20240215WileyWinters_Week5_Assignment

February 16, 2024

0.0.1 Week 5 Assignment

- Wiley Winters
- MSDS 670 — Data Visualization
- 18-FEB-2024

0.0.2 Dataset Information

Dataset: Jobs and Salaries in Data Science Metadata: - `work_year`: Year in which data was recorded. - `job_title`: Specific title of the job role. - `job_category`: Classification of the job role into broader categories for easier analysis - `salary_currency`: Currency in which the salary is paid - `salary`: Annual gross salary of the role in the local currency - `salary_in_usd`: Annual gross salary in USD - `employee_residence`: Country of residence - `experience_level`: Classifies the professional experience level of the employee - `employment_type`: Specifies the type of employment such as *full-time*, *part-time*, *contract*, *etc* - `work_setting`: Work setting or environment such as *remote*, *in-person*, or *hybrid* - `company_location`: Country where the company is located - `company_size`: Size of the employer company categorized as *small (S)*, *medium (M)*, and *large (L)*

Import required packages and libraries. Set global configuration items.

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from matplotlib import rcParams
import numpy as np

# Suppress Warnings
import warnings
warnings.filterwarnings('ignore')

# Set seaborn style and autoconfig
sns.set_style('whitegrid')
rcParams.update({'figure.autolayout': True})
```

Read dataset into a Pandas DataFrame

[2]:
```
jobs_df = pd.read_csv('data/jobs_in_data.csv')
jobs_df.sample(5)
```

[2]:

	work_year	job_title	job_category	salary_currency	salary	salary_in_usd	employee_residence	experience_level	employment_type	work_setting	company_location	company_size
502	2023	Analytics Engineer	Leadership and Management	USD	75000	75000	United States	Mid-level	Full-time	In-person	United States	M
8759	2022	Data Engineer	Data Engineering	USD	160000	160000	United States	Senior	Full-time	Remote	United States	M
4905	2023	Data Analyst	Data Analysis	USD	132000	132000	United States	Senior	Full-time	Remote	United States	M
1952	2023	Data Analyst	Data Analysis	GBP	75000	92280	United Kingdom	Mid-level	Full-time	In-person	United Kingdom	M
6892	2023	Data Scientist	Data Science and Research	USD	209300	209300	United States	Senior	Full-time	Remote	United States	M

[3]:
```
jobs_df.describe().T
```

[3]:

	count	mean	std	min	25%	50%	75%	max
work_year	9355.0	2022.760449	0.519470	2020.0	2023.0	2023.0	2023.0	2023.0
salary	9355.0	149927.981293	63608.835387	14000.0	105200.0	143860.0	187000.0	450000.0
salary_in_usd	9355.0	150299.495564	63177.372024	15000.0	105700.0	143000.0	186723.0	450000.0

The dataset covers years from 2020 to 2023. In order to not double count some values. I will only work with 2023 data

Check some basic items to see if the dataset requires cleaning or not

[4]:
```
print(jobs_df.info())
print('\nNaN Values:\n', jobs_df.isna().sum())
print('\nDuplicates: ', jobs_df.duplicated().sum())
print('\nSize: ', jobs_df.size)
print('\nDistribution:\n', jobs_df.describe().T)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9355 entries, 0 to 9354
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           9355 non-null   int64 
 1   job_title           9355 non-null   object
 2   job_category        9355 non-null   object
 3   salary_currency     9355 non-null   object
 4   salary              9355 non-null   int64 
 5   salary_in_usd       9355 non-null   int64 
 6   employee_residence  9355 non-null   object
 7   experience_level    9355 non-null   object
 8   employment_type     9355 non-null   object
 9   work_setting        9355 non-null   object
 10  company_location    9355 non-null   object
 11  company_size        9355 non-null   object
dtypes: int64(3), object(9)
memory usage: 877.2+ KB
None

NaN Values:
 work_year             0
job_title              0
job_category           0
salary_currency        0
salary                 0
salary_in_usd          0
employee_residence     0
experience_level       0
employment_type        0
work_setting           0
company_location       0
company_size           0
dtype: int64

Duplicates:  4014

Size:  112260

Distribution:
                 count           mean            std      min       25%  \
```

```
work_year       9355.0    2022.760449       0.519470   2020.0    2023.0
salary          9355.0  149927.981293   63608.835387  14000.0  105200.0
salary_in_usd   9355.0  150299.495564   63177.372024  15000.0  105700.0

