30538 Problem Set 4

Nov 3

PS4: Due Sat Nov 2 at 5:00PM Central. Worth 100 points. We use (*) to indicate a problem that we think might be time consuming.

Style Points (10 pts)

Please refer to the minilesson on code style **here**.

Submission Steps (10 pts)

1. This problem set is a paired problem set.
2. Play paper, scissors, rock to determine who goes first. Call that person *Partner 1*.
    - Partner 1 (name and cnet ID): Betsy Shi, betsyshi
    - Partner 2 (name and cnet ID): Joy Wu, joywu
3. Partner 1 will accept the `ps4` and then share the link it creates with their partner. You can only share it with one partner so you will not be able to change it after your partner has accepted.
4. "This submission is our work alone and complies with the 30538 integrity policy." Add your initials to indicate your agreement: BS JW
5. "I have uploaded the names of anyone else other than my partner and I worked with on the problem set **here**" Betsy Shi & Joy Wu (1 point)
6. Late coins used this pset: 1 Late coins left after submission: 2
7. Knit your `ps4.qmd` to an PDF file to make `ps4.pdf`,
    - The PDF should not be more than 25 pages. Use `head()` and re-size figures when appropriate.
8. (Partner 1): push `ps4.qmd` and `ps4.pdf` to your github repo.
9. (Partner 1): submit `ps4.pdf` via Gradescope. Add your partner on Gradescope.
10. (Partner 1): tag your submission in Gradescope

Important: Repositories are for tracking code. **Do not commit the data or shapefiles to your repo.** The best way to do this is with `.gitignore`, which we have covered in class. If you do accidentally commit the data, Github has a guide. The best course of action depends on whether you have pushed yet. This also means that both partners will have to download the initial raw data and any data cleaning code will need to be re-run on both partners' computers.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import geopandas as gpd
import time
import random
from shapely.ops import nearest_points
```

Download and explore the Provider of Services (POS) file (10 pts)

1. Based on the rest of the problem set and the data dictionary, I pulled ZIP_CD, PGM_TRMNTN_CD, STATE_CD, PRVDR_NUM, FAC_NAME, CMPLNC_STUS_CD, CITY_NAME, PRVDR_CTGRY_CD, PRVDR_CTGRY_SBTYP_CD as variables.

2. a.

```
path = "data/pos2016.csv"
df_2016 = pd.read_csv(path)

short_term_hospitals = df_2016[(df_2016['PRVDR_CTGRY_CD'] == 1)
                     & (df_2016['PRVDR_CTGRY_SBTYP_CD'] == 1)]

