What Are the Potential Factors Relating to Employee's Salary in US Tech Companies?

April 14, 2023

1 Project One

1.1 Introduction

The development of technology is closely connecting to and benefiting our daily life. Tech companies are the companies mainly focusing on innovation and manufacture of electronic, technology-intensive products and services, which create a more convenient and connected world only in a few decades. By innovating ideas and inventing products, tech companies earned huge profits, and working in the tech industry is also viewed as having a decent job with considerable income. For example, Amazon, Apple, Google, and Microsoft are some big names among tech companies based in the US. According to Wheelwright (2021), the average salary in US tech industry ranges from 20% to 85% higher than workers in other industries. However, tech companies are facing a long-term shortage in skilled employees, which makes them harder to retain skilled workers and drives up tech company salaries (Harvey Nash Technology Survey, 2015). The survey found 77% employees who changed jobs in 2015 considered a high salary as the main reason of their job change, more important than work/life balance (72%) as the main motivator in the previous year. Additionally, Ayala and Echeverri (2009) found that workers in specific geographical regions, such as the San Francisco Bay Area, earn approximately 17% more than workers in other places across the US. These researches and reports give valuable information about the tech company salary in US in recent years; however, an overall review about the potential factors of tech salary is missing in the current researches.

Contributing to the literature gap, this project focuses on the potential factors relating to US tech company salaries, such as the employee's total year of experience, education attainment ratio, different job categories, and companies' geographic locations. As a conventional wisdom, we know that the longer time one work in a field, the person is more likely to get a higher salary, even promotion. We also know that skill-based positions (such as software engineers) usually make more money compared to non-skilled positions (such as HRs). Collected from the online database Kaggle, an open-source dataset of 2016 tech company salaries (Telle, 2017) is used in the project to test these conventional wisdoms, with two main independent variables (total years of experience and college education attainment), geographic dummies, and a group of job category dummies.

The linear regression and the machine learning model showed that total experience and college education ratio is positively correlated to tech salary, although the results are not very significant, suggesting the conventional wisdom is not validated. Besides, employees working in different job categories earn different salaries, yet most of the differences are insignificant. As for geographic locations, employees in the Western states (e.g. Washington and California) tend to have higher salaries, and their incomes keep increased even after they earn more than 100000 USD per year.

Additionally, an extension to the worldwide tech companies incorporated an outside data and found a slightly positive correlation between female participation ratio in labor force and tech company salary around the world. Some limitations of the project and restrictions of the dataset were also discussed at the end of the project. In the following parts, I will explain how I prepare the data and analyze the summary statistics and regression results, as well as showing visualizations that supports the findings.

1.2 Data Cleaning/Loading

In this part, I prepared the data with relevant information that can be easily analyzed by python packages. First, I selected the data with only US tech companies, deleted some irrelevant, descriptive information in the data, and dropped missing values. Second, I created a new column that shows the geographic region of each company (Eastern/ Western/ Central states). Last and most important, I defined tech company salary as the sum of an employee's annual base, signing bonus, and annual bonus received. During this process, I found an outlier with an extremely high income and dropped it from the dataset. After this step, the data is ready to be analyzed by statistical models.

[1]:
```
# uncomment below to install necessary packages
# %pip install pandas numpy matplotlib seaborn
# %pip install -U scikit-learn statsmodels
# %pip install geopandas xgboost gensim pyLDAvis descartes pycountry us
# %pip install stargazer

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import geopandas as gpd
from shapely.geometry import Point

import requests
import json
import pycountry
import us

import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML

import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
```

```
# read data from the local computer
csv_file = "/Users/cuimengyuan/Desktop/ECO225/ECO225Project/Data/salaries_clean.
 ↪csv"
data = pd.read_csv(csv_file)

# create a copy of the dataframe with US data only
US_data = data.copy()
US_data["location_country"] = US_data["location_country"].astype("string")
US_data = US_data[US_data["location_country"] == "US"]

# drop irrelevant columns
US_data = US_data.drop(columns = ["salary_id", "job_title_rank",␣
 ↪"stock_value_bonus", "comments", "submitted_at"])

# create a new column that shows total salary, clean missing values
salary = US_data["annual_base_pay"] + US_data["signing_bonus"] +␣
 ↪US_data["annual_bonus"]
US_data["salary"] = salary
US_data = US_data.dropna(subset = ["salary"])

# drop missing values in the x variables
US_data = US_data.dropna(subset =␣
 ↪["total_experience_years","employer_experience_years", "location_state"],␣
 ↪inplace = False)

# create a new column that shows geographic information of states (e.g.␣
 ↪coastlines vs. middle/central)
def geographic_info(state_name):
    if state_name in ["ME", "NH", "VT", "MA", "RI", "CT", "NY", "NJ", "PA",␣
 ↪"DE", "MD", "DC", "VA", "WV", "NC", "SC", "GA", "FL"]:
        return "Eastern"
    elif state_name in ["AL", "MS", "LA", "AR", "TN", "KY", "OH", "MI", "IN",␣
 ↪"IL", "WI", "MN", "IA", "MO", "ND", "SD", "NE", "KS", "OK", "TX"]:
        return "Central"
    elif state_name in ["CO", "WY", "MT", "ID", "UT", "NV", "AZ", "NM", "WA",␣
 ↪"OR", "CA", "AK", "HI"]:
        return "Western"
    else:
        return "Unknown"
US_data["geographic_info"] = US_data["location_state"].apply(geographic_info)

# drop an outlier from the y-variable
US_data = US_data[US_data["salary"] < 1000000]
```

1.3 Summary Statistics Tables

After removing missing values, there are 397 observations left in the dataset. The average salary earned among the sample is approximately 113286 USD, and the standard deviation of salary is about 61396 USD. Note some employees in the dataset are earning a zero wage, which prevents me from applying a log transformation to salary. The sample has on average 7 years of total experience (standard deviation = 5.77) and 3 years of experience at current company (standard deviation = 3.52). This pattern indicates employees are young in the technology industry and usually don't stay too long at the same company.

For the dummy variables, I found there are 35% of employees in the sample working in Eastern states, 43% working in Western states, and 22% working in Central states. As for their job categories, I found most of them belong to the "Software" category (52% of the sample), and the least are from the "Applied Science" category (0.25%). However, about 17% of the respondents are having jobs that are difficult to categorize, and thus were put in the "Other" category.

[3]:
```
# add geographic locations as dummy variables
US_data["Eastern"] = np.where(US_data["geographic_info"] == "Eastern", 1, 0)
US_data["Western"] = np.where(US_data["geographic_info"] == "Western", 1, 0)
US_data["Central"] = np.where(US_data["geographic_info"] == "Central", 1, 0)

# add job categories as dummy variables
US_data["Software"] = np.where(US_data["job_title_category"] == "Software", 1,␣
 ↪0)
US_data["Engineering"] = np.where(US_data["job_title_category"] ==␣
 ↪"Engineering", 1, 0)
US_data["Data"] = np.where(US_data["job_title_category"] == "Data", 1, 0)
US_data["Web"] = np.where(US_data["job_title_category"] == "Web", 1, 0)
US_data["Management"] = np.where(US_data["job_title_category"] == "Management",␣
 ↪1, 0)
US_data["Applied Science"] = np.where(US_data["job_title_category"] == "Applied␣
 ↪Science", 1, 0)
US_data["Other"] = np.where(US_data["job_title_category"] == "Other", 1, 0)
```

[4]:
```
# create a summary table for the main variables
summary_stats = {'Total Years of Experience': US_data['total_experience_years'].
 ↪describe(),
                 'Years of Experience at Current Employer':␣
 ↪US_data['employer_experience_years'].describe(),
                 'Salary': US_data['salary'].describe(),
                 'Eastern': US_data['Eastern'].describe(),
                 'Western': US_data['Western'].describe(),
                 'Central': US_data['Central'].describe(),
                 'Software': US_data['Software'].describe(),
                 'Engineering': US_data['Engineering'].describe(),
                 'Data': US_data['Data'].describe(),
                 'Web': US_data['Web'].describe(),
                 'Management': US_data['Management'].describe(),
```

```
                'Applied Science': US_data['Applied Science'].describe(),
                'Other': US_data['Other'].describe()}

# prepare the table
summary = pd.DataFrame(summary_stats).T
summary = summary.rename(columns={'count': 'No. Observations', 'mean': 'Mean',␣
 ↪'std': 'Standard Deviation', 'min': 'Min', 'max': 'Max'})
summary = summary.style.format("{:.2f}")

# add a title to the summary statistic table
summary.set_caption('Table 1 - Summary Statistics.')
```

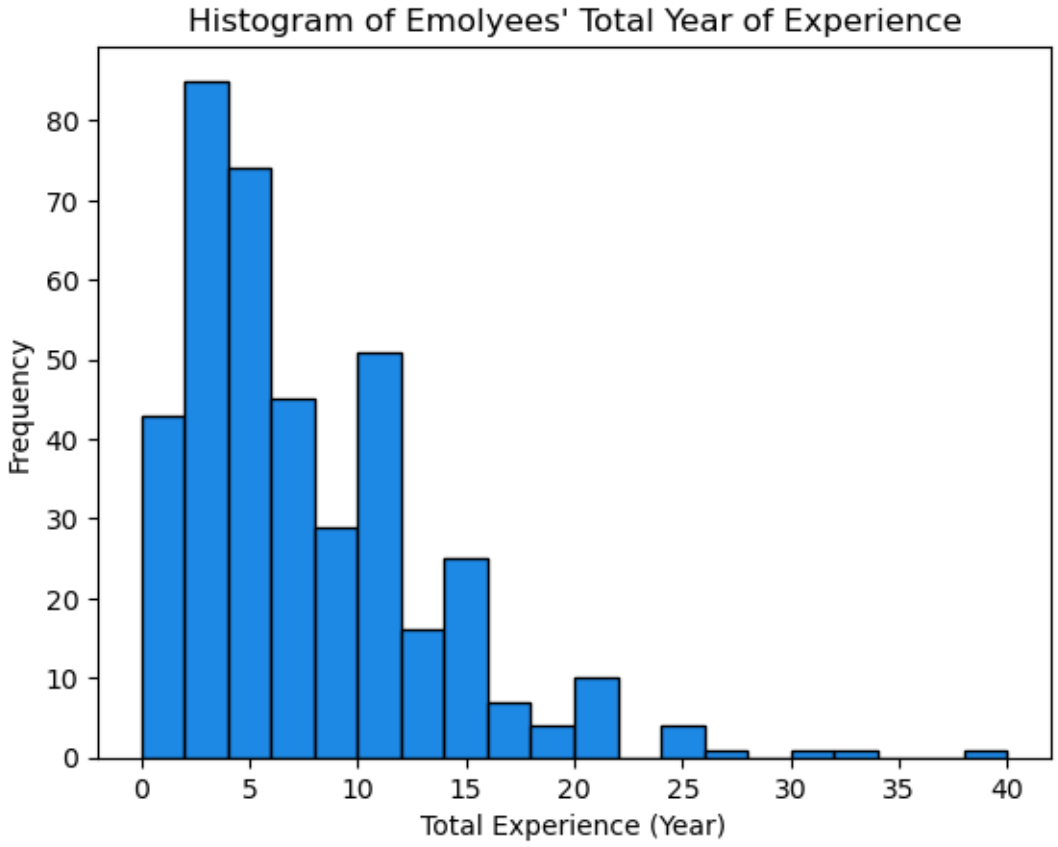
[4]: <pandas.io.formats.style.Styler at 0x7fb0d27dfdf0>

