Q1

November 4, 2019

1 DATASET for Q1

[78]:
```
import pandas as pd
import numpy as np
import datetime
```

[79]:
```
#import all datasets
allergiesCSV = pd.read_csv("../data/allergies.csv")
observationsCSV = pd.read_csv("../data/observations.csv")
acsCSV = pd.read_csv("../data/ACS.csv")
careplansCSV = pd.read_csv("../data/careplans.csv")
encountersCSV = pd.read_csv("../data/encounters.csv")
immunizationsCSV = pd.read_csv("../data/immunizations.csv")
procedutesCSV = pd.read_csv("../data/procedures.csv")
zipCSV = pd.read_csv("../data/zip_to_zcta_2019.csv")
conditionsCSV = pd.read_csv("../data/conditions.csv")
medicationsCSV = pd.read_csv("../data/medications.csv")
patientsCSV = pd.read_csv("../data/patients.csv")
```

[80]:
```
#we only need data from MA, cleaning for necessary data
massZipCSV = zipCSV[zipCSV.STATE == "MA"]                                    ␣
 ↪#contains only MA ZCTA
zipAndIncome = massZipCSV.merge(acsCSV,left_on='ZCTA', right_on='GEO.id2')
zipAndIncome = zipAndIncome.replace(['2,500-',"***",'-','**'],-1)           ␣
 ↪#necessary for converting to int so it can operated on
```

[81]:
```
zipDrop = ['ZIP_CODE','PO_NAME','STATE','ZIP_TYPE','zip_join_type','GEO.
 ↪id','GEO.id2','GEO.
 ↪display-label','HC01_EST_VC13','HC01_MOE_VC13','HC02_EST_VC13','HC02_MOE_VC13','HC03_EST_VC
zaiMerge = zipAndIncome.copy().drop(zipDrop,axis=1)

#Values are imported as strings. Converted to ints to run regression
zaiMerge['HouseholdIncome'] = zipAndIncome['HC01_EST_VC13'].astype(int)
zaiMerge['HouseholdMOE'] = zipAndIncome['HC01_MOE_VC13'].astype(int)        ␣
 ↪#zaiMerge is a table with only ZCTA, Household Median Income, & its MOE

#zaiMerge['Family'] = zipAndIncome[['HC02_EST_VC13','HC02_MOE_VC13']].values.
 ↪tolist()
```

```
#zaiMerge['Married'] = zipAndIncome[['HC03_EST_VC13','HC03_MOE_VC13']].values.
 ↪tolist()
#zaiMerge['Nonfamily'] = zipAndIncome[['HC04_EST_VC13','HC04_MOE_VC13']].values.
 ↪tolist()  #could be interesting for later use
```

[82]:
```
patientsCSV = patientsCSV.drop(['Unnamed: 0',␣
 ↪'SSN','DRIVERS','PASSPORT','PREFIX','FIRST', 'LAST', 'SUFFIX',␣
 ↪'MAIDEN'],axis=1)    #removed data to clean data and make it visually easier␣
 ↪to understand and to ensure privacy
patientsCSV['HOMEZIP'] = patientsCSV['ADDRESS'].map(lambda x:'0'+str(x)[-7:-3])␣
 ↪   #lambda fxn grabs zip code. This works since the format an address is␣
 ↪fixed

#maps each cell to respective income through matching zipcodes
patientsCSV['INCOME'] = patientsCSV['HOMEZIP'].map(lambda x: -1 if x =='0' or x␣
 ↪not in zaiMerge['ZCTA'].unique() else zaiMerge.loc[zaiMerge['ZCTA']==␣
 ↪x]['HouseholdIncome'].values[0])
patientsCSV['MOE'] = patientsCSV['HOMEZIP'].map(lambda x: -1 if x =='0' or x␣
 ↪not in zaiMerge['ZCTA'].unique() else zaiMerge.loc[zaiMerge['ZCTA']==␣
 ↪x]['HouseholdMOE'].values[0])

patientsCSV['DEATHDATE'] = patientsCSV['DEATHDATE'].fillna(0)
```

[83]:
```
emergencyCodes =␣
 ↪[50849002,183460006,183452005,183478001,34285007,183495009,32485007,305408004,305411003]

emergency = pd.read_csv("../data/encounters.csv")
emergency = emergency.drop('Unnamed: 0', axis=1)

#Clean encounters that are too old or not considered an emergency visit
emergency = emergency[emergency['CODE'].isin(emergencyCodes)]
emergency = emergency[emergency['DATE'].between('2008', '2017')]

#last only has most recent encounters for each patient
last = emergency.drop_duplicates("PATIENT",keep='last')
last['DEATHDATE'] = last['PATIENT'].map(lambda x:␣
 ↪patientsCSV[patientsCSV["ID"]==x]['DEATHDATE'].values[0])

#Maps date of death for each patient to their encounter
emergency['DEATHDATE'] = emergency['ID'].map(lambda x: last.
 ↪loc[last['ID']==x]['DEATHDATE'].values[0] if x in last['ID'].unique() else 0)

#I noticed encounters.CSV was missing a lot of reason descriptions that␣
 ↪conditions.CSV had,
#so this for loop fills in missing information
for index,row in emergency.iterrows():
```

```
   if isinstance(row['REASONDESCRIPTION'],float):                          #only␣
↪applied to encounters that are not filled. will not refill a cell.
       date = row['DATE']
       find = conditionsCSV[conditionsCSV['PATIENT'] == row['PATIENT']]  #find␣
↪has all of rows that pertain to patient and reason narrows it to the␣
↪particular encounter
       reason = find[find['START'] == date]['DESCRIPTION']                ␣
↪#assuming someone doesn't visit the hospital in the same day for a different␣
↪reason
       code = find[find['START'] == date]['CODE']                          #also␣
↪assuming the hospital records all visit and didn't visit double visits if it␣
↪occured
       try:
           emergency['REASONDESCRIPTION'][index] = reason.values[0]
           emergency['REASONCODE'][index] = code.values[0]
       except:
           emergency['REASONDESCRIPTION'][index] = np.nan
           emergency['REASONCODE'][index] = np.nan
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:12:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  if sys.path[0] == '':
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:26:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:27:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:29:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:30:
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

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

[84]:
```
#Converted to date_time in order to do calculations on the dates
emergency['DATE']=pd.to_datetime(emergency['DATE'])
emergency['DEATHDATE']=pd.to_datetime(emergency['DEATHDATE'])

#Finds amount of time between emergency visit and death
emergency['DEATHDIFFERENCE'] = emergency['DEATHDATE']-emergency['DATE']
#Dummy variable for all patients who passed away 1 year within visit
emergency['YearDeath'] = emergency['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=365) else 0)
emergency.loc[(emergency['DEATHDIFFERENCE'] < datetime.
 ↪timedelta(days=0)),'DEATHDIFFERENCE'] = np.nan
#DEATH is a dummy variable for deaths post discharge, this removes deaths that␣
 ↪happen on day 0 since that wouldn't be considered post discharge.
emergency['DEATH'] = emergency['DEATHDIFFERENCE'].map(lambda x: 0 if x ==␣
 ↪datetime.timedelta(days=0) or x!=x else 1)
#I think this is more indicative of mortality related to post-discharge.
emergency['DEATH100'] = emergency['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=100) else 0)
emergency['DEATH60']  = emergency['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=60) else 0)
emergency['DEATH30']  = emergency['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=30) else 0)
```

[85]:
```
LOW_LINE = 56763.2 #low income threshole
#Adding patient information for each encounter. For regression and analysis
emergency['RACE'] = emergency['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['RACE'].values[0])
emergency['AGE'] = emergency['PATIENT'].map(lambda x: int(patientsCSV.
 ↪loc[patientsCSV['ID']==x]['AGE'].values[0]))
emergency['HOMEZIP'] = emergency['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['HOMEZIP'].values[0])
emergency['INCOME'] = emergency['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['INCOME'].values[0])
emergency['MOE'] = emergency['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['MOE'].values[0])

