DSE 5002 Project 1

Peter Gyorda April 12 ,2025

We will be analyzing salaries for data scientists and be doing the analysis with pandas and seaborn.

In [96]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os
```

In [97]:
```
import os

os.getcwd()
```

Out[97]:
```
'C:\\Users\\petea'
```

In [98]:
```
# First, install the required PostgreSQL adapter
!pip install psycopg2-binary

#import psycopg2
import sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\petea\anaconda3\lib\site-packages (2.9.10)
```

In [99]:
```
df = pd.read_csv("r project data-1.csv")

num_rows = df.shape[0]
print(f"The DataFrame has {num_rows} rows.")
```

```
The DataFrame has 607 rows.
```

In [100…
```
df.head()
```

Out[100…

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_curr
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	

In [101…

```python
import pandas as pd

def analyze_data_types(csv_file):
    """
    Loads a CSV file and prints the data type of each column.

    Args:
        csv_file (str): Path to the CSV file.
    """
    try:
        df = pd.read_csv(csv_file)
        print("Data Categories and Their Types:")
        print("-" * 30)
        for column in df.columns:
            data_type = df[column].dtype
            print(f"{column}: {data_type}")
    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage (replace 'r project data-1.csv' with your actual file path)
analyze_data_types("r project data-1.csv")
```

```
Data Categories and Their Types:
-----------------------------
Unnamed: 0: int64
work_year: int64
experience_level: object
employment_type: object
job_title: object
salary: int64
salary_currency: object
salary_in_usd: int64
employee_residence: object
remote_ratio: int64
company_location: object
company_size: object
```

In [102…

```python
def convert_to_categorical(csv_file, categories_and_types):
    try:
        # Import pandas library
        import pandas as pd

        # Read the CSV file
        df = pd.read_csv(csv_file)

        # Print original data types
        print("Original Data Types:")
        print("-" * 30)
        for column in df.columns:
            print(f"{column}: {df[column].dtype}")

        # Columns to convert to factors (categorical)
        columns_to_convert = [
            'experience_level',
            'employment_type',
            'job_title',
            'salary_currency',
            'employee_residence',
            'company_location',
            'company_size'
        ]

        # Convert specified columns to factors (categorical)
        for column in columns_to_convert:
            if column in df.columns:  # Check if the column exists
                df[column] = df[column].astype('category')
            else:
                print(f"Warning: Column '{column}' not found in the CSV file.")

        # Print updated data types
        print("\nUpdated Data Types:")
        print("-" * 30)
        for column in df.columns:
            print(f"{column}: {df[column].dtype}")

        return df  # Return the modified DataFrame

    except FileNotFoundError:
```

```
        print(f"Error: File '{csv_file}' not found.")
        return None  # Return None to indicate failure
    except Exception as e:
        print(f"An error occurred: {e}")
        return None  # Return None to indicate failure

# Example usage
csv_file = "r project data-1.csv"  # Replace with your CSV file path
categories_and_types = {
    "Unnamed: 0": "int64",
    "work_year": "int64",
    "experience_level": "object",
    "employment_type": "object",
    "job_title": "object",
    "salary": "int64",
    "salary_currency": "object",
    "salary_in_usd": "int64",
    "employee_residence": "object",
    "remote_ratio": "int64",
    "company_location": "object",
    "company_size": "object"
}

df_modified = convert_to_categorical(csv_file, categories_and_types)
if df_modified is not None:
    # You can now work with the DataFrame df_modified
    # For example, print the first few rows:
    print(df_modified.head())
```

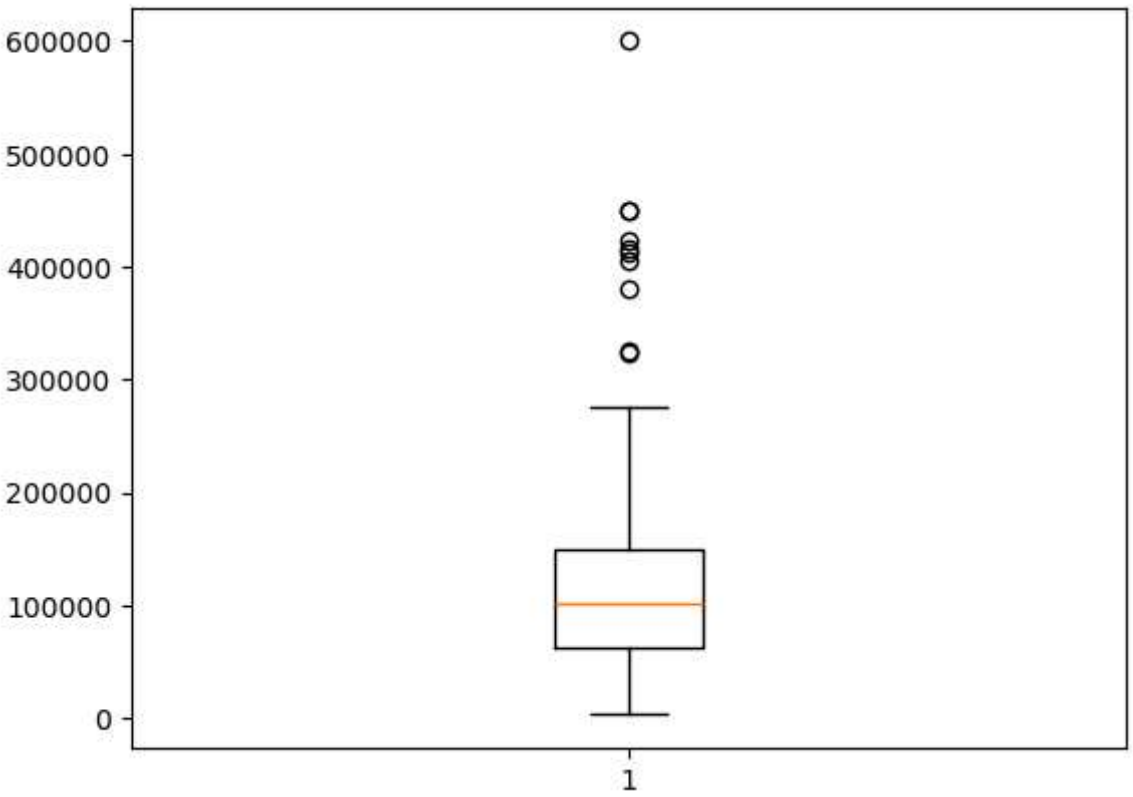
```
Original Data Types:
-----------------------------
Unnamed: 0: int64
work_year: int64
experience_level: object
employment_type: object
job_title: object
salary: int64
salary_currency: object
salary_in_usd: int64
employee_residence: object
remote_ratio: int64
company_location: object
company_size: object

