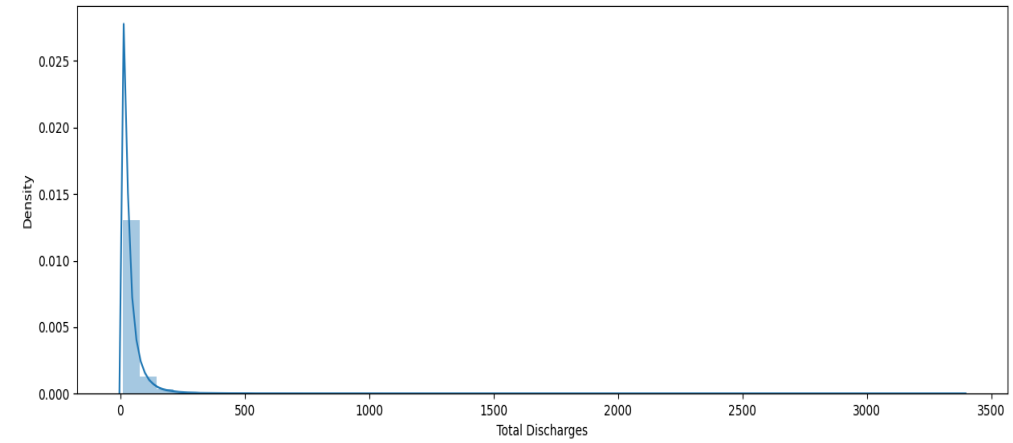
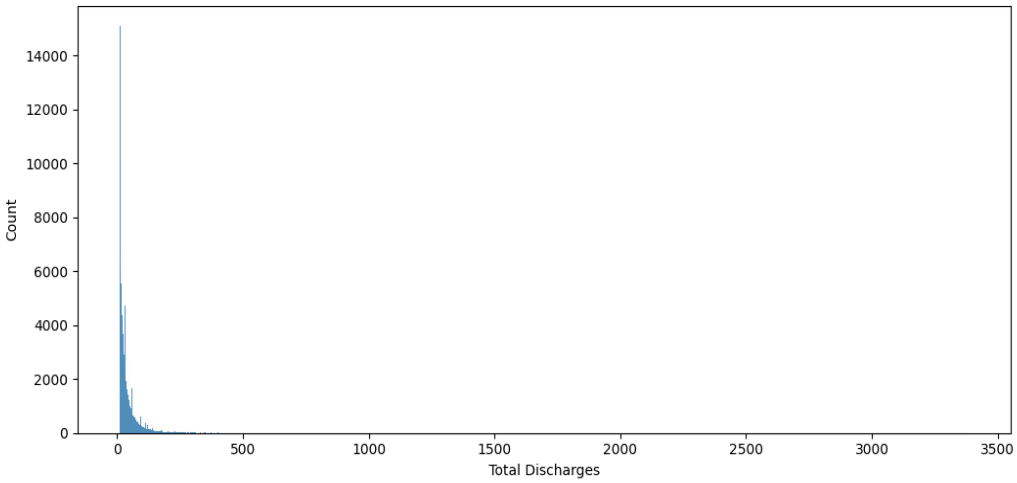
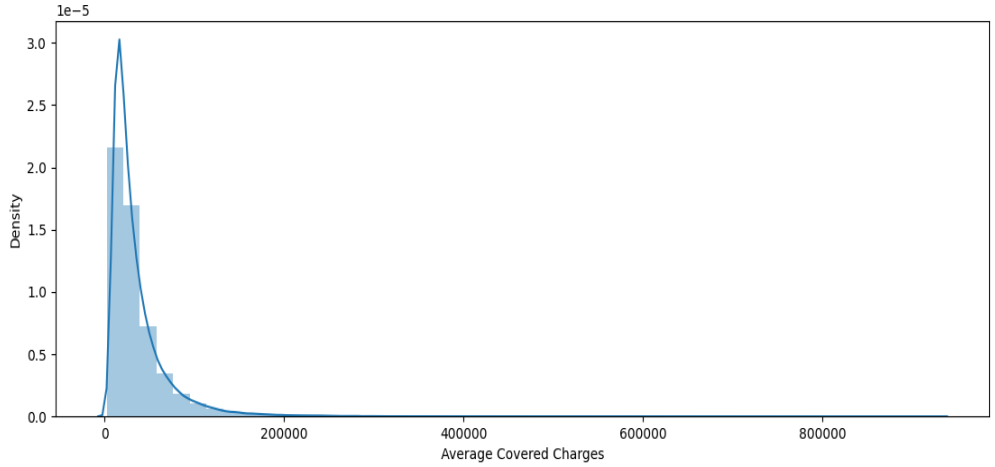
SKKU 3049-41: Project 1 2016312761 Hyuksoo Yeo