#Create dummy variables for race for regression
emergency['BLACK'] = emergency['RACE'].map(lambda x: 1 if x == 'black or␣
 ↪african american' or x=='black' else 0)
emergency['HISPANIC'] = emergency['RACE'].map(lambda x: 1 if x == 'hispanic'␣
 ↪else 0)
emergency['ASIAN'] = emergency['RACE'].map(lambda x: 1 if x == 'asian' else 0)
```

```
emergency['WHITE'] = emergency['RACE'].map(lambda x: 1 if x == 'white' else 0)

#Dummy variable for if patient is considered low income. Low income is 80% of␣
 ↪state median
emergency['LOWINCOME'] = emergency['INCOME'].map(lambda x: 1 if x < LOW_LINE␣
 ↪else 0)
#Replacing NA was necessary for the above lambda fxn, however NA is necessary␣
 ↪for regression since 0 will affect the regression
emergency['INCOME'] = emergency['INCOME'].replace(-1,np.nan)

emergency['SENIOR'] = emergency['AGE'].map(lambda x: 1 if x>=65 else 0)
emergency['MINORITY'] = emergency['RACE'].map(lambda x: 1 if x=='asian' or␣
 ↪x=='hispanic' or x=='black' or x=='black or african american' else 0)
```

[91]:
```
emergency.to_csv('../data/EmergencyVisits.csv')
```

2 1. Extra Analysis

[86]:
```
#Matches patient BMI to patient and encounter
BMI = observationsCSV[observationsCSV['DESCRIPTION']=='Body Mass Index']
BMI["VALUE"] = BMI['VALUE'].astype(float)
patients = BMI['PATIENT'].unique()
emergency['BMI'] = emergency['PATIENT'].map(lambda x:␣
 ↪BMI[BMI['PATIENT']==x]['VALUE'].values[0] if x in patients else -1)
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:3:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[87]:
```
#Homeless dummy variable
homeless = observationsCSV[observationsCSV['DESCRIPTION']=='Housing status']
patients = homeless['PATIENT'].unique()
emergency['HOMELESS'] = emergency['PATIENT'].map(lambda x: 1 if x in patients␣
 ↪else 0)        #Patients with a 'Housing status' description were all␣
 ↪homeless
```

[88]:
```
#Variable for Diastolic Blood Pressure
DBP = observationsCSV[observationsCSV['DESCRIPTION']=='Diastolic Blood␣
 ↪Pressure']
```

```
patient = DBP['PATIENT'].unique()
emergency['DIASTOLICBP'] = emergency['PATIENT'].map(lambda x:␣
↪float(DBP[DBP['PATIENT']==x]['VALUE'].values[0]) if x in patient else -1)
emergency['BADBP'] = emergency['DIASTOLICBP'].map(lambda x: 1 if x > 80 else 0)
```

[89]:
```
#Variable for Calcium levels
Calcium = observationsCSV[observationsCSV['DESCRIPTION']=='Calcium']
patient = Calcium['PATIENT'].unique()
emergency['CALCIUM'] = emergency['PATIENT'].map(lambda x:␣
↪float(Calcium[Calcium['PATIENT']==x]['VALUE'].values[0]) if x in patient␣
↪else -1)
```

[90]:
```
#Finds patients who got the flu shot within 6 months of their most recent visit␣
↪to the hospital.
emergency['FLU']=None        #1 if patient recieved flu shot w/in 6 months of␣
↪visit, 0 otherwise
emergency['FLU_DIFF']=None   #difference between time of flu shot and time of␣
↪visit, if 0 days then they visited the hospital for the flu shot/happened to␣
↪get it

flu = immunizationsCSV[immunizationsCSV['CODE'] == 140]   #CODE 140 is the code␣
↪for influenza immunization                                                  ␣
↪
patients = flu['PATIENT'].unique()

for index, row in emergency.iterrows():
    checked = 0     #keeps track of if the inner for loop found a matching␣
↪immunization
    patientID = row['PATIENT']
    if patientID in patients:    #if patient is in the flu dataset, they had an␣
↪immunization
        date = flu[flu['PATIENT']==patientID]['DATE']   #list of all dates they␣
↪recieved influenza shot
        for d in date:
            d = datetime.datetime.strptime(d, '%Y-%m-%d')
            diff = row['DATE'] - d
            if datetime.timedelta(days=0) <= diff < datetime.
↪timedelta(days=183):    #an influenza shot lasts around 6 months/183 days
                checked = 1
                emergency['FLU'][index]=1
                emergency['FLU_DIFF'][index]=diff    #can help us determine␣
↪cause of visit if there is none (i.e. if =0 days, then the visit was for the␣
↪shot)
        if checked == 0:
            emergency['FLU'][index]=0
            emergency['FLU_DIFF'][index]=-1
    else:
```

```
            emergency['FLU'][index]=0
            emergency['FLU_DIFF'][index]=-1

#I found that a lot of the encounter dates in encounters.csv coincided with
#dates (found in immunizations.csv) that the patient got a flu shot. They all
#coincided with Outpatient Encounters without ReasonDescriptions. So I assume
#the reason they went was for the flu shot and not some hidden sickness/
 ↪emergency
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:21:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:22:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:18:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:19:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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

3 2. Rates

3.1 2.1 Post Discharge Mortality Rate

[24]:
```
seniors = emergency[emergency['SENIOR']==1]
notSeniors = emergency[emergency['SENIOR']==0]
minority = emergency[emergency['MINORITY']==1]
notMinority = emergency[emergency['MINORITY']==0]
hispanic = emergency[emergency['HISPANIC']==1]
notHispanic = emergency[emergency['HISPANIC']==0]
asian = emergency[emergency['ASIAN']==1]
notAsian = emergency[emergency['ASIAN']==0]
lowInc = emergency[emergency['LOWINCOME']==1]
```

```
notLowInc = emergency[emergency['LOWINCOME']==0]

#post-discharge 100 day mortalty rate for seniors
seniorMortRate100 = seniors[seniors['DEATH100']==1].shape[0]/seniors.shape[0]
 ↪   #3.3%
notMortRate100 = notSeniors[notSeniors['DEATH100']==1].shape[0]/notSeniors.
 ↪shape[0]    #0%

#post-discharge 1 year mortalty rate for seniors
seniorMortRate365 = seniors[seniors['YearDeath']==1].shape[0]/seniors.shape[0]
 ↪     #7.1%
notMortRate365 = notSeniors[notSeniors['YearDeath']==1].shape[0]/notSeniors.
 ↪shape[0]   #0.9%

#post-discharge 1 year mortalty rate for minorities
minorityMortRate = minority[minority['YearDeath']==1].shape[0]/minority.
 ↪shape[0]       #2.1%
notMinorityMortRate = notMinority[notMinority['YearDeath']==1].shape[0]/
 ↪notMinority.shape[0]  #1.1%

#post-discharge 100 day mortalty rate for hispanic patients
hispanicMortRate100 = hispanic[hispanic['DEATH100']==1].shape[0]/hispanic.
 ↪shape[0]       #0.67%
notHispanicMortRate100 = notHispanic[notHispanic['DEATH100']==1].shape[0]/
 ↪notHispanic.shape[0]   #0.41%

#post-discharge 1 year mortalty rate for hispanic patients
hispanicMortRate = hispanic[hispanic['YearDeath']==1].shape[0]/hispanic.
 ↪shape[0]       #2%
notHispanicMortRate = notHispanic[notHispanic['YearDeath']==1].shape[0]/
 ↪notHispanic.shape[0]   #2.1%

#post-discharge 100 day mortalty rate for asian patients
asianMortRate100 = asian[asian['DEATH100']==1].shape[0]/asian.shape[0]       #1.
 ↪6%
notAsianMortRate100 = notAsian[notAsian['DEATH100']==1].shape[0]/notAsian.
 ↪shape[0]   #0.54%

