3-none-any.whl (14 kB)
Collecting scipy
  Using cached
scipy-1.7.3-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (38.1 MB)
Collecting pyproj
  Using cached pyproj-3.2.1-cp37-cp37m-manylinux2010_x86_64.whl (6.3 MB)
Collecting snuggs>=1.4.1
  Using cached snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Requirement already satisfied: attrs in /opt/conda/lib/python3.7/site-packages
(from rasterio>=1->pysheds) (21.4.0)
Requirement already satisfied: click>=4.0 in /opt/conda/lib/python3.7/site-
packages (from rasterio>=1->pysheds) (8.0.3)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-
packages (from rasterio>=1->pysheds) (60.5.0)
```

```
Requirement already satisfied: certifi in /opt/conda/lib/python3.7/site-packages
(from rasterio>=1->pysheds) (2021.10.8)
Collecting cligj>=0.5
  Using cached cligj-0.7.2-py3-none-any.whl (7.1 kB)
Collecting click-plugins
  Using cached click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Collecting llvmlite<0.39,>=0.38.0rc1
  Using cached
llvmlite-0.38.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.5
MB)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-
packages (from pandas->pysheds) (2021.3)
Requirement already satisfied: python-dateutil>=2.7.3 in
/opt/conda/lib/python3.7/site-packages (from pandas->pysheds) (2.8.2)
Collecting imageio>=2.4.1
  Using cached imageio-2.14.1-py3-none-any.whl (3.3 MB)
Collecting networkx>=2.2
  Using cached networkx-2.6.3-py3-none-any.whl (1.9 MB)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.7/site-
packages (from scikit-image->pysheds) (21.3)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in
/opt/conda/lib/python3.7/site-packages (from scikit-image->pysheds) (9.0.0)
Collecting tifffile>=2019.7.26
  Using cached tifffile-2021.11.2-py3-none-any.whl (178 kB)
Collecting PyWavelets>=1.1.1
  Using cached
PyWavelets-1.2.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (6.1 MB)
Requirement already satisfied: importlib-metadata in
/opt/conda/lib/python3.7/site-packages (from click>=4.0->rasterio>=1->pysheds)
(4.10.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.7/site-packages (from packaging>=20.0->scikit-
image->pysheds) (3.0.7)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-
packages (from python-dateutil>=2.7.3->pandas->pysheds) (1.16.0)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-
packages (from importlib-metadata->click>=4.0->rasterio>=1->pysheds) (3.7.0)
Requirement already satisfied: typing-extensions>=3.6.4 in
/opt/conda/lib/python3.7/site-packages (from importlib-
metadata->click>=4.0->rasterio>=1->pysheds) (4.0.1)
Installing collected packages: tifffile, snuggs, scipy, PyWavelets, networkx,
llvmlite, imageio, cligj, click-plugins, affine, scikit-image, rasterio, pyproj,
numba, geojson, pysheds
Successfully installed PyWavelets-1.2.0 affine-2.3.0 click-plugins-1.1.1
cligj-0.7.2 geojson-2.5.0 imageio-2.14.1 llvmlite-0.38.0 networkx-2.6.3
numba-0.55.1 pyproj-3.2.1 pysheds-0.3.2 rasterio-1.2.10 scikit-image-0.19.1
scipy-1.7.3 snuggs-1.4.7 tifffile-2021.11.2
```

[2]:
```
import sys
!{sys.executable} -m pip install elevation
```

