STA 141B Final

Nirmal Kaluvai, Riyaadh Buskh, Talha Shafik

December 2023

1 Introduction

In this collaborative effort, our group embarked on a mission of extracting salary data through the intricate process of web scraping. By collectively navigating a targeted website, in this case that site being https://www.salary.com/, we harnessed the power of data collection to compile an extensive dataset, serving as the cornerstone for our joint analysis. Our shared goal was to transform raw salary information into insightful visualizations, weaving a narrative that unveils patterns and trends within the data. Some key filters we took into consideration for this project and used for visual and data analysis were the states and type of jobs in the dataset. To make our project more straightforward and too general we decided to focus only on healthcare professions. We as group have shared interests in the health care industry and decided that this would be a great way to intersect the ideas of this class alongside the passions that we have outside of it. This project not only required technical prowess in web scraping but also fostered creative collaboration to translate data into compelling visuals. Together, we navigated the intersection of technology and storytelling to draw conclusive narratives from the complex web of salary figures, contributing to a deeper comprehension of the economic landscape.

2 Questions

We are interested in a multitude of different key questions that would help give us a deeper understandings of salaries of the healthcare industry taking into account different roles and location.

1. Which type of healthcare profession has the highest average salary?
2. Which locations have the highest salaries in general?
3. Which combo of type of healthcare profession and location has the highest salary?

We plan to answer these questions using descriptive statistics, and general distributions supplemented by graphs and visualizations.

3 Data Acquisition and Processing

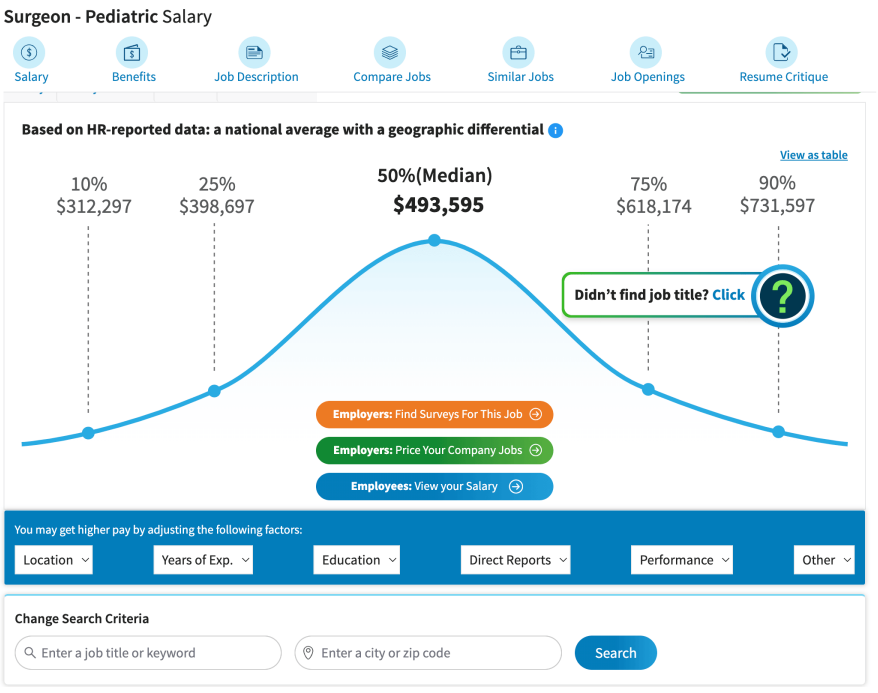
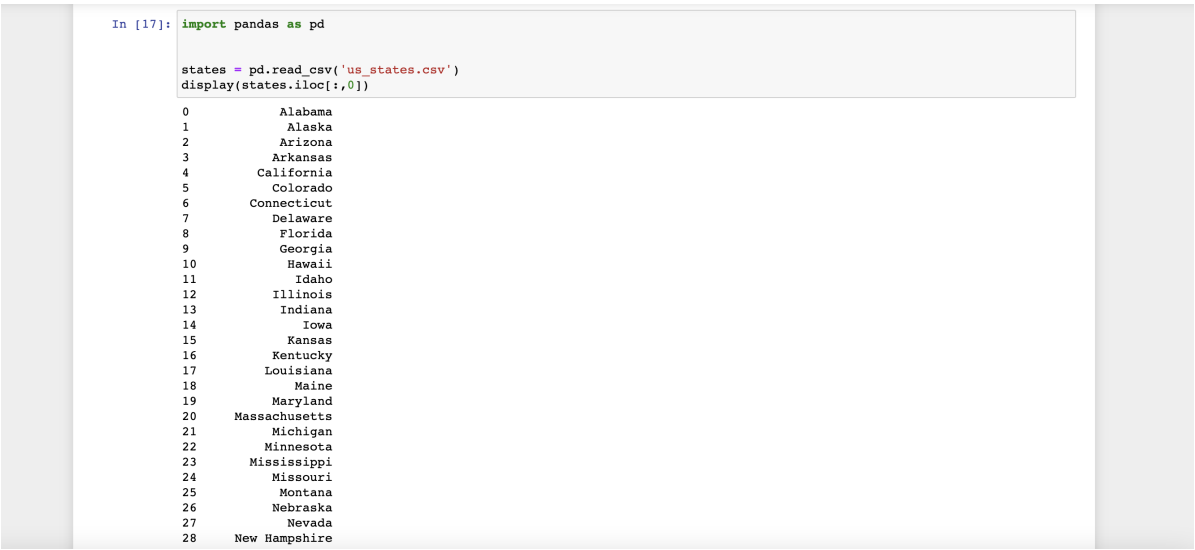