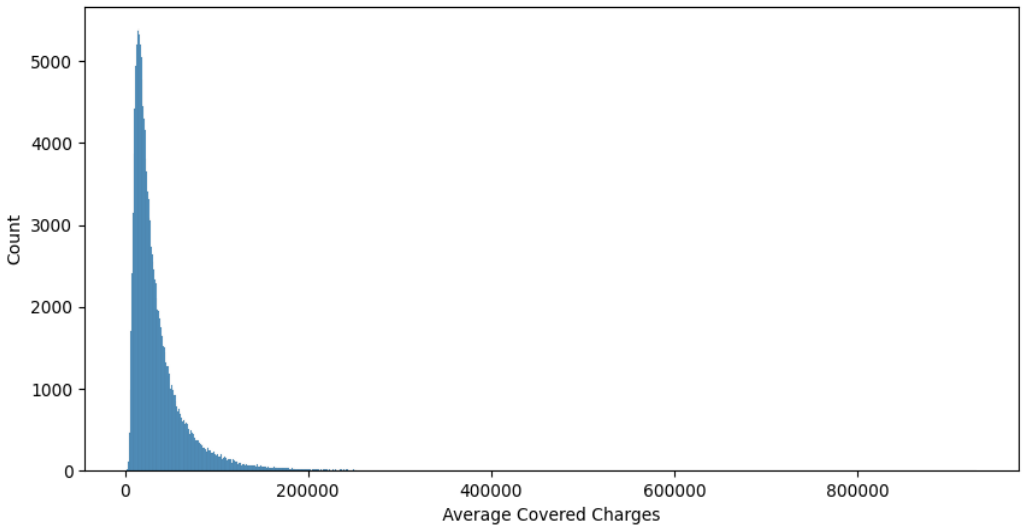
1.



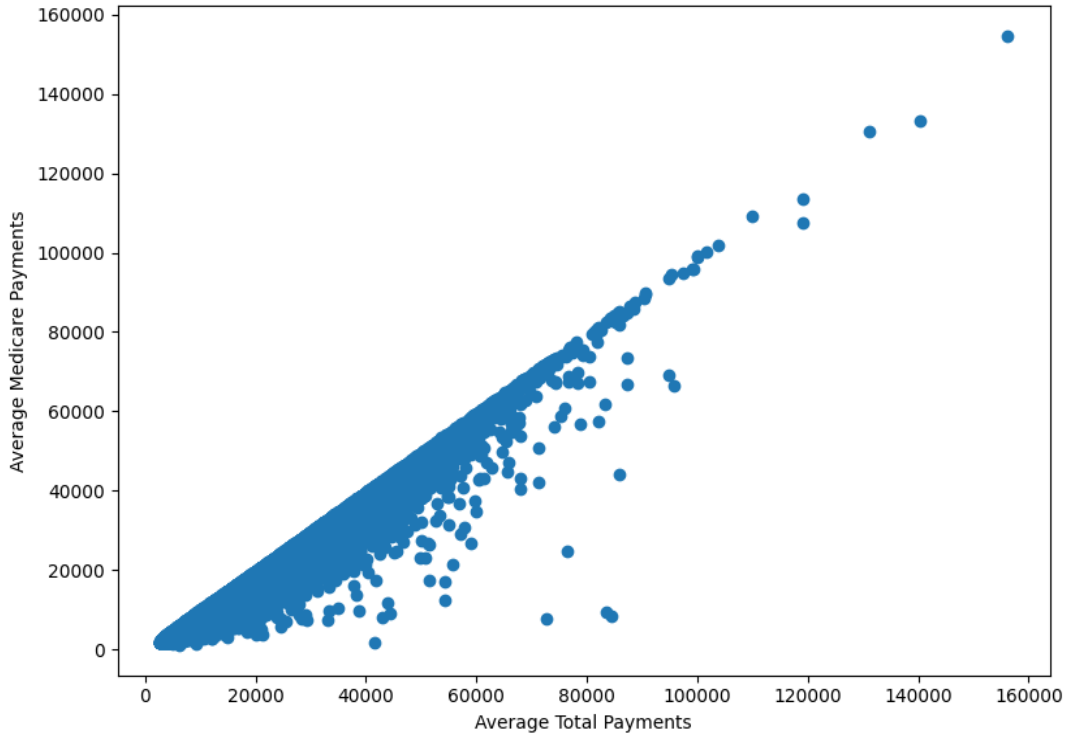


In histograms/densities plot of 'Total Discharges' feature, there is an outlier that has more than 3,000 discharges. This hospital is located in Manhattan, New York. So it is not so surprising because Manhattan has a highest population density in United States and also has many population. More patients in Manhattan are likely to rush to one hospital. Discharges are also expected to be many.