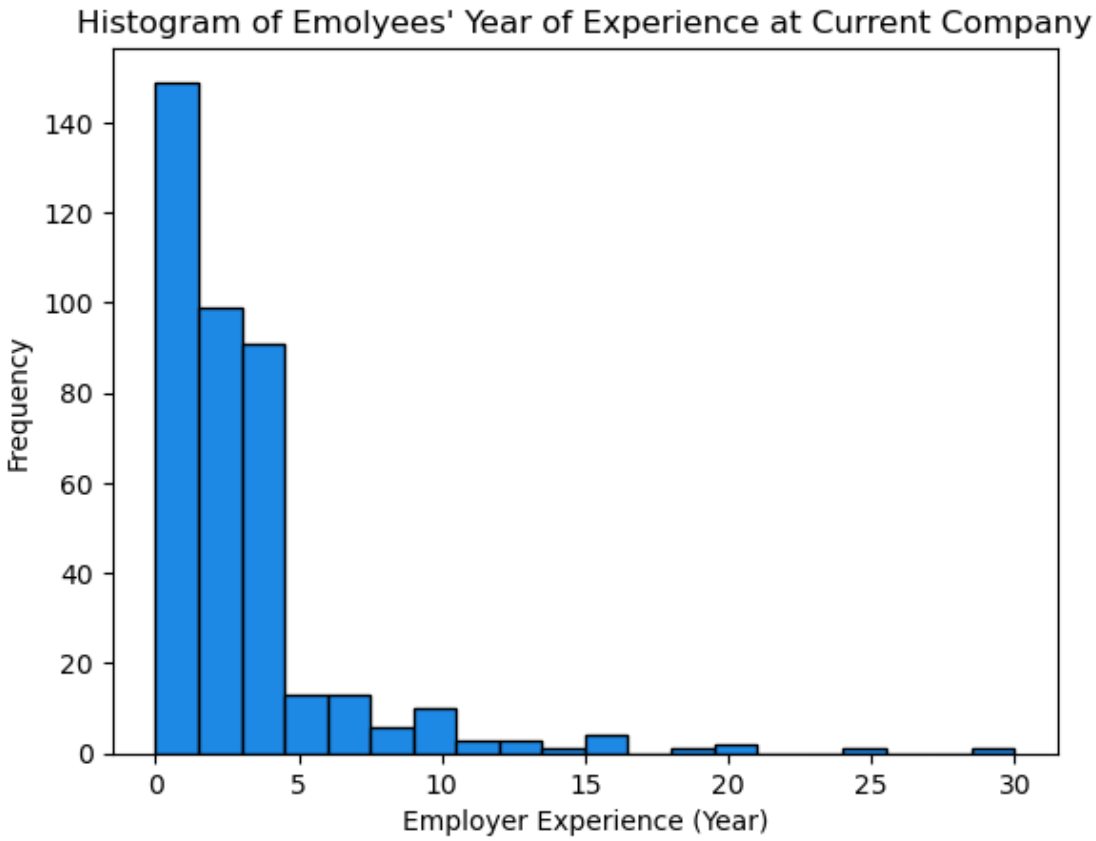
1.4 Plots, Histograms, Figures

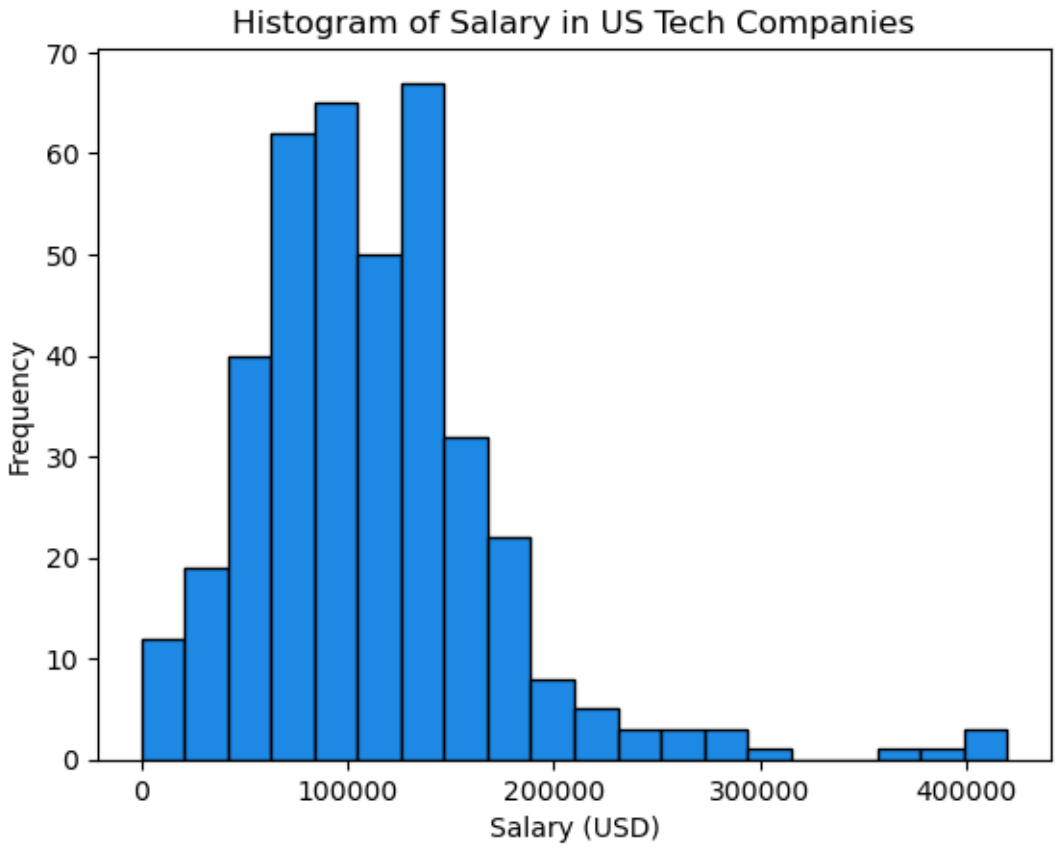
Continuing with summary statistics, let's have a glance at the distributions of the main x-variables and the y-variable, respectively. For the total year of experience, we can see most of the employees in US tech companies have 3~6 years of working experience. For the year of experience at current company, I found most employees stay in a company for less than 5 years. Considering information technology is a profession with relatively young people, these results generated from the histograms are not very surprising. Finally, for the salary paid to these employees, I found most of them have an annual salary of approx. 50000 to 150000 USD, higher than the median income in US (which is approx. 27419 USD in 2016, according to the United States Census Bureau). This finding aligns with our background information that people working in tech companies often have a decent income compared to other professions.

[5]:
```
# create a historgram for my 1st x-variable
plt.hist(US_data["total_experience_years"], bins = 20, color = '#1E88E5',␣
 ↪edgecolor = 'black')

plt.xlabel('Total Experience (Year)')
plt.ylabel('Frequency')
plt.title("Histogram of Emolyees' Total Year of Experience")
plt.show()

# create a historgram for my 2nd x-variable
plt.hist(US_data["employer_experience_years"], bins = 20, color = '#1E88E5',␣
 ↪edgecolor = 'black')

plt.xlabel('Employer Experience (Year)')
plt.ylabel('Frequency')
plt.title("Histogram of Emolyees' Year of Experience at Current Company")
plt.show()

# create a historgram for the y-variable
plt.hist(US_data["salary"], bins = 20, color = '#1E88E5', edgecolor = 'black')
```

```
plt.xlabel('Salary (USD)')
plt.ylabel('Frequency')
plt.title("Histogram of Salary in US Tech Companies")
plt.show()
```




Histogram of Emolyees' Year of Experience at Current Company
Frequency
0
20
40
60
80
100
120
140
0
5
10
15
20
25
30
Employer Experience (Year)



Then, I created two bin scatterplots to show the correlation between our two main x-variables and the y-variable, respectively. I used bin scatterplots to get the average of Xs in each bin and show the scatterplot in a cleaner and nicer way (where dots are not condensed). Both plots indicate a positive relationship between the year of experience (whether total or at the current company) and salary in US tech companies. However, the graph we drew showed two simple linear regressions with only one x-variable and the y-variable, respectively. We might get a different result if we put both the two x-variables into a multiple regression, since the different types of model may change the sign of the coefficient.

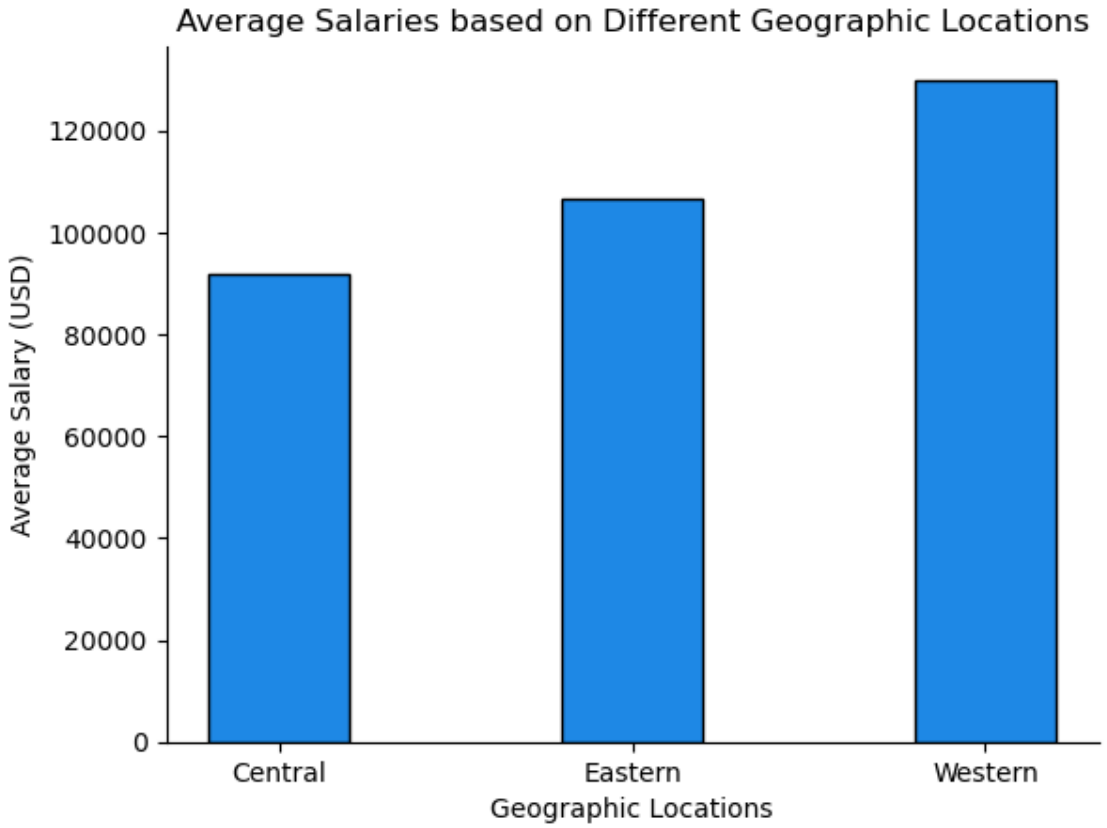
[6]:
```
# separate my 1st x-variable into 200 bins
US_data['binned'] = pd.cut(US_data['total_experience_years'], 200)
df = US_data[['binned', 'salary']].groupby(by='binned').mean().reset_index()
df['bin_mean'] = df.binned.apply(lambda x: x.mid)

# create bin scatterplot with regression line for 1st x-variable
sns.lmplot(data = df, x = "bin_mean", y = "salary")
plt.xlabel("Total Experience (Year)")
plt.ylabel("Salary (USD)")
plt.title("Relationship between Total Experience and Salary")
plt.show()
```

[7]:
```
# separate my 2nd x-variable into 200 bins
US_data['binned_1'] = pd.cut(US_data['employer_experience_years'], 200)
df1 = US_data[['binned_1', 'salary']].groupby(by='binned_1').mean().
 ↪reset_index()
df1['binned_mean'] = df1.binned_1.apply(lambda x: x.mid)

# create bin scatterplot with regression line for 2nd x-variable
sns.lmplot(data = df1, x = "binned_mean", y = "salary")
plt.xlabel("Employer Experience (Year)")
plt.ylabel("Salary (USD)")
plt.title("Relationship between Employer Experience and Salary")
plt.show()
```

Finally, two bar charts are generated to show different salaries in different geographic regions and for different job categories. Since the bar chart has the advantage of comparing different values in the same figure, we can clearly find the employees in Western states and working in the applied science category earn highest salary compared to others, and the employees in Central states and working in the data category have the least income compared to others.

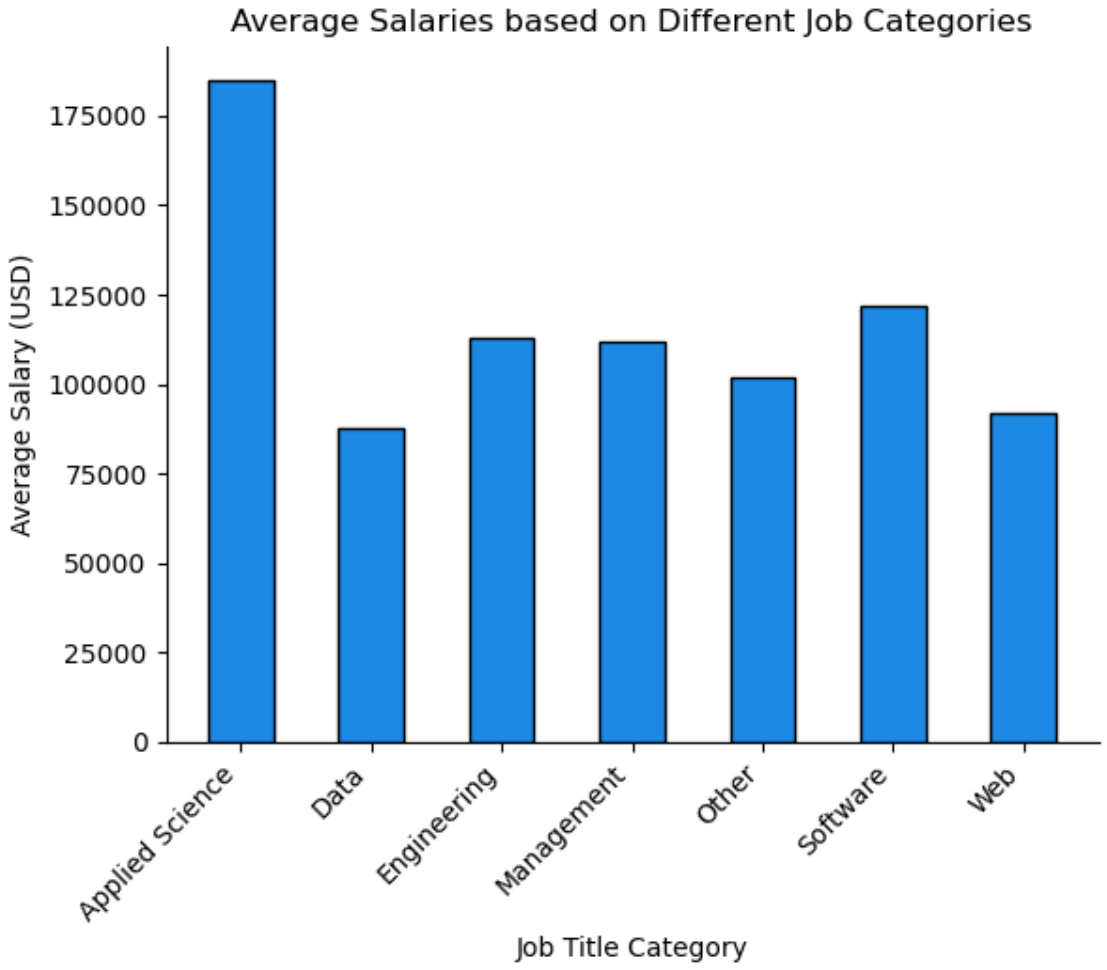
[9]:
```
# calculate average salaries for different geographical locations
average = US_data[['geographic_info', 'salary']].groupby(by="geographic_info").
 ↪mean()["salary"].reset_index()

# generate bar chart
x_axis = average["geographic_info"]
y_axis = average["salary"]
fig, ax = plt.subplots()

bar_width = 0.4
```

```
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
plt.bar(x_axis, y_axis, color = "#1E88E5", width = bar_width, align = 'center',␣
 ↪edgecolor = 'black')

plt.title('Average Salaries based on Different Geographic Locations')
plt.xlabel('Geographic Locations')
plt.ylabel('Average Salary (USD)')
plt.show()
```



