Data Science Project1

Our project hypothesizes that there are significant variances between medical procedure costs across hospitals in United States. In a search to find an explanation of the variance, we pulled data from government data sources such as data.gov, healthdata.gov along with population data from a python library titled “uszipcode 0.2.2”. Other sources of data such as national hospital rankings and scores was gathered by utilizing the free API service *opendata.socrata.com* which pulls data from multiple sources such as medicare.com.

Hypothesis 1: **Hypothesis 1: There is a significant difference between population and medical procedure cost**

Population data utilized for researching the above hypothesis was pulled by the “uszipcode 0.2.2” Python library. From there, all zipcodes in the USA were called via the library and exported to a data file. The code for this process is listed below.

import pandas as pd

from uszipcode import SearchEngine

from pprint import pprint

import json

# In[29]:

df = pd.read\_csv("allzips.csv")

ziplist = df["All Zips"].to\_list()

# LISTS

zipcodelist = []

population = []

population\_density = []

median\_home\_value = []

median\_household\_income = []

major\_city = []

state = []

for zipcode in ziplist:

search = SearchEngine(simple\_zipcode=True)

zipcode = search.by\_zipcode(zipcode)

zipcode = zipcode.to\_dict()

try:

zipcodelist.append(zipcode["zipcode"])

except:

zipcodelist.append("NA")

try:

population.append(zipcode["population"])

except:

population.append("NA")

try:

population\_density.append(zipcode["population\_density"])

except:

population\_density.append("NA")

try:

median\_home\_value.append(zipcode["median\_home\_value"])

except:

median\_home\_value.append("NA")

try:

median\_household\_income.append(zipcode["median\_household\_income"])

except:

median\_household\_income.append("NA")

try:

major\_city.append(zipcode["major\_city"])

except:

major\_city.append("NA")

try:

state.append(zipcode["state"])

except:

state.append("NA")

dfzip = list(zip(major\_city, state, zipcodelist, population, population\_density, median\_home\_value, median\_household\_income))

df = pd.DataFrame(dfzip, columns=["Major City", "State", "Zipcode", "Population", "Population Density", "Median Home Value", "median Household Income"])

df.dropna(inplace=True)

bystate = df.groupby("State").agg({"Population": "sum", "Population Density": "mean", "Median Home Value": "mean", "median Household Income": "mean"})

bystate.to\_csv("All Zip Data\_Grouped (STATE).csv")

df.to\_csv("All Zip Data.csv", index=False)

API Data

Code written to pull data from the socrata open data API is listed below:

import pandas as pd

import requests

from pprint import pprint

import matplotlib.pyplot as plt

import json

import numpy as np

url = "https://opendata.socrata.com/resource/vjsj-36qd.json?$limit=50000"

response = requests.get(url).json()

provider\_id = []

hosname = []

zipcode = []

score = []

scoreraw = []

score\_size = []

rank = []

rank\_size = []

rankraw = []

function = []

city = []

state = []

for dic in range(len(response)):

try:

provider\_id.append(response[dic]["provider\_id"])

except KeyError:

provider\_id.append("")

try:

hosname.append(response[dic]["hospital\_name"])

except KeyError:

hosname.append("")

try:

zipcode.append(response[dic]["zip\_code"])

except KeyError:

zipcode.append("")

try:

scoreraw = response[dic]["score"]

scoreraw = scoreraw.split("\t/\t")

score.append(scoreraw[0])

score\_size.append(scoreraw[1])

except KeyError:

score.append("")

rank\_size.append("")

try:

rankraw = response[dic]["rank"]

rankraw = rankraw.split("\t/\t")

rank.append(rankraw[0])

rank\_size.append(rankraw[1])

except KeyError:

rank.append("")

rank\_size.append("")

try:

function.append(response[dic]["function"])

except:

function.append("")

try:

address = response[dic]["location"]["human\_address"]

address = json.loads(address)

city.append(address["city"])

state.append(address["state"])

except KeyError:

city.append("")

state.append("")