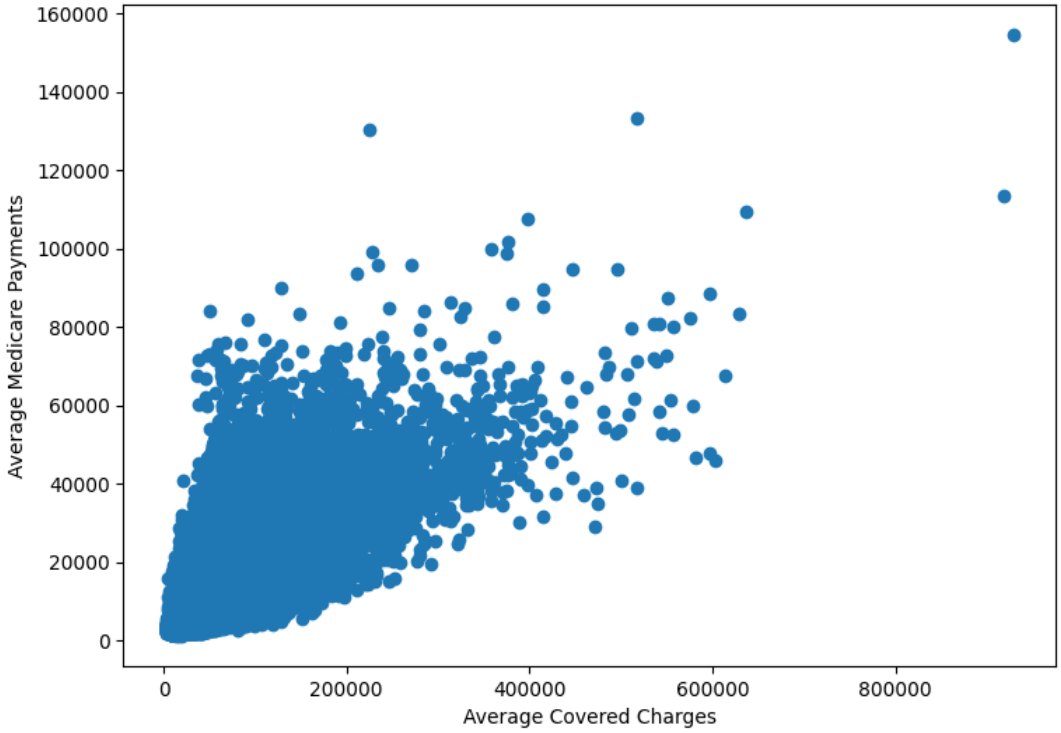




In histograms/densities plot of 'Average Covered Charges' feature, there is an outlier that has more than 900,000$ of charge. This hospital is 'Stanford hospital', located in Stanford, California. I think Stanford hospital is very famous and big hospital, but it is still so surprising because the mean of 'Average Covered Charges' in dataset is about 36133$. It is almost 25 times expensive.



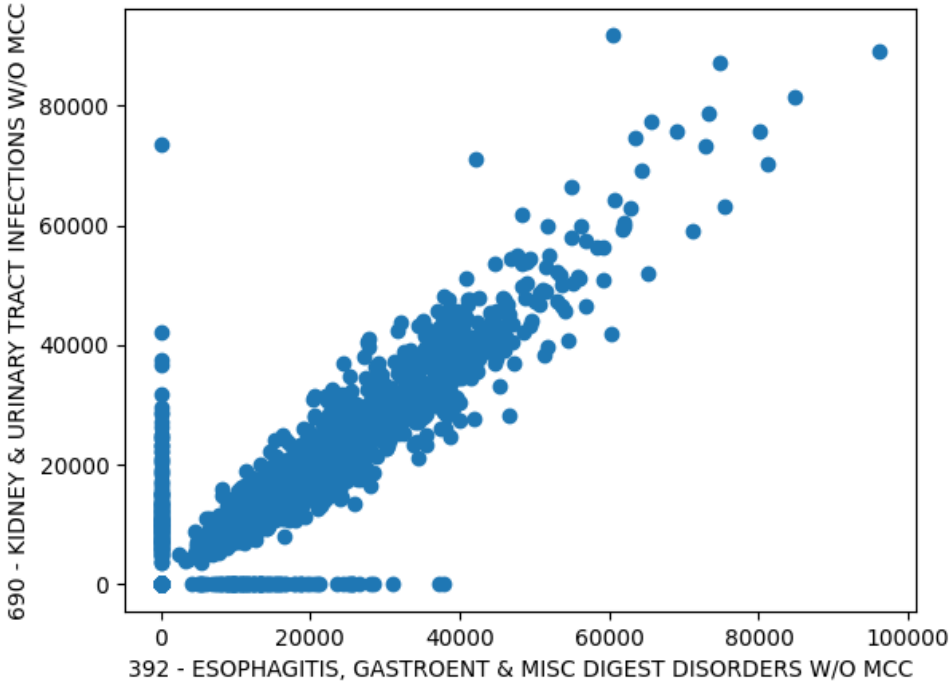
In scatter plot of 'Average Total Payments' and 'Average Medicare payments', most of the data has similar values of 'Average Total Payments' and 'Average Medicare payments'. There is an outlier that has biggest difference with 'Average Total Payments' and 'Average Medicare payments'. In Overlake hospital medical center, located in seattle, Washington, the average total payment is 84499$, while average medicare payment is only 8500$. Medicare payment occupies only 10% of total payment. It is surprising but I think we can find the reason of this high difference in the point that this hospital is nonprofit operating center. It depends upon volunteers to support the operations of the hospital.