Updated Data Types:
-----------------------------
Unnamed: 0: int64
work_year: int64
experience_level: category
employment_type: category
job_title: category
salary: int64
salary_currency: category
salary_in_usd: int64
employee_residence: category
remote_ratio: int64
company_location: category
company_size: category
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
```

In [103…

```
#boxplot using matplotlib

plt.boxplot(df.salary_in_usd)
```

Out[103…

```
{'whiskers': [<matplotlib.lines.Line2D at 0x200d47c38c0>,
  <matplotlib.lines.Line2D at 0x200d1ee7b60>],
 'caps': [<matplotlib.lines.Line2D at 0x200d1ee7ad0>,
  <matplotlib.lines.Line2D at 0x200d1ee5af0>],
 'boxes': [<matplotlib.lines.Line2D at 0x200d1ee56a0>],
 'medians': [<matplotlib.lines.Line2D at 0x200d1ee6e10>],
 'fliers': [<matplotlib.lines.Line2D at 0x200d1ee7050>],
 'means': []}
```



In [104…

```python
import pandas as pd

def calculate_average_salaries(csv_file):
    """
    Calculates and displays average salaries (truncated to whole numbers) in USD by

    Args:
        csv_file (str): Path to the CSV file containing salary data.
    """
    try:
        df = pd.read_csv(csv_file)

        # Ensure salary_in_usd is numeric and handle potential errors
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'experience_level', 'company_locati

        # Calculate average salaries by experience level and location
        average_salaries = df.groupby(['experience_level', 'company_location'])['sa

        # Rename the average salary column
        average_salaries = average_salaries.rename(columns={'salary_in_usd': 'avera
```

```
        # Truncate average salaries to whole numbers
        average_salaries['average_salary_usd'] = average_salaries['average_salary_u