num_short_term = short_term_hospitals.shape[0]
print(num_short_term)
```

7245

Yes, based on the pos2016 data and definitions, the number 7,245 is logical and consistent with the dataset's criteria for short-term hospitals.

b.

Cross-reference with 2024 AHA HOSPITAL STATISTICS. The number of U.S. community hospitals in 2024 is 5,129. (Reference: https://www.aha.org/statistics/fast-facts-us-hospitals) The gap may come from definition difference. The community hospitals in the research are defined as all nonfederal, short-term general, and other special hospitals. This does not fit the short-term designation in pos2016 dataset. And since 2016, there has been a trend of hospital closures, particularly in rural and less profitable areas. These closures could reduce the number of short-term facilities, meaning fewer hospitals are categorized as such in 2024.

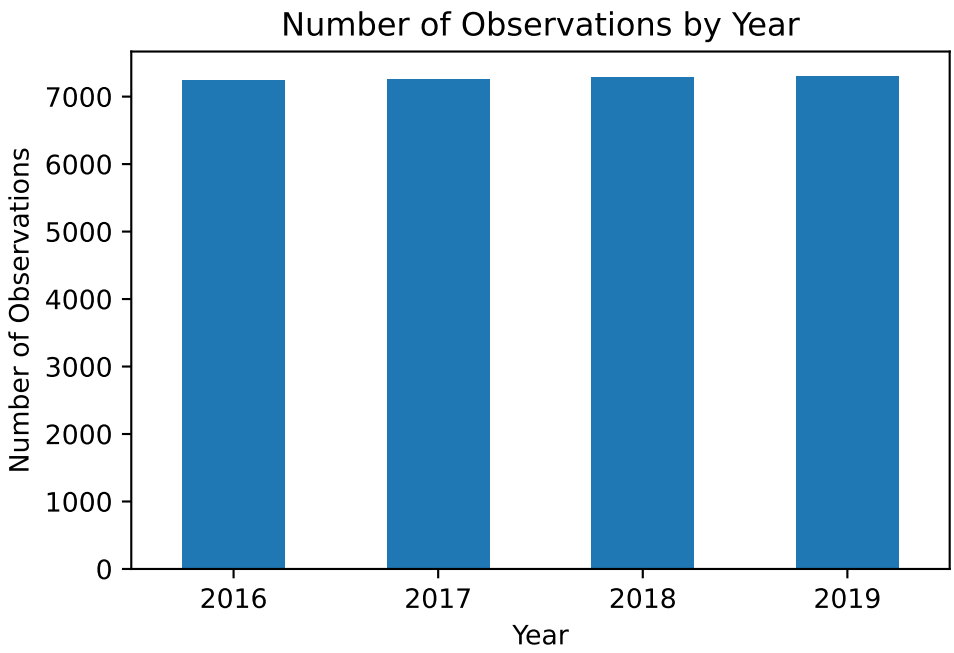
3.

```
years = [2017, 2018, 2019]
data_frames = []

for year in years:
```

```
    file_path = f'data/pos{year}.csv'
    df = pd.read_csv(file_path, encoding='ISO-8859-1')
    df['year'] = year
    all_short_term = df[(df['PRVDR_CTGRY_CD'] == 1) &
↪   (df['PRVDR_CTGRY_SBTYP_CD'] == 1)]
    data_frames.append(all_short_term)

data_all = pd.concat(data_frames, ignore_index=True)

short_term_hospitals['year'] = 2016
data_all = pd.concat([data_all, short_term_hospitals], ignore_index=True)

observation_by_year = data_all.groupby('year').size()
print(observation_by_year)

observation_by_year.plot(kind='bar')
plt.title('Number of Observations by Year')
plt.xlabel('Year')
plt.ylabel('Number of Observations')
plt.xticks(rotation=0)
plt.show()
```

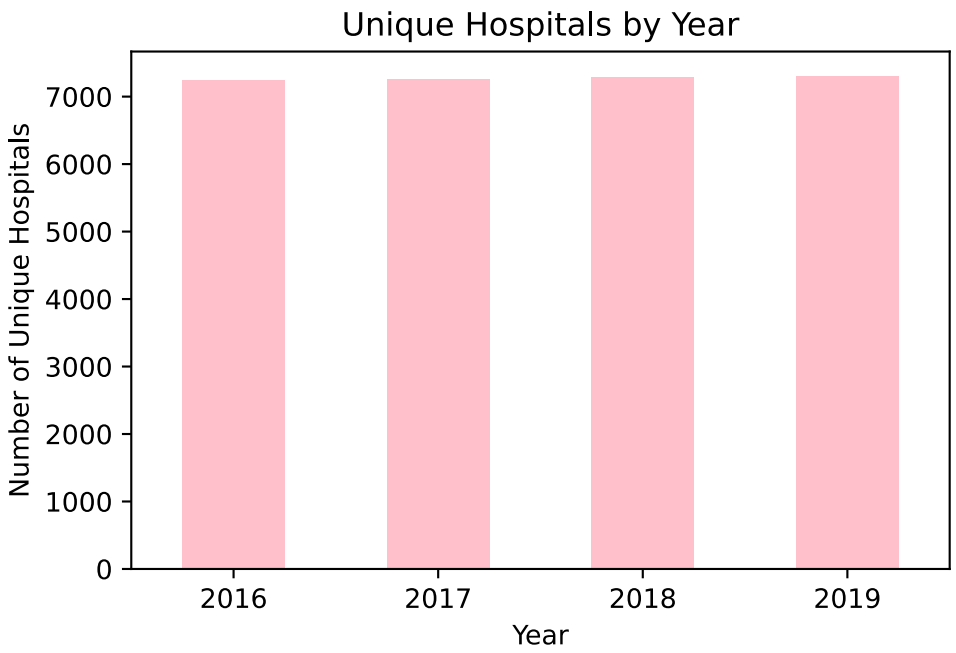
```
/var/folders/29/92n8lbb16qsc1h27rdtp7j8c0000gn/T/ipykernel_11697/1717530138.py:14:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
```