pd.DataFrame({"Provider ID": provider\_id, "Hospital Name": hosname, "Zipcode": zipcode, "Score": score, "Score Range": score\_size, "Rank": rank, "Rank Range": rank\_size, "Function": function, "City": city, "State": state})

df["Score"] = df["Score"].astype(float)

df["Score Range"] = df["Score Range"].astype(int)

df["Score Percentile"] = df["Score"] / df["Score Range"]

df["Rank"] = df["Rank"].astype(int)

df["Rank Range"] = df["Rank Range"].astype(int)

df["Rank Percentile"] = df["Rank"] / df["Rank Range"]

ca = df.loc[df["State"] == "CA"]

md = df.loc[df["State"] == "MS"]

ny = df.loc[df["State"] == "NY"]

df = df.loc[df["State"] != ""]

df.to\_csv("API Exported Data.csv", index=False)

Making sense of data by cleaning and using graphs was done in the code below:

Graph1:

#!/usr/bin/env python

# coding: utf-8

# In[2]:

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

import numpy as np

# In[3]:

zips = pd.read\_csv("All Zip Data.csv")

zips["City-State"] = zips["Major City"] + "-" + zips["State"]

zips["City-State"] = zips["City-State"].str.upper()

zips = zips.groupby("City-State").agg({"Population": "sum", "Population Density": "mean"})

payments = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

zips.to\_csv("zips.csv")

# In[3]:

payments = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

hip = payments.loc[payments["Value of care display name"] == "Value of Care hip/knee replacement"]

hip["City"] = hip["City"].str.upper()

hip["City-State"] = hip["City"] + "-" + hip["State"]

merge = pd.merge(hip, zips, on="City-State")

# merge.to\_csv("merge.csv")

merge["Payment"] = merge["Payment"].str.replace(",", "")

merge["Payment"] = merge["Payment"].str.replace("$", "")

merge["Payment"] = merge["Payment"].astype(int)

byhos = merge.groupby("Hospital name").agg({"Population": "sum", "Population Density": "mean", "Payment": "mean"})

# byhos.to\_csv("byhos.csv")

# In[5]:

fig, ax = plt.subplots()

fig.set\_figwidth(20)

fig.set\_figheight(10)

slope, inter, rvalue, pvalue, std = stats.linregress(byhos["Population"], byhos["Payment"])

yvals = np.linspace(byhos["Population"].min(), byhos["Population"].max())

fit = slope \* yvals + inter

ax.set\_ylim(10000, 35000)

ax.scatter(byhos["Population"], byhos["Payment"])

ax.plot(yvals, fit, color="green", linewidth=2, linestyle="--")

ax.legend(["Regression Line", "Hospital"])

fig.suptitle("United States\nHip / Knee Replacement Cost Per Hospital vs. Population Count",fontsize=20)

ax.set\_xlabel("City Population for Hospital", fontsize=20)