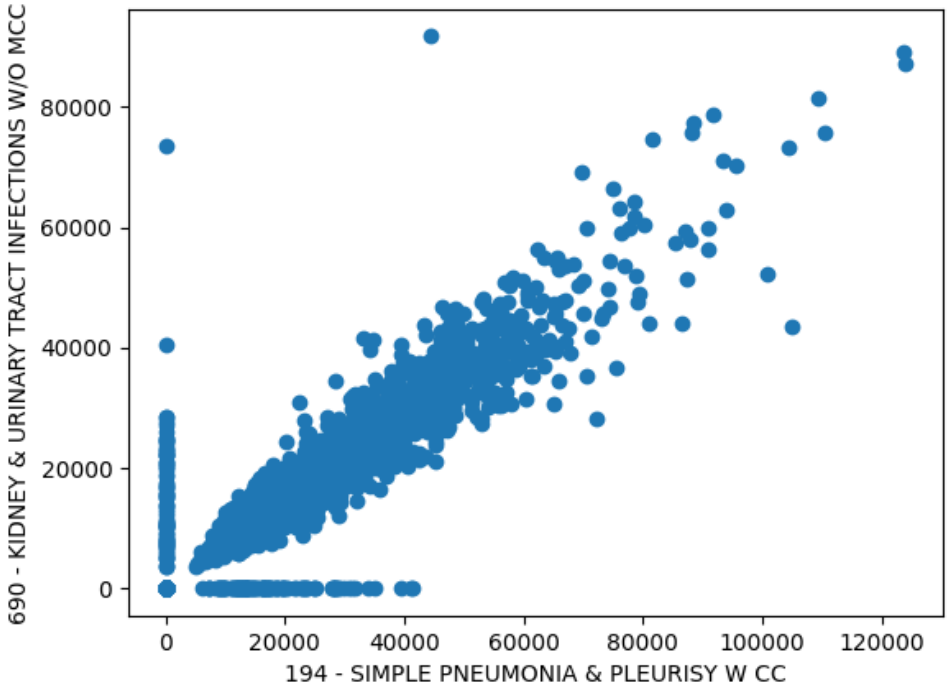
In scatter plot of 'Average Covered Charges' and 'Average Medicare payments', there is an outlier that has highest value in both charges and payments. This hospital is 'Stanford hospital', again. These are quite expected value because it is common sense that the hospital which takes most charges from patients should spends much for medicare payment to repay to patients or develop hospital.

3. – (a)

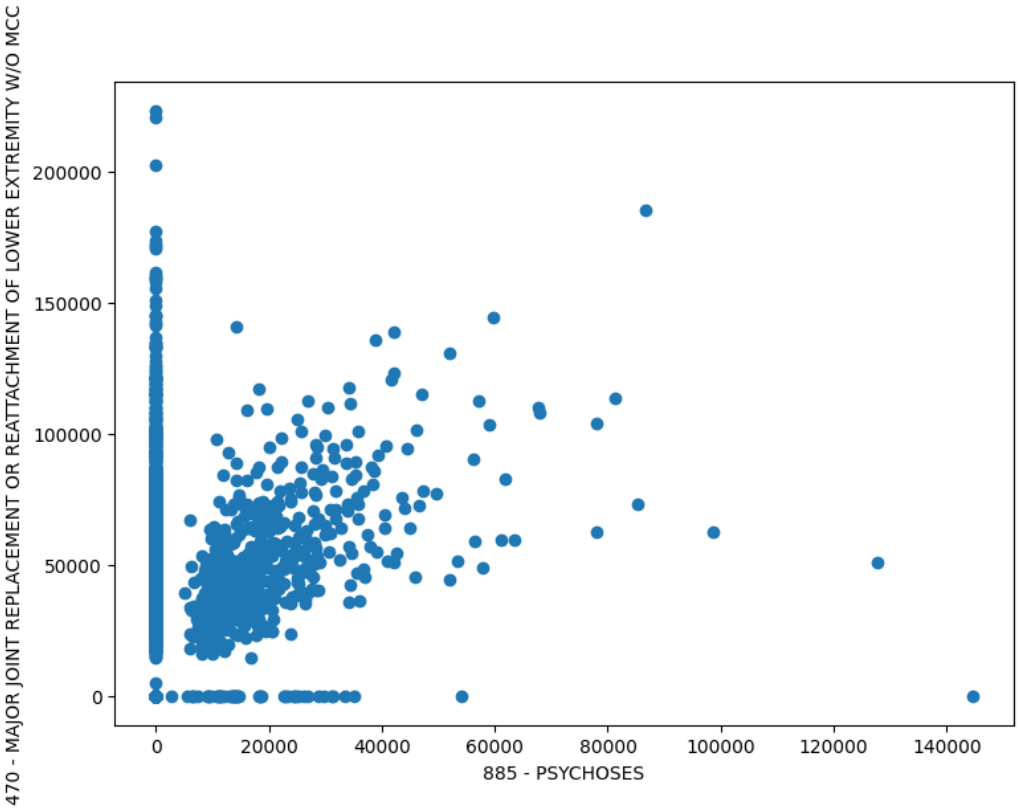
I define the standard of highest positive associations and lowest positive associations as highest and lowest value of correlation derived by pearson method.



