Python 2 Problem Set 4

Yufei Liu

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

Style Points (10 pts)

Submission Steps (10 pts)

Partner 1: Yufei Liu (yufeil) Partner 2: Yufei Liu (yufeil)

This submission is our work alone and complies with the 30538 integrity policy. Add your initials to indicate your agreement: **Yufei Liu**, **Penny Shi**

I have uploaded the names of anyone else other than my partner and I worked with on the problem set. (No other one)

Late coins used this pset: **0** Late coins left after submission: **3** of **Yufei Liu**, ** ** of **Penny Shi**

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

```
import pandas as pd
import altair as alt
import time
import os

import warnings
warnings.filterwarnings('ignore')
```

1. **Selected Variables**:

- FAC_NAME: Facility Name

- PRVDR_CTGRY_SBTYP_CD: Identifies the subtype of the provider, within the primary category. Used in reporting to show the breakdown of provider categories, mainly for hospitals and SNFs.
- PRVDR_CTGRY_CD: Identifies the type of provider participating in the Medicare/Medicaid program.
- PGM_TRMNTN_CD: Termination Code (00=ACTIVE PROVIDER)
- PRVDR_NUM: provider. CMS Certification Number.
- CITY_NAME: City in which the provider is physically located.
- SSA_CNTY_CD: the county where the provider is located.
- SSA_STATE_CD: Social Security Administration geographic code indicating the state where the provider is located.
- STATE_CD: Two-character state abbreviation
- TRMNTN_EXPRTN_DT: Date the provider was terminated. For CLIA providers, date the laboratory's certificate was terminated or the expiration date of the current CLIA certificate
- PSYCH_UNIT_TRMNTN_CD: Indicates the reason that a psychiatric unit of a hospital is no longer exempt from Prospective Payment System (PPS).
- ZIP_CD: Five-digit ZIP code for a provider's physical address
- GNRL_FAC_TYPE_CD: Indicates the category-specific facility type code, for certain provider categories only.

2.

```
# import pos 2016
base_path = "/Users/nancy/Documents/Document/U.S/class/fall
 ↪  2024/dap2/homework/problem-set-4-y/data/pos_1"
path_data_pos2016 = os.path.join(base_path, 'pos2016.csv')
df_pos2016 = pd.read_csv(path_data_pos2016)
print(df_pos2016.shape)
df_pos2016.head(3)
```

```
(141557, 13)
```

	PRVDR_CTGRY_SBTYP_CD	PRVDR_CTGRY_CD	CITY_NAME	SSA_CNTY_CD	PRVDR
0	1.0	1.0	DOTHAN	340.0	10001.0
1	1.0	1.0	BRIDGEPORT	350.0	10004.0
2	1.0	1.0	BOAZ	470.0	10005.0

a. Short-term hospitals in dataset

Short-term facilities are hospitals with type code 01(PRVDR_CTGRY_CD = 01) and subtype code 01(PRVDR_CTGRY_SBTYP_CD = 01).

There are short-term 7245 hospitals according to the selected dataset.

```
# short-term hospitals(type code 01 and subtype code 01)
df_short_pos2016 = df_pos2016[(df_pos2016["PRVDR_CTGRY_CD"] ==
↪  1.0)&(df_pos2016['PRVDR_CTGRY_SBTYP_CD'] == 1)]
print(df_short_pos2016.shape)

print(df_short_pos2016['PRVDR_NUM'].nunique())
```

```
(7245, 13)
7245
```

```
df_short_pos2016_2 = df_pos2016[df_pos2016["GNRL_FAC_TYPE_CD"] == 1.0]
print(df_short_pos2016_2['PRVDR_NUM'].nunique())
```

```
7245
```

b. Cross-reference

However, after checking with the report from the Kaiser Family Foundation, we find only nearly 5,000 short-term, acute care hospitals in the United States.

It's different from what we got in the datasets, might because the 5000 shor-term hospitals are the one provide acute care while there are some counted hospital don't provide such service.