[10]:
```
# calculate average salaries for each job title category
average_1 = US_data[['job_title_category', 'salary']].
 ↪groupby(by="job_title_category").mean()["salary"].reset_index()

# generate bar chart
x_axis = average_1["job_title_category"]
y_axis = average_1["salary"]
fig, ax = plt.subplots()

# specify bar width and distance between bars
```

```
bar_width = 0.5
bar_distance = 1.5

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
plt.bar(x_axis, y_axis, color = "#1E88E5", width = bar_width, align = 'center',␣
  ↪edgecolor = 'black')

# rotate the labels on x-axis
plt.xticks(rotation = 45, ha = 'right')

plt.title('Average Salaries based on Different Job Categories')
plt.xlabel('Job Title Category')
plt.ylabel('Average Salary (USD)')
plt.show()
```



2 Project Two

2.1 The Message

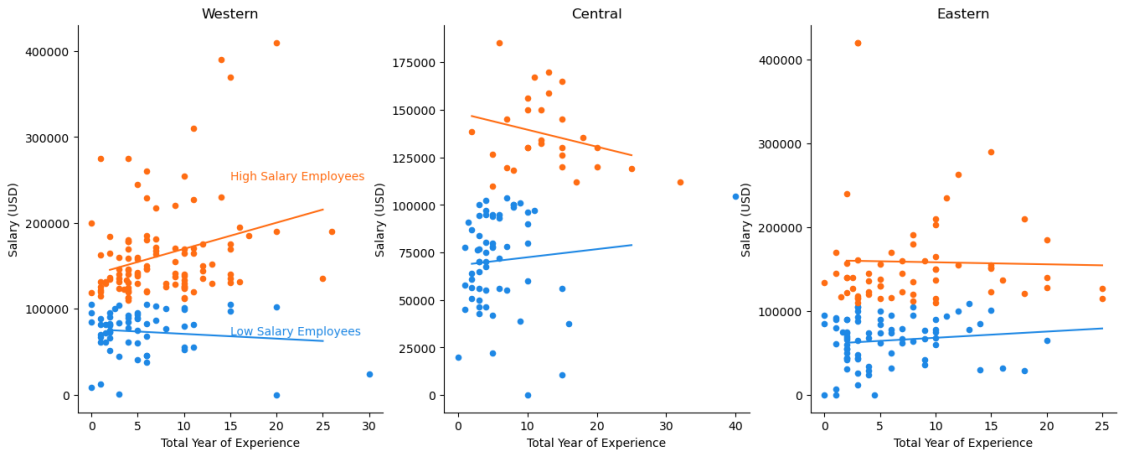
In this project, I want to show the relationships between salary and one of our major independent variables (total years of experience) based on different geographic regions of US tech companies (Eastern/Western/Central states). In the graph showing the main message, I divided salary into two subgroups by 108500 USD (the median salary in our data) to see if any pattern can be found between the high- and low-salary groups.

From the visualization, we clearly found the high-salary group only in Western states has increasing salaries with more years of working experiences, and has decreasing salary with more experiences in Eastern and Central states. However, this trend was reversed in the low-salary group: in Western states, salary is negatively correlated to total experience, while in Eastern and Central states, salary is positively associated to having more years of experience. The underlying reason is currently unclear, yet we can still get the intuition that tech companies in the Western states are somehow different from those in other states. Besides, some limitations need to be acknowledged: the graph has a limited number of variables so I could not add more controls into the regression, and we do not know whether these results are statistically significant.

[11]:
```
from sklearn.linear_model import LinearRegression

# separate salary into 2 groups
US_data['high_salary'] = np.where(US_data['salary'] > 108500, 1, 0)
US_data['high_salary'] = US_data['high_salary'].astype(str)

def single_scatter_plot(US_data, geographic_info, high_salary, ax, color,␣
 ↪plot_type):
    # filter data to keep only the data of interest
    _US_data = US_data.query("(geographic_info == @geographic_info) &␣
 ↪(high_salary == @high_salary)")
    _US_data.plot(
        kind = "scatter", x = "total_experience_years", y = "salary", ax = ax,␣
 ↪color = color
    )

    if plot_type == "low":
        lr = LinearRegression()
        X = _US_data["total_experience_years"].values.reshape(-1, 1)
        y = _US_data["salary"].values.reshape(-1, 1)
        lr.fit(X, y)

        x = np.linspace(2.0, 25.0).reshape(-1, 1)
        y_pred = lr.predict(x)
        if high_salary == "0":
            ax.plot(x, y_pred, color="#1E88E5")

    elif plot_type == "high":
```

```
        lr = LinearRegression()
        X = _US_data["total_experience_years"].values.reshape(-1, 1)
        y = _US_data["salary"].values.reshape(-1, 1)
        lr.fit(X, y)

        x = np.linspace(2.0, 25.0).reshape(-1, 1)
        y_pred = lr.predict(x)
        if high_salary == "1":
            ax.plot(x, y_pred, color="#ff6d13")

    return ax

# create initial plot
fig, ax = plt.subplots(1, 3, figsize = (16, 6), sharey = False)

# enumerate: create both index and value (provide location info about these␣
 ↪values)
for (i, geographic_info) in enumerate(US_data.geographic_info.unique()):
    single_scatter_plot(US_data, geographic_info, "0", ax[i], "#1E88E5", "low")
    single_scatter_plot(US_data, geographic_info, "1", ax[i], "#ff6d13", "high")
    ax[i].set_title(str(geographic_info))

for (i, _ax) in enumerate(ax):
    _ax.set_xlabel("Total Year of Experience")
    _ax.set_ylabel("Salary (USD)")
    _ax.spines['right'].set_visible(False)
    _ax.spines['top'].set_visible(False)

ax[0].annotate("High Salary Employees", (15, 250000), color="#ff6d13")
ax[0].annotate("Low Salary Employees", (15, 70000), color="#1E88E5")
```

[11]: Text(15, 70000, 'Low Salary Employees')



2.2 Maps and Interpretations

In this section, I added more visualizations to better represent the results generated from the data. A new geodataframe is created to plot a map, in which I merged another dataset with the geographic shapes of US continental states. Three maps were created: one for the outcome variable (salary), and two for our variables of interest (total_experience_years and employer_experience_years) in this project. Thanks to having the geographic information of US tech companies in our dataset, I labelled all companies with red dots in the maps.

[12]:
```
# get average salary for each state
average_salary = US_data.groupby('location_state')['salary'].mean()
state_salary = pd.DataFrame({'location_state': average_salary.index,␣
 ↪'average_salary': average_salary.values})

US_data_1 = pd.merge(US_data, state_salary, on = 'location_state', how = 'left')

# get average total and employer experience for each state
avg_total_experience = US_data_1.
 ↪groupby('location_state')['total_experience_years'].mean().reset_index()
avg_employer_experience = US_data_1.
 ↪groupby('location_state')['employer_experience_years'].mean().reset_index()

US_data_1 = US_data_1.merge(avg_total_experience, on = 'location_state', how =␣
 ↪'left')
US_data_1 = US_data_1.merge(avg_employer_experience, on = 'location_state', how␣
 ↪= 'left')

# make a geodataframe
US_data_1["Coordinates"] = list(zip(US_data_1.location_longitude, US_data_1.
 ↪location_latitude))
US_data_1["Coordinates"] = US_data_1["Coordinates"].apply(Point)

gdf = gpd.GeoDataFrame(US_data_1, geometry="Coordinates")
gdf.head()

# read a state map file from local computer
state_df = "/Users/cuimengyuan/Desktop/ECO225/cb_2016_us_state_5m/
 ↪cb_2016_us_state_5m.shp"
states = gpd.read_file(state_df)
to_drop = ["Alaska", "Puerto Rico", "Hawaii", "American Samoa", "Guam",␣
 ↪"Commonwealth of the Northern Mariana Islands", "United States Virgin␣
 ↪Islands"]
states = states[~states.NAME.isin(to_drop)]

# merge the two geodataframes
```

```
salary_w_states = states.merge(gdf, left_on = "STUSPS", right_on =␣
 ↪"location_state", how = "inner")
```

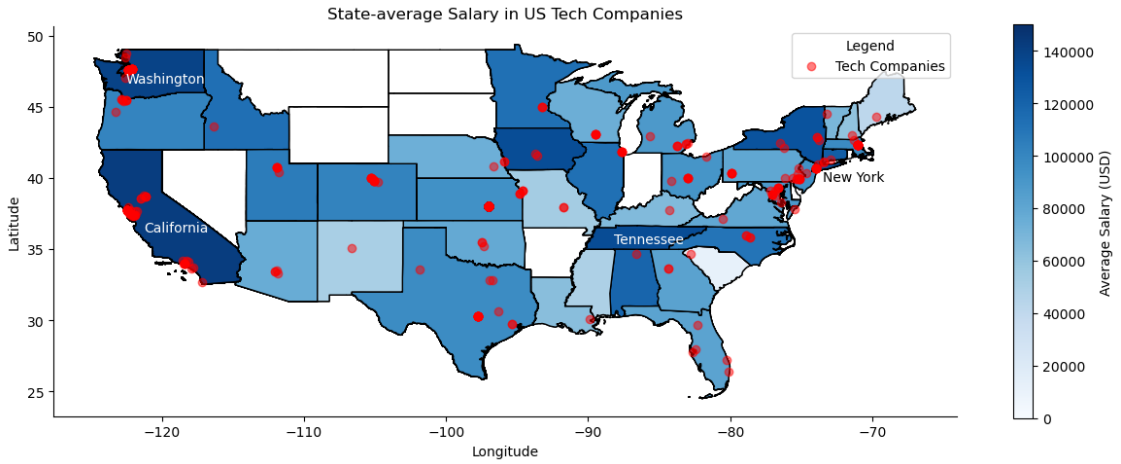
In the first map, tech company salaries are calculated by state and are shown by the values of blue color in the map. Align with our previous findings, Washington and California are the two states having the highest salary at state level, and both are Western states in the US. Some Eastern and Central states (such as New York and Tennessee) also have a high state-average income.

In the second map, total years of working experience are calculated by state and shown in the map. Similar to our previous findings, tech companies have many younger workers usually with less than 15 years of experience. The Western states recruit particularly young employees, with no more than 10 years of prior experience on average. Alabama stands out from the map, showing the companies are having the highest number of year of experience.

In the third map, the state-level number of years at current company are shown in the map. Align with the second graph and previous findings, tech companies have employees usually stay in their company for less than 5 years. Tennessee, Alabama, and Oklahoma stand out from the map, showing companies in these states are having employees stay in the same company for more than 5 years on average. This result is surprising compared to the employee mobility in all other states, and we may get a sense of Central states are having employees who stay longer in the same company.