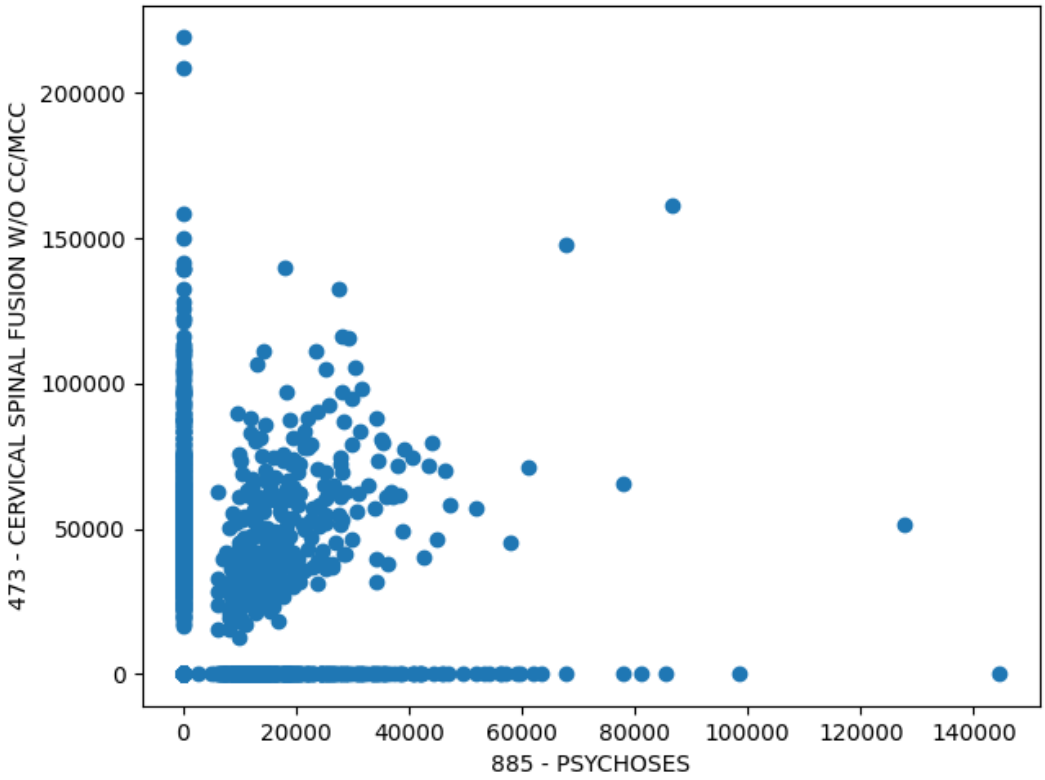
The highest correlation is between ESOPHAGITIS, GASTROENT & MISC DIGEST DISORDERS W/O MCC and KIDNEY & URINARY TRACT INFECTIONS W/O MCC. It is very interesting to me because there is no striking relevance between them. Esophagitis and gastroent are related to neck to stomach. But kidney and urinary is related to kidney and underneath.



Next highest correlation is between SIMPLE PNEUMONIA & PLEURISY W CC and KIDNEY & URINARY TRACT INFECTIONS W/O MCC. It is also very interesting because pneumonia and pleurisy are related to lung and stomach.



The lowest correlation is between PSYCHOSES and MAJOR JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREMITY W/O MCC. It is very expected value because there is no link between psychoses and joint. It is the problem of mental and body.



Next lowest correlation is between PSYCHOSES and CERVICAL SPINAL FUSION W/O CC/MCC. It is also very expected value because it has big difference between mental and human's bone.

3 – (b)

Correlation)

392 - ESOPHAGITIS, GASTROENT & MISC DIGEST DISORDERS W/O MCC 690 - KIDNEY & URINARY TRACT INFECTIONS W/O MCC : 0.9021245491840219

194 - SIMPLE PNEUMONIA & PLEURISY W CC 690 - KIDNEY & URINARY TRACT INFECTIONS W/O MCC : 0.9005407419607516

Two scatter plots of highest correlation, in almost cases, if one charges is high, another is also high. If one charges is quite small, another is also small. There are few cases that one charge is high, but another is zero. So the correlations support observations from the scatterplot enough.

885 - PSYCHOSES 470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREMITY W/O MCC : 0.16127418132602328

885 - PSYCHOSES 473 - CERVICAL SPINAL FUSION W/O CC/MCC : 0.1915474446780618

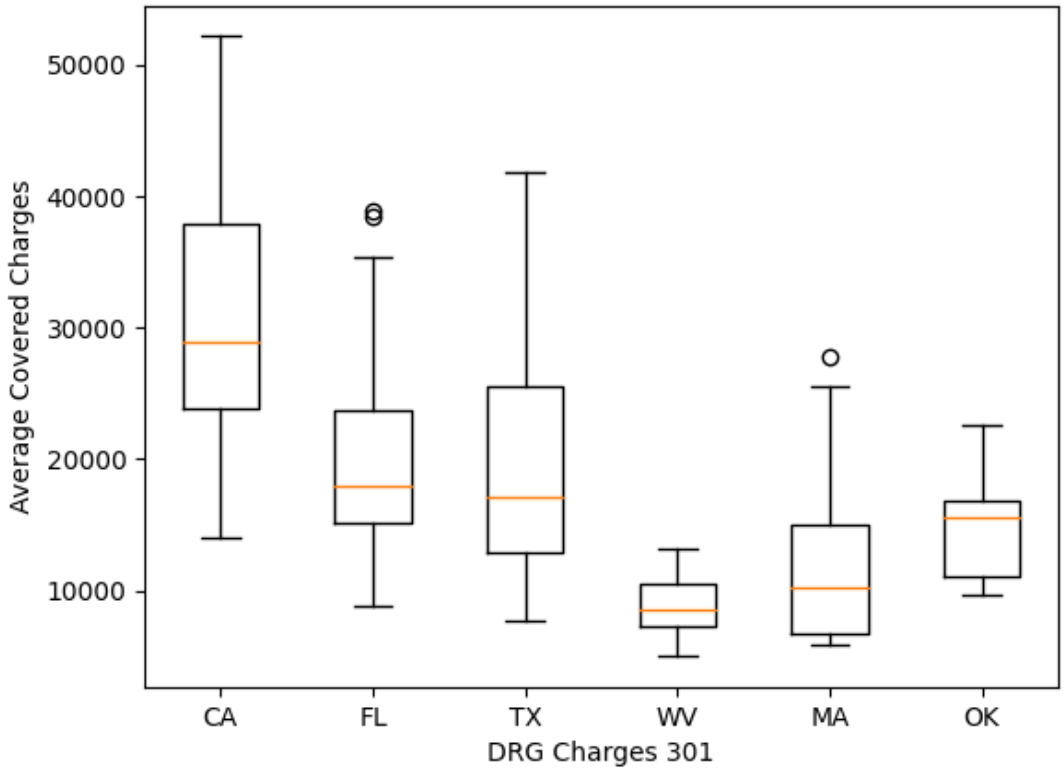
Two scatter plots of lowest correlation, conversely, most of cases have feature that one has high charge, and another has zero value. Even rest of cases do not make rising line in scatterplot, they are located quite spread out. It also means that there is not some relation between them. So the value of correlations support observations from the scatterplot enough.

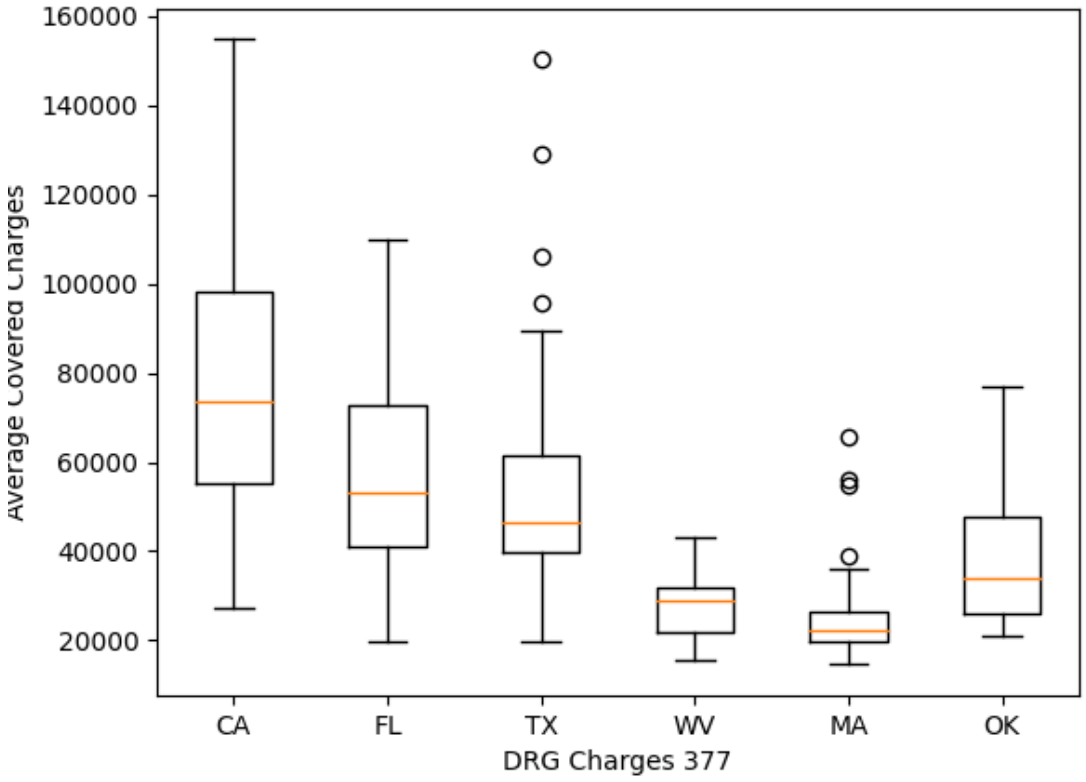
4 - (a)

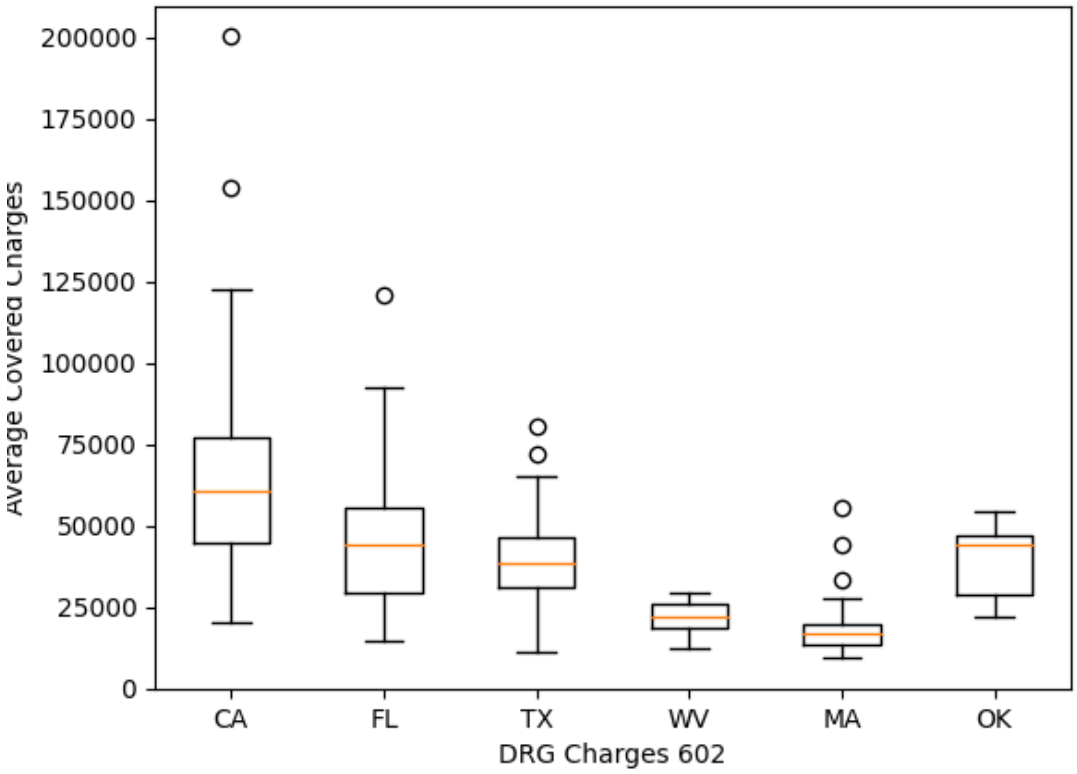
I chose 6 states which are "CA, FL, TE, WV, MA, OK". Because California and Florida are representative city located in west coast and east coast each. And Texas is second biggest state in US next to Alaska, West Virginia is relatively very small state. Lastly, according to statistics of 2020 election US in Wikipedia, Massachusetts is overwhelmingly supporting Biden, while Oklahoma supports trump strongly.