        # Display the results as a table
        print(average_salaries.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage (replace 'r project data-1.csv' with your actual file path)
calculate_average_salaries("r project data-1.csv")
```

experience_level	company_location	average_salary_usd
EN	AS	18053
EN	AU	118351
EN	CA	57132
EN	CH	5882
EN	CN	100000
EN	CO	21844
EN	CZ	31875
EN	DE	57551
EN	DK	37252
EN	DZ	100000
EN	ES	10354
EN	FR	47325
EN	GB	65604
EN	IN	19629
EN	IQ	100000
EN	IT	21669
EN	JP	41689
EN	KE	9272
EN	LU	34551
EN	MY	40000
EN	NG	10000
EN	NL	42000
EN	PK	20000
EN	PT	21983
EN	UA	13400
EN	US	93112
EN	VN	4000
EX	CA	157583
EX	DE	135936
EX	ES	74787
EX	IN	79039
EX	PL	153667
EX	RU	157500
EX	US	243742
MI	AE	115000
MI	AT	66815
MI	AU	87425
MI	BE	88654
MI	BR	12901
MI	CA	83530
MI	CH	122346
MI	CL	40038
MI	CN	43331
MI	CZ	69999
MI	DE	72209
MI	EE	32974
MI	ES	52791
MI	FR	57771
MI	GB	80506
MI	GR	52732
MI	HN	20000
MI	HU	35735
MI	IL	119059
MI	IN	20441
MI	IR	4000

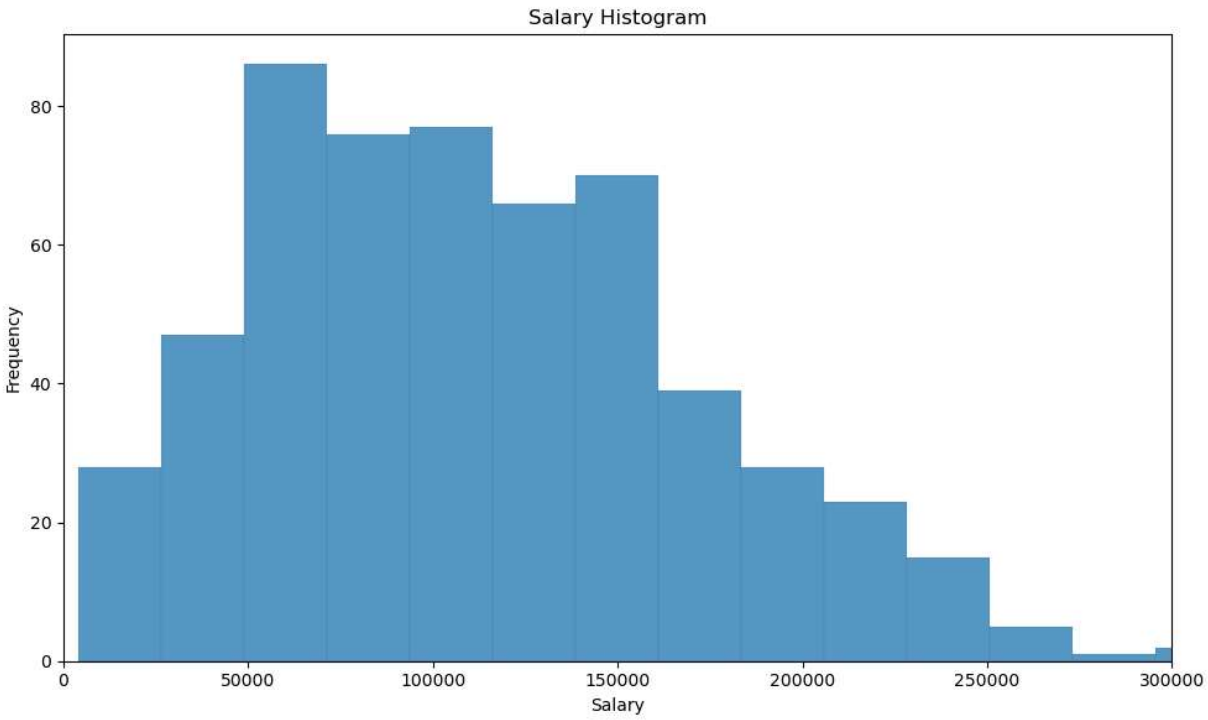
```
MI              IT              51064
MI              JP              71691
MI              LU              62726
MI              MD              18000
MI              MT              28369
MI              MX               2859
MI              NG              50000
MI              NL              57566
MI              PK              10000
MI              PL              36887
MI              PT              54217
MI              RO              60000
MI              SG              89294
MI              SI              24823
MI              TR              20059
MI              US             125780
SE              AE              92500
SE              AT              91237
SE              BE              82744
SE              BR              21453
SE              CA             111523
SE              DE             115745
SE              DK              88654
SE              ES              55000
SE              FR              94075
SE              GB              90931
SE              GR              47899
SE              HR              45618
SE              IE              71444
SE              IN              56460
SE              JP             214000
SE              MX              46755
SE              NL              62651
SE              NZ             125000
SE              PT              60757
SE              SI             102839
SE              TR              20171
SE              US             151527
```

In [105…

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib

def create_adjusted_histogram(csv_file, x_axis_max):
    """
    Creates a histogram of salary with an adjusted x-axis range.
    Includes debugging prints and backend selection.
    """
    try:
        #matplotlib.use('TkAgg') # Try a different backend if needed (e.g., 'Qt5Agg
        df = pd.read_csv(csv_file)
        df['salary'] = pd.to_numeric(df['salary'], errors='coerce')
        df = df.dropna(subset=['salary'])

        print(f"DataFrame length after cleaning: {len(df)}")
```

```
        print("First few rows of DataFrame:")
        print(df.head())
        print("DataFrame data types:")
        print(df.dtypes)

        if df.empty:
            print("DataFrame is empty after cleaning. No histogram to display.")
            return

        plt.figure(figsize=(10, 6))
        sns.histplot(df, x="salary")
        plt.xlim(0, x_axis_max)
        plt.title("Salary Histogram")
        plt.xlabel("Salary")
        plt.ylabel("Frequency")
        plt.tight_layout()
        plt.show()

        print("Graph code ran successfully.")

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

create_adjusted_histogram("r project data-1.csv", 300000)
```

```
DataFrame length after cleaning: 607
First few rows of DataFrame:
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame data types:
Unnamed: 0             int64
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary                 int64
salary_currency       object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```

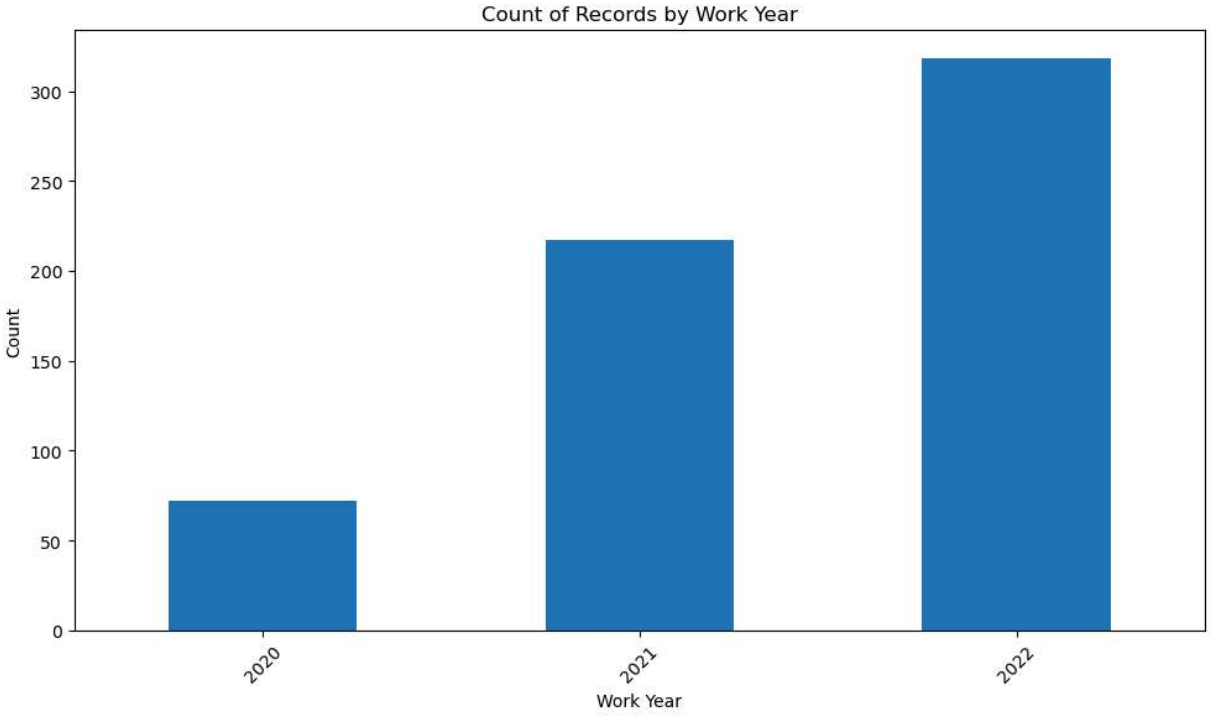


Graph code ran successfully.

In [143…

```
# Assuming df is your DataFrame with the data already loaded
# If not, uncomment the following lines to load your data
# import pandas as pd
# df = pd.read_csv("r project data-1.csv")

# Get counts of each work_year value
work_year_counts = df['work_year'].value_counts()

# Display the counts
print("Work Year Counts:")
print("-" * 20)
print(work_year_counts)

# Optional: Sort by work_year (instead of by count)
work_year_counts_sorted = df['work_year'].value_counts().sort_index()
print("\nWork Year Counts (sorted by year):")
print("-" * 20)
print(work_year_counts_sorted)

# Optional: Create a bar plot of the counts
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
work_year_counts_sorted.plot(kind='bar')
plt.title('Count of Records by Work Year')
plt.xlabel('Work Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Work Year Counts:
--------------------
work_year
2022    318
2021    217
2020     72
Name: count, dtype: int64