                    50%       75%       max
work_year        2023.0    2023.0    2023.0
salary         143860.0  187000.0  450000.0
salary_in_usd  143000.0  186723.0  450000.0
```

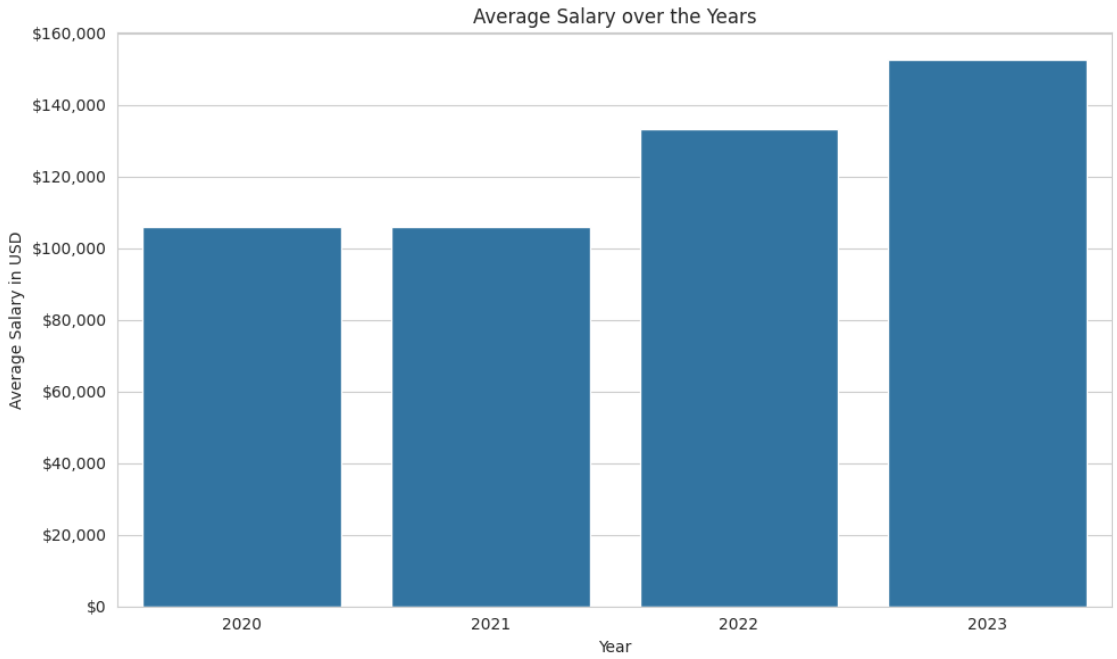
Looks like there is a lot of duplicates. I will remove them.

[5]:
```
jobs_df.drop_duplicates(keep='first', inplace=True)
jobs_df.duplicated().sum()
```

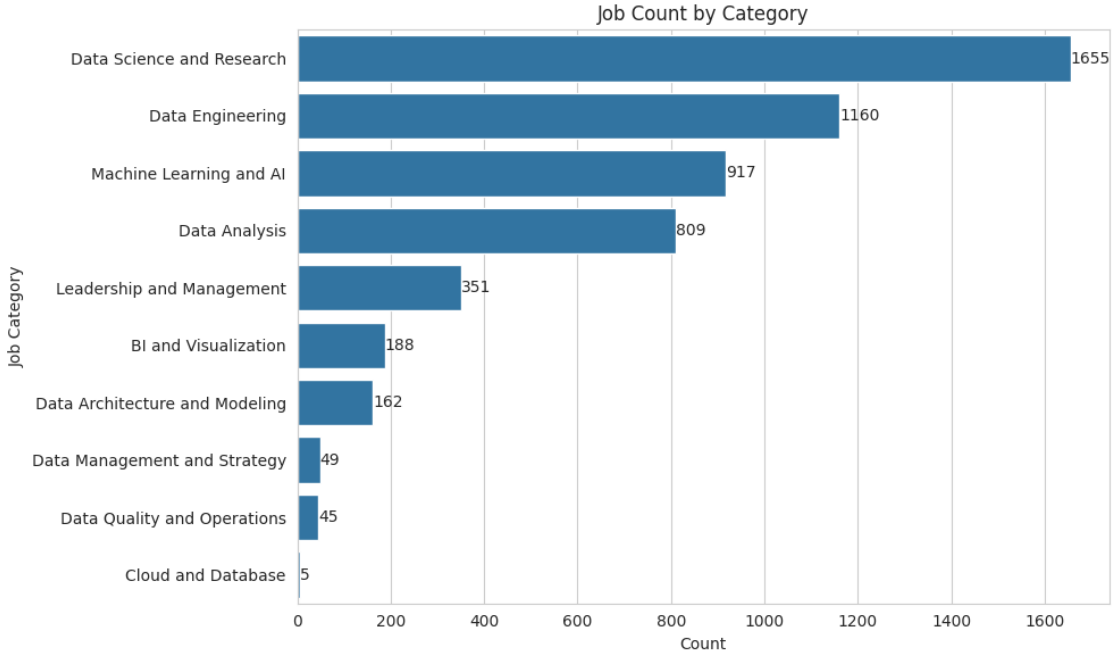
[5]:
```
0
```

0.0.3 Look for interesting items to plot

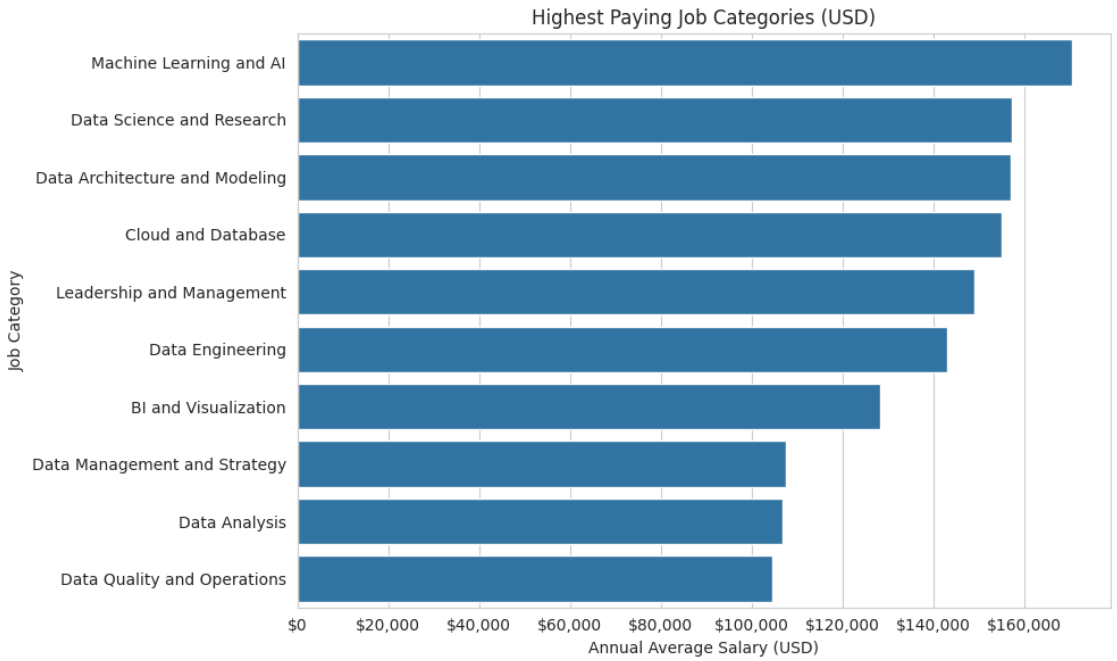
[6]:
```
salary_years = jobs_df.groupby('work_year').agg({'salary_in_usd': 'mean'}). \
                                sort_values('work_year')
fig, ax = plt.subplots(figsize=(10,6))
fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
ax.set(xlabel='Year',ylabel='Average Salary in USD',title='Average Salary over␣
 ↪the Years')
sns.barplot(data=salary_years, y='salary_in_usd', x='work_year')
fig.savefig('images/yearAveSalary.png', bbox_inches='tight', dpi=300)
```