State or district	Biden/Harris Democratic			Trump/Pence Republican		
	Votes	%	EV	Votes	%	EV
Mass.	2,382,202	65.60%	11	1,167,202	32.14%	
Okla.	503,890	32.29%	–	1,020,280	65.37%	

I wanted to choose DRG Charge feature which differentiates distributions of all 6 states. But It seems impossible, so I chose the boxplots based on the variation between CA vs FL, TE vs WV, MA vs OK. Because I chose 3 pairs of states with 3 different standards.


50000
40000
30000
20000
10000
Average Covered Charges
CA
FL
TX
WV
MA
OK
DRG Charges 301


160000
140000
120000
100000
80000
60000
40000
20000
CA
FL
TX
WV
MA
OK
DRG Charges 377



4 – (b)

I thought that the difference between CA and FL at DRG Charges 301 is most significant. I decided my hypothesis that $H_0 : \mu_{CA} = \mu_{FL}$, $H_1 : \mu_{CA} > \mu_{FL}$

After doing two-sample one-sided tTest, I got these values.

t-statistics: 5.91287319, p-value: 0.00000048

p-value/2 is smaller than significance level about 0.05, so I can say that the results support H_1 hypothesis claim.

4 – (c)

I chose TX and WV for showing a significant difference in their charges across all three selected DRG categories.

$H_0 : \mu_{TX} = \mu_{WV}$, $H_1 : \mu_{TX} > \mu_{WV}$

WV state has fewer providers (33), so I downsample TX state providers to WV state. After doing two-sample paired t-test by one-sided test, I got these values.

t-statistics: 3.12597894, p-value: 0.00375579

p-value/2 is smaller than significance level about 0.05, so I can also say that the results support H_1 hypothesis claim.

In case of doing two-sample unpaired t-test, I got these values.

t-statistics: 3.96603627, p-value: 0.00023979

T-statistics increases about 27% than paired t-test, and p-value decreases more than 90% than paired t-test. In my case, the hypothesis which is supported by the results is not changed. However, It can be easily affect to assessment of significance if p-value decreases significantly while the original value is more than 0.05. Hypothesis testing result is depending on p-value.

Code of problem 1.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

data = pd.read_csv('input_data.csv')
td = data[' Total Discharges ']

#1-(a)
sb.distplot(td)
plt.show()
sb.histplot(td)
plt.show()

data[' Average Covered Charges '] = data[' Average Covered Charges
'].apply(lambda x: x[1:])
data[' Average Total Payments '] = data[' Average Total Payments
'].apply(lambda x: x[1:])
data['Average Medicare Payments'] = data['Average Medicare
Payments'].apply(lambda x: x[1:])
data = data.astype({' Average Covered Charges ': 'float'})
data = data.astype({' Average Total Payments ': 'float'})
data = data.astype({'Average Medicare Payments': 'float'})

#1-(b)
sb.distplot(data[' Average Covered Charges '])
plt.show()
sb.histplot(data[' Average Covered Charges '])
plt.show()
#1-(c)
plt.scatter(data[' Average Total Payments '], data['Average Medicare
Payments'])
plt.xlabel('Average Total Payments')
plt.ylabel('Average Medicare Payments')
plt.show()
#1-(d)
plt.scatter(data[' Average Covered Charges '], data['Average Medicare
Payments'])
```