Work Year Counts (sorted by year):
--------------------
work_year
2020     72
2021    217
2022    318
Name: count, dtype: int64
```

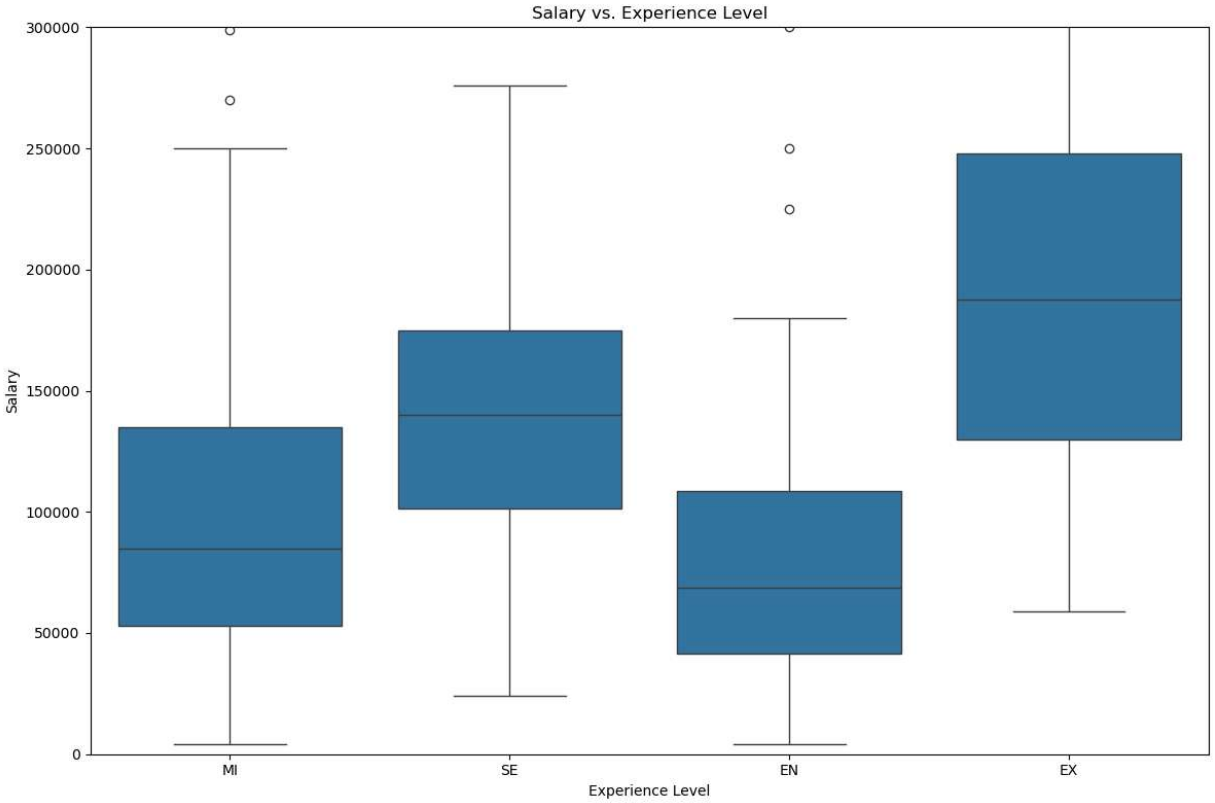


In [106…

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib

def create_salary_boxplot_by_level(csv_file, y_axis_max):
    """
    Creates a boxplot of salary versus experience level with adjusted y-axis.
    Includes debugging prints and backend selection.
    """
    try:
        #matplotlib.use('TkAgg') # Try a different backend if needed (e.g., 'Qt5Agg
        df = pd.read_csv(csv_file)
        df['salary'] = pd.to_numeric(df['salary'], errors='coerce')
        df = df.dropna(subset=['salary', 'experience_level'])
        df.shape

        print(f"DataFrame length after cleaning: {len(df)}")
```

```
        print("First few rows of DataFrame:")
        print(df.head())
        print("DataFrame data types:")
        print(df.dtypes)

        if df.empty:
            print("DataFrame is empty after cleaning. No boxplot to display.")
            return

        plt.figure(figsize=(12, 8))
        sns.boxplot(x="experience_level", y="salary", data=df)
        plt.ylim(0, y_axis_max) #adjust the y axis.
        plt.title("Salary vs. Experience Level")
        plt.xlabel("Experience Level")
        plt.ylabel("Salary")
        plt.tight_layout()
        plt.show()

        print("Boxplot code ran successfully.")
        # Save the Matplotlib boxplot as a PNG file
        plt.savefig('matplotlib_boxplot.png')
        import os
        os.path.abspath(filename)


    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

create_salary_boxplot_by_level("r project data-1.csv", 300000) #adjust 300000 as ne
```

```
DataFrame length after cleaning: 607
First few rows of DataFrame:
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame data types:
Unnamed: 0             int64
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary                 int64
salary_currency       object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```



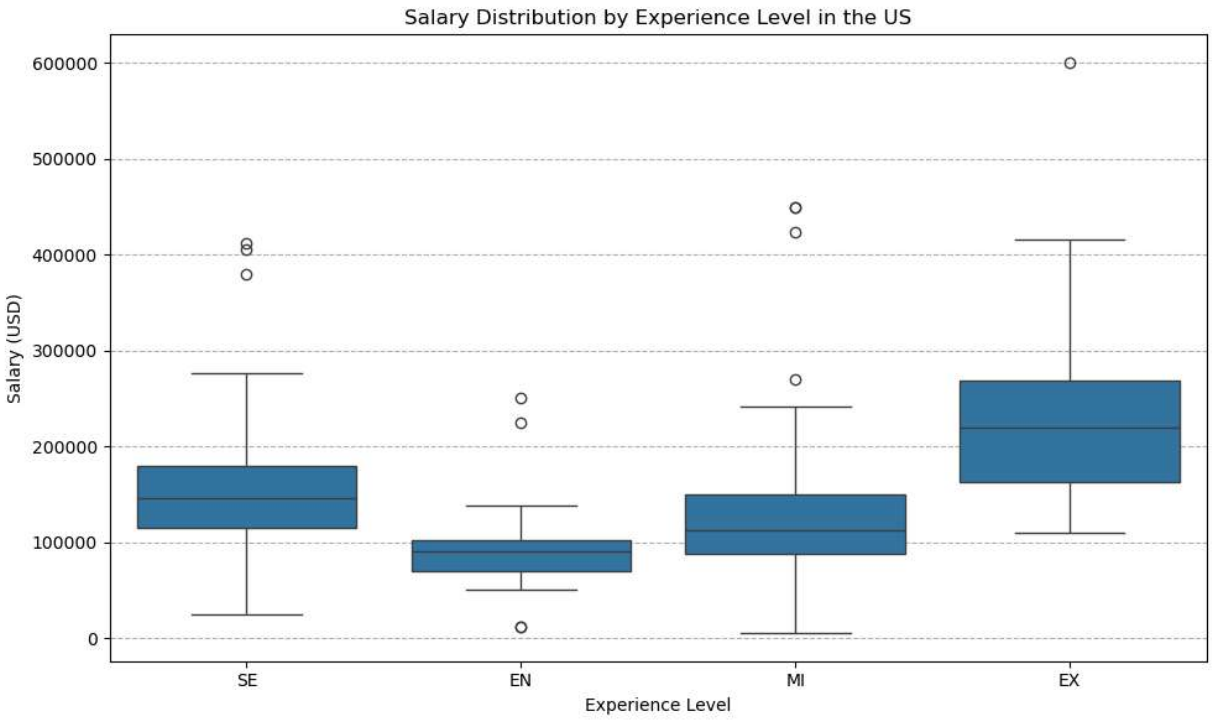

