Problem Set 4

Partner1: Sienna Wang, Partner2: Hengyi Xing

2024-10-28

PS4: Due Sat Nov 2 at 5:00PM Central. Worth 100 points.

Style Points (10 pts)

Submission Steps (10 pts)

```
# Set up
import pandas as pd
import altair as alt
alt.renderers.enable("png")
import geopandas as gpd
import matplotlib.pyplot as plt
import time
from matplotlib.colors import ListedColormap
```

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

1.

We will use `PRVDR_CTGRY_SBTYP_CD` and `PRVDR_CTGRY_CD` to focus on short-term hospitals, `PRVDR_NUM` (CMS certification number) to identify unique hospitals, `PGM_TRMNTN_CD` to identify hospitals that are suspected to have closed, `FAC_NAME` to get the facility name, and `ZIP_CD` to get the ZIP.

2. (a)

```
# To import pos2016
path = ('/Users/wangshiying/Documents/71_Python_Programming_II/'
        'problem-set-4-hengyi-and-sienna')
file = "/data/pos2016.csv"

df_pos2016 = pd.read_csv(path + file)

# To focus on short-term hospitals
df_2016 = df_pos2016[(df_pos2016["PRVDR_CTGRY_SBTYP_CD"] == 1)
                     & (df_pos2016["PRVDR_CTGRY_CD"] == 1)]

# To count the observations
len(df_2016)
```

7245

Therefore, 7,245 hospitals are reported in the data. This number seems a bit larger than expected.

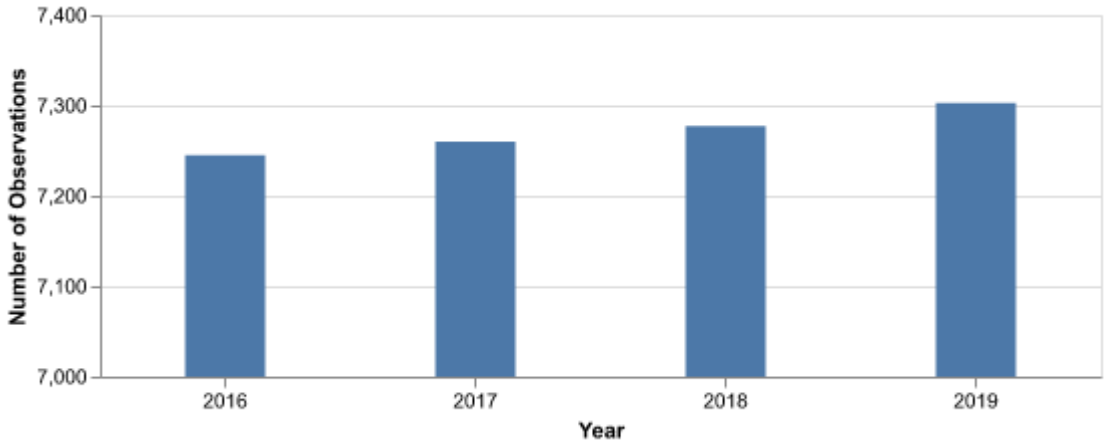
2. (b)

From Definite Healthcare, there are only 3,873 short-term hospitals in the US in 2024. And by American Hospital Association, the total number of all US hospitals is 6,120 in 2024 and only 5,534 in 2016.
We can find from the above information that the number of short-term hospitals we get from our dataset is even larger than the total number of hospitals in the US. As for reasons, firstly, the dataset might include multiple entries for the same hospital, such as separate records for different departments or units within the same facility (there are only 6770 different facility names). Secondly, CMS may categorize certain facilities as "short-term hospitals" even if they wouldn't be considered standalone hospitals in AHA or other national data.

3.