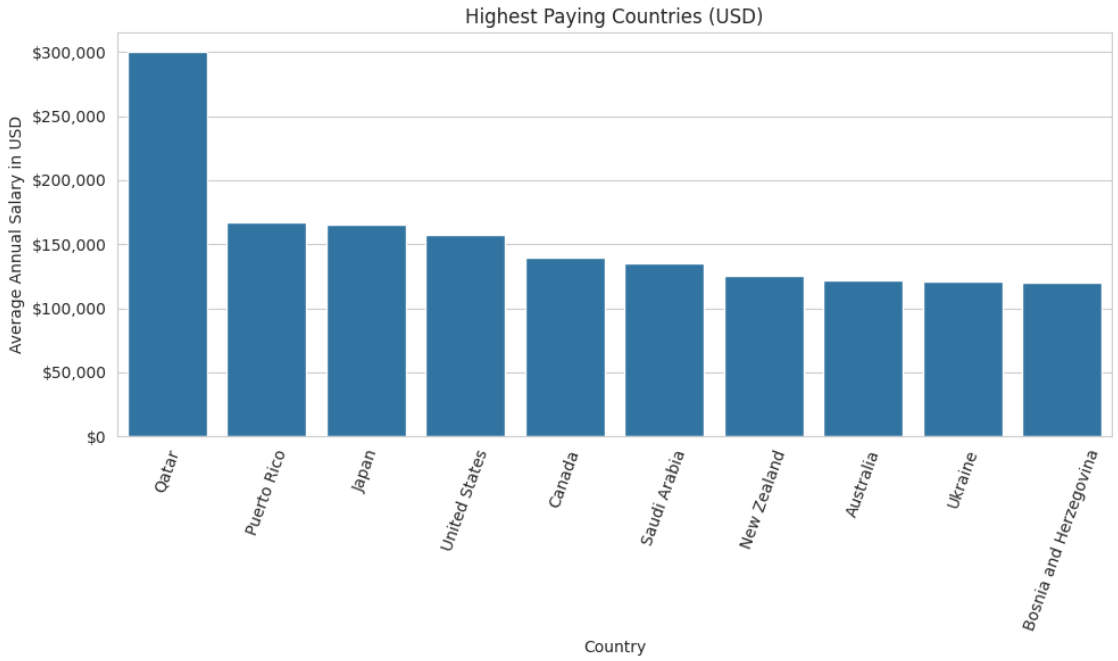
[7]:
```
fig, ax = plt.subplots(figsize=(10,6))
ax.set(ylabel='Job Category',title='Job Count by Category',xlabel='Count')
sns.countplot(ax=ax, data=jobs_df, y='job_category',␣
 ↪order=jobs_df['job_category']. \
             value_counts().index)
ax.bar_label(ax.containers[0])
fig.savefig('images/jobCatCount.png', bbox_inches='tight', dpi=300)
```



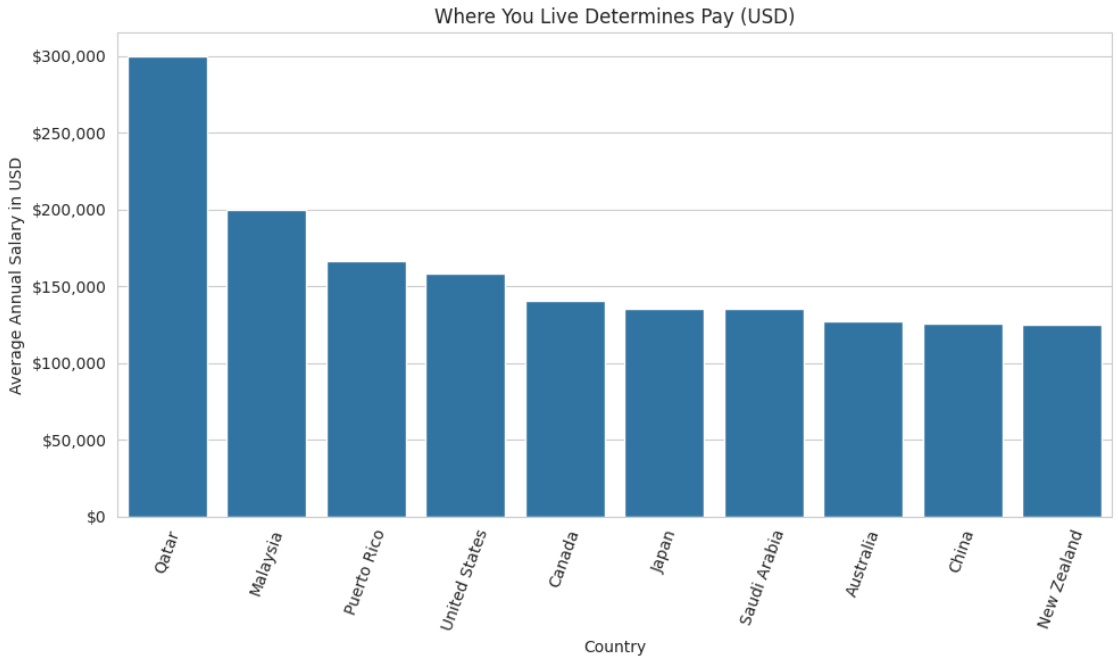
[8]:
```
# Highest paying categories in USD
salary = jobs_df.groupby('job_category').agg({'salary_in_usd': 'mean'}). \
                         sort_values('salary_in_usd', ascending=False)
fig, ax = plt.subplots(figsize=(10,6))
fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.xaxis.set_major_formatter(tick)
ax.set(xlabel='Annual Average Salary (USD)',ylabel='Job Category', \
       title='Highest Paying Job Categories (USD)')
sns.barplot(data=salary, y='job_category', x='salary_in_usd')
fig.savefig('images/highJobCatUSD.png', bbox_inches='tight', dpi=300)