```
year
2016    7245
2017    7260
2018    7277
2019    7303
dtype: int64
```



4. a.

```
unique_hospital = data_all.groupby('year')['PRVDR_NUM'].nunique()

unique_hospital.plot(kind='bar', color='pink')
plt.title('Unique Hospitals by Year')
plt.xlabel('Year')
plt.ylabel('Number of Unique Hospitals')
plt.xticks(rotation=0)
plt.show()
```



b.

The two plots are highly identical, having similar count each year from 2016 to 2019 wich each bar being aroud 7,000. This suggests that each hospital only has one record each year, with no extra records or repeated entries. This consistency shows that the dataset is well organized. And since both the total number of observations an unique hospital counts are stable, there is no sigh of additional data collection periods within each year. This further supports that the deata is collected once a year. The high consistency between the two plots over time suggests that the data has high integrity, with no missing data or big changes in participants.

Identify hospital closures in POS file (15 pts) (*)

1.

```
data_all['PRVDR_NUM'] = pd.to_numeric(data_all['PRVDR_NUM'],
 ↪  errors='coerce')

active_2016 = data_all[(data_all['year'] == 2016) &
 ↪  (data_all['PGM_TRMNTN_CD'] == 0)]
active_2016 = active_2016[['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD']]
original_list = active_2016['PRVDR_NUM'].unique()

closed_hospitals_list = []

for year in [2017, 2018, 2019]:
```

```
    active_year = data_all[(data_all['year'] == year) &
↪   (data_all['PGM_TRMNTN_CD'] == 0)]

    active_year = active_year[active_year['PRVDR_NUM'].isin(original_list)]
    active_list_year = active_year['PRVDR_NUM'].unique()

    closed_in_year =
↪   active_2016[~active_2016['PRVDR_NUM'].isin(active_list_year)]

    closed_in_year = closed_in_year.copy()
    closed_in_year['Suspected_Closure_Year'] = year
    closed_hospitals_list.append(closed_in_year)

    closed_list = closed_in_year['PRVDR_NUM'].unique()
    original_list = original_list[~np.isin(original_list, closed_list)]
    active_2016 =
↪   active_2016[active_2016['PRVDR_NUM'].isin(active_list_year)]


all_closed_hospitals = pd.concat(closed_hospitals_list, ignore_index=True)

num_closed = all_closed_hospitals.shape[0]
print(f'Number of hospitals suspected to have closed between 2016 and 2019:
↪   {num_closed}')
print(all_closed_hospitals[['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD',
↪   'Suspected_Closure_Year']].head())
```

```
Number of hospitals suspected to have closed between 2016 and 2019: 174
   PRVDR_NUM                                  FAC_NAME   ZIP_CD  \
0      30001                     ABRAZO MARYVALE CAMPUS  85031.0
1      50196  ADVENTIST MEDICAL CENTER - CENTRAL VALLEY  93230.0
2      50435               FALLBROOK HOSPITAL DISTRICT  92028.0
3      60036    ARKANSAS VALLEY REGIONAL MEDICAL CENTER  81050.0
4      60043                   KEEFE MEMORIAL HOSPITAL  80810.0

   Suspected_Closure_Year
0                    2017
1                    2017
2                    2017
3                    2017
4                    2017
```

2.

```
sorted_closed = all_closed_hospitals.sort_values(by='FAC_NAME')
```