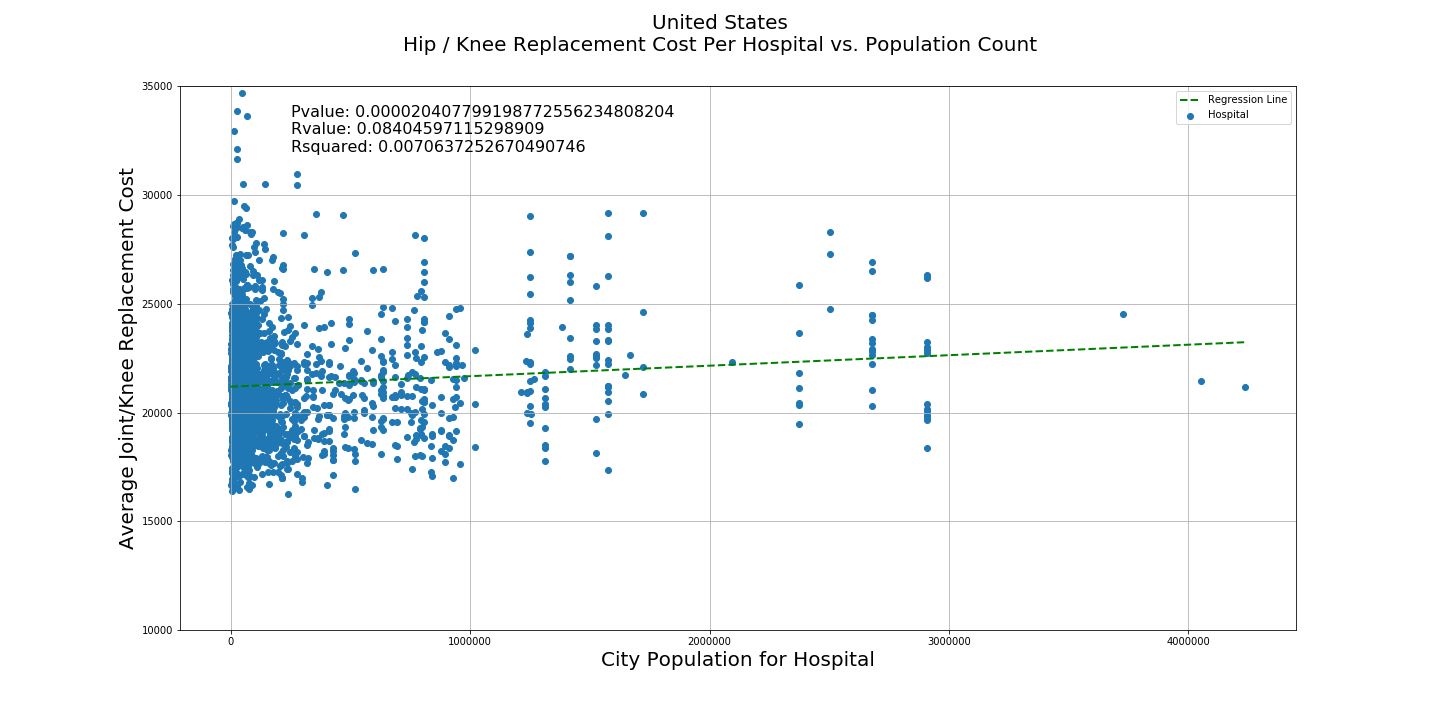
ax.set\_ylabel("Average Joint/Knee Replacement Cost", fontsize=20)

pvalue = '{0:.30f}'.format(pvalue)

plt.text(250000, 32000, "Pvalue: " + str(pvalue) + "\nRvalue: " + str(rvalue) + "\nRsquared: " + str(rvalue\*\*2), fontsize=16)

plt.grid()

fig.savefig("USA population vs payment.png")



Graph2:

#!/usr/bin/env python

# coding: utf-8

# In[3]:

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

import numpy as np

# In[4]:

zips = pd.read\_csv("All Zip Data.csv")

zips = zips.loc[zips["State"] == "NY"]

zips = zips.groupby("Major City").agg({"Population": "sum", "Population Density": "mean"})

payments = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

zips["Major City"] = zips.index

# In[5]:

state = payments.loc[payments["State"] == "NY"]

hip = state.loc[state["Value of care display name"] == "Value of Care hip/knee replacement"]

hip["City"] = hip["City"].str.upper()

zips.rename(columns={"Major City": "City"}, inplace=True)

zips["City"] = zips["City"].str.upper()

merge = pd.merge(hip, zips, on="City")

merge["Payment"] = merge["Payment"].str.replace(",", "")

merge["Payment"] = merge["Payment"].str.replace("$", "")

merge["Payment"] = merge["Payment"].astype(int)

byhos = merge.groupby("Hospital name").agg({"Population": "first", "Population Density": "first", "Payment": "mean"})

# In[6]:

fig, ax = plt.subplots()

fig.set\_figwidth(20)

fig.set\_figheight(10)

slope, inter, rvalue, pvalue, std = stats.linregress(byhos["Population"], byhos["Payment"])

yvals = np.linspace(byhos["Population"].min(), byhos["Population"].max())

fit = slope \* yvals + inter

ax.set\_ylim(10000, 35000)

ax.scatter(byhos["Population"], byhos["Payment"])

ax.plot(yvals, fit, color="green", linewidth=2, linestyle="--")

ax.legend(["Regression Line", "Hospital"])

fig.suptitle("New York\nHip / Knee Replacement Cost Per Hospital vs. Population Count",fontsize=20)