Figure 1: Web scraping code

Our main data source was the salary.com website. This is the UI of the website we are trying to scrap. It has the general descriptive statistics of the role and a filter to search by its location. We were able to acess this data through our python function, extractsalaryinfo, which retrieves salary-related information for a specified job title and city from the website. It constructs a URL based on the provided job title and city, sends an HTTP request to the site, and then parses the HTML content using BeautifulSoup. The function specifically targets a JSON script embedded in the HTML that contains occupation details, including salary percentiles (10th, 25th, median, 75th, and 90th). Finally, it extracts relevant information such as job title, location, description, and the specified salary percentiles, returning a tuple with these details. With this newly scraped data we can make statistical models after some reformatting, and dive into deeper analysis.



```
In [17]: import pandas as pd


         states = pd.read_csv('us_states.csv')
         display(states.iloc[:,0])
```

```
0            Alabama
1             Alaska
2            Arizona
3           Arkansas
4         California
5           Colorado
6        Connecticut
7           Delaware
8            Florida
9            Georgia
10            Hawaii
11             Idaho
12          Illinois
13           Indiana
14              Iowa
15            Kansas
16          Kentucky
17         Louisiana
18             Maine
19          Maryland
20     Massachusetts
21          Michigan
22         Minnesota
23       Mississippi
24          Missouri
25           Montana
26          Nebraska
27            Nevada
28     New Hampshire
```

Figure 2: States dataframe

Above we have a snippet of the data frame created that contains all the states in the united states. This allows us to create models and graphs using

the states to categorize the salary data.

In [18]:
```
salary_data = []
surgeon_types = [
    "general-surgeon",
    "cardiothoracic-surgeon",
    "neurological-surgeon",
    "oral-surgeon",
    "orthopedic-surgeon",
    "burn-surgeon"
]

for state in states.iloc[:,1]:
    for surgeon in surgeon_types:
        result = extract_salary_info(surgeon, state)
        if result:
            salary_data.append(result)
            #sleep(0.5)
```

Save the data to csv

Finally, we'll save our data to a csv file.

In [19]:
```
with open('salary-results_states.csv', 'w', newline='', encoding='utf-8') as f:
    writer = csv.writer(f)
    writer.writerow(['Title','Location', 'Description', 'nTile10', 'nTile25', 'nTile50', 'nTile75', 'nTile90'])
    writer.writerows(salary_data)
```

Figure 3: Extracting state salary data

This Python code iterates through a list of surgeon types, which includes general, cardiothoracic, neurological, oral, orthopedic, and burn surgeons. For each surgeon type, the code further iterates through a list of states. In each iteration, it calls the extractsalaryinfo function to retrieve salary-related information for the specified surgeon type and state. If the result is not empty (indicating successful extraction), the obtained information is appended to the salarydata list. This process collects salary data for various surgeon types across different states, populating the salarydata list with the extracted information.