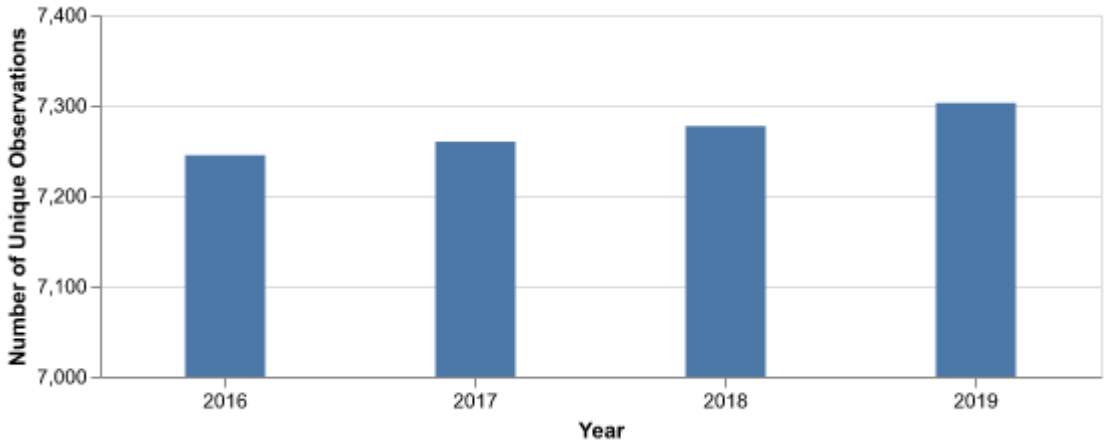
```
# To import datasets
file = "/data/pos2017.csv"
df_pos2017 = pd.read_csv(path + file)
file = "/data/pos2018.csv"
```

```
df_pos2018 = pd.read_csv(path + file, encoding="ISO-8859-1")
file = "/data/pos2019.csv"
df_pos2019 = pd.read_csv(path + file, encoding="ISO-8859-1")

# To focus on short-term hospitals
df_2017 = df_pos2017[(df_pos2017["PRVDR_CTGRY_SBTYP_CD"] == 1)
                     & (df_pos2017["PRVDR_CTGRY_CD"] == 1)]
df_2018 = df_pos2018[(df_pos2018["PRVDR_CTGRY_SBTYP_CD"] == 1)
                     & (df_pos2018["PRVDR_CTGRY_CD"] == 1)]
df_2019 = df_pos2019[(df_pos2019["PRVDR_CTGRY_SBTYP_CD"] == 1)
                     & (df_pos2019["PRVDR_CTGRY_CD"] == 1)]

# To append them together
df_2016["YEAR"] = 2016
df_2017["YEAR"] = 2017
df_2018["YEAR"] = 2018
df_2019["YEAR"] = 2019
df = pd.concat([df_2016, df_2017, df_2018, df_2019], axis=0,
 ↪  ignore_index=True)

# Plot the number of observations by year
alt.data_transformers.enable("vegafusion")
alt.Chart(df).mark_bar(size=40).encode(
    alt.X("YEAR:O", title="Year", axis=alt.Axis(labelAngle=0)),
    alt.Y("count()", title="Number of Observations",
 ↪  scale=alt.Scale(domain=(7000, 7400))) # Set domain to make the trend more
 ↪  obvious
).properties(
  width = 500,
  height = 180
)
```



4. (a)

```
# Find the number of unique hospitals
unique_number_by_year =
 ↪  df.groupby("YEAR")["PRVDR_NUM"].nunique().reset_index()

# Plot the number of unique hospitals
alt.Chart(df).mark_bar(size=40).encode(
    alt.X("YEAR:O", title="Year", axis=alt.Axis(labelAngle=0)),
    alt.Y("distinct(PRVDR_NUM):Q", title="Number of Unique Observations",
 ↪  scale=alt.Scale(domain=(7000, 7400))) # Set domain to make the trend more
 ↪  obvious
).properties(
  width = 500,
  height = 180
)
```



4. (b)

Comparing this with the previous step, we can find that there is no obvious difference. We can learn that in this dataset, each record represents one distinct hospital. Also, it implies that each CMS certification number appears multiple times over years. Therefore, this is an unbalanced panel dataset.

