Brian Mallari - Massachusetts General Hospital Exploratory Analysis - Python (pandas) (SYNTHETIC DATA)

November 25, 2024

1 Preliminary Work

[26]:
```
# Import pandas
import pandas as pd

# Import datetime
import datetime as dt
```

[27]:
```
# Import the encounters.csv file into a dataframe
df_copy_encounters = pd.read_csv('encounters.csv')

# Import the procedures.csv file into a dataframe
df_copy_procedures = pd.read_csv('procedures.csv')

# Import the procedures.csv file into a dataframe
df_copy_payers = pd.read_csv('payers.csv')
```

[28]:
```
# Look at the first rows of the encounters dataframe in order to confirm that␣
 ↪things are going well so far
df_copy_encounters.head()
```

[28]:

	Id	START	STOP	PATIENT
0	32c84703-2481-49cd-d571-3899d5820253	2011-01-02T09:26:36Z	2011-01-02T12:58:36Z	3de74169-7f67-9304-91d4-757e0f3a14d2
1	c98059da-320a-c0a6-fced-c8815f3e3f39	2011-01-03T05:44:39Z	2011-01-03T06:01:42Z	d9ec2e44-32e9-9148-179a-1653348cc4e2
2	4ad28a3a-2479-782b-f29c-d5b3f41a001e	2011-01-03T14:32:11Z	2011-01-03T14:47:11Z	73babadf-5b2b-fee7-189e-6f41ff213e01
3	c3f4da61-e4b4-21d5-587a-fbc89943bc19	2011-01-03T16:24:45Z	2011-01-03T16:39:45Z	3b46a0b7-0f34-9b9a-c319-ace4a1f58c0b
4	a9183b4f-2572-72ea-54c2-b3cd038b4be7	2011-01-03T17:36:53Z	2011-01-03T17:51:53Z	fa006887-d93c-d302-8b89-f3c25f88c0e1

```
                             ORGANIZATION                                 PAYER  \
0  d78e84ec-30aa-3bba-a33a-f29a3a454662  b1c428d6-4f07-31e0-90f0-68ffa6ff8c76
1  d78e84ec-30aa-3bba-a33a-f29a3a454662  b1c428d6-4f07-31e0-90f0-68ffa6ff8c76
2  d78e84ec-30aa-3bba-a33a-f29a3a454662  7caa7254-5050-3b5e-9eae-bd5ea30e809c
3  d78e84ec-30aa-3bba-a33a-f29a3a454662  b1c428d6-4f07-31e0-90f0-68ffa6ff8c76
4  d78e84ec-30aa-3bba-a33a-f29a3a454662  42c4fca7-f8a9-3cd1-982a-dd9751bf3e2a

  ENCOUNTERCLASS       CODE                                DESCRIPTION  \
0     ambulatory  185347001          Encounter for problem (procedure)
1     outpatient  308335008                Patient encounter procedure
2     outpatient  185349003         Encounter for check up (procedure)
3       wellness  162673000  General examination of patient (procedure)
4     ambulatory  390906007                          Follow-up encounter

   BASE_ENCOUNTER_COST  TOTAL_CLAIM_COST  PAYER_COVERAGE  REASONCODE  \
0                85.55           1018.02            0.00         NaN
1               142.58           2619.36            0.00         NaN
2                85.55            461.59          305.27         NaN
3               136.80           1784.24            0.00         NaN
4                85.55            234.72            0.00  55822004.0

  REASONDESCRIPTION
0               NaN
1               NaN
2               NaN
3               NaN
4    Hyperlipidemia
```

[29]:
```
# Look at the first rows of the procedures dataframe in order to confirm that␣
 ↪things are going well so far
df_copy_procedures.head()
```