```
Boxplot code ran successfully.
An error occurred: name 'filename' is not defined
<Figure size 640x480 with 0 Axes>
```

In [107…

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def analyze_us_salary_by_experience_level_with_boxplots(csv_file):
    """
    Analyzes the salary distribution (including box plots), average salary,
    and quartile range (Q1, Q3) by experience level for data located in
    the United States (US).
    """
    try:
        df = pd.read_csv(csv_file)

        # Ensure salary is numeric and handle potential errors
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'experience_level', 'company_locati

        # Filter data for the United States
        us_data = df[df['company_location'] == 'US'].copy()

        if us_data.empty:
            print("No data found for the United States.")
            return

        # Calculate Q1, Q3, and average salary for each experience level in the US
        salary_info_us = us_data.groupby('experience_level')['salary_in_usd'].agg(
            Q1=lambda x: x.quantile(0.25),
            Q3=lambda x: x.quantile(0.75),
```

```
            Average='mean',
            N='count'
        ).reset_index()

        # Format the salary columns as US dollars
        salary_info_us['Q1'] = salary_info_us['Q1'].apply(lambda x: f"${x:,.2f}")
        salary_info_us['Q3'] = salary_info_us['Q3'].apply(lambda x: f"${x:,.2f}")
        salary_info_us['Average'] = salary_info_us['Average'].apply(lambda x: f"${x

        # Print the results table
        print("Salary Analysis for the United States by Experience Level:")
        print(salary_info_us.to_string(index=False))
        print("\n")

        # Create box plots
        plt.figure(figsize=(10, 6))
        sns.boxplot(x='experience_level', y='salary_in_usd', data=us_data)
        plt.title('Salary Distribution by Experience Level in the US')
        plt.xlabel('Experience Level')
        plt.ylabel('Salary (USD)')
        plt.grid(axis='y', linestyle='--')
        plt.tight_layout()
        plt.show()

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
analyze_us_salary_by_experience_level_with_boxplots("r project data-1.csv")
```

```
Salary Analysis for the United States by Experience Level:
experience_level          Q1          Q3     Average   N
              EN  $70,000.00 $102,500.00  $93,112.90  31
              EX $163,406.25 $268,500.00 $243,742.19  16
              MI  $87,750.00 $150,000.00 $125,780.21  90
              SE $115,233.50 $180,000.00 $151,527.63 218
```



In [108…

```python
import pandas as pd

def calculate_and_sort_average_salary_by_location(csv_file):
    """
    Calculates and prints the average salary by location, sorted from highest to lo
    and formats the salaries as US dollars.
    """
    try:
        df = pd.read_csv(csv_file)

        # Ensure salary is numeric and handle potential errors
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_location'])

        # Calculate average salary by location
        average_salaries = df.groupby('company_location')['salary_in_usd'].mean().r

        # Rename the average salary column
        average_salaries = average_salaries.rename(columns={'salary_in_usd': 'avera

        # Sort average salaries from highest to lowest BEFORE formatting.
        sorted_salaries = average_salaries.sort_values(by='average_salary_usd', asc

        # Format average salary as US dollars AFTER sorting.
        sorted_salaries['average_salary_usd'] = sorted_salaries['average_salary_usd

        # Print the sorted table
        print(sorted_salaries.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")
```

```
# Example usage
calculate_and_sort_average_salary_by_location("r project data-1.csv")
```

company_location	average_salary_usd
RU	$157,500.00
US	$144,055.26
NZ	$125,000.00
IL	$119,059.00
JP	$114,127.33
AU	$108,042.67
AE	$100,000.00
DZ	$100,000.00
IQ	$100,000.00
CA	$99,823.73
SG	$89,294.00
BE	$85,699.00
DE	$81,887.21
GB	$81,583.04
AT	$72,920.75
CN	$71,665.50
IE	$71,444.00
PL	$66,082.50
CH	$64,114.00
FR	$63,970.67
SI	$63,831.00
RO	$60,000.00
NL	$54,945.75
DK	$54,386.33
ES	$53,060.14
GR	$52,293.09
CZ	$50,937.00
PT	$47,793.75
HR	$45,618.00
LU	$43,942.67
CL	$40,038.00
MY	$40,000.00
IT	$36,366.50
HU	$35,735.00
EE	$32,974.00
MX	$32,123.33
NG	$30,000.00
IN	$28,581.75
MT	$28,369.00
CO	$21,844.00
TR	$20,096.67
HN	$20,000.00
BR	$18,602.67
AS	$18,053.00
MD	$18,000.00
UA	$13,400.00
PK	$13,333.33
KE	$9,272.00
IR	$4,000.00
VN	$4,000.00

In [109…

```python
import pandas as pd

def analyze_salary_by_location_with_iqr_mean_and_count(csv_file):
    """
```

```
    Calculates and prints the average salary, lower quartile (Q1),
    upper quartile (Q3), and the count of data points for each company location,
    sorted by average salary from highest to lowest. Also counts the total
    number of unique countries.
    """
    try:
        df = pd.read_csv(csv_file)

        # Ensure salary is numeric and handle potential errors
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_location'])

        # Calculate Q1, Q3, average salary, and count for each company location
        salary_info = df.groupby('company_location')['salary_in_usd'].agg(
            Q1=lambda x: x.quantile(0.25),
            Q3=lambda x: x.quantile(0.75),
            Average='mean',
            N='count'
        ).reset_index()

        # Rename the average salary column
        salary_info = salary_info.rename(columns={'Average': 'average_salary_usd'})

        # Sort by average salary from highest to lowest
        sorted_salary_info = salary_info.sort_values(by='average_salary_usd', ascen

        # Format the salary columns as US dollars
        sorted_salary_info['Q1'] = sorted_salary_info['Q1'].apply(lambda x: f"${x:,
        sorted_salary_info['Q3'] = sorted_salary_info['Q3'].apply(lambda x: f"${x:,
        sorted_salary_info['average_salary_usd'] = sorted_salary_info['average_sala

        # Print the sorted table with Q1, Q3, Average Salary, and Count
        print("Salary Information by Company Location (USD - Sorted by Average Sala
        print(sorted_salary_info.to_string(index=False))
        print("\n")

        # Count the number of unique countries
        num_countries = df['company_location'].nunique()
        print(f"Total number of unique countries listed: {num_countries}")

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
analyze_salary_by_location_with_iqr_mean_and_count("r project data-1.csv")
```