#post-discharge 1 year mortalty rate for asian patients
asianMortRate = asian[asian['YearDeath']==1].shape[0]/asian.shape[0]       #6.
 ↪5%
notAsianMortRate = notAsian[notAsian['YearDeath']==1].shape[0]/notAsian.
 ↪shape[0]   #1.7%

#post-discharge 100 day mortalty rate for low income patients
```

```
lowIncMortRate100 = lowInc[lowInc['DEATH100']==1].shape[0]/lowInc.shape[0]
 ↪ #.49
notLowIncMortRate100 = notLowInc[notLowInc['DEATH100']==1].shape[0]/notLowInc.
 ↪shape[0]    #0.69%

#post-discharge 1 year mortalty rate for low income patients
lowIncMortRate = lowInc[lowInc['YearDeath']==1].shape[0]/lowInc.shape[0]
 ↪#2%
notLowIncMortRate = notLowInc[notLowInc['YearDeath']==1].shape[0]/notLowInc.
 ↪shape[0]    #2.1%
```

3.2 2.2 % of Minority, Low Income Patients who are Seniors

[26]:
```
minority = emergency[emergency['MINORITY']==1]
minorityLow = minority[minority['LOWINCOME']==1]
fatalMinority = minorityLow[minorityLow['YearDeath']==1]
fatalMinority30 = minorityLow[minorityLow['DEATH30']==1]
senior = fatalMinority[fatalMinority['SENIOR']==1]
senior30 = fatalMinority30[fatalMinority30['SENIOR']==1]

percentSenior = senior.shape[0]/fatalMinority.shape[0]  #57.1% of minority, low
 ↪income patients who died within 1 year of discharge were elderly
percentSenior30 = senior30.shape[0]/fatalMinority30.shape[0] #100% of minority,
 ↪low income patients who passed away within 30 days of discharge were elderly
```

3.3 2.3 Characteristics of patients who passed away w/in 1 year of discharge

[34]:
```
mortality = emergency[emergency['YearDeath']==1]

hispanicMort = mortality[mortality['HISPANIC']==1].shape[0]/mortality.shape[0]
 ↪#75% of patients who passed away w/in 1 year of discharge were hispanic
hispanicRatio = emergency[emergency['HISPANIC']==1].shape[0]/emergency.shape[0]
 ↪#75.5%

seniorMort = mortality[mortality['SENIOR']==1].shape[0]/mortality.shape[0]
 ↪#65% of patients who passed away w/in 1 year of discharge were seniors
seniorRatio = emergency[emergency['SENIOR']==1].shape[0]/emergency.shape[0]
 ↪#While 65% of YearDeath patients were seniors, only 18.5% of emergency
 ↪encounter patients are seniors

incomeRatio = emergency[emergency['LOWINCOME']==1].shape[0]/emergency.shape[0]
 ↪#41.4%
incomeMort = mortality[mortality['LOWINCOME']==1].shape[0]/mortality.shape[0]
 ↪#40%
```

```
#6.2% of encounters are Asian, while 20% of mortalities were Asian
asianRatio = emergency[emergency['ASIAN']==1].shape[0]/emergency.shape[0]
 ↪#6.2%
asianMort = mortality[mortality['ASIAN']==1].shape[0]/mortality.shape[0]
 ↪#20%

#8.7% of encounters were black, yet 0% of 1 year discharge mortalities were␣
 ↪black
blackRatio = emergency[emergency['BLACK']==1].shape[0]/emergency.shape[0]
 ↪#8.7%
blackMort = mortality[mortality['BLACK']==1].shape[0]/mortality.shape[0]
 ↪#0%

#90.4% of encounters were minority and even more (95%) of mortalities were␣
 ↪minority
minorityRatio = emergency[emergency['MINORITY']==1].shape[0]/emergency.shape[0]␣
 ↪#90.4%
minorityMort = mortality[mortality['MINORITY']==1].shape[0]/mortality.shape[0]
 ↪#95%
```

3.4 2.4 Characteristics of patients who passed away w/in 30 days of discharge

[41]:
```
mortality = emergency[emergency['DEATH30']==1]

#Similar proportion of mortalities to population of Hispanics
hispanicMort = mortality[mortality['HISPANIC']==1].shape[0]/mortality.shape[0]
 ↪#75% of patients who passed away w/in 1 year of discharge were hispanic
hispanicRatio = emergency[emergency['HISPANIC']==1].shape[0]/emergency.shape[0]␣
 ↪#75.5%

#While 18.5% of encounters were with seniors, 100% of 30day post discharge␣
 ↪mortalities were seniors
seniorMort = mortality[mortality['SENIOR']==1].shape[0]/mortality.shape[0]
 ↪#100%
seniorRatio = emergency[emergency['SENIOR']==1].shape[0]/emergency.shape[0]
 ↪#18.5%

#More than expected low income encounters passed away within 30 days of␣
 ↪discharge
incomeRatio = emergency[emergency['LOWINCOME']==1].shape[0]/emergency.shape[0]
 ↪#41.4%
incomeMort = mortality[mortality['LOWINCOME']==1].shape[0]/mortality.shape[0]
 ↪#50%

#More asian patient mortalities than expected (6.2% vs 25%)
```

```
asianRatio = emergency[emergency['ASIAN']==1].shape[0]/emergency.shape[0]
 ↪#6.2%
asianMort = mortality[mortality['ASIAN']==1].shape[0]/mortality.shape[0]
 ↪#25%

#Less black patient mortalities than expected (8.7% vs 0%)
blackRatio = emergency[emergency['BLACK']==1].shape[0]/emergency.shape[0]
 ↪#8.7%
blackMort = mortality[mortality['BLACK']==1].shape[0]/mortality.shape[0]
 ↪#0%

#All 30-day post discharge mortalities were minorities
minorityRatio = emergency[emergency['MINORITY']==1].shape[0]/emergency.shape[0]
 ↪#90.4%
minorityMort = mortality[mortality['MINORITY']==1].shape[0]/mortality.shape[0]
 ↪#100%
```

3.5 2.5 Characteristics of patients who passed away w/in 60 days of discharge

[44]:
```
mortality = emergency[emergency['DEATH60']==1]

hispanicMort = mortality[mortality['HISPANIC']==1].shape[0]/mortality.shape[0]
 ↪#75% of patients who passed away w/in 1 year of discharge were hispanic
hispanicRatio = emergency[emergency['HISPANIC']==1].shape[0]/emergency.shape[0]
 ↪#75.5%

#100% of 60day post discharge mortalitis were seniors, they only make up 18.5%
 ↪of the encounters
seniorMort = mortality[mortality['SENIOR']==1].shape[0]/mortality.shape[0]
 ↪#100% of patients who passed away w/in 1 year of discharge were seniors
seniorRatio = emergency[emergency['SENIOR']==1].shape[0]/emergency.shape[0]
 ↪#While 18.5% of YearDeath patients were seniors, only 31.3% of patients are
 ↪seniors

incomeRatio = emergency[emergency['LOWINCOME']==1].shape[0]/emergency.shape[0]
 ↪#41.4%
incomeMort = mortality[mortality['LOWINCOME']==1].shape[0]/mortality.shape[0]
 ↪#50%

#More asian patient mortalities than expected
asianRatio = emergency[emergency['ASIAN']==1].shape[0]/emergency.shape[0]
 ↪#6.2%
asianMort = mortality[mortality['ASIAN']==1].shape[0]/mortality.shape[0]
 ↪#25%
```

```
#Less black patient mortalities than expected
blackRatio = emergency[emergency['BLACK']==1].shape[0]/emergency.shape[0]
 ↪#8.7%
blackMort = mortality[mortality['BLACK']==1].shape[0]/mortality.shape[0]
 ↪#0%