3.

```
# import datasets from 2017Q4 ~ 2019Q4
def load_yearly_data(base_path, start_year=2017, end_year=2019):
    # Dictionary to store DataFrames for each year
    data_frames = {}

    # Loop through each year in the range
    for year in range(start_year, end_year + 1):
        # Construct the file path for the specific year
        file_path = os.path.join(base_path, f'pos{year}.csv')
        df = pd.read_csv(file_path)

        # Filter for short-term facilities
        df = df[(df["PRVDR_CTGRY_CD"] == 1) & (df["PRVDR_CTGRY_SBTYP_CD"] ==
↪  1.0)]

        # Add a 'Year' column
```

```
        df["YEAR"] = year

        # Store the filtered DataFrame with the year column in the dictionary
        data_frames[year] = df
        print(f"Loaded data for {year}: {data_frames[year].shape[0]} rows,
         ↪  {data_frames[year].shape[1]} columns")

    return data_frames

data_frames = load_yearly_data(base_path)

df_short_pos2017 = data_frames[2017]
df_short_pos2018 = data_frames[2018]
df_short_pos2019 = data_frames[2019]

print(df_short_pos2017.columns)
print(df_short_pos2018.columns)
print(df_short_pos2019.columns)
```

```
Loaded data for 2017: 7260 rows, 14 columns
Loaded data for 2018: 7277 rows, 14 columns
Loaded data for 2019: 7303 rows, 14 columns
Index(['PRVDR_CTGRY_SBTYP_CD', 'PRVDR_CTGRY_CD', 'CITY_NAME', 'SSA_CNTY_CD',
       'PRVDR_NUM', 'STATE_CD', 'SSA_STATE_CD', 'TRMNTN_EXPRTN_DT', 'ZIP_CD',
       'GNRL_FAC_TYPE_CD', 'PSYCH_UNIT_TRMNTN_CD', 'YEAR', 'FAC_NAME',
       'PGM_TRMNTN_CD'],
      dtype='object')
Index(['PRVDR_CTGRY_SBTYP_CD', 'PRVDR_CTGRY_CD', 'CITY_NAME', 'SSA_CNTY_CD',
       'PRVDR_NUM', 'STATE_CD', 'SSA_STATE_CD', 'TRMNTN_EXPRTN_DT', 'ZIP_CD',
       'GNRL_FAC_TYPE_CD', 'PSYCH_UNIT_TRMNTN_CD', 'YEAR', 'FAC_NAME',
       'PGM_TRMNTN_CD'],
      dtype='object')
Index(['PRVDR_CTGRY_SBTYP_CD', 'PRVDR_CTGRY_CD', 'CITY_NAME', 'SSA_CNTY_CD',
       'PRVDR_NUM', 'STATE_CD', 'SSA_STATE_CD', 'TRMNTN_EXPRTN_DT', 'ZIP_CD',
       'GNRL_FAC_TYPE_CD', 'PSYCH_UNIT_TRMNTN_CD', 'FAC_NAME',
       'PGM_TRMNTN_CD',
       'YEAR'],
      dtype='object')
```

```
# append datset
df_short_pos2016['YEAR'] = 2016
```

```
df_short = pd.concat([df_short_pos2016] + list(data_frames.values()),
 ↪  ignore_index=True)
df_short.head(3)
```

	PRVDR_CTGRY_SBTYP_CD	PRVDR_CTGRY_CD	CITY_NAME	SSA_CNTY_CD	PRVDR
0	1.0	1.0	DOTHAN	340.0	10001.0
1	1.0	1.0	BRIDGEPORT	350.0	10004.0
2	1.0	1.0	BOAZ	470.0	10005.0

The Plot for observations from 2016 to 2019

```
obs_year = df_short.groupby('YEAR')['PRVDR_NUM'].count().reset_index()
obs_year.rename(columns={'PRVDR_NUM':'NUM_HOS'},inplace=True)

bar_obs = alt.Chart(obs_year).mark_bar(color='#9966CC').encode(
  x = alt.X('YEAR:O',title='Year'),
  y = alt.Y('NUM_HOS:Q',title='Count of Observations'),
  tooltip=['YEAR','NUM_HOS']
)

text_obs = bar_obs.mark_text(
    align='center',
    baseline='bottom',
    dy=-5
).encode(
    text='NUM_HOS:Q'
)
chart_obs = (bar_obs+text_obs).properties(
  title = 'Plot 1:Numbers of Short-term Hospitals by Year',
  width='container',
  height=400
)
chart_obs
```

```
alt.LayerChart(...)
```

4. a.

```
obs_year_unique =
 ↪  df_short.groupby('YEAR')['PRVDR_NUM'].nunique().reset_index()
obs_year_unique.rename(columns={'PRVDR_NUM':'NUM_HOS_UQ'},inplace=True)

bar_obs_unique = alt.Chart(obs_year_unique).mark_bar(color='#9966CC').encode(
  x = alt.X('YEAR:O',title='Year'),
  y = alt.Y('NUM_HOS_UQ:Q',title='Unique Count of Observations'),
  tooltip=['YEAR','NUM_HOS_UQ']
)

text_obs_unique = bar_obs_unique.mark_text(
    align='center',
    baseline='bottom',
    dy=-5
).encode(
    text='NUM_HOS_UQ:Q'
)
chart_obs_unique = (bar_obs_unique+text_obs_unique).properties(
  title = 'Plot 1:Unique Numbers of Short-term Hospitals by Year',
  width='container',
  height=400
)
chart_obs_unique
```

```
alt.LayerChart(...)
```

b.

The numbers of observations in each year are the same as the unique counts. Therefore, we can find each row of data in each year is one unique hospital. The dataset likely has a panel structure, where each PRVDR_NUM is tracked over multiple years but appears only once in each year.

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

1.

A hospital is suspected to have closed if its Termination Code('PGM_TRMNTN_CD') in the POS file lists them as an "Active Provider" ('PGM_TRMNTN_CD' = 0) in 2016 and then they are either not active or do not appear in the data at all in a subsequent year. We will also record suspected closed hospital's zip code ('ZIP_CD'), facility name ('FAC_NAME') and year of closure.

```
active_short2016 = df_short[(df_short['PGM_TRMNTN_CD'] == 0) &
↪  (df_short['YEAR'] == 2016)]['PRVDR_NUM'].unique()

def find_closure_csm(active_short2016, df_short):
  closure_csm = []

  for csm in active_short2016:
    # Flag to stop searching once added to closure_csm
    added_to_closure = False

    # Retrieve 2016 ZIP code and facility name for fallback use
    info_2016 = df_short[(df_short['PRVDR_NUM'] == csm) & (df_short['YEAR']
↪  == 2016)]
    zip_2016 = info_2016['ZIP_CD'].iloc[0] if not info_2016.empty else None
    fac_name_2016 = info_2016['FAC_NAME'].iloc[0] if not info_2016.empty else
↪  None

    for year in [2017,2018,2019]:
      year_check =
↪  df_short[(df_short['YEAR']==year)&(df_short['PRVDR_NUM']==csm)]

      # Check conditions:
      # 1. If the provider exists and PGM_TRMNTN_CD is not 0, add to
       ↪  closure_csm
      # 2. If the provider does not exist for this year, also add to
       ↪  closure_csm
      if not year_check.empty and (year_check['PGM_TRMNTN_CD'].iloc[0]!=0):
        closure_csm.append({
          'PRVDR_NUM':csm,
          'YEAR_CLO':year,
          'ZIP_CD':year_check['ZIP_CD'].values[0],
          'FAC_NAME':year_check['FAC_NAME'].values[0]
        })
        added_to_closure = False
        break

      elif year_check.empty:
        closure_csm.append({
            'PRVDR_NUM': csm,
            'YEAR_CLO': year,
            'ZIP_CD': zip_2016,
            'FAC_NAME': fac_name_2016
```

```
        })
        added_to_closure = True
        break

      if added_to_closure:
        continue

  return closure_csm

closure_csm = find_closure_csm(active_short2016, df_short)
print(f"There are {len(closure_csm)} hospitals suspected to close.")
```

```
There are 174 hospitals suspected to close.
```

2.

```
# Sort the list by 'FAC_NAME'
closure_csm.sort(key=lambda x: x['FAC_NAME'])

# Print the first 10 items by 'FAC_NAME' and 'YEAR_CLO'
for entry in closure_csm[:10]:
    print(f"The facility named '{entry['FAC_NAME']}' was closed in
    ↪ {entry['YEAR_CLO']}.")
```

```
The facility named 'ABRAZO MARYVALE CAMPUS' was closed in 2017.
The facility named 'ADVENTIST MEDICAL CENTER - CENTRAL VALLEY' was closed in
2017.
The facility named 'AFFINITY MEDICAL CENTER' was closed in 2018.
The facility named 'ALBANY MEDICAL CENTER / SOUTH CLINICAL CAMPUS' was closed
in 2017.
The facility named 'ALLEGIANCE SPECIALTY HOSPITAL OF KILGORE' was closed in
2017.
The facility named 'ALLIANCE LAIRD HOSPITAL' was closed in 2019.
The facility named 'ALLIANCEHEALTH DEACONESS' was closed in 2019.
The facility named 'ARKANSAS VALLEY REGIONAL MEDICAL CENTER' was closed in
2017.
The facility named 'ASCENSION NE WISCONSIN MERCY CAMPUS' was closed in 2018.
The facility named 'ATRIUM HEALTH KINGS MOUNTAIN' was closed in 2019.
```

3. a.
   b.
   c.

Download Census zip code shapefile (10 pt)

1. a.
   b.
2.

Calculate zip code's distance to the nearest hospital (20 pts) (*)

1.
2.
3.
4. a.
   b.
   c.
5. a.
   b.
   c.

Effects of closures on access in Texas (15 pts)

1.
2.
3.
4.

Reflecting on the exercise (10 pts)