```
Collecting elevation
  Using cached elevation-1.1.3-py3-none-any.whl (16 kB)
Collecting appdirs
  Using cached appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting fasteners
  Using cached fasteners-0.17.3-py3-none-any.whl (18 kB)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages
(from elevation) (8.0.3)
Requirement already satisfied: importlib-metadata in
/opt/conda/lib/python3.7/site-packages (from click->elevation) (4.10.1)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-
packages (from importlib-metadata->click->elevation) (3.7.0)
Requirement already satisfied: typing-extensions>=3.6.4 in
/opt/conda/lib/python3.7/site-packages (from importlib-
metadata->click->elevation) (4.0.1)
Installing collected packages: fasteners, appdirs, elevation
Successfully installed appdirs-1.4.4 elevation-1.1.3 fasteners-0.17.3
```

[3]:
```
# Preliminaries
import matplotlib.pyplot as plt
import numpy as np
import pandas
import gdal
from pysheds.grid import Grid
from matplotlib import colors
```

[4]:
```
filename = 'SantaBarbara_frontrange_DEM_90m.tif'  # The name of the file to␣
 ↪read.
                                                   # This must be in the same␣
 ↪folder as the ipynb file
grid = Grid.from_raster(filename)  # Create the grid
dem = grid.read_raster(filename)   # Put DEM data into the grid
```

[5]:
```
grid
```

[5]:
```
'affine' : Affine(0.0008333333333333334, 0.0, -120.0,
       0.0, -0.0008333333333333334, 34.6)
'shape' : (264, 528)
'nodata' : -32768
'crs' : <Other Coordinate Operation Transformer: longlat>
Description: PROJ-based coordinate operation
Area of Use:
- undefined
'mask' : array([[ True,  True,  True, …,  True,  True,  True],
```

```