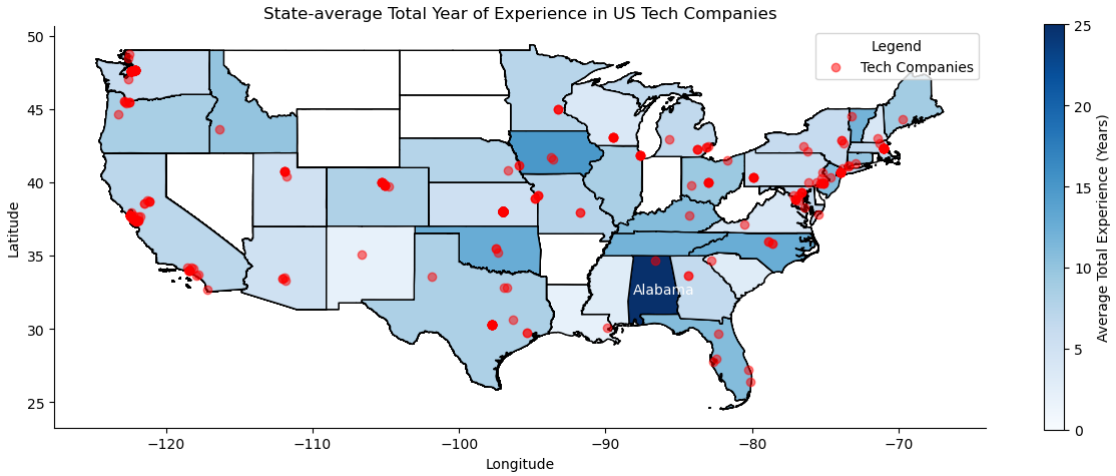
[13]:
```
# plotting the states map
fig, gax = plt.subplots(figsize = (15,15))

states.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

salary_w_states.plot(
    ax = gax, edgecolor = 'black', column = 'average_salary', legend = True,␣
 ↪cmap = 'Blues',
    vmin = 0, vmax = 150000, legend_kwds = {'label': "Average Salary (USD)",␣
 ↪'shrink': 0.35}
)

# add a second legend to label tech companies
gdf.plot(ax = gax, color = 'red', alpha = 0.5, label = 'Tech Companies')
plt.legend(frameon=True, title='Legend')

gax.annotate('Washington', xy = (0.08,0.88), ha = 'left', va = 'top', xycoords␣
 ↪= 'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('California', xy = (0.1,0.5), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('New York', xy = (0.85,0.63), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points')
gax.annotate('Tennessee', xy = (0.62,0.47), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')

gax.set_title('State-average Salary in US Tech Companies', fontsize=12)
```

```
gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



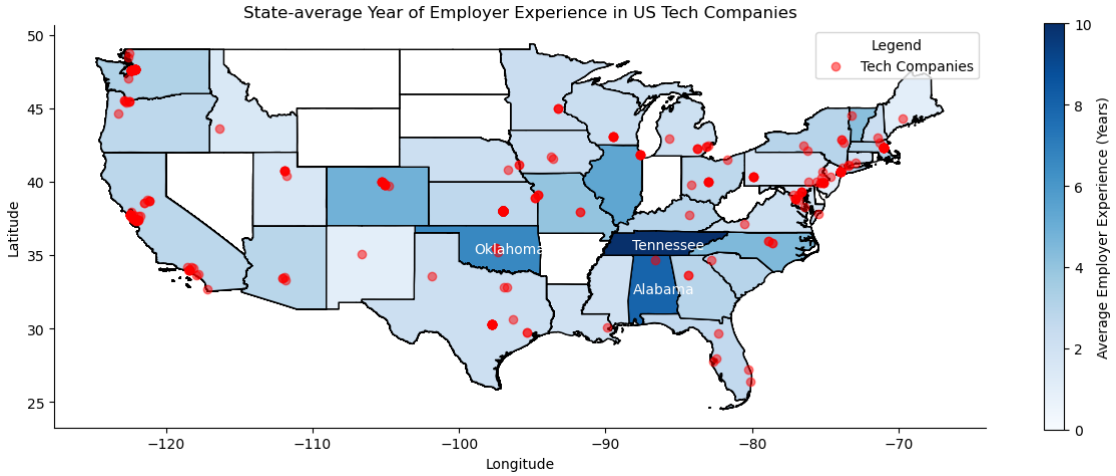
[14]:
```
fig, gax = plt.subplots(figsize = (15,15))

states.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

salary_w_states.plot(
    ax = gax, edgecolor = 'black', column = 'total_experience_years_y', legend␣
 ↪= True, cmap = 'Blues',
    vmin = 0, vmax = 25, legend_kwds = {'label': "Average Total Experience␣
 ↪(Years)", 'shrink': 0.35}
)

# add a second legend to label tech companies
gdf.plot(ax = gax, color = 'red', alpha = 0.5, label = 'Tech Companies')
plt.legend(frameon=True, title='Legend')

gax.annotate('Alabama', xy = (0.62,0.36), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')

gax.set_title('State-average Total Year of Experience in US Tech Companies',␣
 ↪fontsize=12)
gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')
```

```
gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



[15]:
```
fig, gax = plt.subplots(figsize = (15,15))

states.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

salary_w_states.plot(
    ax = gax, edgecolor = 'black', column = 'employer_experience_years_y',␣
 ↪legend = True, cmap = 'Blues',
    vmin = 0, vmax = 10, legend_kwds = {'label': "Average Employer Experience␣
 ↪(Years)", 'shrink': 0.35}
)

# add a second legend to label tech companies
gdf.plot(ax = gax, color = 'red', alpha = 0.5, label = 'Tech Companies')
plt.legend(frameon=True, title='Legend')

gax.annotate('Tennessee', xy = (0.62,0.47), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('Alabama', xy = (0.62,0.36), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('Oklahoma', xy = (0.45,0.46), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')

gax.set_title('State-average Year of Employer Experience in US Tech Companies',␣
 ↪fontsize=12)
gax.set_xlabel('Longitude')
```

```
gax.set_ylabel('Latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



3 Project Three

3.1 Potential Data to Scrape

The main dataset in this project is lacking two aspects of information: one is the demographic info about our population of interest, such as gender and education attainment ratio among employees in tech salaries; another is the information of employees' annual salary in years other than 2016, which can change the structure of our data to a panel data. Having knowledge in either aspect could provide more comprehensive information about tech company salary, resulting in more comprehensive analyses and more valuable insights.

Since the original dataset is extracted from a survey, I found it is difficult to find another survey with the same topic but in a different year. On the other hand, I found a a potential dataset from the US Bureau of Labor Statistics with demographic data that can be used to scrape and analyze. This dataset contains US employment data at state-level, with detailed information of employment ratio, age, race, and other demographics in each state over a long period of time. The data is called "State and Metro Area Employment" from the website: https://www.bls.gov/sae/data/.

I plan to scrape this data by using API-based scraping, since the website has provided public data API and detailed instruction on how to scrape the data. I will collect the demographic data only in year 2016 and merge with our main US dataset. The new dataset should contain additional information of employees' age, gender, race, and employment rate in the US labor force. By adding these variables into the regression model, I expect the multivariate model could provide a stronger prediction of salary in US tech companies. If these variables could better explain tech company salary, then the variables would become influential factors relating to salary.

3.2 Potential Challenges

Although the employment data can largely satisfy our needs for additional information, I found the data has a population of entire labor force in US, which cannot perfectly match to our population of interest (employees in tech companies). Besides, this data does not contain a clear classification for jobs in tech industry. For example, "Information" and "Business" can both describe some positions in a tech company, depending on the nature of each position. Considering the 7 job categories we used in previous sections, I found it difficult to include them into one of the eight industries given in the employment dataset, such as the difference between data analysts and managers. As a result, although the Bureau of Labor Statistics has Public Data API allowing users to access its data, I chose to not scrape this data and use it into the analysis.

3.3 Scraping Data from a Website

Instead of the data I sourced and introduced above, I used the data of female participation rate in labor force provided by professor and incorporated it with our main dataset. This data is provided by the World Bank with panel data on country-level female participation in the labor force. In other words, it has data of US as well as other countries. Thus, in the following sections, I will diverge from our focus on US tech companies and extend to discuss the relationship between female participation rate and tech company salary around the world.

The first step is to scrape data from the World Bank website. Here I used API-based scraping to access the data for year 2016 (since our main dataset only contains information in 2016) and stored it into a dataframe/ csv file with the name "fem_ratio".

[16]:
```
# set up the API endpoint and parameters from website provided: https://data.
 ↪worldbank.org/indicator/SL.TLF.CACT.FE.ZS
endpoint = 'http://api.worldbank.org/v2/country/all/indicator/SL.TLF.CACT.FE.ZS'
params = {
    'format': 'json',
    'per_page': 16492, # showing all 16492 observations in one page
    'page': 1
}

# send a request to the API endpoint
response = requests.get(endpoint, params = params)

# parse the JSON response into python object
info = response.json()

# extract data from the response object
result = []
for entry in info[1]:
  country = entry['country']['id']
  year = entry['date']
  if year == '2016':  # only keep data for year 2016
        value = entry['value']
        if value is not None:
            result.append({'country_code': country, 'female_ratio': value})
```

```
# create a dataframe to store the results
fem_ratio = pd.DataFrame(result, columns = ['country_code', 'female_ratio'])
print(fem_ratio)

# save the dataframe as a CSV file
fem_ratio.to_csv('female_ratio.csv', index = False)
```

```
    country_code  female_ratio
0             ZH     64.564024
1             ZI     54.813358
2             1A     21.727881
3             S3     53.833807
4             B8     48.744986
..           …           …
230           VI     48.749001
231           PS     17.416000
232           YE      5.924000
233           ZM     51.918999
234           ZW     60.617001

[235 rows x 2 columns]
```

3.4 Merging the Scraped Dataset

In the second step, I did an outer merge to combine the female ratio data with the main dataset of this project based on the two-letter country code (which is the ISO Alpha-2 country code). To differentiate the current world salary dataset from the US data, I reread the dataset, cleaned the data, and named the new dataset after merging as "world_ratio". Before merging the datasets, I have a total of 1181 observations in the world salary data, which is almost tripled observations compared to the previous US data (which has only 397 observations). However, doing the outer merge with 'world_ratio' dataset did not increase the number of observations in my main dataset. This is because all countries listed in the female ratio data are already included in the world salary dataset, and nothing new were added into the data.

[17]:
```
# in this part, we will use the world data, not the US data in the previous␣
 ↪sections
data = pd.read_csv("/Users/cuimengyuan/Desktop/ECO225/ECO225Project/Data/
 ↪salaries_clean.csv")

# prepare the world data
data = data.drop(columns = ["salary_id", "job_title_rank", "stock_value_bonus",␣
 ↪"comments", "submitted_at"])

salary = data["annual_base_pay"] + data["signing_bonus"] + data["annual_bonus"]
data["salary"] = salary
data = data.dropna(subset = ["salary"])
```

```
data = data.dropna(subset =␣
 ↪["total_experience_years","employer_experience_years"], inplace = False)

# change the country column into string type
data["location_country"] = data["location_country"].astype("string")

# merge the world dataset with web-scraping data
world_ratio = data.merge(fem_ratio, left_on = 'location_country', right_on =␣
 ↪'country_code', how = 'outer')

# look at the number of observations in the data
world_ratio.count()
```

[17]:
```
index                        1181
employer_name                1178
location_name                1181
location_state                408
location_country              562
location_latitude             562
location_longitude            562
job_title                    1181
job_title_category           1181
total_experience_years       1181
employer_experience_years    1181
annual_base_pay              1181
signing_bonus                1181
annual_bonus                 1181
salary                       1181
country_code                  747
female_ratio                  747
dtype: int64
```

3.5 Visualizing the Scraped Dataset

In the third step, I created some visualizations for the "world_ratio" dataset containing worldwide tech company salaries as well as the percentage of female workers in labor force. Similar methods were used as in project 1 and 2, here I generated a geodataframe to plot world maps, hoping to get some insights about female participation rate across countries.

However, due to limitations of the data sourced, we only have country-level data of female participation. We need to acknowledge that the female participation ratio in labor force is measuring a different thing from the female ratio in tech industry only. (To give you a preview: after having some practice with web scraping, we will go back to analyze the US data in the next section).

[18]:
```
# make a geodataframe for the world data
```

```
world_ratio["Coordinates"] = list(zip(world_ratio.location_longitude,␣
 ↪world_ratio.location_latitude))
world_ratio["Coordinates"] = world_ratio["Coordinates"].apply(Point)

geodf = gpd.GeoDataFrame(world_ratio, geometry="Coordinates")

# read the world map with ISO alpha-2 country code
world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))
world["iso_a2"] = world["iso_a3"].apply(lambda x: pycountry.countries.
 ↪get(alpha_3=x).alpha_2 if pycountry.countries.get(alpha_3=x) else None)
world = world.set_index("iso_a2")