#All 60-day mortalities were minorities
minorityRatio = emergency[emergency['MINORITY']==1].shape[0]/emergency.shape[0]
 ↪#90.4%
minorityMort = mortality[mortality['MINORITY']==1].shape[0]/mortality.shape[0]
 ↪#100%
```

3.6 3. Extra Information (based off of entire dataset (2008-16), not just emergencies)

[52]:
```
encounters = pd.read_csv("../data/encounters.csv")
encounters = encounters.drop('Unnamed: 0', axis=1)

#Clean encounters that are too old
encounters = encounters[encounters['DATE'].between('2008', '2017')]
encounters = encounters[encounters['CODE']!=308646001]

#last only has most recent encounters for each patient
last = encounters.drop_duplicates("PATIENT",keep='last')
last['DEATHDATE'] = last['PATIENT'].map(lambda x:
 ↪patientsCSV[patientsCSV["ID"]==x]['DEATHDATE'].values[0])

#Maps date of death for each patient to their encounter
encounters['DEATHDATE'] = encounters['ID'].map(lambda x: last.
 ↪loc[last['ID']==x]['DEATHDATE'].values[0] if x in last['ID'].unique() else 0)

#I noticed encounters.CSV was missing a lot of reason descriptions that
 ↪conditions.CSV had,
#so this for loop fills in missing information
for index,row in encounters.iterrows():
    if isinstance(row['REASONDESCRIPTION'],float):                    #only
 ↪applied to encounters that are not filled. will not refill a cell.
        date = row['DATE']
        find = conditionsCSV[conditionsCSV['PATIENT'] == row['PATIENT']]  #find
 ↪has all of rows that pertain to patient and reason narrows it to the
 ↪particular encounter
        reason = find[find['START'] == date]['DESCRIPTION']
 ↪#assuming someone doesn't visit the hospital in the same day for a different
 ↪reason
        code = find[find['START'] == date]['CODE']                    #also
 ↪assuming the hospital records all visit and didn't visit double visits if it
 ↪occured
```

```
        try:
            encounters['REASONDESCRIPTION'][index] = reason.values[0]
            encounters['REASONCODE'][index] = code.values[0]
        except:
            encounters['REASONDESCRIPTION'][index] = np.nan
            encounters['REASONCODE'][index] = np.nan

#Converted to date_time in order to do calculations on the dates
encounters['DATE']=pd.to_datetime(encounters['DATE'])
encounters['DEATHDATE']=pd.to_datetime(encounters['DEATHDATE'])

#Finds amount of time between emergency visit and death
encounters['DEATHDIFFERENCE'] = encounters['DEATHDATE']-encounters['DATE']
#Dummy variable for all patients who passed away 1 year within visit
encounters['YearDeath'] = encounters['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=365) else 0)
encounters.loc[(encounters['DEATHDIFFERENCE'] < datetime.
 ↪timedelta(days=0)),'DEATHDIFFERENCE'] = np.nan
#DEATH is a dummy variable for deaths post discharge, this removes deaths that␣
 ↪happen on day 0 since that wouldn't be considered post discharge.
encounters['DEATH'] = encounters['DEATHDIFFERENCE'].map(lambda x: 0 if x ==␣
 ↪datetime.timedelta(days=0) or x!=x else 1)
#I think this is more indicative of mortality related to post-discharge.
encounters['DEATH100'] = encounters['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=100) else 0)
encounters['DEATH60']  = encounters['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=60) else 0)
encounters['DEATH30']  = encounters['DEATHDIFFERENCE'].map(lambda x: 1 if␣
 ↪datetime.timedelta(days=0)<x<=datetime.timedelta(days=30) else 0)

LOW_LINE = 56763.2 #low income threshole
#Adding patient information for each encounter. For regression and analysis
encounters['RACE'] = encounters['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['RACE'].values[0])
encounters['AGE'] = encounters['PATIENT'].map(lambda x: int(patientsCSV.
 ↪loc[patientsCSV['ID']==x]['AGE'].values[0]))
encounters['HOMEZIP'] = encounters['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['HOMEZIP'].values[0])
encounters['INCOME'] = encounters['PATIENT'].map(lambda x: patientsCSV.
 ↪loc[patientsCSV['ID']==x]['INCOME'].values[0])

#Dummy variable for if patient is considered low income. Low income is 80% of␣
 ↪state median
encounters['LOWINCOME'] = encounters['INCOME'].map(lambda x: 1 if x < LOW_LINE␣
 ↪else 0)
```

```
#Replacing NA was necessary for the above lambda fxn, however NA is necessary␣
 ↪for regression since 0 will affect the regression
encounters['INCOME'] = encounters['INCOME'].replace(-1,np.nan)

encounters['SENIOR'] = encounters['AGE'].map(lambda x: 1 if x>=65 else 0)
encounters['MINORITY'] = encounters['RACE'].map(lambda x: 1 if x=='asian' or␣
 ↪x=='hispanic' or x=='black' or x=='black or african american' else 0)

#Matches patient BMI to patient and encounter
BMI = observationsCSV[observationsCSV['DESCRIPTION']=='Body Mass Index']
BMI["VALUE"] = BMI['VALUE'].astype(float)
patients = BMI['PATIENT'].unique()
encounters['BMI'] = encounters['PATIENT'].map(lambda x:␣
 ↪BMI[BMI['PATIENT']==x]['VALUE'].values[0] if x in patients else -1)

#Homeless dummy variable
homeless = observationsCSV[observationsCSV['DESCRIPTION']=='Housing status']
patients = homeless['PATIENT'].unique()
encounters['HOMELESS'] = encounters['PATIENT'].map(lambda x: 1 if x in patients␣
 ↪else 0)        #Patients with a 'Housing status' description were all␣
 ↪homeless

#Variable for Diastolic Blood Pressure
DBP = observationsCSV[observationsCSV['DESCRIPTION']=='Diastolic Blood␣
 ↪Pressure']
patient = DBP['PATIENT'].unique()
encounters['DIASTOLICBP'] = encounters['PATIENT'].map(lambda x:␣
 ↪float(DBP[DBP['PATIENT']==x]['VALUE'].values[0]) if x in patient else -1)
encounters['BADBP'] = encounters['DIASTOLICBP'].map(lambda x: 1 if x > 80 else␣
 ↪0)

#Variable for Calcium levels
Calcium = observationsCSV[observationsCSV['DESCRIPTION']=='Calcium']
patient = Calcium['PATIENT'].unique()
encounters['CALCIUM'] = encounters['PATIENT'].map(lambda x:␣
 ↪float(Calcium[Calcium['PATIENT']==x]['VALUE'].values[0]) if x in patient␣
 ↪else -1)

#Finds patients who got the flu shot within 6 months of their most recent visit␣
 ↪to the hospital.
encounters['FLU']=None        #1 if patient recieved flu shot w/in 6 months of␣
 ↪visit, 0 otherwise
encounters['FLU_DIFF']=None   #difference between time of flu shot and time of␣
 ↪visit, if 0 days then they visited the hospital for the flu shot/happened to␣
 ↪get it
```

```
flu = immunizationsCSV[immunizationsCSV['CODE'] == 140]   #CODE 140 is the code␣
 ↪for influenza immunization                                                  ␣
 ↪
patients = flu['PATIENT'].unique()

for index, row in encounters.iterrows():
    checked = 0     #keeps track of if the inner for loop found a matching␣
 ↪immunization
    patientID = row['PATIENT']
    if patientID in patients:    #if patient is in the flu dataset, they had an␣
 ↪immunization
        date = flu[flu['PATIENT']==patientID]['DATE']   #list of all dates they␣
 ↪recieved influenza shot
        for d in date:
            d = datetime.datetime.strptime(d, '%Y-%m-%d')
            diff = row['DATE'] - d
            if datetime.timedelta(days=0) <= diff < datetime.
 ↪timedelta(days=183):    #an influenza shot lasts around 6 months/183 days
                checked = 1
                encounters['FLU'][index]=1
                encounters['FLU_DIFF'][index]=diff    #can help us determine␣
 ↪cause of visit if there is none (i.e. if =0 days, then the visit was for the␣
 ↪shot)
        if checked == 0:
            encounters['FLU'][index]=0
            encounters['FLU_DIFF'][index]=-1
    else:
            encounters['FLU'][index]=0
            encounters['FLU_DIFF'][index]=-1

#I found that a lot of the encounter dates in encounters.csv coincided with
#dates (found in immunizations.csv) that the patient got a flu shot. They all
#coincided with Outpatient Encounters without ReasonDescriptions. So I assume
#the reason they went was for the flu shot and not some hidden sickness/
 ↪emergency
```

```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:27:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:28:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:24:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:25:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:63:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:100:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:101:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:103:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:104:
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:106:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:107:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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

[51]:
```
encounters.to_csv("En.csv")
```

46% of senior patients who passed away w/in 30 days of their visit did not visit for care. They visited for a flu shot.

39% of senior patients who passed away w/in 30 days of their visit passed away due to Pneumonia/Viral Illness.

[56]:
```
seniorPatients = encounters[encounters['SENIOR']==1]
seniorPatients = seniorPatients.drop_duplicates('PATIENT',keep='last')
totalSeniors = seniorPatients.shape[0]
Death30 = seniorPatients[seniorPatients['DEATH30']==1]

print(str(Death30[Death30['FLU_DIFF']==datetime.timedelta(days=0)].shape[0]/
 ↪Death30.shape[0] * 100) + "% of senior patients who passed away within 30␣
 ↪days of their hospital visit visited the hospital for a flu shot")
print(str((Death30[Death30['REASONDESCRIPTION']=='Pneumonia'].
 ↪shape[0]+Death30[Death30['REASONDESCRIPTION']=='Viral sinusitis (disorder)'].
 ↪shape[0])/Death30.shape[0]*100) + "% of senior patients who passed away␣
 ↪within 30 days of visit, visited for Pneumonia/Viral Illness")

Death30
```

```
46.15384615384615% of senior patients who passed away within 30 days of their
hospital visit visited the hospital for a flu shot
38.46153846153847% of senior patients who passed away within 30 days of visit,
visited for Pneumonia/Viral Illness
```

[56]:
```
                                      ID       DATE  \
 669    46b94251-1039-42a9-ae54-3ca75f25e75c 2012-07-07
```

```
1924   d8ed4921-783c-4e10-bbc9-e8dbd5913c20 2013-04-09
2149   b297ac62-23e4-4fe7-8f72-8277722eaf52 2009-02-25
3407   ca457bec-18f4-4ce0-81f9-0800b346428d 2016-07-13
4180   b8a9dcac-1b61-49ae-9e06-4c793be6e29d 2009-04-16
5761   508aebb9-5b6c-42fc-990d-3f399b7ba386 2011-11-02
6063   e7d1b9ab-a5c4-4d0b-ac86-f6582496195e 2014-01-25
7120   977f504f-1ec5-45c7-aa16-b0010a62745f 2015-02-02
10994  bb71193a-8aa3-499a-8d18-223e50978588 2014-09-22
11942  beb50d63-b424-4b33-8315-9c1e4a7ad479 2016-07-02
13148  2925ea65-0f13-41e7-9e5e-d6165167f831 2015-12-10
13285  d9a9ddf9-5a40-49b1-9a0d-693bc98e1431 2009-04-26
20345  736e13f7-4dc5-4e49-8e0e-2e28115fca12 2013-01-05

                                     PATIENT       CODE  \
669    c383f814-9ee4-4ca8-a0ff-ee1369f8d2ee  185349003
1924   6d66c6a6-8b0d-47c0-a573-b78ec8ceec63  185349003
2149   481373a7-79df-429d-b3da-e971116c1df1  185349003
3407   bc6fbe62-116e-424f-943c-bae29fa9f319  185345009
4180   bc2f58a2-ea29-45e7-93c4-99c03cbfcc4c  185349003
5761   90e7f959-f0ec-4e9e-bce8-ca0a7d0e6a1f  185349003
6063   cdbe1927-b954-4f37-9294-ac0ca277d147   34285007
7120   8b3cf0c5-affd-4cbb-a397-e63484ba4d47   34285007
10994  d871e7f4-b18d-4e4a-8fbb-f1d2c101efc5  185349003
11942  e6e63adb-9c2a-4f7a-ade9-3b89ff862a03   34285007
13148  af7c4d94-3524-4c7b-b8c0-fb774c9e5b6d   34285007
13285  e3c8d2c8-570d-4c69-a81e-bd88d6c6b0ca  185347001
20345  2bbaedc8-d765-4950-bfc2-32b16713934c  185349003

                   DESCRIPTION   REASONCODE           REASONDESCRIPTION  \
669        Outpatient Encounter          NaN                         NaN
1924       Outpatient Encounter          NaN                         NaN
2149       Outpatient Encounter          NaN                         NaN
3407      Encounter for symptom  444814009.0  Viral sinusitis (disorder)
4180       Outpatient Encounter          NaN                         NaN
5761       Outpatient Encounter          NaN                         NaN
6063         Hospital admission  233604007.0                   Pneumonia
7120         Hospital admission  233604007.0                   Pneumonia
10994      Outpatient Encounter          NaN                         NaN
11942        Hospital admission  233604007.0                   Pneumonia
13148        Hospital admission  233604007.0                   Pneumonia
13285     Encounter for problem          NaN                         NaN
20345  Encounter for 'check-up'   65966004.0        Fracture of forearm

       DEATHDATE DEATHDIFFERENCE  YearDeath  ...  LOWINCOME  SENIOR  MINORITY  \
669   2012-08-02         26 days          1  ...          0       1         1
1924  2013-05-01         22 days          1  ...          0       1         1
2149  2009-03-18         21 days          1  ...          1       1         1
```

```
3407   2016-07-22          9 days          1  ...          1       1         0
4180   2009-05-11         25 days          1  ...          0       1         0
5761   2011-11-12         10 days          1  ...          0       1         1
6063   2014-02-06         12 days          1  ...          1       1         1
7120   2015-02-15         13 days          1  ...          0       1         1
10994  2014-10-01          9 days          1  ...          0       1         1
11942  2016-07-10          8 days          1  ...          0       1         1
13148  2015-12-21         11 days          1  ...          1       1         1
13285  2009-05-23         27 days          1  ...          0       1         1
20345  2013-01-25         20 days          1  ...          0       1         1

          BMI HOMELESS  DIASTOLICBP BADBP  CALCIUM  FLU          FLU_DIFF
669     37.40        0         72.0     0    -1.00    1    0 days 00:00:00
1924    39.06        0        116.0     1     9.45    1    0 days 00:00:00
2149    30.24        0        111.0     1    10.09    1    0 days 00:00:00
3407    36.44        0        114.0     1     9.19    0                -1
4180    35.59        0        111.0     1     9.51    1    0 days 00:00:00
5761    31.78        0         75.0     0     9.30    1    0 days 00:00:00
6063    25.15        0         80.0     0     8.65    1  121 days 00:00:00
7120    37.09        0         97.0     1     9.27    1  101 days 00:00:00
10994   25.89        0         71.0     0    -1.00    1    0 days 00:00:00
11942   35.13        0        111.0     1     9.04    0                -1
13148   35.71        0         77.0     0    -1.00    0                -1
13285   41.13        0         82.0     1    10.04    0                -1
20345   23.84        0         72.0     0     9.09    1   63 days 00:00:00

[13 rows x 28 columns]
```

57% of senior patients who passed away w/in 60 days of their hospital visit visited the hospital for a flu shot. An additional 32% passed away after visiting for Pneumonia/Viral Sinusitis.