4 Job Data

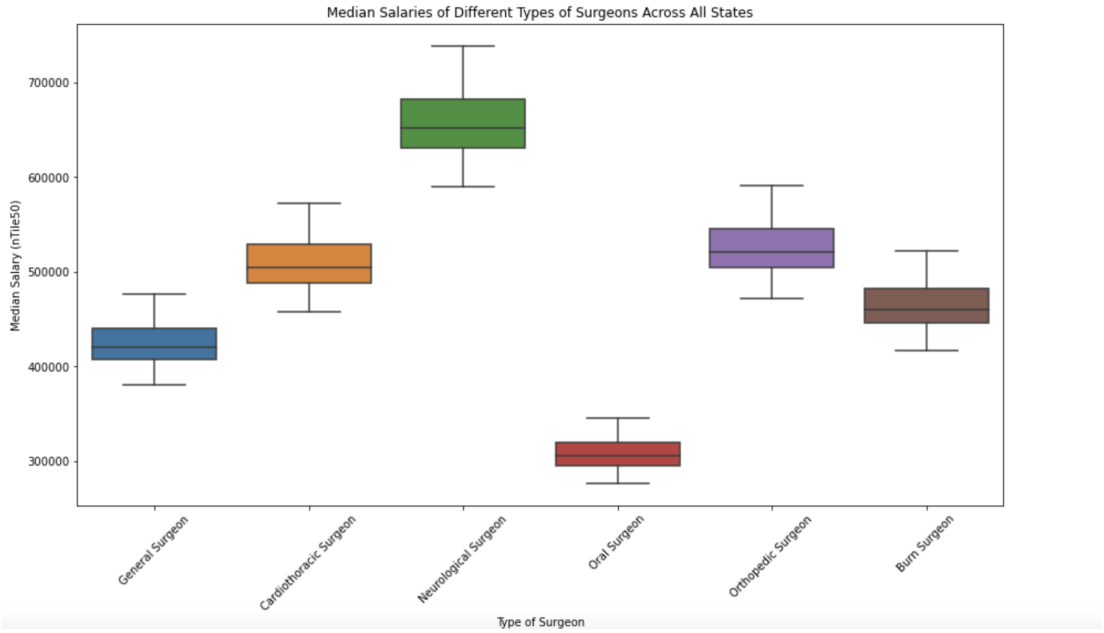


Figure 4: box and whisker plot of the data

Above we have a box and whisker plot depicting the average salaries of us surgeons across the united states. As seen from the graph the neurosuergeon makes the most on average with a median salary around 650,000 dollars a year. Cardiothoracic and orthopedic surgery follow right behind neurological surgery, with both median salaries being slightly higher than 500,000 a year. One interesting seen from the graph, is that oral surgery salaries are much less than its counterparts. The average salary for an oral surgeon is about 300,000 which more than 2 times less than average neurosurgeon.

The correlation plot allows us to analyze patterns and variations in salary distribution across different job titles. It serves as a valuable tool for understanding the salary landscape within our dataset, highlighting the dispersion and central tendencies associated with each role. This analysis aids in identi-