Salary Information by Company Location (USD - Sorted by Average Salary):

company_location	Q1	Q3	average_salary_usd	N
RU	$121,250.00	$193,750.00	$157,500.00	2
US	$100,000.00	$170,000.00	$144,055.26	355
NZ	$125,000.00	$125,000.00	$125,000.00	1
IL	$119,059.00	$119,059.00	$119,059.00	1
JP	$66,283.25	$145,341.00	$114,127.33	6
AU	$87,064.00	$118,712.50	$108,042.67	3
AE	$90,000.00	$117,500.00	$100,000.00	3
DZ	$100,000.00	$100,000.00	$100,000.00	1
IQ	$100,000.00	$100,000.00	$100,000.00	1
CA	$69,730.00	$117,916.25	$99,823.73	30
SG	$89,294.00	$89,294.00	$89,294.00	1
BE	$84,221.50	$87,176.50	$85,699.00	2
DE	$58,986.00	$90,734.00	$81,887.21	28
GB	$57,575.00	$103,931.00	$81,583.04	47
AT	$64,003.50	$78,406.75	$72,920.75	4
CN	$57,498.25	$85,832.75	$71,665.50	2
IE	$71,444.00	$71,444.00	$71,444.00	1
PL	$33,811.50	$73,364.50	$66,082.50	4
CH	$34,998.00	$93,230.00	$64,114.00	2
FR	$48,202.50	$69,143.00	$63,970.67	15
SI	$44,327.00	$83,335.00	$63,831.00	2
RO	$60,000.00	$60,000.00	$60,000.00	1
NL	$44,543.25	$64,423.50	$54,945.75	4
DK	$37,252.50	$67,275.00	$54,386.33	3
ES	$40,073.50	$68,793.00	$53,060.14	14
GR	$42,077.50	$60,453.00	$52,293.09	11
CZ	$41,406.00	$60,468.00	$50,937.00	2
PT	$43,130.75	$58,880.50	$47,793.75	4
HR	$45,618.00	$45,618.00	$45,618.00	1
LU	$34,551.00	$60,914.00	$43,942.67	3
CL	$40,038.00	$40,038.00	$40,038.00	1
MY	$40,000.00	$40,000.00	$40,000.00	1
IT	$29,017.75	$43,715.25	$36,366.50	2
HU	$35,735.00	$35,735.00	$35,735.00	1
EE	$32,974.00	$32,974.00	$32,974.00	1
MX	$18,185.00	$46,755.50	$32,123.33	3
NG	$20,000.00	$40,000.00	$30,000.00	2
IN	$16,735.00	$32,163.25	$28,581.75	24
MT	$28,369.00	$28,369.00	$28,369.00	1
CO	$21,844.00	$21,844.00	$21,844.00	1
TR	$16,137.00	$24,093.50	$20,096.67	3
HN	$20,000.00	$20,000.00	$20,000.00	1
BR	$15,904.00	$21,453.50	$18,602.67	3
AS	$18,053.00	$18,053.00	$18,053.00	1
MD	$18,000.00	$18,000.00	$18,000.00	1
UA	$13,400.00	$13,400.00	$13,400.00	1
PK	$10,000.00	$16,000.00	$13,333.33	3
KE	$9,272.00	$9,272.00	$9,272.00	1
IR	$4,000.00	$4,000.00	$4,000.00	1
VN	$4,000.00	$4,000.00	$4,000.00	1

Total number of unique countries listed: 50

In [110…

```python
import pandas as pd

def calculate_and_sort_average_salary_and_level_counts(csv_file):
    """
    Calculates average salary by location, counts occurrences of each experience le
    sorts by average salary (highest to lowest), and formats salaries as US dollars
    """
    try:
        df = pd.read_csv(csv_file)

        # Ensure salary is numeric and handle potential errors
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_location', 'experience_lev

        # Calculate average salary by location
        average_salaries = df.groupby('company_location')['salary_in_usd'].mean().r
        average_salaries = average_salaries.rename(columns={'salary_in_usd': 'avera

        # Count occurrences of each experience level by location
        level_counts = df.groupby(['company_location', 'experience_level']).size().

        # Merge average salaries and level counts
        result = pd.merge(average_salaries, level_counts, on='company_location')

        # Sort average salaries from highest to lowest BEFORE formatting.
        result_sorted = result.sort_values(by='average_salary_usd', ascending=False

        # Format average salary as US dollars AFTER sorting.
        result_sorted['average_salary_usd'] = result_sorted['average_salary_usd'].a

        # Print the sorted table
        print(result_sorted.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
calculate_and_sort_average_salary_and_level_counts("r project data-1.csv")
```