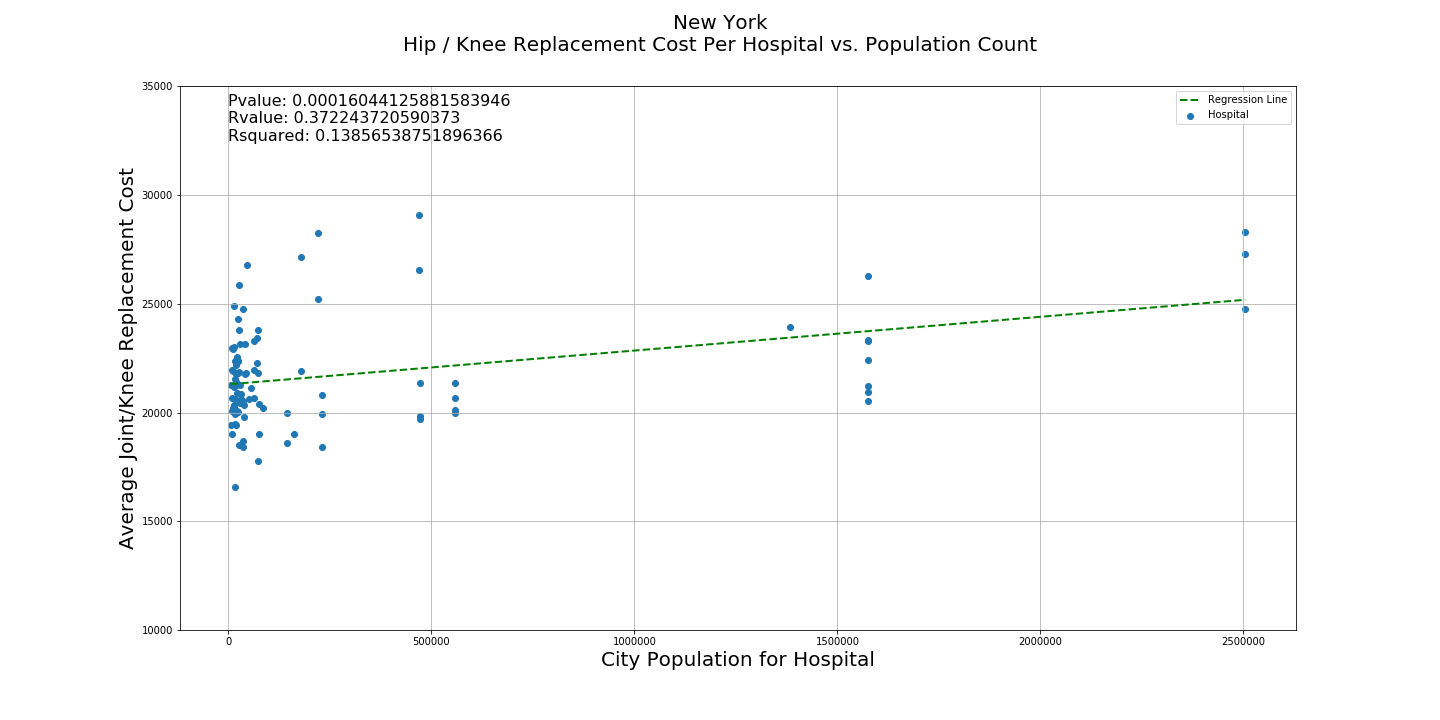
ax.set\_xlabel("City Population for Hospital", fontsize=20)

ax.set\_ylabel("Average Joint/Knee Replacement Cost", fontsize=20)

plt.text(0, 32500, "Pvalue: " + str(pvalue) + "\nRvalue: " + str(rvalue) + "\nRsquared: " + str(rvalue\*\*2), fontsize=16)

plt.grid()

fig.savefig("New York population vs payment")