[59]:
```
Death60 = seniorPatients[seniorPatients['DEATH60']==1]

print(str(Death60[Death60['FLU_DIFF']==datetime.timedelta(days=0)].shape[0]/
 ↪Death60.shape[0] * 100) + "% of senior patients who passed away within 60␣
 ↪days of their hospital visit visited the hospital for a flu shot")
print(str(6/Death60.shape[0]*100) +"% of senior patients passed away after␣
 ↪visiting the hospital for a virus")
Death60
```

```
57.89473684210527% of senior patients who passed away within 60 days of their
hospital visit visited the hospital for a flu shot
31.57894736842105% of senior patients passed away after visiting the hospital
for a virus
```

[59]:
```
                                      ID       DATE  \
669    46b94251-1039-42a9-ae54-3ca75f25e75c 2012-07-07
```

```
1924   d8ed4921-783c-4e10-bbc9-e8dbd5913c20 2013-04-09
2149   b297ac62-23e4-4fe7-8f72-8277722eaf52 2009-02-25
3334   b57f2d3d-e9fc-4d21-be1e-e9b2f84fc714 2014-06-15
3407   ca457bec-18f4-4ce0-81f9-0800b346428d 2016-07-13
4180   b8a9dcac-1b61-49ae-9e06-4c793be6e29d 2009-04-16
4530   12b94e93-09fe-4dcc-98d0-ac67ce07e674 2009-04-03
5761   508aebb9-5b6c-42fc-990d-3f399b7ba386 2011-11-02
6063   e7d1b9ab-a5c4-4d0b-ac86-f6582496195e 2014-01-25
6821   04e0b540-cfaf-4875-895f-8cdb3235a48e 2008-06-20
6851   4f64dddf-122d-4da0-9b4b-eeaf7e1e0f83 2009-02-11
7120   977f504f-1ec5-45c7-aa16-b0010a62745f 2015-02-02
9856   62662cef-3923-43ac-a677-ca370e852dc0 2013-12-15
10994  bb71193a-8aa3-499a-8d18-223e50978588 2014-09-22
11942  beb50d63-b424-4b33-8315-9c1e4a7ad479 2016-07-02
12676  956178d5-c90e-4f0b-8713-263fef438716 2016-03-26
13148  2925ea65-0f13-41e7-9e5e-d6165167f831 2015-12-10
13285  d9a9ddf9-5a40-49b1-9a0d-693bc98e1431 2009-04-26
20345  736e13f7-4dc5-4e49-8e0e-2e28115fca12 2013-01-05

                                      PATIENT       CODE  \
669    c383f814-9ee4-4ca8-a0ff-ee1369f8d2ee  185349003
1924   6d66c6a6-8b0d-47c0-a573-b78ec8ceec63  185349003
2149   481373a7-79df-429d-b3da-e971116c1df1  185349003
3334   18a225df-378e-419a-8aff-bc03ab654103  185349003
3407   bc6fbe62-116e-424f-943c-bae29fa9f319  185345009
4180   bc2f58a2-ea29-45e7-93c4-99c03cbfcc4c  185349003
4530   a77d3e00-6f1a-4bfd-9333-3439be45b15f  185349003
5761   90e7f959-f0ec-4e9e-bce8-ca0a7d0e6a1f  185349003
6063   cdbe1927-b954-4f37-9294-ac0ca277d147   34285007
6821   6ddea441-48d4-4ef1-8213-04ca25c78497  185349003
6851   6f45458c-8658-449d-a583-effa43cbd8a8  185345009
7120   8b3cf0c5-affd-4cbb-a397-e63484ba4d47   34285007
9856   28627be3-81b6-4917-bd4a-d2af28c5ac5a  185349003
10994  d871e7f4-b18d-4e4a-8fbb-f1d2c101efc5  185349003
11942  e6e63adb-9c2a-4f7a-ade9-3b89ff862a03   34285007
12676  a83957db-3e38-4fd4-a4bb-aca5f2090b80  185349003
13148  af7c4d94-3524-4c7b-b8c0-fb774c9e5b6d   34285007
13285  e3c8d2c8-570d-4c69-a81e-bd88d6c6b0ca  185347001
20345  2bbaedc8-d765-4950-bfc2-32b16713934c  185349003

                   DESCRIPTION   REASONCODE  \
669        Outpatient Encounter          NaN
1924       Outpatient Encounter          NaN
2149       Outpatient Encounter          NaN
3334       Outpatient Encounter          NaN
3407      Encounter for symptom  444814009.0
4180       Outpatient Encounter          NaN
```

```
4530       Outpatient Encounter          NaN
5761       Outpatient Encounter          NaN
6063         Hospital admission  233604007.0
6821       Outpatient Encounter          NaN
6851      Encounter for symptom  195662009.0
7120         Hospital admission  233604007.0
9856       Outpatient Encounter          NaN
10994      Outpatient Encounter          NaN
11942        Hospital admission  233604007.0
12676      Outpatient Encounter          NaN
13148        Hospital admission  233604007.0
13285     Encounter for problem          NaN
20345  Encounter for 'check-up'   65966004.0

                      REASONDESCRIPTION  DEATHDATE DEATHDIFFERENCE  \
669                                 NaN 2012-08-02         26 days
1924                                NaN 2013-05-01         22 days
2149                                NaN 2009-03-18         21 days
3334                                NaN 2014-07-26         41 days
3407          Viral sinusitis (disorder) 2016-07-22          9 days
4180                                NaN 2009-05-11         25 days
4530                                NaN 2009-05-15         42 days
5761                                NaN 2011-11-12         10 days
6063                          Pneumonia 2014-02-06         12 days
6821                                NaN 2008-07-24         34 days
6851   Acute viral pharyngitis (disorder) 2009-03-21         38 days
7120                          Pneumonia 2015-02-15         13 days
9856                                NaN 2014-01-23         39 days
10994                               NaN 2014-10-01          9 days
11942                         Pneumonia 2016-07-10          8 days
12676                               NaN 2016-05-24         59 days
13148                         Pneumonia 2015-12-21         11 days
13285                               NaN 2009-05-23         27 days
20345               Fracture of forearm 2013-01-25         20 days

       YearDeath  ...  LOWINCOME  SENIOR  MINORITY    BMI HOMELESS  \
669            1  ...          0       1         1  37.40        0
1924           1  ...          0       1         1  39.06        0
2149           1  ...          1       1         1  30.24        0
3334           1  ...          0       1         0  25.35        0
3407           1  ...          1       1         0  36.44        0
4180           1  ...          0       1         0  35.59        0
4530           1  ...          0       1         1  31.78        0
5761           1  ...          0       1         1  31.78        0
6063           1  ...          1       1         1  25.15        0
6821           1  ...          0       1         1  27.58        0
6851           1  ...          0       1         1  30.56        0
```

```
7120           1  ...           0       1         1  37.09        0
9856           1  ...           0       1         1  39.09        0
10994          1  ...           0       1         1  25.89        0
11942          1  ...           0       1         1  35.13        0
12676          1  ...           0       1         1  38.02        0
13148          1  ...           1       1         1  35.71        0
13285          1  ...           0       1         1  41.13        0
20345          1  ...           0       1         1  23.84        0

       DIASTOLICBP BADBP  CALCIUM  FLU           FLU_DIFF
669           72.0     0    -1.00    1    0 days 00:00:00
1924         116.0     1     9.45    1    0 days 00:00:00
2149         111.0     1    10.09    1    0 days 00:00:00
3334          87.0     1     8.55    1    0 days 00:00:00
3407         114.0     1     9.19    0                -1
4180         111.0     1     9.51    1    0 days 00:00:00
4530          86.0     1    -1.00    1    0 days 00:00:00
5761          75.0     0     9.30    1    0 days 00:00:00
6063          80.0     0     8.65    1  121 days 00:00:00
6821          88.0     1    -1.00    1    0 days 00:00:00
6851          91.0     1     8.60    1   33 days 00:00:00
7120          97.0     1     9.27    1  101 days 00:00:00
9856          78.0     0     9.84    1    0 days 00:00:00
10994         71.0     0    -1.00    1    0 days 00:00:00
11942        111.0     1     9.04    0                -1
12676         85.0     1    -1.00    1    0 days 00:00:00
13148         77.0     0    -1.00    0                -1
13285         82.0     1    10.04    0                -1
20345         72.0     0     9.09    1   63 days 00:00:00

[19 rows x 28 columns]
```

Finding areas/zipcodes that might benefit most/first from the program

[501]:
```
yeardeath = encounters[encounters['DEATH']==1]
yeardeath = yeardeath.drop_duplicates('PATIENT',keep='last')

locations = yeardeath.groupby('HOMEZIP').count()
locations.sort_values('PATIENT').tail(10)

emergency_drop = encounters.drop_duplicates('PATIENT', keep='last')

m = emergency_drop[emergency_drop['MINORITY']==1]
m = m.groupby("HOMEZIP").count()
top = m.sort_values('PATIENT').tail(10)
```

83.6% of YearDeath==1 seniors were not w/in the healthy BMI range for seniors.