company_location	average_salary_usd	EN	EX	MI	SE
RU	$157,500.00	0	2	0	0
US	$144,055.26	31	16	90	218
NZ	$125,000.00	0	0	0	1
IL	$119,059.00	0	0	1	0
JP	$114,127.33	1	0	3	2
AU	$108,042.67	2	0	1	0
AE	$100,000.00	0	0	1	2
DZ	$100,000.00	1	0	0	0
IQ	$100,000.00	1	0	0	0
CA	$99,823.73	3	2	10	15
SG	$89,294.00	0	0	1	0
BE	$85,699.00	0	0	1	1
DE	$81,887.21	11	2	8	7
GB	$81,583.04	5	0	30	12
AT	$72,920.75	0	0	3	1
CN	$71,665.50	1	0	1	0
IE	$71,444.00	0	0	0	1
PL	$66,082.50	0	1	3	0
CH	$64,114.00	1	0	1	0
FR	$63,970.67	5	0	6	4
SI	$63,831.00	0	0	1	1
RO	$60,000.00	0	0	1	0
NL	$54,945.75	1	0	2	1
DK	$54,386.33	2	0	0	1
ES	$53,060.14	1	2	10	1
GR	$52,293.09	0	0	10	1
CZ	$50,937.00	1	0	1	0
PT	$47,793.75	1	0	2	1
HR	$45,618.00	0	0	0	1
LU	$43,942.67	2	0	1	0
CL	$40,038.00	0	0	1	0
MY	$40,000.00	1	0	0	0
IT	$36,366.50	1	0	1	0
HU	$35,735.00	0	0	1	0
EE	$32,974.00	0	0	1	0
MX	$32,123.33	0	0	1	2
NG	$30,000.00	1	0	1	0
IN	$28,581.75	9	1	10	4
MT	$28,369.00	0	0	1	0
CO	$21,844.00	1	0	0	0
TR	$20,096.67	0	0	2	1
HN	$20,000.00	0	0	1	0
BR	$18,602.67	0	0	1	2
AS	$18,053.00	1	0	0	0
MD	$18,000.00	0	0	1	0
UA	$13,400.00	1	0	0	0
PK	$13,333.33	1	0	2	0
KE	$9,272.00	1	0	0	0
IR	$4,000.00	0	0	1	0
VN	$4,000.00	1	0	0	0

In [111…

```
import pandas as pd
import os

def calculate_and_sort_average_salary_per_level_by_location(csv_file):
```

```
    """
    Calculates average salary per experience level by location,
    sorts by overall average salary (highest to lowest), and formats salaries as US
    truncating decimals.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_location', 'experience_lev
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate overall average salary by location
        overall_average_salaries = df.groupby('company_location')['salary_in_usd'].
        overall_average_salaries = overall_average_salaries.rename(columns={'salary

        # Calculate average salary per experience level by location
        level_average_salaries = df.groupby(['company_location', 'experience_level'

        # Merge overall and level average salaries
        result = pd.merge(overall_average_salaries, level_average_salaries, on='com

        # Sort overall average salaries from highest to lowest BEFORE formatting.
        result_sorted = result.sort_values(by='average_salary_usd', ascending=False

        # Truncate and format average salaries as US dollars AFTER sorting.
        result_sorted['average_salary_usd'] = result_sorted['average_salary_usd'].a
        for col in level_average_salaries.columns:
            result_sorted[col] = result_sorted[col].apply(lambda x: f"${int(x):,}")

        # Print the sorted table
        print(result_sorted.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
calculate_and_sort_average_salary_per_level_by_location("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607
company_location average_salary_usd       EN       EX       MI       SE
              RU           $157,500       $0 $157,500       $0       $0
              US           $144,055  $93,112 $243,742 $125,780 $151,527
              NZ           $125,000       $0       $0       $0 $125,000
              IL           $119,059       $0       $0 $119,059       $0
              JP           $114,127  $41,689       $0  $71,691 $214,000
              AU           $108,042 $118,351       $0  $87,425       $0
              AE           $100,000       $0       $0 $115,000  $92,500
              DZ           $100,000 $100,000       $0       $0       $0
              IQ           $100,000 $100,000       $0       $0       $0
              CA            $99,823  $57,132 $157,583  $83,530 $111,523
              SG            $89,294       $0       $0  $89,294       $0
              BE            $85,699       $0       $0  $88,654  $82,744
              DE            $81,887  $57,551 $135,936  $72,209 $115,745
              GB            $81,583  $65,604       $0  $80,506  $90,931
              AT            $72,920       $0       $0  $66,815  $91,237
              CN            $71,665 $100,000       $0  $43,331       $0
              IE            $71,444       $0       $0       $0  $71,444
              PL            $66,082       $0 $153,667  $36,887       $0
              CH            $64,114   $5,882       $0 $122,346       $0
              FR            $63,970  $47,325       $0  $57,771  $94,075
              SI            $63,831       $0       $0  $24,823 $102,839
              RO            $60,000       $0       $0  $60,000       $0
              NL            $54,945  $42,000       $0  $57,566  $62,651
              DK            $54,386  $37,252       $0       $0  $88,654
              ES            $53,060  $10,354  $74,787  $52,791  $55,000
              GR            $52,293       $0       $0  $52,732  $47,899
              CZ            $50,937  $31,875       $0  $69,999       $0
              PT            $47,793  $21,983       $0  $54,217  $60,757
              HR            $45,618       $0       $0       $0  $45,618
              LU            $43,942  $34,551       $0  $62,726       $0
              CL            $40,038       $0       $0  $40,038       $0
              MY            $40,000  $40,000       $0       $0       $0
```

IT	$36,366	$21,669	$0	$51,064	$0
HU	$35,735	$0	$0	$35,735	$0
EE	$32,974	$0	$0	$32,974	$0
MX	$32,123	$0	$0	$2,859	$46,755
NG	$30,000	$10,000	$0	$50,000	$0
IN	$28,581	$19,629	$79,039	$20,441	$56,460
MT	$28,369	$0	$0	$28,369	$0
CO	$21,844	$21,844	$0	$0	$0
TR	$20,096	$0	$0	$20,059	$20,171
HN	$20,000	$0	$0	$20,000	$0
BR	$18,602	$0	$0	$12,901	$21,453
AS	$18,053	$18,053	$0	$0	$0
MD	$18,000	$0	$0	$18,000	$0
UA	$13,400	$13,400	$0	$0	$0
PK	$13,333	$20,000	$0	$10,000	$0
KE	$9,272	$9,272	$0	$0	$0
IR	$4,000	$0	$0	$4,000	$0
VN	$4,000	$4,000	$0	$0	$0

In [112…

```python
import pandas as pd
import os

def list_unique_job_titles(csv_file):
    """
    Reads a CSV file and prints a list of unique job titles found in the 'job_title
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())

        if 'job_title' in df.columns:
            unique_titles = df['job_title'].unique()
            print("\nUnique Job Titles:")
            for title in unique_titles:
                print(f"- {title}")
        else:
            print("Error: 'job_title' column not found in the CSV file.")

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
list_unique_job_titles("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L

Unique Job Titles:
- Data Scientist
- Machine Learning Scientist
- Big Data Engineer
- Product Data Analyst
- Machine Learning Engineer
- Data Analyst
- Lead Data Scientist
- Business Data Analyst
- Lead Data Engineer
- Lead Data Analyst
- Data Engineer
- Data Science Consultant
- BI Data Analyst
- Director of Data Science
- Research Scientist
- Machine Learning Manager
- Data Engineering Manager
- Machine Learning Infrastructure Engineer
- ML Engineer
- AI Scientist
- Computer Vision Engineer
- Principal Data Scientist
- Data Science Manager
- Head of Data
- 3D Computer Vision Researcher
- Data Analytics Engineer
- Applied Data Scientist
- Marketing Data Analyst
- Cloud Data Engineer
- Financial Data Analyst
- Computer Vision Software Engineer
- Director of Data Engineering
```