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

1.

```
# Find active facilities in each year
active_2016 = (df_2016[df_2016['PGM_TRMNTN_CD'] == 0]
               [['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD']].reset_index(drop=True))
active_2017 = (df_2017[df_2017['PGM_TRMNTN_CD'] == 0]
               [['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD']].reset_index(drop=True))
active_2018 = (df_2018[df_2018['PGM_TRMNTN_CD'] == 0]
               [['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD']].reset_index(drop=True))
active_2019 = (df_2019[df_2019['PGM_TRMNTN_CD'] == 0]
               [['PRVDR_NUM', 'FAC_NAME', 'ZIP_CD']].reset_index(drop=True))


# Define a function to identify suspected closure in each year
def find_closed_facilities(active_year_before, active_year_after):
    '''
```

```
    Identify records in one year yet missing in the next year, using merge.
    Input arguments are two dataframes above and the suspected closure year.
    '''
    result = pd.merge(active_year_before, active_year_after,
                      on=['PRVDR_NUM'], how='outer', indicator=True)
    result = result[result['_merge'] == 'left_only']
    return result


# Apply to each year from 2017 to 2019
closed_2017 = find_closed_facilities(active_2016, active_2017)
closed_2018 = find_closed_facilities(active_2016, active_2018)
closed_2019 = find_closed_facilities(active_2016, active_2019)

# Rectify 2018 and 2019 data to include only the exact year
closed_2018 = pd.concat([closed_2017,
 ↪  closed_2018]).drop_duplicates(keep=False)
closed_2019 = pd.concat([closed_2017,
 ↪  closed_2019]).drop_duplicates(keep=False)
closed_2019 = pd.concat([closed_2018,
 ↪  closed_2019]).drop_duplicates(keep=False)
closed_2017['SUS_YEAR'] = 2017
closed_2018['SUS_YEAR'] = 2018
closed_2019['SUS_YEAR'] = 2019

# Connect all three years together
closed_2016_to_2019 = pd.concat([closed_2017, closed_2018, closed_2019])
closed_count = len(closed_2016_to_2019)
print(closed_count,
      'facilities active in 2016 were suspected to have closed by 2019.')
```

```
174 facilities active in 2016 were suspected to have closed by 2019.
```

2.

```
# Modify the column name to FAC_NAME
closed_2016_to_2019['FAC_NAME'] = closed_2016_to_2019['FAC_NAME_x']
print(closed_2016_to_2019[['FAC_NAME', 'SUS_YEAR']]
      .sort_values('FAC_NAME', ignore_index=True).head(10))
```

```
                                      FAC_NAME  SUS_YEAR
0                         ABRAZO MARYVALE CAMPUS      2017
1      ADVENTIST MEDICAL CENTER - CENTRAL VALLEY      2017
2                         AFFINITY MEDICAL CENTER      2018
3  ALBANY MEDICAL CENTER / SOUTH CLINICAL CAMPUS      2017
4       ALLEGIANCE SPECIALTY HOSPITAL OF KILGORE      2017
5                         ALLIANCE LAIRD HOSPITAL      2019
6                       ALLIANCEHEALTH DEACONESS      2019
7                  ANNE BATES LEACH EYE HOSPITAL      2019
8        ARKANSAS VALLEY REGIONAL MEDICAL CENTER      2017
9            BANNER CHURCHILL COMMUNITY HOSPITAL      2017
```

3. (a)