[60]:
```
yearSeniors = seniorPatients[seniorPatients['YearDeath']==1]

totalYrSeniors = yearSeniors.shape[0]
under = yearSeniors[yearSeniors['BMI']<25].shape[0]
over =yearSeniors[yearSeniors['BMI']>27].shape[0]
print(str((under+over)/totalYrSeniors *100) +"% of 1 year mortality senior␣
 ↪patients were not within the healthy BMI range for seniors")
print(str(under/totalYrSeniors*100)+"% of 1 year mortality senior patients were␣
 ↪under the healthy BMI range for seniors")
print(str(over/totalYrSeniors*100)+"% of 1 year mortality senior patients were␣
 ↪over the healthy BMI range for seniors")


yearSeniors = seniorPatients[seniorPatients['YearDeath']==1]

totalYrSeniors = yearSeniors.shape[0]
under = yearSeniors[yearSeniors['BMI']<25].shape[0]
over =yearSeniors[yearSeniors['BMI']>27].shape[0]
print(str((under+over)/totalYrSeniors *100) +"% of 1 year mortality senior␣
 ↪patients were not within the healthy BMI range for seniors")
print(str(under/totalYrSeniors*100)+"% of 1 year mortality senior patients were␣
 ↪under the healthy BMI range for seniors")
print(str(over/totalYrSeniors*100)+"% of 1 year mortality senior patients were␣
 ↪over the healthy BMI range for seniors")
```

```
83.56164383561644% of 1 year mortality senior patients were not within the
healthy BMI range for seniors
17.80821917808219% of 1 year mortality senior patients were under the healthy
BMI range for seniors
65.75342465753424% of 1 year mortality senior patients were over the healthy BMI
range for seniors
83.56164383561644% of 1 year mortality senior patients were not within the
healthy BMI range for seniors
17.80821917808219% of 1 year mortality senior patients were under the healthy
BMI range for seniors
65.75342465753424% of 1 year mortality senior patients were over the healthy BMI
range for seniors
```

66.9% of YearDeath==1 seniors did not have normal diastolic blood pressure.

[62]:
```
print(str(yearSeniors[yearSeniors['BADBP']==1].shape[0]/totalYrSeniors*100)+"%␣
 ↪of 1 year mortality seniors did not have normal diastolic blood pressure")
print(str(seniorPatients[seniorPatients['BADBP']==1].shape[0]/
 ↪totalSeniors*100)+"% of the entire senior patient population does not have␣
 ↪normal diastolic blood pressure")
```

```
print(str(yearSeniors[yearSeniors['DIASTOLICBP']>=90].shape[0]/
 ↪totalYrSeniors*100) +"% of 1 year mortality seniors had high diastolic blood␣
 ↪pressure")
print(str(seniorPatients[seniorPatients['DIASTOLICBP']>=90].shape[0]/
 ↪totalSeniors*100) +"% of the entire senior patient population has high␣
 ↪diastolic blood pressure")
```

```
64.38356164383562% of 1 year mortality seniors did not have normal diastolic
blood pressure
62.447257383966246% of the entire senior patient population does not have normal
diastolic blood pressure
35.61643835616438% of 1 year mortality seniors had high diastolic blood pressure
33.755274261603375% of the entire senior patient population has high diastolic
blood pressure
```

11/15 YearDeaths for minority, senior patients were related to curable viruses or injuries sustained from falling.

[70]:
```
e = encounters[encounters['YearDeath']==1]
g = e[e['MINORITY']==1]
g = g[g['SENIOR']==1]
h = g.drop_duplicates('PATIENT',keep='last')
g = h.groupby('REASONDESCRIPTION').count()

print("8/14 of known reasons for passing was due to curable viruses and␣
 ↪sicknesses")
print("3/14 of known reasons were related to injuries from falling")
```

```
8/14 of known reasons for passing was due to curable viruses and sicknesses
3/14 of known reasons were related to injuries from falling
```

The most recent hospital visit for 9/16 senior, minority, patients who passed away w/in 60 days of their visits went to the hospital for a flu shot

[74]:
```
e = encounters[encounters['DEATH60']==1]
g = e[e['MINORITY']==1]
g = g[g['SENIOR']==1]
h = g.drop_duplicates('PATIENT',keep='last')
h.groupby('REASONDESCRIPTION').count()
print("9/16 senior, minority, patients who passed away w/in 30 days visited the␣
 ↪hospital for flu shots")
```

```
9/16 senior, minority, patients who passed away w/in 30 days visited the
hospital for flu shots
```

92.8% of Outpatient Encounters were influenza immunizations/included an influenza immunization

[76]:
```
e_drop = encounters.drop_duplicates("PATIENT", keep='last')
outpatient = e_drop[e_drop['CODE']==185349003]
outFluSize = outpatient[outpatient["FLU_DIFF"]==datetime.timedelta(days=0)].
 ↪shape[0]
print(str(outFluSize/outpatient.shape[0]*100)+'% of Outpatient Encounters were/
 ↪included an influenza immunization')
```

```
92.81767955801105% of Outpatient Encounters were/included an influenza
immunization
```

Avg. Household Income for a Homeless Patient = 76,309.20. That's 34.4% more than the low income threshold.

[95]:
```
homeless = observationsCSV[observationsCSV['DESCRIPTION']=='Housing status']

homeless = homeless.drop_duplicates('PATIENT', keep='last')
homeless['INCOME'] = homeless['PATIENT'].map(lambda x:␣
 ↪patientsCSV[patientsCSV['ID']==x]["INCOME"].values[0])
avg = homeless['INCOME'].sum()/25


print('The average estimated household income for a homeless patient is $'␣
 ↪+str(avg) +' which is ' +str((avg-LOW_LINE)/LOW_LINE*100)+"% more than the␣
 ↪low income threshold")
```

```
The average estimated household income for a homeless patient is $76309.28 which
is 34.43442230177298% more than the low income threshold
```

[95]:

