Lily Kuentz – Lab 2 Submission

Question 1: Oregon Wildfires

```
# Import modules
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt

# Define data filepath
pathname = '/Users/lily/Documents/GitHub/geospatial-data-science/labs/lab2/lab2_data/'

# Read data
df = gpd.read_file(pathname + 'or_1992-2018.shp') # 'df' stands for DataFrame

# Find column labels
df.columns

# Find columns datatypes
df.dtypes
```

Question 1.a

```
# answering: Which county had the most human caused wildfires >50 acres?
# Filter for wildfires greater than 50 acres
df_fire50 = df[df['FIRE_SIZE'] > 50]

# Filter for wildfires caused  by humans
df_fire50_human = df_fire50[df_fire50['NWCG_CAUSE'] == "Human"]

#Identify the unique counties in the FIPS_NAME column
counties = df_fire50_human['FIPS_NAME'].unique()
counties

# Count the number of items that occur for each county in FIPS_NAME then print the output
next to the county name
for i in counties:
    number = df_fire50_human[df_fire50_human['FIPS_NAME'] == i].count()
    print(i, number)
```

After filtering the data for fires larger than 50 acres and caused by humans, the script iterates through the FIPS_NAME data column and counts the number of items within each county. Finally, the forloop prints out each iteration with the county name at the start of each iteration. Scrolling through this output we can see that Wasco County has the highest number of human caused wildfires >50 acres at 71.

Question 1.b

```
# answer: Which month had the most natural caused wildfires >100 acres?
# For now, I'm going to use a similar structure to that of 1.a
# Filter for wildfires larger than 100 acres
df_fire100 = df[df['FIRE_SIZE'] > 100]

# Filter for wildfires caused by natural forces
df_fire100_nat = df_fire100[df_fire100['NWCG_CAUSE'] == "Natural"]

# Convert dates from DISCOVERY_ to datetime64 type
datetime = pd.to_datetime(df_fire100_nat['DISCOVERY_'], format='%Y/%m/%d
%H:%M:%S.%f')

# Add the formatted datetime column to the dataframe
df_fire100_nat['datetime'] = datetime

# Add a new column with just the month of the wildfire indicated
df_fire100_nat['month'] = pd.DatetimeIndex(df_fire100_nat['datetime']).month

# Identify the months in which fires take place
months = df_fire100_nat['month'].unique()
months

# Count the number of items that occur in each month in 'month' column then print the output
next to the month number
for i in months:
    number = df_fire100_nat[df_fire100_nat['month'] == i].count()
    print(i, number)
```

After filtering the data for fires larger than 100 acres and caused by natural sources, the script iterates through the months column and counts the number of items within each month. Finally, the forloop prints out each iteration with the month number at the start of each iteration. Scrolling through this output we can see that month 8 (August) has the highest number of natural wildfires >100 acres at 549.

Question 1.c

```
# answering: How many fires >200 acres have an undetermined cause (e.g. Missing data/not
specified/undetermined?
# Filter for fires >200 acres
df_fire200 = df[df['FIRE_SIZE'] > 200]

df_fire200_unk = df_fire200[df_fire200['NWCG_CAUSE'] == 'Missing data/not
specified/undetermined']

# Count the number of items in the filtered dataset
df_fire200_unk.count()
```

There are 13 fires larger than 200 acres whose causes are unknown/missing/unspecified.

Question 1.d

```
# answering: What is the name, date, and county of the largest sized fire?
# Identify the Object ID/row number of the largest fire
print(df[df.FIRE_SIZE == df.FIRE_SIZE.max()])

# Print by the specific row of the largest fire (more legible format)
print(df.iloc[66964,:])
```

The largest fire was 558198.3 acres. It was called the Long Draw Fire, it took place on July 8th, 2012 in Malheur County.

Question 1.e

```
# answering: How many wildfires in Lane County were >50 acres?
# Filter dataset for fires that occurred in Lane County
df_lane = df[df['FIPS_NAME'] == 'Lane County']

df_lane[df_lane['FIRE_SIZE'] > 50].count()
```

There have been 33 fires larger than 50 acres in Lane County.

Question 2: Census Data

```
# Import modules
from cenpy import products
import matplotlib.pyplot as plt

# Define product
acs = products.ACS(2019)
```

Lily Kuentz – Lab 2 Submission

```
# Download data
ohio = products.ACS(2019).from_state('Ohio', level ='county', variables = ['B01003_001E'])

# Calculate Ohio population density and convert to people/square km
ohio['pop_density'] = ohio['B01003_001E'] / (ohio['geometry'].area / 1e+6)

from mpl_toolkits.axes_grid1 import make_axes_locatable

f, (ax1, ax2) = plt.subplots(ncols = 2, figsize=(10,10),sharex=True, sharey=True)

# These two lines make the colorbar the same size as the axes.
divider1 = make_axes_locatable(ax1)
divider2 = make_axes_locatable(ax2)
cax1 = divider1.append_axes("right", size="5%", pad=0.5)
cax2 = divider2.append_axes("right", size="5%", pad=0.5)

# Plot
ohio.plot('B01003_001E', ax=ax1, cmap='cividis', legend=True, cax=cax1)
ohio.plot('pop_density', ax=ax2, cmap='cividis', legend=True, cax=cax2)

plt.show()
```

The above figures show the population (left, millions of people) and population density (right, population/km^2) of each Ohio county across the whole state. Note: there is some distortion of the map along the Northern coast because the county lines are being drawn across Lake Erie.

```
### [**Ohio 2020 Election Results**](https://www.wlwt.com/article/election-2020-ohio-historical-maps-oct-23/34463242)

<div>
<img src="images/OH_2020election.jpg" width="400"/>
</div>

### [**Ohio Electoral College History**](https://www.270towin.com/states/Ohio)
<div>

<img src="images/OH_electoralcollege.png" width="400"/>
</div>
```

Lily Kuentz – Lab 2 Submission

I wanted to look at this data because I am from Ohio and I wanted to recreate the comparisons I had seen for other states. Ohio is well known around election time as a "swing state" because it does not have a consistent voting history. When we compare the 2019 US Census population data to the electoral college voting pattern from the most recent election, we can see that the regions with the most people and the highest densities of people voted for Biden while those with lower populations and lower population density voted for Trump.