Firstly, check whether the closure is a potential merger/acquisition by a simple "non-decrease" approach.

```
# Check the suspected closure in 2017
zip_closed_2017 = closed_2017['ZIP_CD_x'].tolist()
ma_zip_2017 = []
for zip_code in zip_closed_2017:
  zip_count_2017 = len(active_2017[active_2017['ZIP_CD'] == zip_code])
  zip_count_2018 = len(active_2018[active_2018['ZIP_CD'] == zip_code])
  if zip_count_2017 <= zip_count_2018:
    ma_zip_2017.append(zip_code)
ma_2017 = closed_2017[closed_2017['ZIP_CD_x'].isin(ma_zip_2017)]

# Check the suspected closure in 2018
zip_closed_2018 = closed_2018['ZIP_CD_x'].tolist()
ma_zip_2018 = []
for zip_code in zip_closed_2018:
  zip_count_2018 = len(active_2018[active_2018['ZIP_CD'] == zip_code])
  zip_count_2019 = len(active_2019[active_2019['ZIP_CD'] == zip_code])
  if zip_count_2018 <= zip_count_2019:
    ma_zip_2018.append(zip_code)
ma_2018 = closed_2018[closed_2018['ZIP_CD_x'].isin(ma_zip_2018)]

ma_2017_2018_a = pd.concat([ma_2017, ma_2018])
ma_count_a = len(ma_2017_2018_a)
print(f'By a simple "non-decrease" approach, {ma_count_a} facilities fit the
 ↪  definition.')
```

```
By a simple "non-decrease" approach, 97 facilities fit the definition.
```

Secondly, by further consideration, it is reasonable to remove records where the number of hospitals is still 0 in the year after. These ZIP codes experienced “non-decrease”, but there is no hospitals reappear in the year after.

```
# Check the suspected closure in 2017
zip_closed_2017 = closed_2017['ZIP_CD_x'].tolist()
ma_zip_2017 = []
for zip_code in zip_closed_2017:
  zip_count_2017 = len(active_2017[active_2017['ZIP_CD'] == zip_code])
  zip_count_2018 = len(active_2018[active_2018['ZIP_CD'] == zip_code])
  if (zip_count_2017 <= zip_count_2018) & (zip_count_2018 != 0):
    ma_zip_2017.append(zip_code)
ma_2017 = closed_2017[closed_2017['ZIP_CD_x'].isin(ma_zip_2017)]

# Check the suspected closure in 2018
zip_closed_2018 = closed_2018['ZIP_CD_x'].tolist()
ma_zip_2018 = []
for zip_code in zip_closed_2018:
  zip_count_2018 = len(active_2018[active_2018['ZIP_CD'] == zip_code])
  zip_count_2019 = len(active_2019[active_2019['ZIP_CD'] == zip_code])
  if (zip_count_2018 <= zip_count_2019) & (zip_count_2019 != 0):
    ma_zip_2018.append(zip_code)
ma_2018 = closed_2018[closed_2018['ZIP_CD_x'].isin(ma_zip_2018)]

ma_2017_2018_b = pd.concat([ma_2017, ma_2018])
ma_count_b = len(ma_2017_2018_b)
print(f'Removing ZIP codes with no hospitals reappearing, {ma_count_b}
 ↪  facilities fit the definition.')
```

```
Removing ZIP codes with no hospitals reappearing, 31 facilities fit the
definition.
```

3. (b)

```
left_count = closed_count - ma_count_a
print(f'By a simple "non-decrease" approach, {left_count} hospitals are
 ↪  left.')
left_count = closed_count - ma_count_b
print(f'Take ZIP codes with no hospitals reappearing into consideration,
 ↪  {left_count} hospitals are left.')
```

```
By a simple "non-decrease" approach, 77 hospitals are left.
Take ZIP codes with no hospitals reappearing into consideration, 143
hospitals are left.
```

3. (c)