Graph3:

#!/usr/bin/env python

# coding: utf-8

# In[2]:

import warnings

warnings.filterwarnings('ignore')

# In[3]:

get\_ipython().run\_line\_magic('matplotlib', 'inline')

import pandas as pd

from matplotlib import pyplot as plt

import numpy as np

import scipy.stats as stats

# In[5]:

plt.figure(figsize=(20, 10))

payments = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

hipcare = payments.loc[payments["Value of care display name"] == "Value of Care hip/knee replacement"]

hipcare['Payment'].replace(",", "", inplace=True, regex=True)

hipcare['Payment'] = hipcare['Payment'].str.replace("$", "", regex=True)

hipcare["Payment"] = hipcare["Payment"].astype(int)

maxpayment = hipcare.groupby("State").agg({"Payment": "max"})

maxpayment.rename(columns={"Payment": "Max Payment"}, inplace=True)

minpayment = hipcare.groupby("State").min()

maxpayment["Min Payment"] = minpayment["Payment"]

maxav = maxpayment["Max Payment"].mean()

minav = maxpayment["Min Payment"].mean()

maxpayment["Max Average"] = maxav

maxpayment["Min Average"] = minav

plt.title("Highest vs. Lowest Payments for Hip/Knee Replacement", fontsize=20)

plt.xlabel("State", fontsize=20)

plt.ylabel("Payment Amount", fontsize=20)

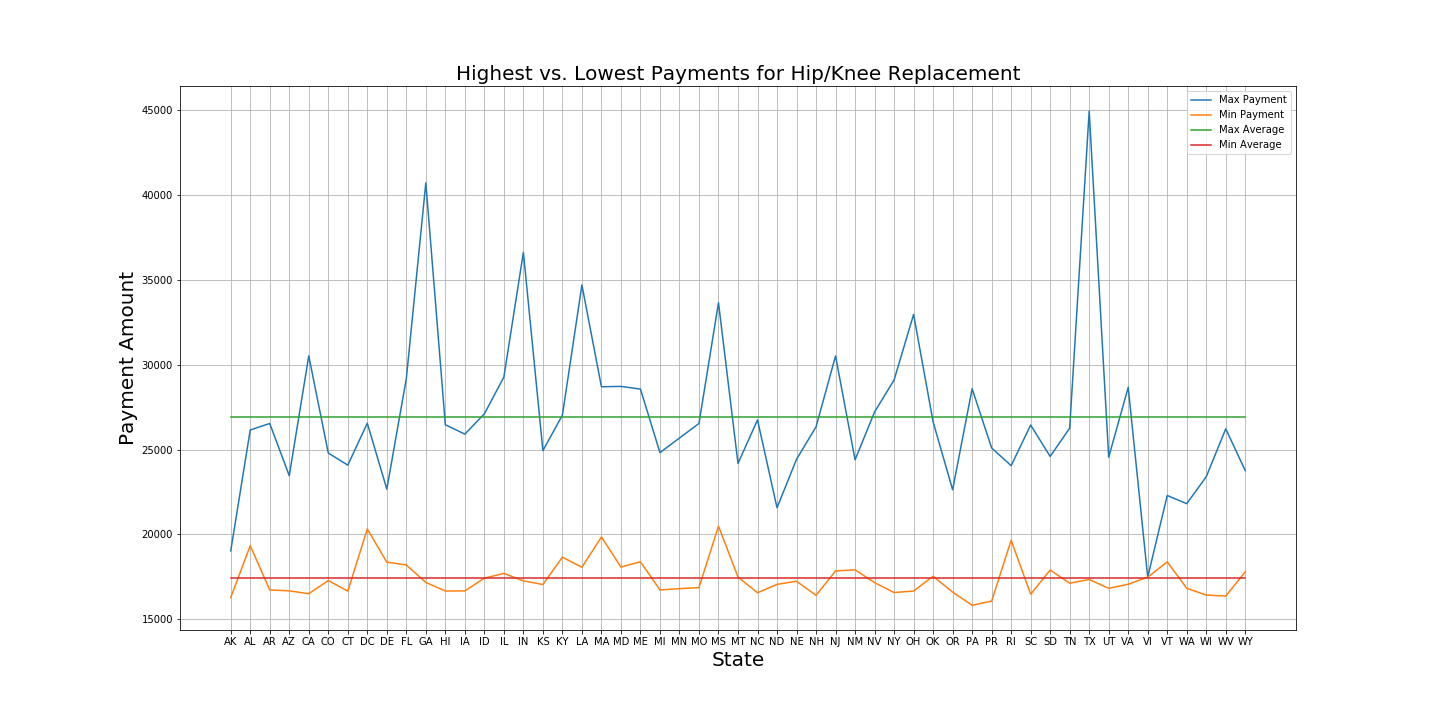
plt.plot(maxpayment)