# merge the world map with the world geodataframe
world_w_fem = world.merge(geodf, left_on="iso_a2", right_on="country_code",␣
 ↪how="inner")
```

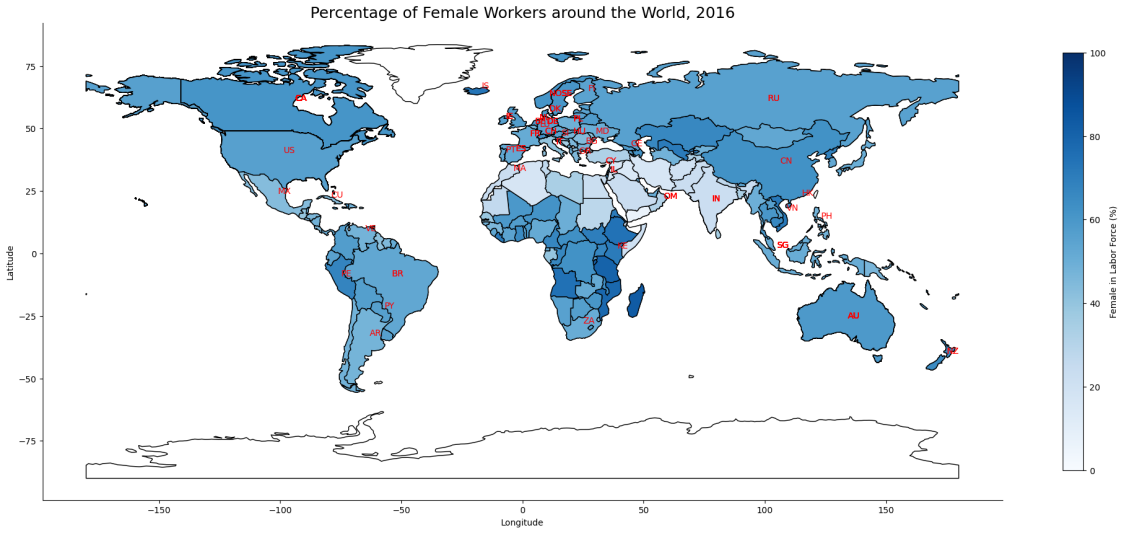
From the first world map, we found countries along the Southeast coast of Africa (e.g. Kenya) are having the highest percentage of female in labor force. On the other hand, India and some North Africa and Middle East countries are having the lowest female participation ratio, possibly due to religious and cultural influence in these regions. Countries in Asia, Europe, North and South Americas, and Oceania show a relatively balanced labor force participation by gender.

From the second world map, we found tech companies are not equally distributed around the world. According to the dataset, tech companies are widely found in North and South America, Europe, Oceania; some are located in Asian countries (e.g. China, India, Philippines, and Singapore), or located in only a few of African and Middle East countries. While studying tech company salaries, I found some outliers with extremely large values, and applied a log transformation to salary to create a better visualization. From this map I found India has the highest salary at country-level, yet this result may be driven by having extremely large outliers. Note: since the world dataset does not have precise geographic locations for tech companies outside of the US, we have only one latitude and longitude information for countries other than US, resulting having one red dot labelled on the countries.

Note in both maps, I labeled countries having tech companies in our dataset with country names in red.

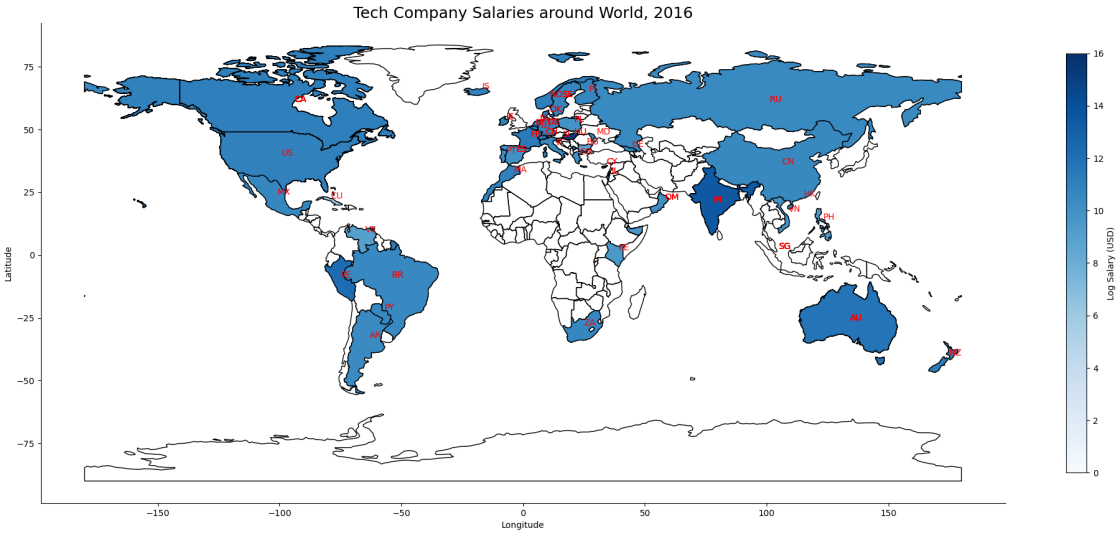
[19]:
```
# plotting the world map with female participation ratio in labor force
fig, gax = plt.subplots(figsize = (25,25))

world.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

world_w_fem.plot(
    ax = gax, edgecolor = 'black', column = 'female_ratio', legend = True, cmap␣
 ↪= 'Blues',
    vmin = 0, vmax = 100, legend_kwds = {'label': "Female in Labor Force (%)",␣
 ↪'shrink': 0.35}
)
```

```
# label US with a single indicator
us_geodf = geodf[geodf['country_code'] == 'US']
if not us_geodf.empty:
    x_mean = us_geodf['Coordinates'].x.mean()
    y_mean = us_geodf['Coordinates'].y.mean()
    gax.annotate('US', xy=(x_mean, y_mean), xytext=(3,3), color='red',␣
 ↪textcoords='offset points')

for x, y, label in zip(geodf['Coordinates'].x, geodf['Coordinates'].y,␣
 ↪geodf['country_code']):
    if not pd.isna(label):
        if label != 'US':
            gax.annotate(label, xy=(x,y), xytext=(3,3), color='red',␣
 ↪textcoords='offset points')

gax.set_title('Percentage of Female Workers around the World, 2016',␣
 ↪fontsize=18)
gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



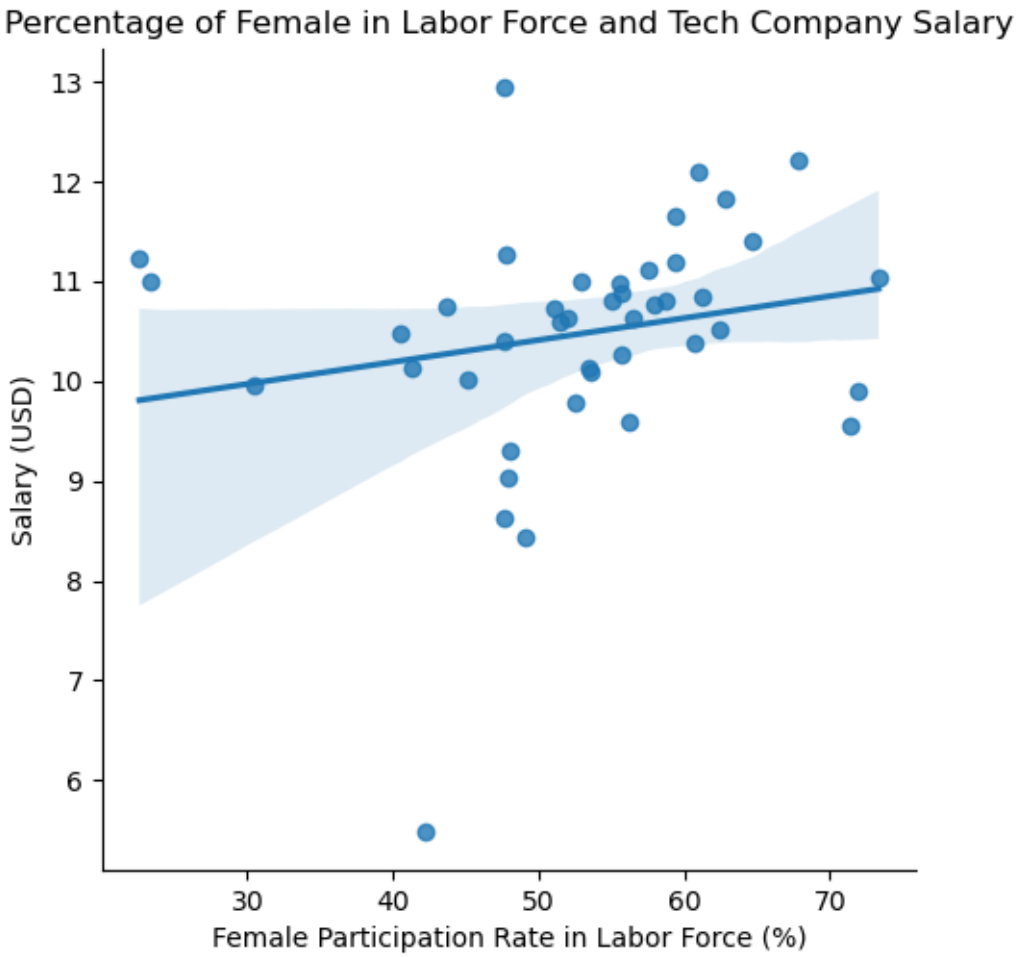
[20]:
```
# because of extremely big outliers, we apply log transformation to salary
world_w_fem["salary_log"] = np.log(world_w_fem["salary"])

# drop an outlier in salary (as well as in log salary) and plot the map again
```

```
world_w_fem = world_w_fem[world_w_fem["salary_log"] < 20]

# plotting the world map
fig, gax = plt.subplots(figsize = (25,25))

world.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

world_w_fem.plot(
    ax = gax, edgecolor = 'black', column = 'salary_log', legend = True, cmap =␣
 ↪'Blues',
    vmin = 0, vmax = 16, legend_kwds = {'label': "Log Salary (USD)", 'shrink':␣
 ↪0.35}
)

# label US with a single indicator
us_geodf = geodf[geodf['country_code'] == 'US']
if not us_geodf.empty:
    x_mean = us_geodf['Coordinates'].x.mean()
    y_mean = us_geodf['Coordinates'].y.mean()
    gax.annotate('US', xy=(x_mean, y_mean), xytext=(1,1), color='red',␣
 ↪textcoords='offset points')

for x, y, label in zip(geodf['Coordinates'].x, geodf['Coordinates'].y,␣
 ↪geodf['country_code']):
    if not pd.isna(label):
        if label != 'US':
            gax.annotate(label, xy=(x,y), xytext=(3,3), color='red',␣
 ↪textcoords='offset points')

gax.set_title('Tech Company Salaries around World, 2016', fontsize=18)
gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



After looking at the maps, I took a step further to see if there is a relationship between the percentage of female in labor force and salary, after applying log transformation to salary. Here I also used a bin scatter plot to group the female participation ratio into different bins to make the plot looks cleaner and nicer. In the plot below, I found female participation ratio seems to have a slightly positive correlation with log salary, yet some heavy outliers are messing the regression. Again, this can be caused by the limitation of the female ratio dataset, which only has information of female workers compared to the entire labor force, not among tech companies around the world.

[21]:
```
# draw a bin scatterplot for salary and female participation ratio
df2 = world_w_fem[['female_ratio', 'salary_log']].groupby(by=['female_ratio']).
  ↪mean().reset_index()

sns.lmplot(x = "female_ratio", y = "salary_log", data = df2)

plt.title('Percentage of Female in Labor Force and Tech Company Salary')
plt.xlabel('Female Participation Rate in Labor Force (%)')
plt.ylabel('Salary (USD)')
plt.show()
```



3.6 Adding a New Dataset

In this part, we will move from the extension and focus again on the US tech company data. In previous part - Potential data to scrape - I discussed the possibility of adding some demographic data into our main dataset, and one of them is educational attainment ratio among tech company employees. In this part, I added a data of college education attainment (people having their first college degree) among population of each state, as a proxy for the education attainment among tech company employees. Since higher education is often correlates with a higher salary in the labor force (as a conventional wisdom), I hope there is a positive relationship between the newly added variable and annual salary, and if so, I may conclude that education attainment is an explanatory factor of tech company salary.

Specifically, I merged two datasets then added them into the US data: one is the number of people who have a college degree (named "Field of Bachelor's Degree for First Major"), and the other is the 2016 census data showing the number of populations in each state. Both data are from the American Community Survey and have been download from the website of US Census Bureau. However, as I mentioned earlier, we need to acknowledge the dataset is having some limitations such as focusing on the entire state population (i.e. with unemployed workers and people out of

labor force), instead of having a specified group of people working in tech industry.

[22]:
```
# use the educational attainment data that I downloaded
education = pd.read_csv('/Users/cuimengyuan/Desktop/ECO225/ECO225Project/Data/
 ↪education_attainment_2016.csv')

# drop irrelevant rows
education = education.iloc[:1]

# drop irrelevant columns
need_drop = education.filter(regex='Margin of Error')
education = education.drop(need_drop.columns, axis=1)

# switch rows and columns
education = education.transpose()
education = education.iloc[1:]

# add an index and clean the data
education = education.rename(columns={0: "educ_pop"})
education.insert(0, "states", education.index)
education = education.reset_index(drop=True)
education['states'] = education['states'].str.replace('!!Total!!Estimate', '')

education.head()
```

[22]:

	states	educ_pop
0	Alabama	815,522
1	Alaska	141,577
2	Arizona	1,335,894
3	Arkansas	445,833
4	California	8,660,470

[23]:
```
# read the 2016 census data to get population
population = pd.read_csv('/Users/cuimengyuan/Desktop/ECO225/ECO225Project/Data/
 ↪population_2016.csv')

# prepare the data
population = population.iloc[:2]

population = population.transpose()
population = population.iloc[1:]

population = population.drop(population.columns[0], axis=1)

population = population.rename(columns={1: "population"})
population.insert(0, "states", population.index)
population = population.reset_index(drop=True)
```