```



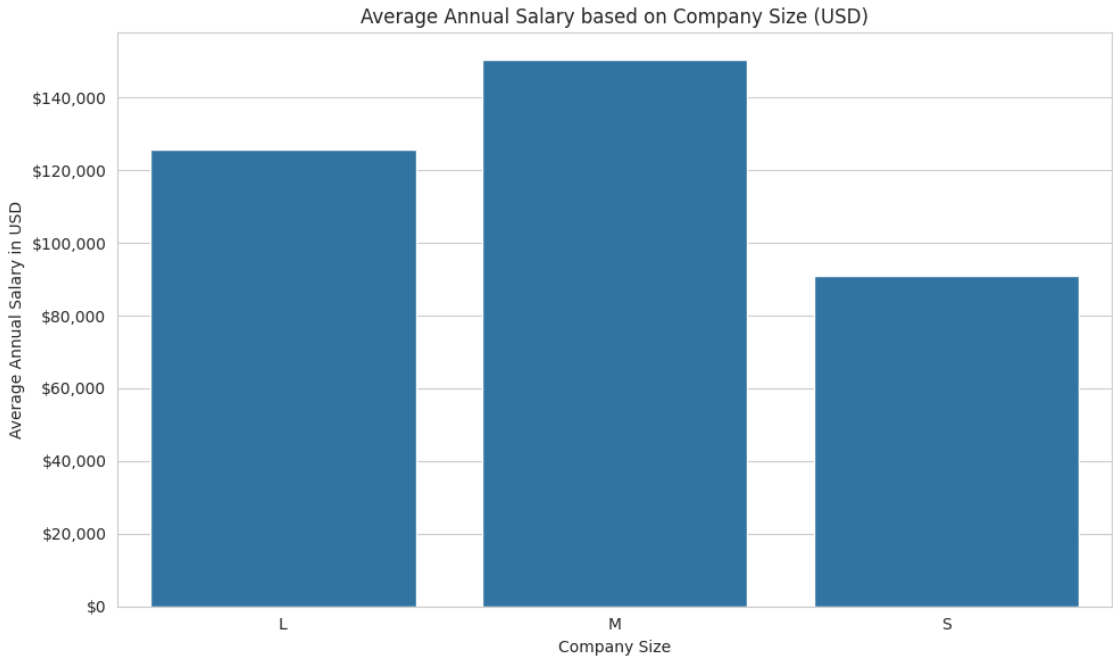
[9]:
```
# Highest pay by company_location in USD. Top 10
pay_country = jobs_df.groupby('company_location').agg({'salary_in_usd':␣
 ↪'mean'}). \
                                sort_values('salary_in_usd', ascending=False).
 ↪head(10)
fig, ax = plt.subplots(figsize=(10,6))
fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
ax.set(xlabel='Country',ylabel='Average Annual Salary in USD', \
       title='Highest Paying Countries (USD)')
sns.barplot(data=pay_country, x='company_location', y='salary_in_usd')
plt.xticks(rotation=70)
fig.savefig('images/highCountryUSD.png', bbox_inches='tight', dpi=300)
```



[13]:
```
# Highest pay by employee_residence in USD. Top 10
pay_residence = jobs_df.groupby('employee_residence').agg({'salary_in_usd':␣
 ↪'mean'}). \
                                sort_values('salary_in_usd', ascending=False).
 ↪head(10)
fig, ax = plt.subplots(figsize=(10,6))
fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
ax.set(xlabel='Country',ylabel='Average Annual Salary in USD', \
       title='Where You Live Determines Pay (USD)')
sns.barplot(data=pay_residence, x='employee_residence', y='salary_in_usd')
plt.xticks(rotation=70)
fig.savefig('images/highResidenceUSD.png', bbox_inches='tight', dpi=300)
```



[18]:
```
# Average Salary by company_size
size = jobs_df.groupby('company_size').agg({'salary_in_usd': 'mean'})
fig, ax = plt.subplots(figsize=(10,6))
fmt = '${x:,.0f}'
tick = mtick.StrMethodFormatter(fmt)
ax.yaxis.set_major_formatter(tick)
ax.set(xlabel='Company Size', ylabel='Average Annual Salary in USD', \
       title='Average Annual Salary based on Company Size (USD)')
sns.barplot(data=size, x='company_size', y='salary_in_usd')
fig.savefig('images/aveCompanySize.png', bbox_inches='tight', dpi=300)
```


Average Annual Salary based on Company Size (USD)
$140,000
$120,000
$100,000
$80,000
$60,000
$40,000
$20,000
$0
Average Annual Salary in USD
L
M
S
Company Size

[ ]: