DATA 608 - Assignment 4: How much do we get paid?

I have introduced the term "Data Practitioner" as a generic job descriptor because we have so many different job role titles for individuals whose work activities overlap including Data Scientist, Data Engineer, Data Analyst, Business Analyst, Data Architect, etc.

For this story we will answer the question, "How much do we get paid?" Your analysis and data visualizations must address the variation in average salary based on role descriptor and state.

Kaggle Link: https://www.kaggle.com/datasets/juanmerinobermejo/data-jobs-dataset/code

Load Dependencies

In [170…
```
# load dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import ticker
```

Data Loading

In [150…
```
# load job data pulled from Kaggle's US Data Jobs Salaries Dataset
job_url = "https://raw.githubusercontent.com/eddiexunyc/data_608_story_4/refs/heads/main/Resources/jobs.csv"
job_data = pd.read_csv(job_url, encoding = "utf-16", error_bad_lines=False).drop(['ID'], axis=1)

# check data type
job_data.dtypes
```

```
/var/folders/h4/zjq554hs0b57vqfcrc5738wh0000gn/T/ipykernel_95322/2629541512.py:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version.
Use on_bad_lines in the future.


  job_data = pd.read_csv(job_url, encoding = "utf-16", error_bad_lines=False).drop(['ID'], axis=1)
```

Out[150]:
```
Job                object
Jobs_Group         object
Profile            object
Remote             object
Company            object
Location           object
City               object
State              object
Salary             object
Frecuency_Salary   object
Low_Salary        float64
High_Salary       float64
Mean_Salary       float64
Skills             object
dtype: object
```

In [201…
```
# remove Virgin Island and Guams since they are not realy US states
location_remove = ['GU', 'VI', 'PR']
# job_data = job_data.loc[job_data['State'] != 'GU']
job_data = job_data[~job_data['State'].isin(location_remove)]

job_data.head()
```

Out[201]:

	Job	Jobs_Group	Profile	Remote	Company	Location	City	State	Salary	Frecuency_Salary	Low_Salary	High_Salary	Mean_Salary	Skills
0	Business Analyst	Business Analyst	NaN	NaN	CyberCoders	Torrington, CT 06790	Torrington	CT	80,000–110,000 por año	año	80000.0	110000.0	95000.0	[]
1	RPA Business Systems Analyst	Business Analyst	NaN	NaN	Amerihealth	Philadelphia, PA 19107 (City Center East area)...	Philadelphia	PA	NaN	NaN	NaN	NaN	NaN	['Office', 'SQL', 'Bachelor']
2	Quantitive Business Analyst - Strategic Data S...	Business Analyst	NaN	NaN	Apple	Austin, TX+1 location	Austin	TX	NaN	NaN	NaN	NaN	NaN	['Python', 'SQL', 'Bachelor']
3	Business Line Product Lifecycle Management (PL...	Business Analyst	Junior	NaN	NXP Semiconductors	Austin, TX (West Oak Hill area)	Austin	TX	NaN	NaN	NaN	NaN	NaN	['Bachelor']
4	Global Markets Operations Asset Services Ops S...	Operations Analyst	Senior	NaN	Bank of America	Jacksonville, FL 32246 (Windy Hill area)+4 loc...	Jacksonville	FL	NaN	NaN	NaN	NaN	NaN	['Excel']

Data Wrangling

In [202…
```
#List of states
abbrev2state = {'AK': 'Alaska',
                'AL': 'Alabama',
                'AR': 'Arkansas',
                'AZ': 'Arizona',
                'CA': 'California',
                'CO': 'Colorado',
                'CT': 'Connecticut',
                'DC': 'District of Columbia',
                'DE': 'Delaware',
                'FL': 'Florida',
                'GA': 'Georgia',
                'HI': 'Hawaii',
                'IA': 'Iowa',
                'ID': 'Idaho',
                'IL': 'Illinois',
                'IN': 'Indiana',
                'KS': 'Kansas',
                'KY': 'Kentucky',
                'LA': 'Louisiana',
                'MA': 'Massachusetts',
                'MD': 'Maryland',
                'ME': 'Maine',
                'MI': 'Michigan',
                'MN': 'Minnesota',
                'MO': 'Missouri',
                'MS': 'Mississippi',
                'MT': 'Montana',
                'NC': 'North Carolina',
                'ND': 'North Dakota',
                'NE': 'Nebraska',
                'NH': 'New Hampshire',
                'NJ': 'New Jersey',
                'NM': 'New Mexico',
                'NV': 'Nevada',
                'NY': 'New York',
                'OH': 'Ohio',
                'OK': 'Oklahoma',
                'OR': 'Oregon',
                'PA': 'Pennsylvania',
                'RI': 'Rhode Island',
                'SC': 'South Carolina',
                'SD': 'South Dakota',
                'TN': 'Tennessee',
                'TX': 'Texas',
                'UT': 'Utah',
                'VA': 'Virginia',
                'VT': 'Vermont',
                'WA': 'Washington',
                'WI': 'Wisconsin',
                'WV': 'West Virginia',
                'WY': 'Wyoming'}
```

The dictionary of state names is needed for the state name replacement. After reviewing the data set, it appears that there are certain job that does not fit the data practitioner roles. So they need to be removed/filtered.

In [203…
```
# find out distinct job groups in the dataframe
unique_job_group_set = set(job_data['Jobs_Group'])
to_remove = ['Others', 'Controller', 'CFO', 'Statistician/Mathemathics', 'Finance']
data_practitioner_jobs_set = [x for x in unique_job_group_set if x not in to_remove]

# filter out other job groups that are not in data practitioners list
job_filtered_data = job_data[job_data['Jobs_Group'].isin(data_practitioner_jobs_set)]

# drop the frequency salary column
job_filtered_data = job_filtered_data.drop(['Frecuency_Salary'], axis=1)

# filter out missing salary and jobs with remote location
# add a new column with the state full name
job_filtered_data = job_filtered_data[job_filtered_data['Salary'].notnull()]
job_filtered_data = job_filtered_data[job_filtered_data['Remote'] != 'Remote']
job_filtered_data['State'] = job_filtered_data['State'].replace(abbrev2state)

# calculate the mean salary sum group by state and jobs group
job_state_salary_data_average = round(job_filtered_data.groupby(['State', 'Jobs_Group'])['Mean_Salary'].mean(),2)
job_state_salary_data_mean = round(job_filtered_data.groupby(['State', 'Jobs_Group'])['Mean_Salary'].mean().reset_index(),2)
job_state_salary_data_low = round(job_filtered_data.groupby(['State', 'Jobs_Group'])['Low_Salary'].mean().reset_index(),2)
job_state_salary_data_high = round(job_filtered_data.groupby(['State', 'Jobs_Group'])['High_Salary'].mean().reset_index(),2)

# combine all data into one dataframe and pivot from long to wide format dataset
job_state_combined_data = pd.merge(job_state_salary_data_low, job_state_salary_data_mean, on = ['State', 'Jobs_Group'], how = 'left')
job_state_combined_data = pd.merge(job_state_combined_data, job_state_salary_data_high, on = ['State', 'Jobs_Group'], how = 'left')
job_state_combined_data.head()
```

Out[203]:

	State	Jobs_Group	Low_Salary	Mean_Salary	High_Salary
0	Alabama	Analyst	67421.00	76463.86	85506.71
1	Alabama	Business Analyst	84074.55	95847.65	107620.75
2	Alabama	Business Intelligence	65920.00	68312.00	70704.00
3	Alabama	Data Analyst	66954.37	78748.95	90543.53
4	Alabama	Data Engineer	167191.40	183589.30	199987.20

Data Visualization

The term 'Data Practitioners' is growing in popularity in recent decades. Many companies across America offer to pay a decent salary for job positions under that term. As shown below, District of Columbias offer highest salary overall for data practitioner jobs, followed by California and Washington. It appears that both Business and Data Analysts are most popular data practitioner titles as they are being offered in every states.

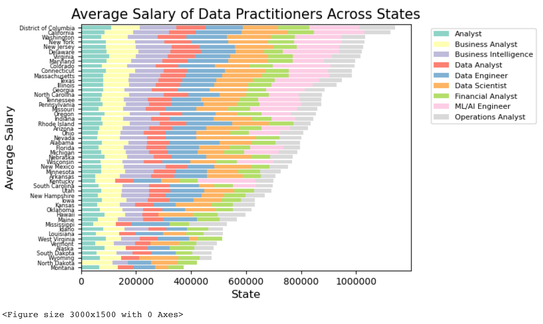
In [214…
```
# get the total sum index and apply to the dataframe so the bar plot can be sorted by the total sum
job_sort = job_state_salary_data_average.groupby(level=[0]).sum().sort_values(ascending=True)

# job_state_salary_data_mean.reindex(index=job_sort.index, level=0).unstack().plot.bar(stacked=True)
sorted_job_state_salary_data = job_state_salary_data_average.reindex(index=job_sort.index, level=0).unstack()

# stacked bar plot
fig, ax = plt.subplots()
sorted_job_state_salary_data.plot(kind = 'barh', stacked = True, width = 0.8, color=plt.cm.Set3(np.arange(len('Job_Group'))), ax = ax)
plt.title('Average Salary of Data Practitioners Across States', size = 15)
plt.legend(bbox_to_anchor=(1.05, 1.0), fontsize = '8', loc='upper left')
plt.figure(figsize=(30, 15))

# customize both x and y labels
ax.set_xlabel('State', size = 12)
ax.set_ylabel('Average Salary', size = 12)
plt.setp(ax.yaxis.get_majorticklabels(), size = 6)
ax.xaxis.get_major_formatter().set_scientific(False)
ax.xaxis.get_major_formatter().set_useOffset(False)

# show plot
plt.show()
```




```
<Figure size 3000x1500 with 0 Axes>
```