[29]:
```
                  START                   STOP  \
0  2011-01-02T09:26:36Z  2011-01-02T12:58:36Z
1  2011-01-03T05:44:39Z  2011-01-03T06:01:42Z
2  2011-01-04T14:49:55Z  2011-01-04T15:04:55Z
3  2011-01-05T04:02:09Z  2011-01-05T04:17:09Z
4  2011-01-05T12:58:36Z  2011-01-05T16:42:36Z

                                PATIENT                              ENCOUNTER  \
0  3de74169-7f67-9304-91d4-757e0f3a14d2  32c84703-2481-49cd-d571-3899d5820253
1  d9ec2e44-32e9-9148-179a-1653348cc4e2  c98059da-320a-c0a6-fced-c8815f3e3f39
2  d856d6e6-4c98-e7a2-129b-44076c63d008  2cfd4ddd-ad13-fe1e-528b-15051cea2ec3
3  bc9d59c3-0a30-6e3b-f47d-022e4f03c8de  17966936-0878-f4db-128b-a43ae10d0878
4  3de74169-7f67-9304-91d4-757e0f3a14d2  9de5f0b0-4ba4-ce6f-45fb-b55c202f31a5

       CODE                                    DESCRIPTION  BASE_COST  \
```

```
0  265764009                          Renal dialysis (procedure)        903
1   76601001                             Intramuscular injection       2477
2  703423002  Combined chemotherapy and radiation therapy (p…      11620
3  173160006     Diagnostic fiberoptic bronchoscopy (procedure)       9796
4  265764009                          Renal dialysis (procedure)       1255

    REASONCODE                  REASONDESCRIPTION
0          NaN                                NaN
1          NaN                                NaN
2  363406005.0          Malignant tumor of colon
3  162573006.0  Suspected lung cancer (situation)
4          NaN                                NaN
```

[30]:
```
# Look at the first rows of the payers dataframe in order to confirm that␣
 ↪things are going well so far
df_copy_payers.head()
```

[30]:
```
                                     Id                    NAME  \
0  b3221cfc-24fb-339e-823d-bc4136cbc4ed           Dual Eligible
1  7caa7254-5050-3b5e-9eae-bd5ea30e809c                Medicare
2  7c4411ce-02f1-39b5-b9ec-dfbea9ad3c1a                Medicaid
3  d47b3510-2895-3b70-9897-342d681c769d                  Humana
4  6e2f1a2d-27bd-3701-8d08-dae202c58632  Blue Cross Blue Shield

               ADDRESS        CITY STATE_HEADQUARTERED      ZIP           PHONE
0  7500 Security Blvd   Baltimore                  MD  21244.0  1-877-267-2323
1  7500 Security Blvd   Baltimore                  MD  21244.0  1-800-633-4227
2  7500 Security Blvd   Baltimore                  MD  21244.0  1-877-267-2323
3    500 West Main St  Louisville                  KY  40018.0  1-844-330-7799
4      Michigan Plaza     Chicago                  IL  60007.0  1-800-262-2583
```

2 Number of Patients Who Have Been Admitted or Readmitted Over Time

[32]:
```
# Create a new, empty dataframe to hold data for this analysis
df_copy_1a = pd.DataFrame()

# Retain only the rows where encounter description contains the word "admission"
df_copy_encounters_filtered = df_copy_encounters[
    df_copy_encounters['DESCRIPTION'].str.contains("admission", case = False)
]

# Convert the start dates from ISO 8601 UTC date format to datetime format, and␣
 ↪then retain the year from the datetime
df_copy_1a['Admission Year'] = pd.
 ↪to_datetime(df_copy_encounters_filtered['START']).dt.year
```

```
# Convert the start dates from ISO 8601 UTC date format to datetime format, and then retain the month from the datetime
df_copy_1a['Admission Month'] = pd.to_datetime(df_copy_encounters_filtered['START']).dt.month

# Copy over the patient IDs
df_copy_1a['Patient ID'] = df_copy_encounters_filtered['PATIENT']

# Group the rows of this dataframe first by admission year and then admission month,
# count the unique patient IDs for each group, and convert the series to a dataframe
df_copy_1final = df_copy_1a.groupby(['Admission Year', 'Admission Month'])['Patient ID'].nunique().to_frame()

# Configure pandas to dislay all rows of dataframe for this analysis
pd.set_option('display.max_rows', None)

df_copy_1final
```

[32]:

Admission Year	Admission Month	Patient ID
2011	1	2
	2	3
	3	4
	4	1
	6	1
	7	5
	8	3
	9	1
	11	1
	12	3
2012	1	3
	2	6
	3	7
	4	7
	5	9
	6	9
	7	3
	8	7
	9	9
	10	3
	11	5
	12	5
2013	1	5
	2	10

	3	4
	4	7
	5	10
	6	6
	7	5
	8	11
	9	7
	10	7
	11	9
	12	7
2014	1	7
	2	8
	3	12
	4	10
	5	6
	6	7
	7	1
	8	5
	9	11
	10	4
	11	8
	12	8
2015	1	7
	2	7
	3	8
	4	5
	5	5
	6	5
	7	4
	8	7
	9	5
	10	7
	11	6
	12	8
2016	1	9
	2	4
	3	5
	4	4
	5	7
	6	4
	7	9
	8	7
	9	7
	10	7
	11	6
	12	3
2017	1	5

	2	8
	3	10
	4	6
	5	10
	6	5
	7	8
	8	5
	9	6
	10	5
	11	6
	12	13
2018	1	10
	2	6
	3	7
	4	6
	5	10
	6	9
	7	2
	8	6
	9	4
	10	8
	11	7
	12	5
2019	1	6
	2	3
	3	5
	4	6
	5	8
	6	4
	7	4
	8	4
	9	5
	10	9
	11	4
	12	5
2020	1	6
	2	5
	3	7
	4	6
	5	4
	6	9
	7	15
	8	10
	9	9
	10	10
	11	7
	12	16

2021	1	11
	2	5
	3	4
	4	4
	5	5
	6	6
	7	8
	8	7
	9	3
	10	3
	11	6
	12	6
2022	1	6

3 Average Duration of Patient Stay in the Hospital

[34]:
```
# Create a new, empty dataframe to hold data for this analysis
df_copy_2a = pd.DataFrame()

# Convert the start dates from ISO 8601 UTC data formate to datetime format
df_copy_2a['Start Date and Time'] = pd.to_datetime(df_copy_encounters['START'])
df_copy_2a['Stop Date and Time'] = pd.to_datetime(df_copy_encounters['STOP'])

# Create a new column to hold the differences between starting and ending dates␣
 ↪and times in hours
df_copy_2a['Duration of Stay'] = df_copy_2a['Stop Date and Time'] -␣
 ↪df_copy_2a['Start Date and Time']

# Retrieve the average duration of stay
df_copy_2b = pd.DataFrame().from_dict({'Avg. Stay in Hours' :␣
 ↪[round(df_copy_2a['Duration of Stay'].mean().total_seconds()/3600, 2)] })

# Eliminate the index of the dataframe in order to make the output a little bit␣
 ↪neater
blank_index = [''] * len(df_copy_2b)
df_copy_2final = df_copy_2b
df_copy_2final.index = blank_index
df_copy_2final
```

[34]:

Avg. Stay in Hours
7.27

4 Average Cost per Visit

[36]:
```
# Create a new, empty dataframe to hold data for this analysis
df_copy_3a = pd.DataFrame()

# Calculate the sum of total claim costs
sum_of_total_claim_costs = df_copy_encounters['TOTAL_CLAIM_COST'].sum()

# Calcuate the count of encounter IDs
count_of_encounter_IDs = df_copy_encounters['Id'].count()

# Calculate the average cost per visit, and round that value to the nearest␣
 ↪dollar
avg_cost_per_visit = round(sum_of_total_claim_costs/count_of_encounter_IDs, 2)

# Modify the existing  dataframe to hold that average
df_copy_3a['Avg. Cost per Visit (USD)'] = [avg_cost_per_visit]

# Eliminate the index of this dataframe in order to make the output look tidier
blank_index = ['']*len(df_copy_3a)
df_copy_3final = df_copy_3a
df_copy_3final.index = blank_index
df_copy_3final
```

[36]:

	Avg. Cost per Visit (USD)
	3639.68

5 Count of Procedures Covered by Insurance

[38]:
```
# Merge two of the starting dataframes into one dataframe
df_copy_4a = pd.merge(left = df_copy_procedures, right = df_copy_encounters,␣
 ↪how = 'inner', left_on = 'ENCOUNTER', right_on = 'Id')

# Look at the names of the columns for this resultant dataframe
# print(df_copy_4a.columns.to_list())

# Merge the third dataframe to the previous resultant dataframe
df_copy_4b = pd.merge(left = df_copy_4a, right = df_copy_payers, how = 'inner',␣
 ↪left_on = 'PAYER', right_on = 'Id')

# Look at the names of the columns for this resultant dataframe
# print(df_copy_4b.columns.to_list())

# Filter out the rows that have payer name as NO_INSURANCE from the latest␣
 ↪resultant dataframe
df_copy_4c = df_copy_4b[
```

```
    df_copy_4b['NAME'] != 'NO_INSURANCE'
]

# Retrieve the count of procedures covered by insurance
count_of_covered_procedures = df_copy_4c['ENCOUNTER'].count()

# create a new, empty dataframe to hold the data for this analysis
df_copy_4c = pd.DataFrame()
df_copy_4c['Procedures Covered by Insurance'] = [count_of_covered_procedures]
df_copy_4c

# Eliminate the index of this dataframe in order to make the output look tidier
blank_index = ['']*len(df_copy_4c)
df_copy_4final = df_copy_4c
df_copy_4final.index = blank_index
df_copy_4final
```

[38]:

Procedures Covered by Insurance
32599