Without considering ZIP codes with no hospitals reappearing, the table is shown below:

```
# Remove ma_2017_2018 from closed_2016_to_2019
combined = pd.concat([closed_2016_to_2019, ma_2017_2018_a])
corrected_closed = combined.drop_duplicates('PRVDR_NUM', keep=False)
# Modify the column name to ZIP_CD
corrected_closed['ZIP_CD'] = corrected_closed['ZIP_CD_x']
print(corrected_closed[['FAC_NAME', 'SUS_YEAR']]
      .sort_values('FAC_NAME', ignore_index=True).head(10))
```

```
                                        FAC_NAME  SUS_YEAR
0                          ALLIANCE LAIRD HOSPITAL      2019
1                         ALLIANCEHEALTH DEACONESS      2019
2                    ANNE BATES LEACH EYE HOSPITAL      2019
3                    BARIX CLINICS OF PENNSYLVANIA      2019
4                  BAYLOR EMERGENCY MEDICAL CENTER      2019
5  BAYLOR SCOTT & WHITE EMERGENCY MEDICAL CENTER ...      2019
6                        BELMONT COMMUNITY HOSPITAL      2019
7                            BIG SKY MEDICAL CENTER      2019
8              BLACK RIVER COMMUNITY MEDICAL CENTER      2019
9                      CARE REGIONAL MEDICAL CENTER      2019
```

Take ZIP codes with no hospitals reappearing into consideration, the table is shown below:

```
# Remove ma_2017_2018 from closed_2016_to_2019
combined = pd.concat([closed_2016_to_2019, ma_2017_2018_b])
corrected_closed = combined.drop_duplicates('PRVDR_NUM',keep=False)
# Modify the column name to ZIP_CD
corrected_closed['ZIP_CD'] = corrected_closed['ZIP_CD_x']
print(corrected_closed[['FAC_NAME', 'SUS_YEAR']]
      .sort_values('FAC_NAME', ignore_index=True).head(10))
```

```
                                  FAC_NAME  SUS_YEAR
0                    ABRAZO MARYVALE CAMPUS      2017
1                   AFFINITY MEDICAL CENTER      2018
2  ALLEGIANCE SPECIALTY HOSPITAL OF KILGORE      2017
3                    ALLIANCE LAIRD HOSPITAL      2019
4                  ALLIANCEHEALTH DEACONESS      2019
5             ANNE BATES LEACH EYE HOSPITAL      2019
6   ARKANSAS VALLEY REGIONAL MEDICAL CENTER      2017
7       BANNER CHURCHILL COMMUNITY HOSPITAL      2017
8              BANNER PAYSON MEDICAL CENTER      2018
9             BARIX CLINICS OF PENNSYLVANIA      2019
```

Download Census zip code shapefile (10 pt)

1. (a)

- **`.shp`** is the shape file storing geometric data, such as points, lines, or polygons.
- **`.shx`** is an index file for quick access to `.shp` records.
- **`.dbf`** holds attribute data related to each geographic feature.
- **`.prj`** defines the coordinate system and map projection for accurate spatial representation.
- **`.xml`** contains metadata describing the dataset's content, source, and creation details.

1. (b)

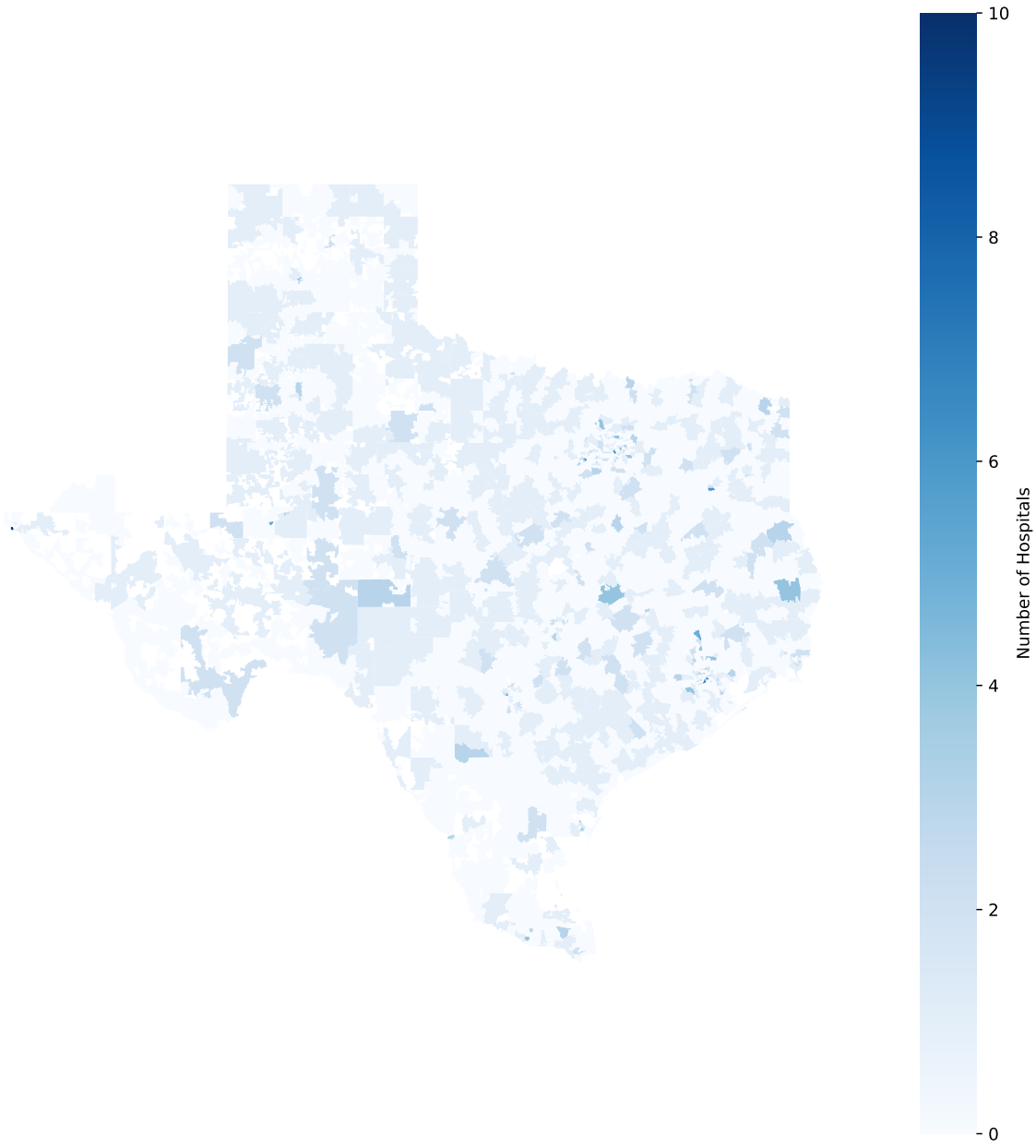
After unzipping, `.shp` file is 837.5 MB, `.shx` file is 265 KB, `.dbf` file is 6.4 MB, `.prj` file is 165 bytes, and `.xml` file is 16 KB.

2.

```
file = "/data/gz_2010_us_860_00_500k/gz_2010_us_860_00_500k.shp"
gdf_whole = gpd.read_file(path + file)
```

It can be found that for TX, the first 3 digits of ZIP code should be 750-799.

```
# Filter the geodataframe for TX
gdf = gdf_whole[gdf_whole["ZCTA5"].str.startswith(
    ("75", "76", "77", "78", "79"))]

# Count the number of hospitals by ZIP code
hospitals_number =
 ↪  df_2016.groupby("ZIP_CD")["PRVDR_NUM"].count().reset_index()
hospitals_number.columns = ["ZCTA5", "COUNTS"]

# Merge into the geodataframe
gdf["ZCTA5"] = pd.to_numeric(gdf["ZCTA5"])
gdf = gdf.merge(hospitals_number, on="ZCTA5", how="left").fillna(0)

# Make the plot
fig, ax = plt.subplots(figsize=(12, 12))

ax = gdf.plot(column="COUNTS",
         cmap="Blues",
         legend=True,
         legend_kwds={"label": "Number of Hospitals"},
         linewidth=0,
         ax=ax
         )
cbar = ax.get_figure().get_axes()[-1]
cbar.spines[:].set_visible(False)
plt.axis("off")
plt.show()
```


10
8
6
4
2
0
Number of Hospitals