```
population['states'] = population['states'].str.replace('!!Estimate', '')

population.head()
```

[23]:

	states	population
0	Alabama	4,863,300
1	Alaska	741,894
2	Arizona	6,931,071
3	Arkansas	2,988,248
4	California	39,250,017

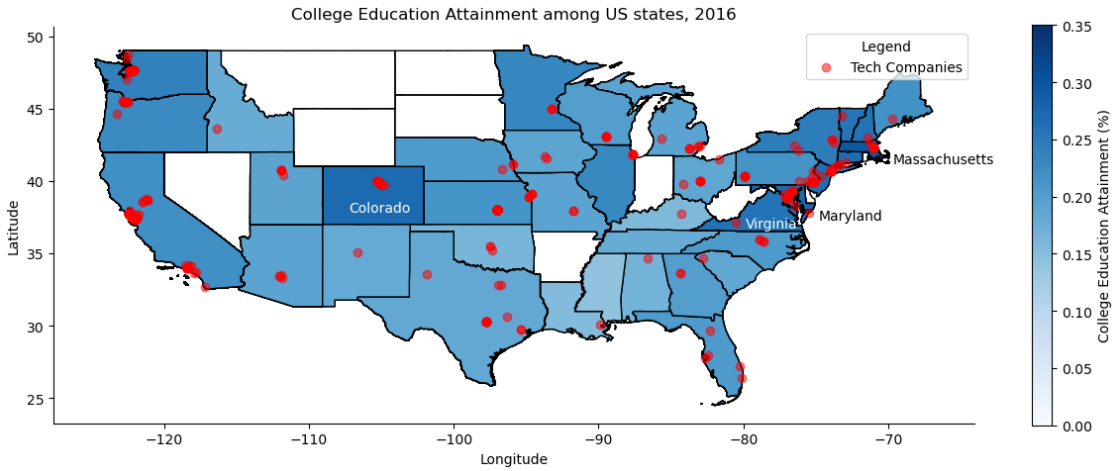
After getting the two datasets, I merged them to get the percentage of people having college degree among state population. Then, I merged the dataset with the percentage ratio with the US dataset that have been used in project 1 and 2. Note the US dataset "salary_w_states" is already a geodataframe, so we don't need additional data for visualization in the next step (i.e. can directly use to plot a US map).

[24]:
```
# merge the two datasets to get educational attainment rate
educ_attain = pd.merge(education, population, on = 'states', how = 'left')

# formatting the values in two columns
educ_attain['educ_pop'] = educ_attain['educ_pop'].str.replace(',', '').
 ↪astype(float)
educ_attain['population'] = educ_attain['population'].str.replace(',', '').
 ↪astype(float)

# add a new column to get the percentage
educ_attain['percent'] = educ_attain['educ_pop'] / educ_attain['population']

# add state codes
educ_attain['state_code'] = educ_attain['states'].apply(lambda x: us.states.
 ↪lookup(x).abbr)

# finally merge the education rate data with the US data
US_w_educ = salary_w_states.merge(educ_attain, left_on = 'location_state',␣
 ↪right_on = 'state_code', how = 'left')
```

In the visualization part, I plot a US map based on the percentage of people attaining college education among state population in each state. Note the red dots in the graph are showing the location of tech companies, similar to in previous graphs. Generally speaking, Eastern states have relatively higher education attainment ratio, especially in the 13 founding states such as Massachusetts, New York, Maryland, and Virginia. The three west-coast states (Washington, Oregon, California) also have relatively higher education attainment compared to the central area. Among the Central states, Colorado stands out from the map with a higher percent of people having college degrees. Remember what we conclude in project 1 and 2? Tech companies in the Eastern and Western states are paying on average a higher salary to their employees, compared to those in the Central states. According to Kulkarni et al. (2022), tech unicorns (i.e. companies

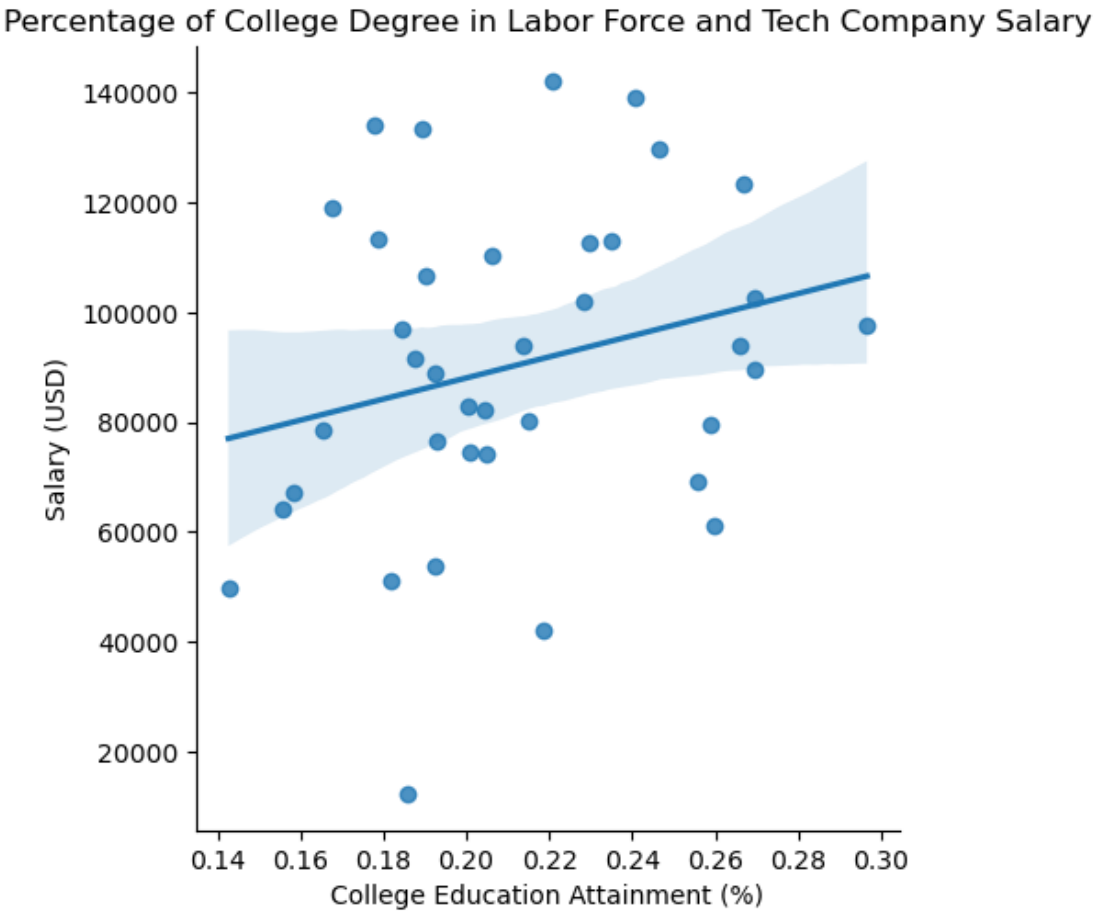
with market value over $1 billion without being listed on the stock market) are actively expanding and looking for locations where they can absorb the most talented workers. The map here may also reveal this finding that employers would like to choose states with relatively more educated population as their companies' location, which may benefit the company by easier reaching out to and recruiting people with higher education.

[25]:
```
# plotting the states map
fig, gax = plt.subplots(figsize = (15,15))

states.plot(ax = gax, edgecolor = 'black', facecolor = 'none')

US_w_educ.plot(
    ax = gax, edgecolor = 'black', column = 'percent', legend = True, cmap =␣
 ↪'Blues',
    vmin = 0, vmax = 0.35, legend_kwds = {'label': "College Education␣
 ↪Attainment (%)", 'shrink': 0.35}
)

# add a second legend to label tech companies
gdf.plot(ax = gax, color = 'red', alpha = 0.5, label = 'Tech Companies')
plt.legend(frameon=True, title='Legend')

gax.annotate('Colorado', xy = (0.32,0.56), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('Massachusetts', xy = (0.91,0.68), ha = 'left', va = 'top',␣
 ↪xycoords = 'axes fraction', textcoords = 'offset points')
gax.annotate('Virginia', xy = (0.75,0.52), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points', color = 'white')
gax.annotate('Maryland', xy = (0.83,0.54), ha = 'left', va = 'top', xycoords =␣
 ↪'axes fraction', textcoords = 'offset points')

gax.set_title('College Education Attainment among US states, 2016', fontsize=12)
gax.set_xlabel('Longitude')
gax.set_ylabel('Latitude')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

plt.show()
```



Finally, I drew a bin scatter plot to further investigate the relationship between educational attainment and tech company salaries, and indeed found a positive relationship between the two variables. This result matches the Harvey Nash Survey (2015) by saying tech companies would like to attract and retain the best workers with required skills to catch up, embrace, even lead the change in technology. Besides, the conventional wisdom of having higher education linked to a higher income is also tested to be true in the context of US tech industry. However, outliers in the data are messing the correlation, making the result less significant.

[26]:
```
# draw a bin scatterplot for salary and education attaiment ratio in US
df3 = US_w_educ[['percent', 'salary']].groupby(by=['percent']).mean().
 ↪reset_index()

sns.lmplot(x = "percent", y = "salary", data = df3)

plt.title('Percentage of College Degree in Labor Force and Tech Company Salary')
plt.xlabel('College Education Attainment (%)')
plt.ylabel('Salary (USD)')
plt.show()
```



4 Final Project

4.1 Linear Regression

In this part, I will investigate the relationship between potential factors (e.g. total years of experience, education attainment, job categories, etc.) of salary and the annual salary of employees working in US tech industry. Since total year of experience and employer year of experience has some common characteristics, I dropped the later one from the regression to avoid any collinearity issue. There are two main x-variables in the regression models: one is employee's total year of experience in tech industry, and the other is the percentage of people having their first college degree among population (as a proxy of employee's educational attainment). For the first variable, from the conventional wisdom we know the longer one works in an industry, the more likely one will earn higher salary. For the second variable, we know people with higher education are more likely to get a higher pay. According to these conventional wisdoms, I expect to find positive relationships between each explanatory variable and the salary. Before making any model specifications, I assume there is a linear relationship between these factors and the outcome variable (salary). Additionally, the geographic state fixed effects and the job category fixed effects will be included in some models to better interpret the main variables of interest. Finally, I will try a model specification with

quadratic terms and compare the results with linear models to see if the specification could lead to a higher predictability of the model.

[27]:
```
# add constant term to dataset
US_w_educ['Const'] = 1

# add quadratic term to dataset
US_w_educ['total_sqre'] = US_w_educ['total_experience_years_y']**2
US_w_educ['educ_sqre'] = US_w_educ['percent']**2

# define different sets of X-variables in each regression
X1 = ['Const', 'total_experience_years_y', 'percent']
X2 = ['Const', 'total_experience_years_y', 'percent', 'Western', 'Central']
X3 = ['Const', 'total_experience_years_y', 'percent', 'Western', 'Central',␣
 ↪'Software', 'Engineering', 'Data', 'Management', 'Applied Science', 'Other']
X4 = ['Const', 'total_experience_years_y', 'percent', 'total_sqre',␣
 ↪'educ_sqre', 'Western', 'Central', 'Software', 'Engineering', 'Data',␣
 ↪'Management', 'Applied Science', 'Other']

# estimate the regressions for each set of variables
reg1 = sm.OLS(US_w_educ['salary'], US_w_educ[X1], missing='drop').fit()
reg2 = sm.OLS(US_w_educ['salary'], US_w_educ[X2], missing='drop').fit()
reg3 = sm.OLS(US_w_educ['salary'], US_w_educ[X3], missing='drop').fit()
reg4 = sm.OLS(US_w_educ['salary'], US_w_educ[X4], missing='drop').fit()
```

Starting with the baseline model, I use total years of experience and the college educational attainment as two independent variables in a multiple regression model to study the relationship between these variables and the outcome variable (salary). The baseline model has the following equation:

Note: for some reason my LaTex is not working properly, so I manually add the notations and labelled them in red.

$$Salary_{ij} = \beta_0 + \beta_1 * TotalExperience_i + \beta_2 * EducationAttainment_j + \epsilon_{ij}$$

Where $ β1 $ and $$\beta$2 $ *are the coefficients for the two independent variables, and* $$\epsilon${ij} $ is the error term. The subscript $ {i} $ indicates the observation is on individual level and the subscript $ {j} $ indicates the observation is at state-level.

In Table 2 Model (1), I found the annual salary of one with zero years of experience working in US tech company is approximately 60654 USD. Note this is a statistically and economically significant result: the coefficient has p-value smaller than 0.05 (passing the 5% significant level) and has practical interpretations showing the entry salary in the US tech industry. After controlling education attainment, having one more year of total experience is associated with on average a 653 USD increase in salary. This result matches our conventional wisdom that the longer time stayed in an industry, the more income will earn. However, the coefficient is not significant. It is not statistically significant because the p-value failed to pass even the 10% significant level, and not the economically significant once we considered the average salary shown in the summary statistic table: a one standard deviation difference in the x-variable is associated with only 6.14% of one standard deviation difference in salary.

As for the other independent variable, I found a 1 percentage point increase in employee's college

education attainment ratio is correlated with approximately 2102 USD increase in salary. Note this result is statistically significant at 5% significance level (p-value less than 0.05) but economically insignificant due to only having 10.27% difference in one standard deviation of salary. Overall, the coefficient of determination (R-square) in this model is close to zero (0.01), indicating there is almost no correlation and we should not be surprised by having inconclusive result, also suggesting more variables and controls should be added into the regression to better explain the y-variable.

Continue with the previous model, in Model (2) I added the geographic dummies (Eastern/Western/Central) into the regression, hoping to increase the explaining power of the overall model. Note the Eastern states dummy was left-out from the regression to make comparison. Here is the regression formula:

$$Salary_{ij} = \beta_0 + \beta_1 * TotalExperience_i + \beta_2 * EducationAttainment_j + \delta_1 * Central_i + \delta_2 * Western_i + \epsilon_{ij}$$

Where $ δ1 $ and $ δ2 $ are the coefficients for the two geographic dummies.

After controlling for geographic effects, the annual salary of one with zero experience and no college degree has slightly increased to approximately 63736 USD, which is statistically significant at 10% significance level (p-value < 0.10). After controlling all other variables, having an extra year of total experience is now associated with on average a 1611 USD increase in salary, higher than the previous estimation in Model (1). Note the result is again statistically and economically insignificant. As for education variable, I surprisingly found an insignificant result, in which increasing 1 percentage point of college education attainment correlates with a 1280 USD salary increase, lower than in Model (1). The R-square in this model has increased to 0.07 (with adjusted R-square increased to 0.06), suggesting an improvement in model explanation power. Although the results are not very informative, at least I am on the right track of explaining what relates to tech company salaries.

To have a closer look at the difference in geographic regions, I found the tech companies in Western states are paying way more salaries as employees work longer, with approximately 2831 USD higher than the average salary paid in Eastern states. The companies in the Central states pay the least to their employees, which is about 9814 USD less than the average salary in Eastern states. However, note only the result of the Western states is statistically significant (with p-value smaller than 0.01). This notable difference is likely because Western states have some regions famous for developed tech industries and big-name companies, such as Meta in San Francisco Bay Area and Microsoft in Seattle. This result can be supported by Echeverri-Carroll & Ayala (2009) study, in which they found a spatial concentration of tech companies in specific regions, such as the Silicon Valley, where companies are competing on human resources and are willing to pay more salary to their employees.

Similar to the previous step, in Model (3) I added a group of 7 job category dummies (such as engineering, software, management, etc.) to control for the job category fixed effects. Note the dummy "web" is used as the omitted variable in the regression to compare the results. The regression formula can be found below:

Apologize for the inconvenience again!

$$Salary_{ij} = \beta_0 + \beta_1 * TotalExperience_i + \beta_2 * EducationAttainment_j + \delta_1 * Central_i + \delta_2 * Western_i + \delta_3 * AppliedScience_i + \ldots + \delta_8 * Software_i + \epsilon_{ij}$$

Where $ δ3 $ to $ δ8 $ are the coefficients for the job category dummies.

After controlling all other variables, the annual salary of one with zero year of experience is now decreased to on average 55737 USD, and an extra year of total experience is now associated with an approximately 1356 USD increase in salary. Having 1 percentage point increase in education attainment is associated with an even-lower, on average 919 USD increase in salary. However, all these coefficients are not significant. After controlling all other variables, employees working in the web category earn the least salary compared to in other categories, and those work as applied scientists have the highest salary, which is about 75791 USD more than 'web' employees. However, only the coefficient of "software" category is statistically significant (with p-values passed the 10% significant level), suggesting working as a software engineer may be an influential factor to tech salary. Although the overall model has R-square further increased to 0.09 (adjusted R-square increased to 0.07), the F-statistic in this model is lower than Model (2), also suggesting an overall not very effective improvement in model's predictability.

Since Model (3) doesn't perform a substantial improvement in our regression model, I reconsidered the model and chose a nonlinear specification with quadratic terms (total experience squared and education attainment squared) to further explore the results. Note this is still a linear regression model but having non-linear features (the quadratic terms).

Apologize for the inconvenience again!

$$Salary_{ij} = \beta_0 + \beta_1 * TotalExperience_i + \beta_2 * EducationAttainment_j + \gamma_1 * (TotalExperience_i)^2 + \gamma_2 * (EducationAttainment_j)^2 + \delta_1 * Central_i + \delta_2 * Western_i + \delta_3 * AppliedScience_i + \ldots + \delta_8 * Software_i + \epsilon_{ij}$$

Where $ γ1 $ and $ γ2 $ are the coefficients of the quadratic terms.

In the final model (Model 4), the model intercept is uninterpretable because it changed to a negative value due to the model specification (no salary is negative). After controlling all other variables, an extra year of total experience is now correlated with on average 2646 USD increase in salary, and a 1 percentage point increase in education attainment is associated with on average 42541 USD increase in salary (very surprising!). Note the coefficients for total experience and its squared variable are both statistically and economically insignificant, and the coefficients for education attainment and its squared value are both significant (with p-value < 0.01; a one standard deviation difference in the education is associated with more than 2 standard deviation differences in salary). As for the controls, the Western state dummy and the software category dummy are still statistically significant as in the previous model. The overall model also shows an improvement in R-square (further increased to 0.12) and adjusted R-square (increased to 0.09), suggesting the nonlinear specification better fits the data and enhance the explaining power of the model.

[28]:
```
# add number of observations and MSE in output table
info_dict={'No. of Observations' : lambda x: f"{int(x.nobs):d}", 'Mean Squared␣
 ↪Error': lambda x: f"{x.mse_resid:.3f}"}

# create the output table, set model names and variables order
results_table = summary_col(results=[reg1,reg2,reg3,reg4],
                            float_format='%0.2f',
                            stars = True,
                            model_names=['Baseline',
                                         'Geographic FE',
                                         'Job Category FE',
                                         'Nonlinear Specification'],
                            info_dict=info_dict,
```

```
                            regressor_order=['Const',
                                             'total_experience_years_y',
                                             'total_sqre',
                                             'percent',
                                             'educ_sqre',
                                             'Central',
                                             'Western'])

results_table.add_title('Table 2 - Regression Outputs.')

print(results_table)
```

```
                                Table 2 - Regression Outputs.
=============================================================================
==============
                            Baseline    Geographic FE  Job Category FE Nonlinear
Specification
-----------------------------------------------------------------------------
--------------
Const                     60653.97**     63735.57*      55737.34
-436424.86***
                          (29007.19)     (36458.02)     (38225.85)
(154717.59)
total_experience_years_y 652.91          1611.25        1356.17         2645.66
                          (1559.43)      (1536.15)      (1542.57)       (4469.00)
total_sqre                                                               -24.35
                                                                         (217.38)
percent                   210186.59**    128001.46      91926.30
4254151.43***
                          (105113.16)    (134288.23)    (134226.42)
(1292773.38)
educ_sqre
-8831092.47***
(2720996.90)
Central                                  -9813.60       -11339.36       -1804.32
                                         (10997.41)     (11043.23)
(11423.64)
Western                                  25831.45***    23003.33***
18519.97**
                                         (7271.88)      (7317.19)       (7407.28)
Applied Science                                         75790.72        75253.04
                                                        (60999.38)
(60340.34)
Data                                                    1781.47         3720.57
                                                        (18684.34)
(18517.64)
Engineering                                             19460.59        22092.70
```

```
                                                          (15967.78)
(15856.33)
Management                                                18936.17        21893.18
                                                          (17431.65)
(17260.92)
Other                                                     8699.95         10465.21
                                                          (15540.11)
(15394.23)
Software                                                  26671.62*
29123.85**
                                                          (14278.53)
(14148.08)
R-squared                 0.01            0.07            0.09            0.12
R-squared Adj.            0.01            0.06            0.07            0.09
No. of Observations       397             397             397             397
Mean Squared Error        3750448983.666  3551119058.994  3515708482.159
3437167251.346
================================================================================
==============
Standard errors in parentheses.
* p<.1, ** p<.05, ***p<.01
```

[29]:
```
# use stargazer to generate regression outputs
stargazer_1 = Stargazer([reg1, reg2, reg3, reg4])
stargazer_1.title('Table 2 - Regression Outputs.')

# rename the models
stargazer_1.custom_columns(['Baseline', 'Geographic FE', 'Job Category FE',␣
 ↪'Nonlinear'], [1, 1, 1, 1])

# set variables order, rename some variables
stargazer_1.covariate_order(['Const', 'total_experience_years_y', 'total_sqre',␣
 ↪'percent', 'educ_sqre', 'Central', 'Western', 'Applied Science', 'Data',␣
 ↪'Engineering', 'Management', 'Other', 'Software'])
stargazer_1.rename_covariates({'total_experience_years_y': 'Total year of␣
 ↪experience', 'percent': 'Education Attainment'})

# show two decimal points in the output
stargazer_1.significant_digits(2)
stargazer_1.show_degrees_of_freedom(False)

HTML(stargazer_1.render_html())
```

[29]: <IPython.core.display.HTML object>

4.2 Machine Learning

In the final part of the project, I used machine learning to provide accurate mapping from my independent variables to the output variable (which is salary). The regression model is similar to the linear regression models in the previous part, but with location longitude and latitude variables added. And since the decision tree is a non-parametric model, here I do not need to drop the employer years of experience, Eastern states dummy, and Web category dummy to avoid collinearity. My goal is to minimize the mean squared errors (MSE) in the regression (i.e. the difference between predicted value and the actual value). The objective function of the regression analysis is shown below:

$$\min_{j,s}\left[\sum_{i:x_{i,j}\leq s, x_i\in R1}(y_i-\hat{y}_{R1})^2+\sum_{i:x_{i,j}>s, x_i\in R2}(y_i-\hat{y}_{R2})^2\right]$$

In this function, $ R1 $ and $ R2 $ are both from a rectangular region $ R $ that contains all values of X. By iteration, a feature and location ($ {j} $ and $ {s} $) is chosen to split $ R $ into two parts to minimize MSE, and this process is repeated with $ R1 $ and $ R2 $ and the smaller rectangles they produced. By iterating this process, finally I got a regression tree with all branches generated by machine learning. How does the model know when to stop iterating? This is because I chose a maximum depth of the tree, which is 3. The iteration would stop when the depth of tree reached this value.

A relating concept of the stopping rule is regularization and pruning. Regularization process penalizes a regression tree if it has extra nodes and branches, but does not increase the model predictability (i.e. it's overfitting the data). Pruning is a way to regularize a regression and avoid model overfit. A penalty function often includes regularization parameters such as the minimum leaf size, maximum depth of tree, and $$\alpha$$ in the pruning process. Below is the penalty function used in my regression:

$$\min_{tree\subset T}\sum(\hat{f}(x)-y)^2+\alpha|\text{terminal nodes in tree}|$$

Where $$\alpha$$ is the pruning parameter, usually between 0 and 1. Increasing the value of $$\alpha$$ would lead to a smaller tree and vice versa. As for another important regularization parameter, the maximum tree depth in our regression is set to be 3, which limits the number of terminal nodes in the tree. Increasing the maximum tree depth would lead to a larger tree, often with smaller MSE, but may be overfitting the data. With the regression tree created, I found the mean squared error (MSE) of the tree with depth of 3 is 3082745152, which is better (smaller) than the MSE of linear regression models in the previous part (see Table 2). Although I can further decrease the MSE in regression by increasing the tree depth, I chose not to do so to avoid overfitting because I don't want the model to have a low prediction power.

[30]:
```
# set X and Y for the regression tree
X = US_w_educ[['location_latitude', 'location_longitude',␣
 ↪'total_experience_years_y', 'employer_experience_years_y', 'percent',␣
 ↪'Eastern', 'Western', 'Central', 'Software', 'Engineering', 'Data', 'Web',␣
 ↪'Management', 'Applied Science', 'Other']]
Y = US_w_educ['salary']
```

```
# form a regression tree
from sklearn import tree
regtree = tree.DecisionTreeRegressor(max_depth = 3).fit(X,Y) # set no more than␣
 ↪3 levels of the tree

# use the fitted tree to predict
y_pred_tree = regtree.predict(X)

# find the error of prediction (MSE)
from sklearn import metrics
print('Mean Squared Error:', metrics.mean_squared_error(Y, y_pred_tree))
```

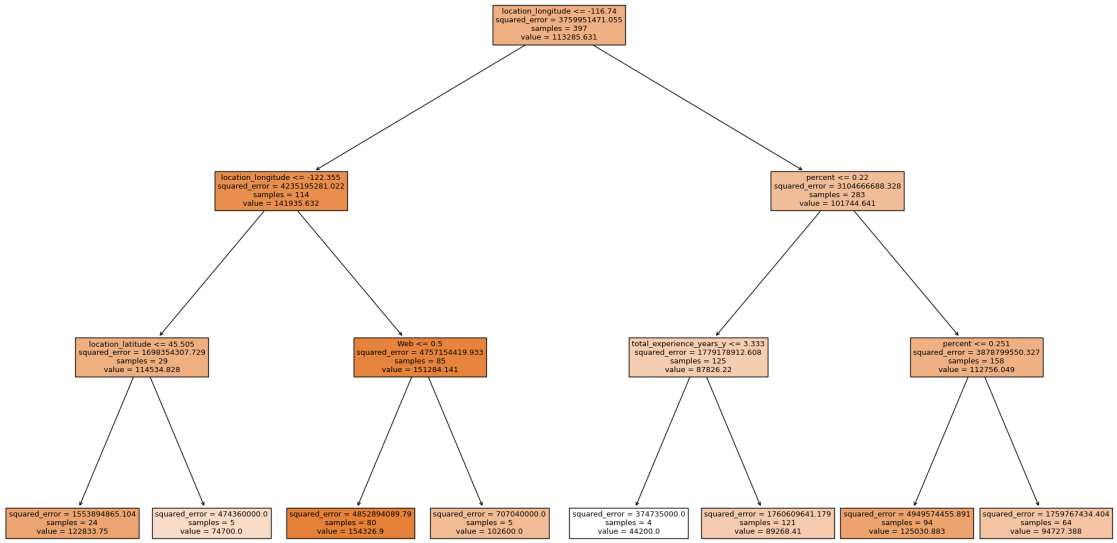
```
Mean Squared Error: 3082745151.5968194
```

In the regression tree below (with max tree depth of 3), I found the most contributed variable in regression is the location longitude of US tech companies. The data is separated into two groups, one with longitude less than or equal to -116.74 (i.e. US West Coast including Washington, Oregon, and California), and the other group with longitude more than -116.74. This result matches the previous finding in linear regression, suggesting a 40191 USD difference in salary between Western and other states (for example, the Bay Area is in the West!). Note the longitude variable is not included in the linear regressions because its numerical value has no meanings while interpreting, and I changed the company locations into 3 categorical, dummy variables (Eastern/ Western/ Central states). However, since regression trees do not impose linearity or monotonicity, the longitude and latitude variables can be used as a feature to separate data.

According to the regression tree, educational attainment further separates the sample at the East of longitude -116.74. Employees in states with college education attainment ratio greater than 22% are earning a 24930 USD higher salary on average. This finding also matches the result in the linear regression models, suggesting the importance of having a higher education/ living in a state with more educated people. As for the sample at the West of longitude -116.74, the longitude -122.36 further divide the sample into two groups. This line of longitude separates the well-famous high-tech regions such as Seattle and the Silicon Valley from the East of the states. However, due to the small sample size (one group with 85 observations and another group has only 29 observations), I cannot conclude an insightful result.

Speaking of the number of observations in each subgroup, a regression tree can show the divisions of data by showing the size of the subgroups. For example, the longitude variable separates the sample into two groups: one has 114 observations and the other contains 283 observations. However, the number of observations at each terminal node is no greater than 121. Having such a small sample size at each node is not good for the data analysis, making me difficult to determine if a particular result is true and unbiased.

[31]:
```
# draw the regression tree
regfig = plt.figure(figsize=(25,15))
regfig = tree.plot_tree(regtree, feature_names=X.columns, filled=True)
```



Finally, I used a random forest to find the best regression that minimizes the mean squared error. Random forest is a machine learning algorithm combining the output of several decision trees to attain a single result. A random sample draw from the set of independent variables are selected at each node, then the tree chooses the best split variable from the sample. By having different groups of x-variables, random forest provides a solution of having similar/ correlated trees, which may result in little improvement of the regression.

In this model, a group of 5 x-variables were randomly selected at each node. By iteration I found the best model with MSE of 1917663837, which is only 2/3 of the error in the previous tree! This substantial decrease in MSE shows a great benefit of having a random forest, which forces the trees to split on different variables and avoid having common nodes.

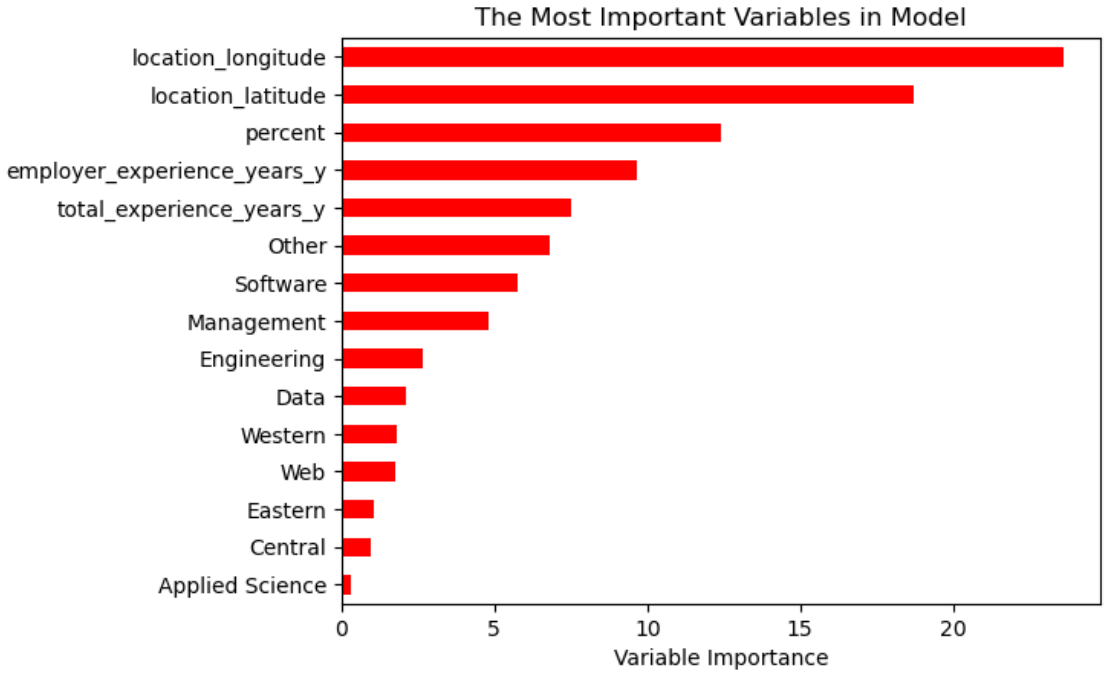
[32]:
```
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier,␣
 ↪BaggingRegressor, RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error,confusion_matrix,␣
 ↪classification_report

# select 5 variables in X to create a random forest
regr2 = RandomForestRegressor(max_features=5, random_state=1)
regr2.fit(X, Y)
pred = regr2.predict(X)
mean_squared_error(Y, pred)
```

[32]:
```
1917663837.1937025
```

From the random forest result, the algorithm records the amount of averaged reduction in mean

squared error due to splits, among all regression trees. If a splitting variable chosen from the set of x-variables leads to a large reduction in MSE, we conclude this variable is important. Thus I plotted a variable importance figure to show which variables are important in reducing the MSE in the regression. From this graph I found the location variables (longitude and latitude data) are mostly important in explaining tech company salary. Following the geographic variables, the third important variable is college education attainment ratio among population, similar to the linear regression results. The experience-related variables (total year of experience and the year of experience at current employer) are not the most important factors relating to salary, ranking at 4th and 5th on the variable importance graph. Although the linear regression also generates insignificant results for the two variables, this result is still surprising because it contradicts our conventional wisdom: it's not always the case that the longer one stayed in an industry/ a position, the more money one earned. Following the two experience variables, two job category dummies ('Other' and 'Software') are ranking the 6th and 7th on variable importance, partially matched to the linear regression results (where 'Software' category has statistically significant coefficients).

[33]:
```
# draw a variable importance graph to show which variables are more important
Importance = pd.DataFrame({'Importance':regr2.feature_importances_*100},␣
 ↪index=X.columns)
Importance.sort_values('Importance', axis=0, ascending=True).plot(kind='barh',␣
 ↪color='r')

plt.xlabel('Variable Importance')
plt.title('The Most Important Variables in Model')
plt.gca().legend_ = None
```



4.3 Limitation and Conclusion

This project investigates the potential factors (such as total year of experiences, education attainment, company locations, and job categories) of US tech company salaries, aiming to find any meaningful relationship between these factors and salary. With the two independent variables and a group of geographic and job category dummies, the linear regression and machine learning results showed that geographic locations and education attainment are the two main factors of tech company salary, suggesting companies are willing to pay a higher salary in Western states and/or to employees with a college degree. Doing a software job is also found to have a higher income. To my surprise, there is no substantial results showed that experience is correlated to salary, which contradicted to the conventional wisdom that the longer one worked, the more money one earned.

However, this project has some caveats and many results are inconclusive. One possible reason of the insignificant results can be having a small sample size (with only 397 observations in the dataset), leading to large standard errors. A further study can increase the sample size by adding more observations into the data. For example, one can find the tech salary data in different years to create a panel data, which would improve the significance of regression results and show a trend over time.

The second limitation is lacking control variables (such as age, gender, race and ethnicity among employees) in the regressions, and the regressions may suffer from omitted variable bias. For example, according to Lee (2013), Asian immigrants working in tech industry are experiencing some patterns between ethnic niches (i.e. companies are disproportionately owned and/or staffed by ethnic minorities) and their salary. To address the omitted variable bias, future studies can add related information and run regressions with the more comprehensive data. A potential dataset, State and Metro Area Employment from the US Bureau of Labor Statistics, has state-level demographic data of employees' age, gender, race, and employment rate in US labor force that can be used to scrape and analyze. By adding these variables into the regression model, the multivariate model should provide a stronger prediction of salary in US tech companies. However, this data has a population of entire US labor force, which cannot perfectly match to the population of employees in tech companies. Future researchers can also use a dataset with population that can better match to the topic.

The third limitation is also about the data itself. Since the data only has tech salary in year 2016, it could not give us the most recent trend in the field, resulting in less valuable insights. For example, what are the changes in salary during the pandemic? In this project I found a relatively competitive labor market in tech industry, where companies are paying high salary to more educated people. Is this trend become more prevalence during Covid, when more people are working from home? According to Normandin Beaudry (2022), IT companies in Canada are planning to increase average salaries to 5.8% increase in 2023, as an evidence of adapting to the competitive labor market. Would US tech companies do the same and would these firms continue to increase their salary in future? Future research could source a similar dataset containing tech salaries in the post-Covid era, and use the panel data to implement a difference in difference design to study the effect of the pandemic on tech salary and answer the above questions.

Last but most importantly, since the data does not randomly assign the independent variables to salary (the survey respondents are not a random sample), this project only captured some correlations between the potential factors and salary, but failed to investigate any causal effects. To solve this problem, future studies can collect the data from a random sample with individual-level education data to investigate if the potential factors are actually affecting tech salary. For example,

education attainment may have a causal effect on tech salary, but it could not be randomly assigned. Besides, motivation is possibly a confounding variable affecting both education attainment and salary: people with higher education tend to be highly motivated, yet this motivation can also lead them to more competitive positions and thus higher salaries. If researchers can find a proper instrumental variable correlated with education attainment, the instrumental variable can control for the confounding motivation and thus can estimate the true treatment effect of education on salary.

4.4 Reference

Echeverri-Carroll, E., & Ayala, S. G. (2009). Wage Differentials and the Spatial Concentration of High-technology Industries. Papers in Regional Science, 88(3), 623–641. https://doi.org/10.1111/j.1435-5957.2008.00199.x

IT Skills Crisis Drives Tech Salaries Up, Companies Struggle to Retain Talent, Says Harvey Nash Survey. (2015). PR Newswire Association LLC.

Kulkarni, A. U., Murkute, P. A., Kamble, H., & Sinha, K. (2022). AnExploratory Study on Salary Bubble & its repercussions on employee retention ontechies of Indian Edutech and Tech Companies. International Journal of Early Childhood Special Education, 14(5), 2281–2286. https://doi.org/10.9756/INTJECSE/V14I5.238

Lee, J. C. (2013). Employment and Earnings in High-Tech Ethnic Niches. Social Forces, 91(3), 747–784. https://doi.org/10.1093/sf/sos199

Normandin Beaudry. (2022). Salary increases: Organizations are constantly adapting. Cision Canada. Retrieved from https://www.newswire.ca/news-releases/salary-increases-organizations-are-constantly-adapting-868298820.html

Telle, B. (2017). 2016 Hacker News Salary Survey Results. [Data set]. Retrieved 2023, from https://data.world/brandon-telle/2016-hacker-news-salary-survey-results.

United States Census Bureau. (2022). American Community survey 5-year data (2009-2021). Retrieved 2023, from https://www.census.gov/data/developers/data-sets/acs-5year.html.

U.S. Bureau of Labor Statistics. (2023). State and Metro Area Employment, Hours & Earnings. [Data set]. Featured SAE Searchable Databases. Retrieved 2023, from https://www.bls.gov/sae/data/.

U.S. Census Bureau. (2023). American Community Survey (ACS) Demographic and Housing Estimates. [Data set]. Retrieved 2023, from https://data.census.gov/table?q=population%2Bin%2B2016&g=010XX00US%240400000.

U.S. Census Bureau. (2023). Field of Bachelor's Degree for First Major. [Data set]. Retrieved 2023, from https://data.census.gov/table?q=s1502&g=010XX00US$0400000.

Wheelwright, T. (2021). Top tech salaries in the US. Business.org. Retrieved from https://www.business.org/hr/benefits/highest-tech-salaries/.

The World Bank. (2023). Labor force participation rate, female (% of female population ages 15+) (modeled ILO estimate). The World Bank Data. Retrieved 2023, from https://data.worldbank.org/indicator/SL.TLF.CACT.FE.ZS.