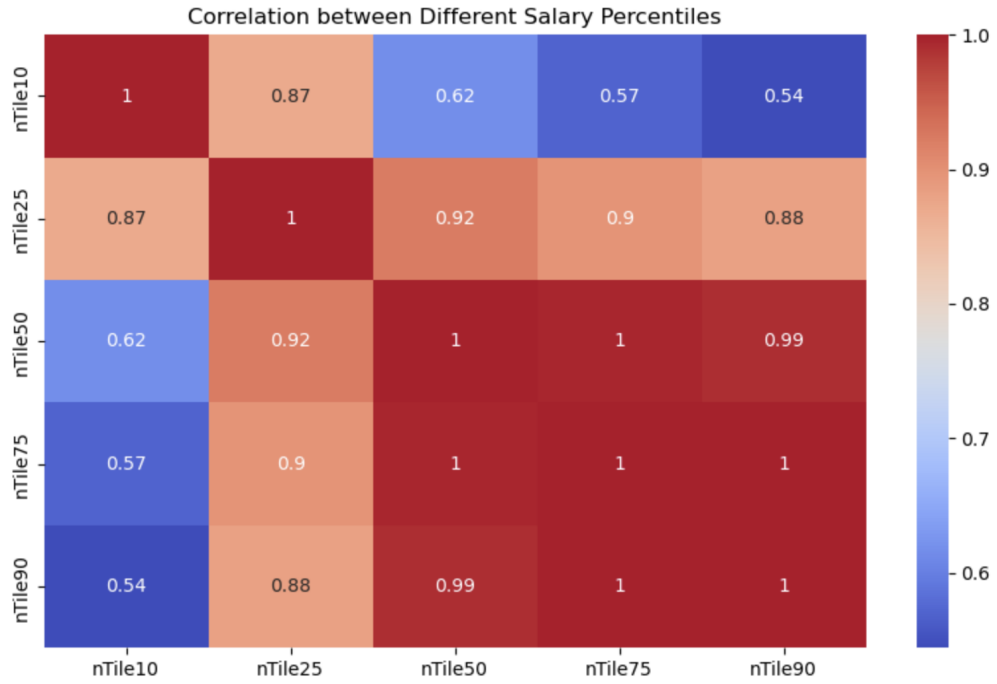


Figure 5: correlation plot between salary and job title

fying not only the median salaries for various job titles but also the range and variability in compensation, providing a nuanced perspective on the relationship between job titles and salary levels.

We analyzed one specific role in general to visualize the distribution of the salary. We found the peak salary to be between 450,000 and 475,000.

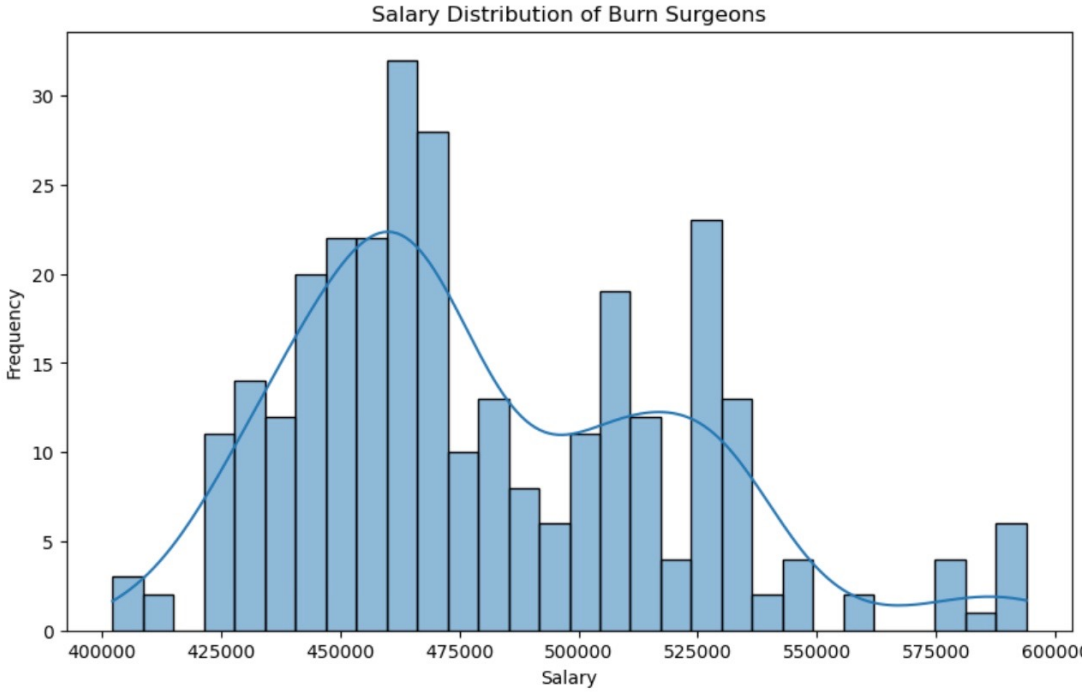


Figure 6: Salary distribution of burn surgeons

5 Location Data

Now that we have depicted surgeon salary data in correlation to the states, we as group thought it would be interesting if we could be more specific. So we decided to run the analysis based off of cities as well as this would give us a more precise analysis of the salaries, by being more specific with our parameters, rather than generalizing the results with states.

We extracted the salary information for surgeons in all the major cities across the United States. One major problem we ran into during this was the run time, as scraping this much data, resulted in a large amount of server requests, but after waiting for over a half hour we were able to get the data we needed.