	Unnamed: 0	DATE	PATIENT	\
202	202	2013-12-16	96b24072-e1fe-49cd-a22a-6dfb92c3994c	
823	823	2014-07-04	38364c57-80ce-4749-aed6-878cdff95379	
1465	1465	2007-11-04	abf99602-7cb7-49db-9251-0499968c472f	
3986	3986	2008-05-20	9943efb3-15d2-4225-ba54-0cb4d4478f6a	
4256	4256	2015-02-21	41ec5505-df3f-4e0c-9e4c-2cc7e152031e	
10881	10881	2013-04-16	63b2d7b3-c597-4ede-b477-9b0515799cc8	
12216	12216	2010-12-29	bc8af009-ec20-409e-813d-fba6a952a2ab	
16128	16128	2016-11-29	24de5840-c471-4436-93ef-3e5b3e905353	
16216	16216	2009-07-28	d2e9efc1-431e-4736-8823-e86c16dcb141	
17213	17213	2011-11-24	af9cd97a-f11a-469a-8781-84e88ac83774	
20485	20485	2017-02-14	9016a427-917f-4729-ac9c-ef865f17f4d7	
20737	20737	2010-10-24	186ce071-096a-4eed-9c61-dd773ccf3830	
22775	22775	2014-07-30	932563cc-7bc0-46b3-8b36-d687202f0705	
24061	24061	2014-09-05	86082f75-6af5-4df0-a8f7-effd9bd68a58	
27000	27000	2013-05-13	dbd3794c-5ecc-47bc-8d6a-c8d60fbe9d31	

```
27506       27506  2009-08-09  4243c357-7791-4eac-a6db-4d18f561c910
29256       29256  2012-12-27  6e8b250f-3ce4-442c-96d7-08786c74a966
31687       31687  2017-10-03  c95416a1-5c72-4588-9dd4-ff1959ef7687
33982       33982  2013-02-17  260e1f37-18b0-4f2b-9a2c-78645485bf1e
35790       35790  2008-11-14  0ed686b8-877a-430b-a1a6-5d0be2d6bebc
37416       37416  2012-01-13  de5deae8-3379-4bd1-8cf7-e48b5e61212b
48080       48080  2014-04-19  fae141ca-fa4e-487d-8562-38a1d4b27ad8
48128       48128  2009-07-22  1b74791c-4223-4c9c-81ab-c8427e07484e
48218       48218  2008-03-08  16014a34-8dee-4402-89da-edf7884ae808
53557       53557  2008-01-28  1db96fae-3f95-4e2d-9e44-49b36d31d734
59809       59809  2012-11-18  ab163027-f86b-4571-8fff-2fd20391001b
62911       62911  2016-08-29  ecdfa659-f3d3-44d2-870c-a6c6a7fc68e1
65472       65472  2012-07-16  94f98532-269c-4a9f-8a34-186d80a16a8e
66999       66999  2008-01-25  7a36dd06-0363-4a3e-9d08-47feb4b2c3ce
67676       67676  2017-07-14  13cda82a-3cfe-4ad7-9c04-5b324f148340
69415       69415  2017-07-14  06c36d78-53eb-45e6-936e-96145a959c14
69893       69893  2016-10-28  b5fcea3e-2248-4045-945d-531fa938541a
75427       75427  2015-05-12  f9b7b2f0-146d-48d4-ba99-49ba9db0a02f
76058       76058  2014-11-22  f7ccb69b-54e3-4969-9d8c-4f75d481256b

                                    ENCOUNTER     CODE     DESCRIPTION  \
202    ab703308-e57a-4aa9-b8ef-d0fb07b60ab4  71802-3  Housing status
823    ddde7e19-206f-455f-99c0-98f73def8965  71802-3  Housing status
1465   fa874ebb-3cf7-4140-a168-806c98720029  71802-3  Housing status
3986   a1b7f3f5-cbac-4b0a-a371-65f2ec0a4f1f  71802-3  Housing status
4256   931ff5a5-7e2b-4f98-8f26-23aa8584b9af  71802-3  Housing status
10881  eea684d8-febf-4099-8533-a56663c87960  71802-3  Housing status
12216  c2ae4110-443b-4cca-baf4-775ffd4689be  71802-3  Housing status
16128  691a9847-3c9e-457b-9ac6-6bf1874898bc  71802-3  Housing status
16216  c93595bd-c911-4c10-8255-4466b34b4b78  71802-3  Housing status
17213  bbc7bd6d-3897-4f93-a1e0-abfa645ec208  71802-3  Housing status
20485  5bb890bb-30f3-4131-9eef-73f346bf0889  71802-3  Housing status
20737  558cf428-6f6a-48b4-8bf3-8e39a67d4d08  71802-3  Housing status
22775  32521ff1-da2d-4cd5-b35f-580e047caf3f  71802-3  Housing status
24061  59ed3e8e-da68-4e6d-974e-0cb62739e1f9  71802-3  Housing status
27000  d9168ca9-474f-4e10-a5c2-60993ce81d08  71802-3  Housing status
27506  57bd77e0-66e2-43e5-a3be-50247052cd38  71802-3  Housing status
29256  ed270417-914c-42de-a9bc-74d06a6919d9  71802-3  Housing status
31687  ed98e33d-75d3-4f6f-8fb8-dfc16ee74f36  71802-3  Housing status
33982  4f008b65-00aa-4260-9f31-f508d6d10530  71802-3  Housing status
35790  0bbf3b0a-f231-428a-b0ff-ed0680215673  71802-3  Housing status
37416  b9556eac-4bfb-4591-b275-bbf0cbaa6466  71802-3  Housing status
48080  86cf252c-8dca-41aa-b677-c2254038f79c  71802-3  Housing status
48128  50cba3d2-4d40-4f2b-8672-1dc2d75c9606  71802-3  Housing status
48218  a9841df6-f3d4-4228-bc36-68dc786a7237  71802-3  Housing status
53557  22aae102-fd7c-4c4f-8b8e-849ac30e70e8  71802-3  Housing status
59809  08f2cf51-98cc-407c-a33c-0b750d1228ca  71802-3  Housing status
```

```
62911  f2d50f79-c5c3-4859-9945-6ac5b32b806a  71802-3  Housing status
65472  642d5994-0807-4b70-98b7-8cbb8a27825a  71802-3  Housing status
66999  ea661a39-ffd4-440a-925b-a7e3d4145516  71802-3  Housing status
67676  0a6bd323-63d3-4bc9-84aa-bf8c65e87867  71802-3  Housing status
69415  c42832f2-3d9f-46e2-bf63-0df8f81fa556  71802-3  Housing status
69893  51229a5e-b92a-4266-9141-ff8217df87c1  71802-3  Housing status
75427  286859b9-7a6d-44d4-a2a4-1a9c07619988  71802-3  Housing status
76058  d4f5951e-0047-4613-9d63-8954c49518c5  71802-3  Housing status
```

	VALUE	UNITS	INCOME
202	Patient is homeless	{nominal}	102219
823	Patient is homeless	{nominal}	60755
1465	Patient is homeless	{nominal}	-1
3986	Patient is homeless	{nominal}	102301
4256	Patient is homeless	{nominal}	-1
10881	Patient is homeless	{nominal}	56714
12216	Patient is homeless	{nominal}	-1
16128	Patient is homeless	{nominal}	47115
16216	Patient is homeless	{nominal}	-1
17213	Patient is homeless	{nominal}	108815
20485	Patient is homeless	{nominal}	71065
20737	Patient is homeless	{nominal}	70574
22775	Patient is homeless	{nominal}	49083
24061	Patient is homeless	{nominal}	-1
27000	Patient is homeless	{nominal}	73839
27506	Patient is homeless	{nominal}	40876
29256	Patient is homeless	{nominal}	51229
31687	Patient is homeless	{nominal}	-1
33982	Patient is homeless	{nominal}	110043
35790	Patient is homeless	{nominal}	88958
37416	Patient is homeless	{nominal}	70873
48080	Patient is homeless	{nominal}	102090
48128	Patient is homeless	{nominal}	56577
48218	Patient is homeless	{nominal}	62114
53557	Patient is homeless	{nominal}	83283
59809	Patient is homeless	{nominal}	91273
62911	Patient is homeless	{nominal}	61527
65472	Patient is homeless	{nominal}	104600
66999	Patient is homeless	{nominal}	50044
67676	Patient is homeless	{nominal}	130739
69415	Patient is homeless	{nominal}	-1
69893	Patient is homeless	{nominal}	61035
75427	Patient is homeless	{nominal}	-1
76058	Patient is homeless	{nominal}	-1

[508]:
```
emergency['BMIUp'] = emergency['BMI'].map(lambda x: 1 if x > 27 else 0)
emergency['BMIDown'] = emergency['BMI'].map(lambda x: 1 if x < 25 else 0)
```

```
emergency['BMIOverall'] = emergency['BMI'].map(lambda x: 1 if x>27 or x<25 else␣
↪0)
```

I am a computer science major and have experience coding in Python and Javascript and using Stata. I am currently in Econometrics and becoming better at analyzing data with statistics and equations (vs. how I usually analyze data by questioning it, comparing it to the world around me, sometimes following my gut, asking for outside opinions. . .)