```
plt.xlabel('Average Covered Charges')
plt.ylabel('Average Medicare Payments')
plt.show()
```

Code of problem 2~4.

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('input_data.csv')
data[' Average Covered Charges '] = data[' Average Covered Charges
'].apply(lambda x: x[1:])
data = data.astype({' Average Covered Charges ': 'float'})

info = data[['DRG Definition', 'Provider Id', 'Provider State', ' Average
Covered Charges ']]

new_info = {}
col_list = ['Provider Id']

drg_list = []
for i in range(info.shape[0]):
    if info['DRG Definition'][i] not in drg_list:
        drg_list.append(info['DRG Definition'][i])
        col_list.append('DRG Charges '+str(info['DRG Definition'][i][:3]))

prov_id = []
prov_st = []

for i in range(data.shape[0]):
    if info['Provider Id'][i] not in prov_id:
        prov_id.append(info['Provider Id'][i])
        prov_st.append(info['Provider State'][i])

for c in col_list:
    new_info[c] = [0 for i in range(len(prov_id))]

#2 - Transforming Data~
df = pd.DataFrame({'Provider Id': prov_id, 'Provider State': prov_st})
df = df.astype({'Provider Id': 'int'})
new_info['Provider Id'] = prov_id
new_df = pd.DataFrame(new_info, columns= col_list)
new_df = new_df.astype({'Provider Id': 'int'})
for i in range(data.shape[0]):
    new_df.loc[new_df['Provider Id'] == info['Provider Id'][i], 'DRG
Charges ' + str(info['DRG Definition'][i][:3])] = info[' Average Covered
Charges '][i]

new_df = pd.merge(df, new_df, on='Provider Id')
# ~Transforming Data

#3 - Correlation and Scatterplots
corr_df = new_df
corr_df = corr_df.drop(['Provider Id', 'Provider State'], axis=1)
lists = []
for c in col_list[1:]:
    lists.append(list(corr_df[c]))
```

```
corr_df = pd.DataFrame(lists).T
corr = corr_df.corr(method = 'pearson') # compute correlations

corr_dic = {}
for i in range(len(corr)):
    for j in range(len(corr)):
        if float(corr[i][j]) != 1.0:
            corr_dic[len(corr)*i + j] = float(corr[i][j])

corr_list = sorted(corr_dic.items(), key=lambda x: x[1], reverse=True)

#3-(a) plot scatterplots
plt.scatter(list(new_df['DRG Charges ' + str(drg_list[int(corr_list[0][0] /
100)][:3])]), list(new_df['DRG Charges ' +
str(drg_list[int(corr_list[0][0] % 100)][:3])]))
plt.xlabel(drg_list[int(corr_list[0][0] / 100)])
plt.ylabel(drg_list[int(corr_list[0][0] % 100)])
plt.show()

plt.scatter(list(new_df['DRG Charges ' + str(drg_list[int(corr_list[2][0] /
100)][:3])]), list(new_df['DRG Charges ' +
str(drg_list[int(corr_list[2][0] % 100)][:3])]))
plt.xlabel(drg_list[int(corr_list[2][0] / 100)])
plt.ylabel(drg_list[int(corr_list[2][0] % 100)])
plt.show()

plt.scatter(list(new_df['DRG Charges ' +
str(drg_list[int(corr_list[len(corr_list)-1][0] / 100)][:3])]),
list(new_df['DRG Charges ' + str(drg_list[int(corr_list[len(corr_list)-
1][0] % 100)][:3])]))
plt.xlabel(drg_list[int(corr_list[len(corr_list)-1][0] / 100)])
plt.ylabel(drg_list[int(corr_list[len(corr_list)-1][0] % 100)])
plt.show()

plt.scatter(list(new_df['DRG Charges ' +
str(drg_list[int(corr_list[len(corr_list)-3][0] / 100)][:3])]),
list(new_df['DRG Charges ' + str(drg_list[int(corr_list[len(corr_list)-
3][0] % 100)][:3])]))
plt.xlabel(drg_list[int(corr_list[len(corr_list)-3][0] / 100)])
plt.ylabel(drg_list[int(corr_list[len(corr_list)-3][0] % 100)])
plt.show()
# ~Correlation and Scatterplots

#4 - Boxplots and T-tests
ca = new_df['Provider State'] == 'CA'
fl = new_df['Provider State'] == 'FL'
tx = new_df['Provider State'] == 'TX'
wv = new_df['Provider State'] == 'WV'
ma = new_df['Provider State'] == 'MA'
ok = new_df['Provider State'] == 'OK'
state_label = ['CA', 'FL', 'TX', 'WV', 'MA', 'OK']
#4-(a) Boxplots

# make boxplots for all DRG Charge feature to choose three features
"""for c in col_list[1:]:
    valid = new_df[c] != 0.0
    state6 = [new_df[ca & valid][c],
              new_df[fl & valid][c],
              new_df[tx & valid][c],
              new_df[wv & valid][c],
```

```
            new_df[ma & valid][c],
            new_df[ok & valid][c]]
    plt.boxplot(state6, labels=state_label)
    plt.xlabel(c)
    plt.ylabel(' Average Covered Charges ')
    plt.show()"""
# make boxplots for all DRG Charge feature to choose three features

from scipy import stats
valid = new_df['DRG Charges 301'] != 0.0
tTestResultDiffVar = stats.ttest_ind(new_df[ca & valid]['DRG Charges 301'],
new_df[fl & valid]['DRG Charges 301'], equal_var=False)
print("t-statistics: %.8f, p-value: %.8f" % tTestResultDiffVar) # 4-(b)

# 4-(c) concatenate three selected DRG categories
tx_df = new_df[tx & valid]['DRG Charges 301']
valid = new_df['DRG Charges 377'] != 0.0
tx_df = pd.concat([tx_df, new_df[tx & valid]['DRG Charges 377']])
valid = new_df['DRG Charges 602'] != 0.0
tx_df = pd.concat([tx_df, new_df[tx & valid]['DRG Charges 602']])

wv_df = new_df[wv & valid]['DRG Charges 301']
valid = new_df['DRG Charges 377'] != 0.0
wv_df = pd.concat([wv_df, new_df[wv & valid]['DRG Charges 377']])
valid = new_df['DRG Charges 602'] != 0.0
wv_df = pd.concat([wv_df, new_df[wv & valid]['DRG Charges 602']])

#downsample and testing
tTestResultDiffVar2 = stats.ttest_ind(tx_df.sample(n=len(wv_df)), wv_df,
equal_var=False)
print("t-statistics: %.8f, p-value: %.8f" % tTestResultDiffVar2) # 4-(c)
```