plt.legend(maxpayment)

plt.grid()

plt.savefig("Hip Knee Replacement Cost Variances.png")

maxav - minav



**Hypothesis 2: There is a significant difference between consumer satisfaction hospital ranks and the cost of medical procedures.**

Graph4:

#!/usr/bin/env python

# coding: utf-8

# In[2]:

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

import numpy as np

# In[3]:

ratings = pd.read\_csv("API Exported Data.csv")

c\_s = ratings.loc[ratings["Function"] == "Consumer Satisfaction"]

valueofcare = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

dbdf = pd.read\_csv("diabetes\_only\_2017\_chrgs.csv")

zips = pd.read\_csv("All Zip Data.csv")

dbdf = pd.merge(dbdf, c\_s, on="Provider ID")

dbdf = pd.merge(dbdf, zips, on="Zipcode")

dbdf.to\_csv("dbdf.csv")

# In[4]:

fig, ax = plt.subplots()

fig.set\_figwidth(20)

fig.set\_figheight(10)

slope, inter, rvalue, pvalue, std = stats.linregress(dbdf["Rank"], dbdf["Average Total Payments"])

yvals = np.linspace(dbdf["Rank"].min(), dbdf["Rank"].max())

fit = slope \* yvals + inter

ax.set\_ylim(0, 30000)

ax.set\_xlim(0, 35)

ax.plot(yvals, fit, color="green", linewidth=2, linestyle="--")

ax.scatter(dbdf["Rank"], dbdf["Average Total Payments"])

ax.set\_xlabel("Hospital Consumer Satisfaction Rank (Best Ranking Hospitals = 1)", fontsize=20)

ax.set\_ylabel("Cost of Diabetes Procedure: '638 - DIABETES WCC'", fontsize=20)

plt.legend(["Regression Line", "Hospitals"])

pvalue = '{0:.30f}'.format(pvalue)