This code enhances a DataFrame (dfsurgeonsalarycities) containing surgeon salary information in different cities. It utilizes the geocoder library to obtain

```
: import geocoder

df_surgeon_salary_cities = pd.read_csv('salary-results_cities.csv')

display(df_surgeon_salary_cities.head())

# Define a function to get latitude and longitude
def get_lat_lng(location):
    g = geocoder.osm(location)
    if g.latlng:
        return g.latlng
    else:
        return (None, None)

# Apply the function to each row in the DataFrame
df_surgeon_salary_cities['Coordinates'] = df_surgeon_salary_cities['Location'].apply(get_lat_lng)

# Split the coordinates into two separate columns
df_surgeon_salary_cities['Latitude'] = df_surgeon_salary_cities['Coordinates'].apply(lambda x: x[0] if x else None)
df_surgeon_salary_cities['Longitude'] = df_surgeon_salary_cities['Coordinates'].apply(lambda x: x[1] if x else None)

# Drop the temporary Coordinates column
df_surgeon_salary_cities = df_surgeon_salary_cities.drop(columns=['Coordinates'])

# Display the updated DataFrame
print(df_surgeon_salary_cities.head())
```

	Title	Location	Description	nTile10	nTile25	nTile50	nTile75	nTile90
0	General Surgeon	New York, NY	The General Surgeon reviews patient history an...	358321	429700	508100	603000	689401
1	Cardiothoracic Surgeon	New York, NY	The Cardiothoracic Surgeon determines which in...	378536	488700	609700	764300	905055
2	Neurological Surgeon	New York, NY	The Neurological Surgeon operates on the brain...	415518	592600	787100	996000	1186193
3	Oral Surgeon	New York, NY	The Oral Surgeon reviews patient history and c...	322467	344500	368700	402700	433655
4	Orthopedic Surgeon	New York, NY	The Orthopedic Surgeon performs surgical proce...	372209	494710	629260	814070	982330

Figure 7: City hot spots

latitude and longitude coordinates for each city listed in the 'Location' column of the DataFrame. The function getlatlng is defined to retrieve these coordinates using the OpenStreetMap (OSM) geocoding service. The obtained coordinates are then split into two separate columns ('Latitude' and 'Longitude') and added to the DataFrame. Finally, the temporary 'Coordinates' column is dropped, resulting in an updated DataFrame with additional geographical information.

Now moving on lets make an interactive map so that we can see salaries across the country and see which states have higher salaries.

```
]:
burn_surgeon_salary_cities = df_surgeon_salary_cities.loc[df_surgeon_salary_cities['Title'] == 'Burn Surgeon',:]

# Create a Folium map
m = folium.Map(location=[39.8283, -98.5795], zoom_start=4)

# Add markers to the map
for index, row in burn_surgeon_salary_cities.iterrows():
    props = get_marker_properties(row['nTile50'])
    folium.Marker(
        [row['Latitude'], row['Longitude']],
        popup=f"Positon: {row['Title']}: {row['Location']}: Median Salary - {row['nTile50']}",
        icon=folium.Icon(color=props['color'], icon=props['icon'])
    ).add_to(m)

# Display the map inline in a Jupyter Notebook
display(m)
```

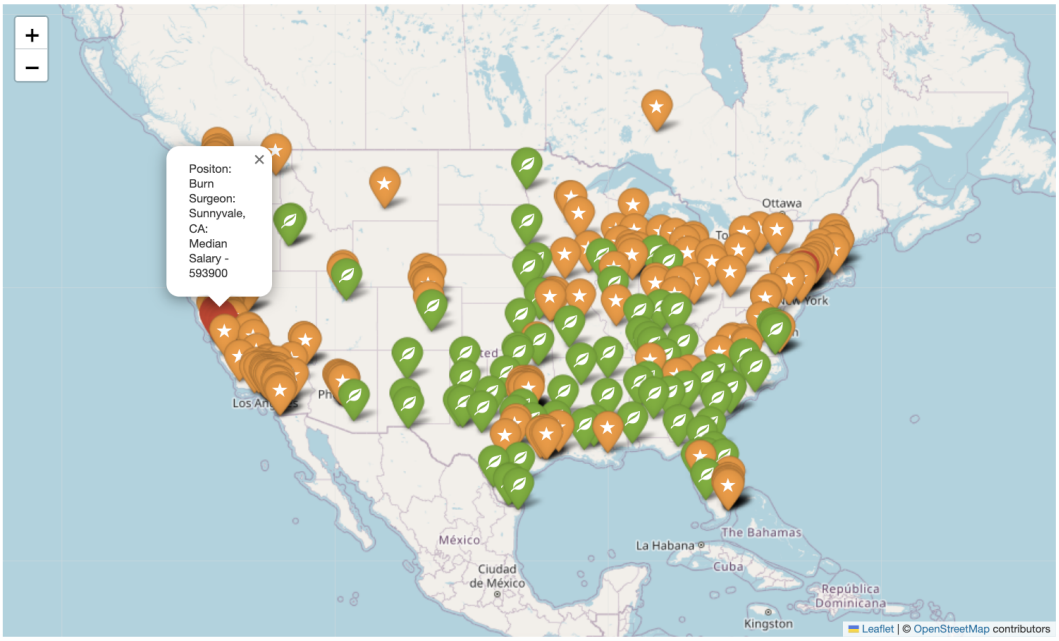


Figure 8: Interactive Map for burn surgeons

In this code segment we create an interactive plot where you can scroll around the united states and select the leaf icons to display the state, city, and salary of a burn surgeon. We decided to only do the burn surgeon to not over flood the map and keep the visual as clean as possible. As we can see it appears that surgeons in major cities such San Francisco and New York make the most money. This can be due cost of living plus a higher demand of surgeries with a lower supply surgeons.

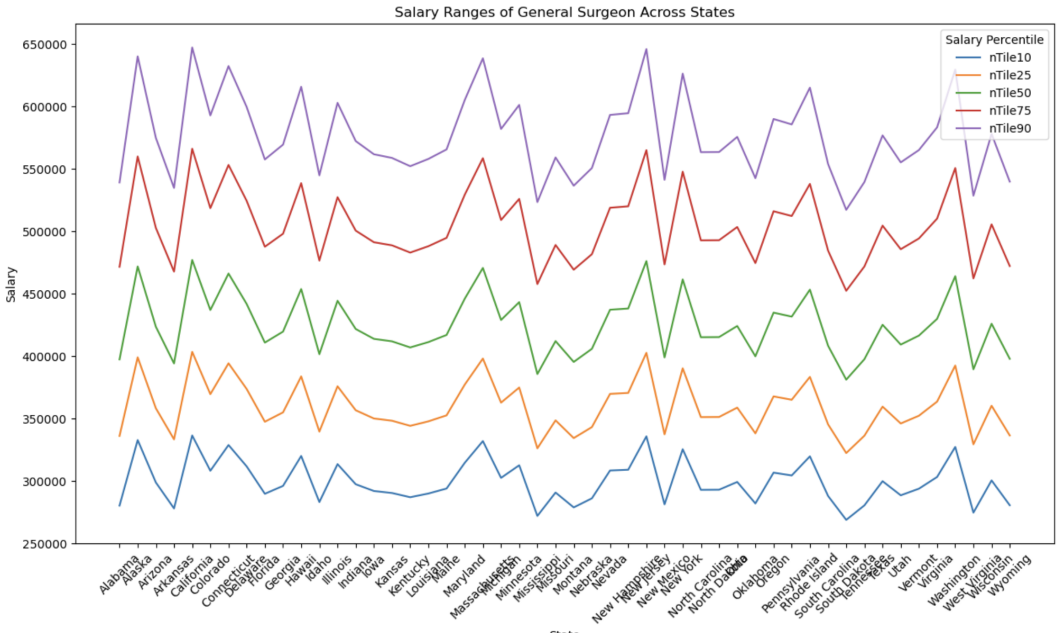


Figure 9: Salary Ranges of surgeons across states

In this visual representation, we explore the distribution of surgeon salaries across various states, employing a quantile analysis to unveil key insights into the salary landscape. This was our original location analysis but upon further discussion we decided it was not as insightful as a city based analysis.

6 Final remarks and Conclusion

Throughout this whole process we ran into many obstacles. One of these included being kicked out of the salary.com site due to a large influx of requests. In order to combat this we needed to put sleep timers in order to delay the requests and overtask the sites servers. Additionally, when we originally started we were quantifying our data with states rather than cities. This created a pretty sound model, but there could be a lot of variance in this data, as each state is so diverse and one part of the state could result in higher salaries than

the other part. So to deal with this we filtered out results based on cities to determine more specific analysis's and made our judgements based off that.

In terms of the original questions asked at the start we were able to answer them using the code we made.

1. Which type of healthcare profession has the highest average salary?

The answer to this was Neurosurgeon as seen in our box and whisker plot

2. Which locations have the highest salaries in general?

For this question, we coursed though the city data frame as well as our interactive plot at the end and saw that big cities such as New York, Los Angeles, and San Francisco yielded the highest salaries for healthcare workers.

3. Which combo of type of healthcare profession and location has the highest salary?

Piggy backing off of our previous answers a combination being a Neurosurgeon and practicing in New York, Los Angeles, and San Francisco would on average give you a much higher salary than any other profession and city.