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True],
        …,
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True]])
```

[6]:
```
dem
```

[6]:
```
Raster([[   255,    257,    255, …,   1296,   1279,   1263],
        [   244,    246,    245, …,   1251,   1212,   1200],
        [   236,    239,    242, …,   1210,   1166,   1172],
        …,
        [-32768, -32768, -32768, …, -32768, -32768, -32768],
        [-32768, -32768, -32768, …, -32768, -32768, -32768],
        [-32768, -32768, -32768, …, -32768, -32768, -32768]],
       dtype=int16)
```

[7]:
```
dem.viewfinder
```

[7]:
```
'affine' : Affine(0.0008333333333333334, 0.0, -120.0,
       0.0, -0.0008333333333333334, 34.6)
'shape' : (264, 528)
'nodata' : -32768
'crs' : <Other Coordinate Operation Transformer: longlat>
Description: PROJ-based coordinate operation
Area of Use:
- undefined
'mask' : array([[ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True],
        …,
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True],
        [ True,  True,  True, …,  True,  True,  True]])
```

[8]:
```
from pysheds.view import Raster, ViewFinder
land_dem = Raster(np.where(dem > -32768, dem, np.nan), viewfinder=grid.
 ↪viewfinder)
    # pull data from dem, if > -32768, then keep the dem value, and if it's␣
 ↪less than that, make it NAN
```

[9]:
```
from matplotlib.pyplot import figure
```

1.3 Goleta bay Plot

[11]:
```
Goleta_Slough_Lat = 34.4208388 # Coords for Goleta Slough
```

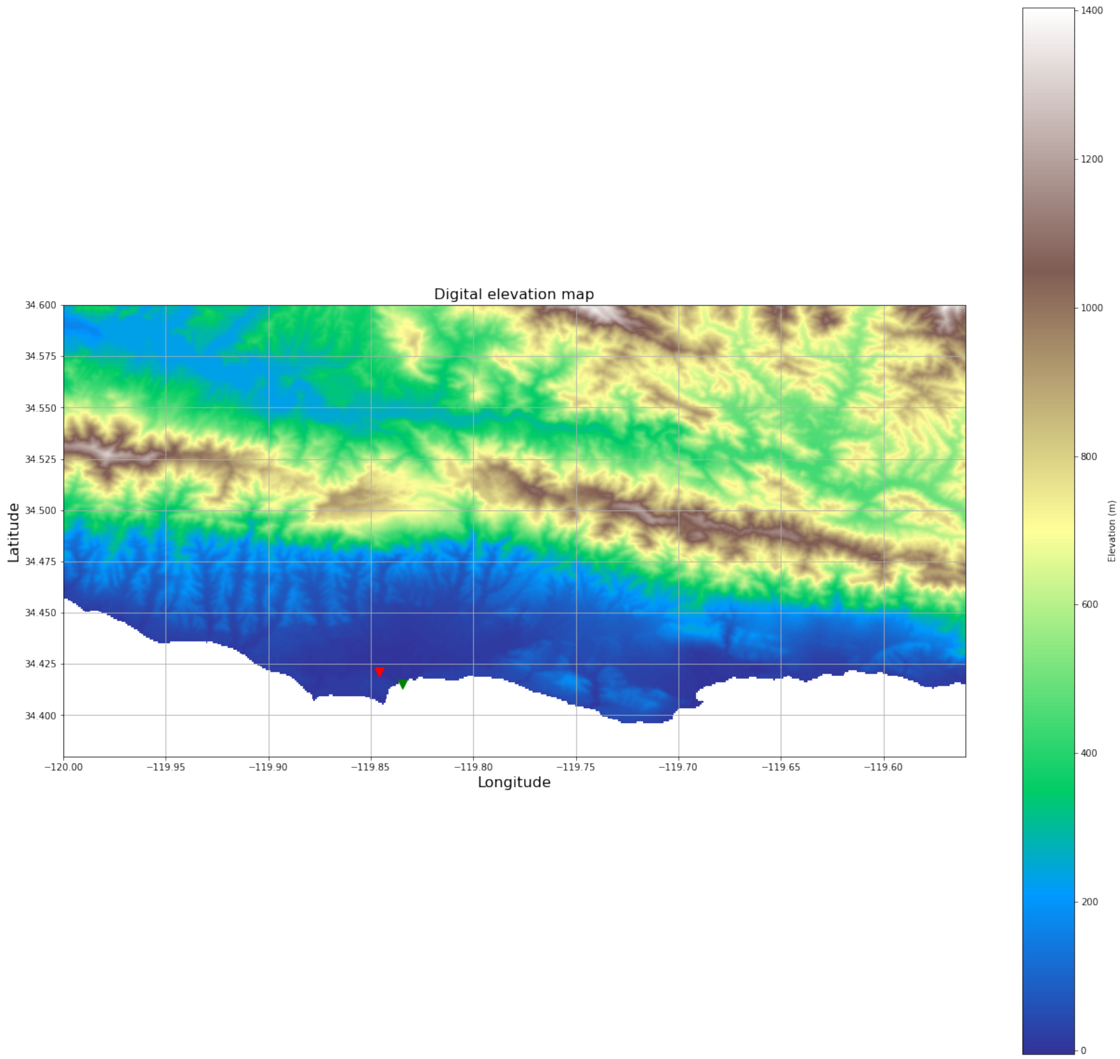
[12]:
```
Goleta_Slough_Long = -119.8457206
```

[13]:
```
Goleta_Pier_Lat = 34.4152131 # Coords for Goleta Pier
```

[14]:
```
Goleta_Pier_Long = -119.8348058
```

[15]:
```
fig, ax = plt.subplots(figsize=(18,16))
fig.patch.set_alpha(0)

plt.imshow(land_dem, extent=grid.extent, cmap='terrain', zorder=1)
plt.colorbar(label='Elevation (m)')
plt.grid(zorder=0)
plt.title('Digital elevation map', size=16)
plt.xlabel('Longitude' , size=16)
plt.ylabel('Latitude', size=16)

plt.tight_layout()

ax.plot(Goleta_Slough_Long, Goleta_Slough_Lat, 'rv' , markersize=10) # red␣
↪triangle point for the Goleta Slough

ax.plot(Goleta_Pier_Long, Goleta_Pier_Lat, 'gv' , markersize=10) # Green␣
↪triangle point for the Goleta Pier
```

[15]: [<matplotlib.lines.Line2D at 0x7f584401fa10>]

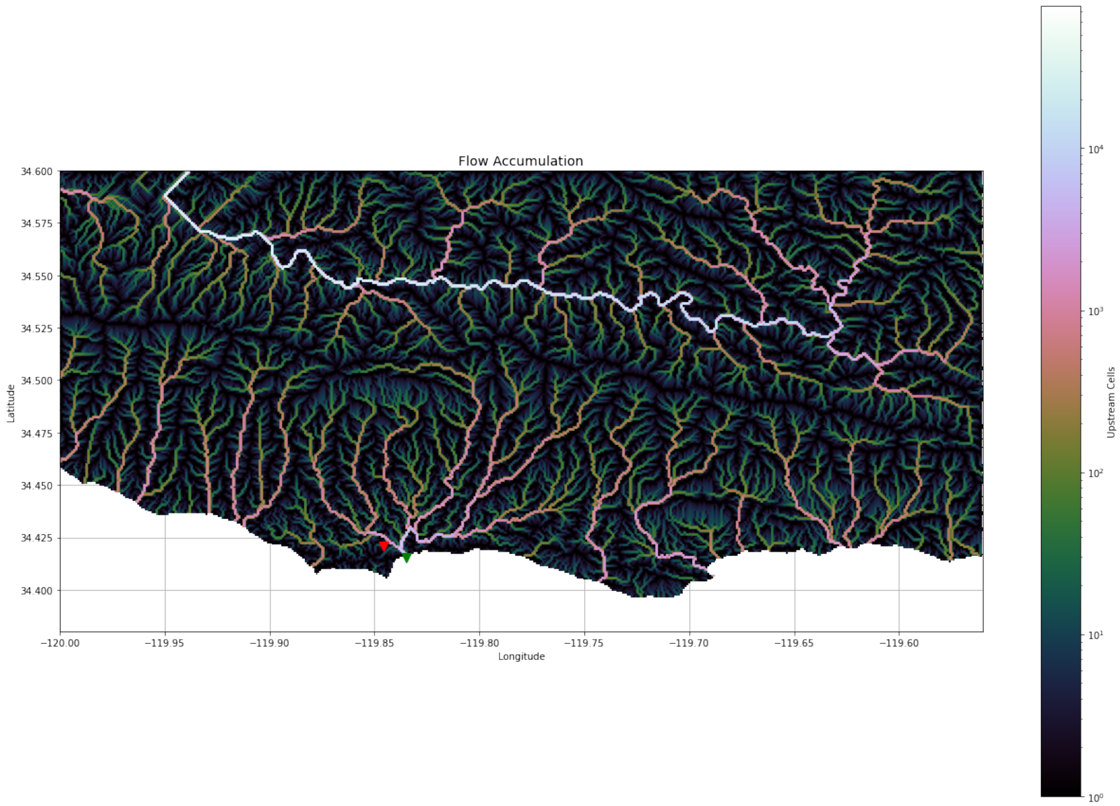


[16]:
```
# Condition DEM
# ----------------------
# Fill pits in DEM
pit_filled_dem = grid.fill_pits(land_dem) # function from pysheds

# Fill depressions in DEM
flooded_dem = grid.fill_depressions(pit_filled_dem)

# Resolve flats in DEM
inflated_dem = grid.resolve_flats(flooded_dem)
```

[17]:
```
# Determine D8 flow directions from DEM
# ------------------------------------
```

```python
# Compute flow directions
fdir = grid.flowdir(inflated_dem) # function from pysheds

# Calculate flow accumulation
acc = grid.accumulation(fdir)

# Mask out the ocean on this accumulation
accumulation_map = Raster(np.where(dem > -32768, acc, np.nan), viewfinder=grid.
 ↪viewfinder)
```

[18]:
```python
fig, ax = plt.subplots(figsize=(18,12)) # Matplotlib (graphing functions)
fig.patch.set_alpha(0)
plt.grid('on', zorder=0)
im = ax.imshow(accumulation_map, extent=grid.extent, zorder=2,
               cmap='cubehelix',
               norm=colors.LogNorm(1, acc.max()),
               interpolation='bilinear')
plt.colorbar(im, ax=ax, label='Upstream Cells')
plt.title('Flow Accumulation', size=14)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
ax.plot(Goleta_Slough_Long, Goleta_Slough_Lat, 'rv' , markersize=10) # red␣
 ↪triangle point for the Goleta Slough

ax.plot(Goleta_Pier_Long, Goleta_Pier_Lat, 'gv' , markersize=10) # Green␣
 ↪triangle point for the Goleta Pier

plt.tight_layout()
```



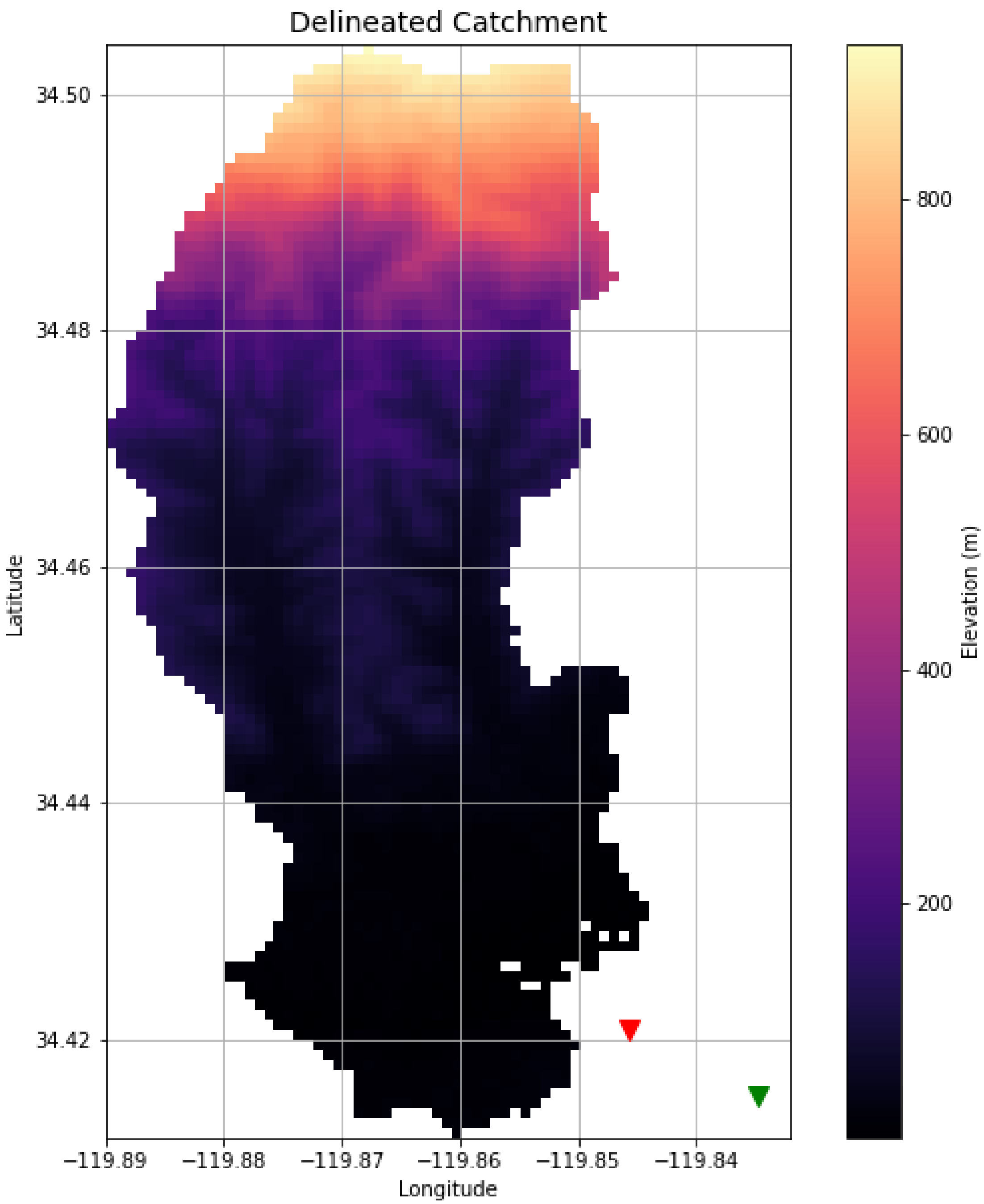
[19]:

```
# Delineate a catchment
# ----------------------
# Specify pour point
x, y = -119.8495492, 34.4223271 # Goleta Slough.  34.4223271,-119.8495492

# Snap pour point to high accumulation cell
x_snap, y_snap = grid.snap_to_mask(acc > 1000, (x, y)) # pick highest␣
↪accumulation point closest to coordinates

# Delineate the catchment
catch = grid.catchment(x=x_snap, y=y_snap, fdir=fdir, xytype='coordinate') #␣
↪from pysheds
                # lon       lat     flow direction      coordinate system

# Make a Raster that fills in land_dem data within the catchment
catchment_map = Raster(np.where(catch, land_dem, np.nan), viewfinder=grid.
↪viewfinder)
```

1.4 Goleta Slough and Pier Plot

[22]:

```
# Create a cropped verison, that is tight to the watershed
cgrid = grid
cgrid.clip_to(catch) # defining a new grid that aligns with the catchment data

ccatchment = cgrid.view(catchment_map)

# Put in NaNs instead of zeros
ccatchment = Raster(np.where(ccatchment > 0, ccatchment, np.nan),␣
 ↪viewfinder=cgrid.viewfinder)

# Plot the catchment
fig, ax = plt.subplots(figsize=(10,10))
fig.patch.set_alpha(0)

plt.grid('on', zorder=0)
im = ax.imshow(ccatchment, extent=cgrid.extent,
               zorder=1, cmap='magma')
plt.colorbar(im, ax=ax, label='Elevation (m)')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Delineated Catchment', size=14)

ax.plot(Goleta_Slough_Long, Goleta_Slough_Lat, 'rv' , markersize=10) # red␣
 ↪triangle point for the Goleta Slough

ax.plot(Goleta_Pier_Long, Goleta_Pier_Lat, 'gv' , markersize=10) # Green␣
 ↪triangle point for the Goleta Pier

ax.ticklabel_format(useOffset = False)
```



1.5 3. Mass balance for PBDE-47 loading to Goleta Bay

(35%) Develop a mass balance for PBDE-47 loading from Goleta Slough to Goleta Bay . You should report your answer as the mass flux of PDBE per unit area in ng/m^2/yr. For the purposes of this calculation, assume that the discharge from the Slough covers a 1 km^2 area of seafloor.

For the purposes of your calculation, you will need the following values: * Assume a PBDE sediment concentration of 0.7 ng/g. (That's 0.7 ng of PBDE per g of sediment.) * Sediment that is carried into the water column by turbulence is called "suspended sediment". This sediment is then moved along with the same speed as the water. Assume a suspended sediment concentration of 100 mg/L. (That's 100 mg of sediment suspended in every liter of water) * Annual streamflow statistics from San Jose Creek. You can find these here: https://waterdata.usgs.gov/nwis/annual/?search_site_no=11120500&agency_cd=USGS&referred_module

You must show your work two ways. 1. Make a table in Markdown (see e.g. https://www.tablesgenerator.com/markdown_tables) that clearly shows your unit conversions. You table should have three rows. Row one should be a heading indicating in words what the term is. Row two should be the numerator units. Row three should be the denominator units. You should have as many columns as necssary to reach the desired units. 2. Compute the loading rate using Python. You must define at least three variables in your calculation. Put your code in the cell below.

Miligrams/grams	Liters/cubic feet	Seconds/Minutes	Minutes/Hour	Hour/days	days/year	Kilometers$^{2/meters}$2
1g	2.83 L	60 s	60 min	24 hr	365 days	1 km^2
1000 mg	1 ft3	1 min	1hr	1 day	1 yr	10^6 m^2

[67]:
```
Sediment_Concentration = 0.7 # ng/g

Suspended_Sed_Conc = 100 # mg/L

Stream_flow = 0.148 # flow rate from USGS for 2021

Area_of_Seafloor = 1 # km^2
```

[85]:
```
# Converting 100 mg/L to g/ft^3
format(Suspended_Sed_Conc/35.315, '.2e')
    # Formatted in scientific Notation
```

[85]: '2.83e+00'

[87]:
```
# Converting 0.148 ft^3/s to ft^3/yr
format(Stream_flow/3.17, '2e')
        # Formatted in scientific Notation
```

[87]: '4.668770e-02'

[101]:
```
# Converting 1 km^2 to m^2
format(Area_of_Seafloor*1e+6, '2e')
```

[101]: '1.000000e+06'

[103]:
```
format(Sediment_Concentration * 4.67 * 2.83/10e+6, '2e')
```

[103]: '9.251270e-07'

1.6 9.25 ng/m^2/yr is Final Answer