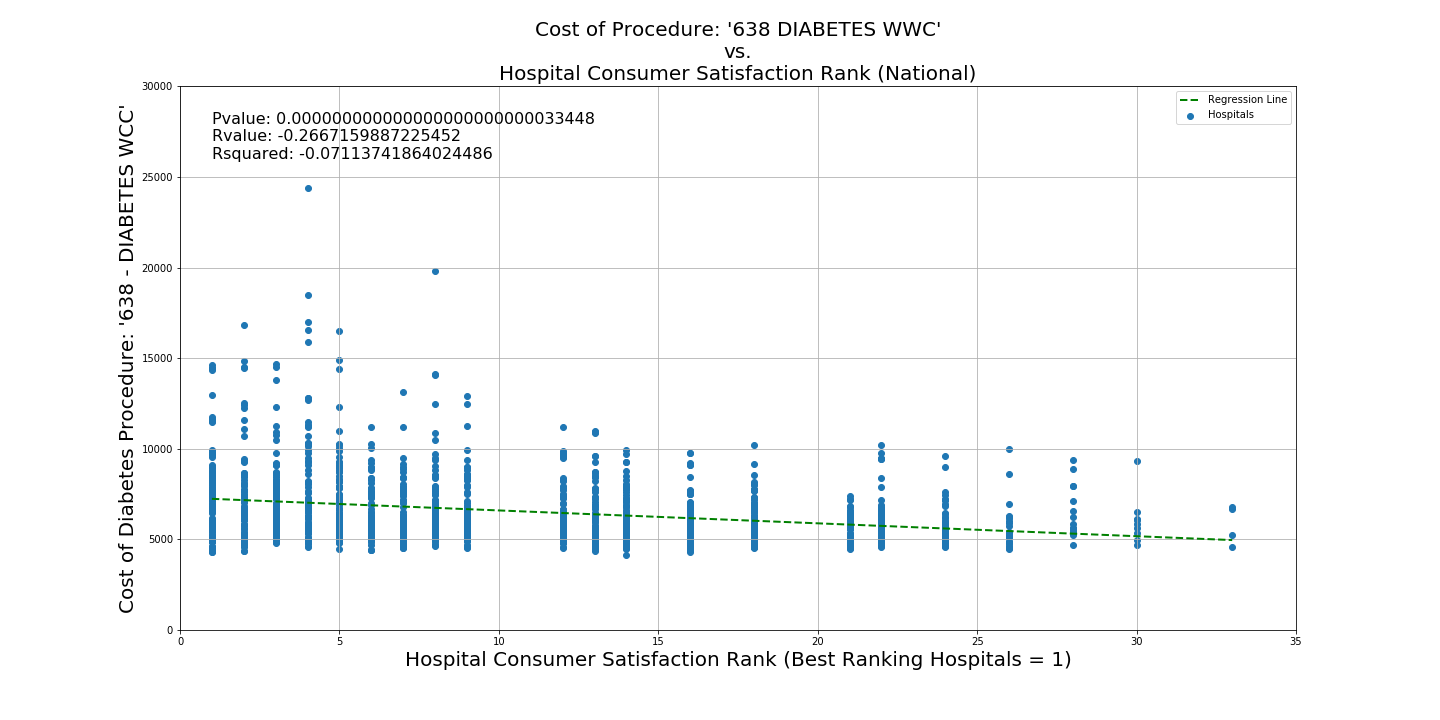
plt.title("Cost of Procedure: '638 DIABETES WWC'\nvs.\nHospital Consumer Satisfaction Rank (National)", fontsize=20)

plt.text(1, 26000, "Pvalue: " + str(pvalue) + "\nRvalue: " + str(rvalue) + "\nRsquared: -" + str(rvalue\*\*2), fontsize=16)

plt.grid()

plt.savefig("Rank vs Diabeties cost.png")

# In[ ]:



Graph5:

#!/usr/bin/env python

# coding: utf-8

# In[1]:

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

import numpy as np

# In[3]:

ratings = pd.read\_csv("API Exported Data.csv")

c\_s = ratings.loc[ratings["Function"] == "Consumer Satisfaction"]

valueofcare = pd.read\_csv("Payment\_and\_value\_of\_care\_-\_Hospital\_CLEAN.csv")

dbdf = pd.read\_csv("joint\_only\_2017\_chrgs.csv")

zips = pd.read\_csv("All Zip Data.csv")

dbdf = pd.merge(dbdf, c\_s, on="Provider ID")

dbdf = pd.merge(dbdf, zips, on="Zipcode")

dbdf.to\_csv("dbdf.csv")

# In[11]:

fig, ax = plt.subplots()

fig.set\_figwidth(20)

fig.set\_figheight(10)

slope, inter, rvalue, pvalue, std = stats.linregress(dbdf["Rank"], dbdf["Average Total Payments"])

yvals = np.linspace(dbdf["Rank"].min(), dbdf["Rank"].max())

fit = slope \* yvals + inter

ax.set\_ylim(0, 60000)

ax.set\_xlim(0, 35)

ax.plot(yvals, fit, color="green", linewidth=2, linestyle="--")

ax.scatter(dbdf["Rank"], dbdf["Average Total Payments"])

ax.set\_xlabel("Hospital Consumer Satisfaction Rank (Best Ranking Hospitals = 1)", fontsize=20)

ax.set\_ylabel("Cost of Procedure", fontsize=20)

plt.legend(["Regression Line", "Hospitals"])

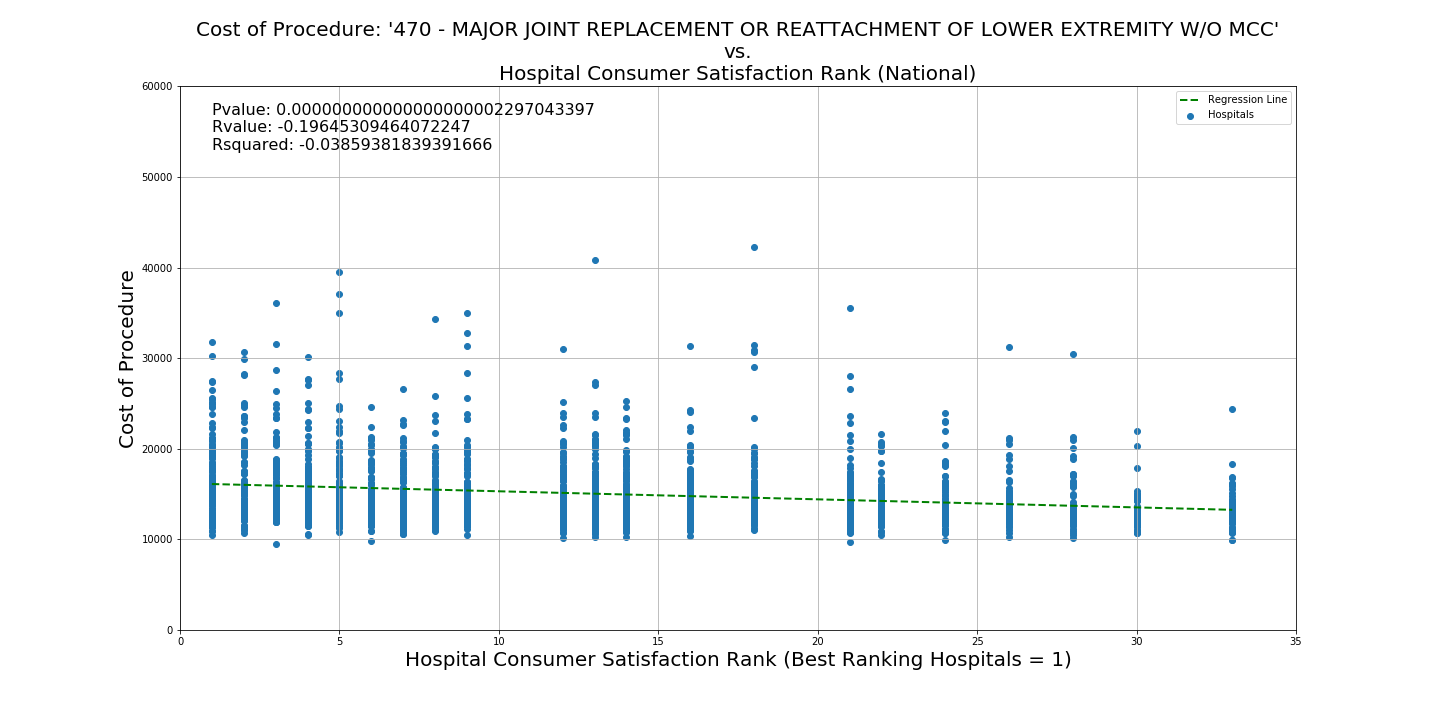
pvalue = '{0:.30f}'.format(pvalue)

plt.title("Cost of Procedure: '470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREMITY W/O MCC'\nvs.\nHospital Consumer Satisfaction Rank (National)", fontsize=20)

plt.text(1, 53000, "Pvalue: " + str(pvalue) + "\nRvalue: " + str(rvalue) + "\nRsquared: -" + str(rvalue\*\*2), fontsize=16)

plt.grid()

plt.savefig("Rank vs joint cost.png")



Conclusion

We were able to reject both of the above null hypothesizes since p-values < 0.05, however the correlation coefficients (r-value) are too low to make practical conclusions.