- Data Science Engineer
- Principal Data Engineer
- Machine Learning Developer
- Applied Machine Learning Scientist
- Data Analytics Manager
- Head of Data Science
- Data Specialist
- Data Architect
- Finance Data Analyst
- Principal Data Analyst
- Big Data Architect
- Staff Data Scientist
- Analytics Engineer
- ETL Developer
- Head of Machine Learning
- NLP Engineer
- Lead Machine Learning Engineer
- Data Analytics Lead

In [113…

```python
import pandas as pd
import os

def list_job_titles_by_experience_level(csv_file):
    """
    Reads a CSV file and creates a table listing job titles for each experience lev
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())

        if 'experience_level' in df.columns and 'job_title' in df.columns:
            job_titles_by_level = df.groupby('experience_level')['job_title'].apply
            job_titles_by_level['job_title'] = job_titles_by_level['job_title'].app

            print("\nJob Titles by Experience Level:")
            print(job_titles_by_level.to_string(index=False))
        else:
            print("Error: Both 'experience_level' and 'job_title' columns are requi

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
list_job_titles_by_experience_level("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L

Job Titles by Experience Level:
experience_level
job_title
              EN
Research Scientist, Business Data Analyst, Machine Learning Engineer, BI Data Analys
t, Data Science Consultant, Machine Learning Scientist, Computer Vision Software Eng
ineer, ML Engineer, AI Scientist, Financial Data Analyst, Data Engineer, Big Data En
gineer, Data Scientist, Computer Vision Engineer, Data Analytics Engineer, Machine L
earning Developer, Applied Data Scientist, Data Analyst, Applied Machine Learning Sc
ientist
              EX
BI Data Analyst, Principal Data Engineer, Data Engineer, Data Science Consultant, Le
ad Data Engineer, Principal Data Scientist, Head of Machine Learning, Head of Data S
cience, Data Analyst, Director of Data Science, Analytics Engineer, Data Engineering
Manager, Head of Data
              MI
Product Data Analyst, NLP Engineer, Lead Data Engineer, Research Scientist, Principa
l Data Scientist, Head of Data Science, Data Architect, Principal Data Analyst, Busi
ness Data Analyst, Machine Learning Engineer, BI Data Analyst, Data Science Consulta
nt, Machine Learning Scientist, Data Science Manager, Computer Vision Software Engin
eer, ML Engineer, AI Scientist, Financial Data Analyst, Data Engineer, Big Data Engi
neer, ETL Developer, Data Science Engineer, Data Scientist, Lead Data Scientist, Clo
ud Data Engineer, Data Analytics Engineer, Machine Learning Developer, Data Engineer
ing Manager, 3D Computer Vision Researcher, Machine Learning Infrastructure Enginee
r, Lead Data Analyst, Applied Data Scientist, Data Analyst, Applied Machine Learning
Scientist, Head of Data
              SE Lead Data Engineer, Principal Data Scientist, Research Scientist, D
ata Specialist, Principal Data Analyst, Director of Data Science, Data Architect, Ma
chine Learning Engineer, Big Data Architect, Staff Data Scientist, Machine Learning
Scientist, Data Science Manager, ML Engineer, AI Scientist, Machine Learning Manage
r, Marketing Data Analyst, Principal Data Engineer, Data Engineer, Big Data Enginee
r, Data Science Engineer, Data Scientist, Computer Vision Engineer, Lead Data Scient
ist, Cloud Data Engineer, Lead Machine Learning Engineer, Data Analytics Manager, Di
```

```
rector of Data Engineering, Data Analytics Engineer, Machine Learning Developer, Dat
a Engineering Manager, Lead Data Analyst, Machine Learning Infrastructure Engineer,
Applied Data Scientist, Finance Data Analyst, Data Analytics Lead, Analytics Enginee
r, Data Analyst, Head of Data
```

In [114…

```python
import pandas as pd
import os

def calculate_and_sort_average_salary_by_company_size(csv_file):
    """
    Calculates the average salary for each company size and sorts the results
    first by company size (alphabetically) and then by average salary
    (highest to lowest). Formats salaries as US dollars, truncating decimals.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_size'])
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate average salary per company size
        average_salaries_by_size = df.groupby('company_size')['salary_in_usd'].mean
        average_salaries_by_size = average_salaries_by_size.rename(columns={'salary

        # Sort first by company size (alphabetically) and then by average salary (d
        sorted_results = average_salaries_by_size.sort_values(by=['company_size', '

        # Truncate and format average salaries as US dollars AFTER sorting.
        sorted_results['average_salary_usd'] = sorted_results['average_salary_usd']

        # Print the sorted table
        print("\nAverage Salary by Company Size (Sorted):")
        print(sorted_results.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
calculate_and_sort_average_salary_by_company_size("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607

Average Salary by Company Size (Sorted):
company_size average_salary_usd
           L           $119,242
           M           $116,905
           S            $77,632
```

In [115…

```python
import pandas as pd
import os

def calculate_and_sort_average_salary_by_company_size_location(csv_file):
    """
    Calculates the average salary for each combination of company size and location
    and sorts the results first by company size (alphabetically), then by location
    (alphabetically), and finally by average salary (highest to lowest).
    Formats salaries as US dollars, truncating decimals.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_size', 'company_location'])
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate average salary per company size and location
        average_salaries_by_size_location = df.groupby(['company_size', 'company_lo
        average_salaries_by_size_location = average_salaries_by_size_location.renam

        # Sort by company size, then location, then average salary (descending)
        sorted_results = average_salaries_by_size_location.sort_values(
            by=['company_size', 'company_location', 'average_salary_usd'],
```

```
            ascending=[True, True, False]
        )

        # Truncate and format average salaries as US dollars AFTER sorting.
        sorted_results['average_salary_usd'] = sorted_results['average_salary_usd']

        # Print the sorted table
        print("\nAverage Salary by Company Size and Location (Sorted):")
        print(sorted_results.to_string(index=False))

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
calculate_and_sort_average_salary_by_company_size_location("r project data-1.csv")
```

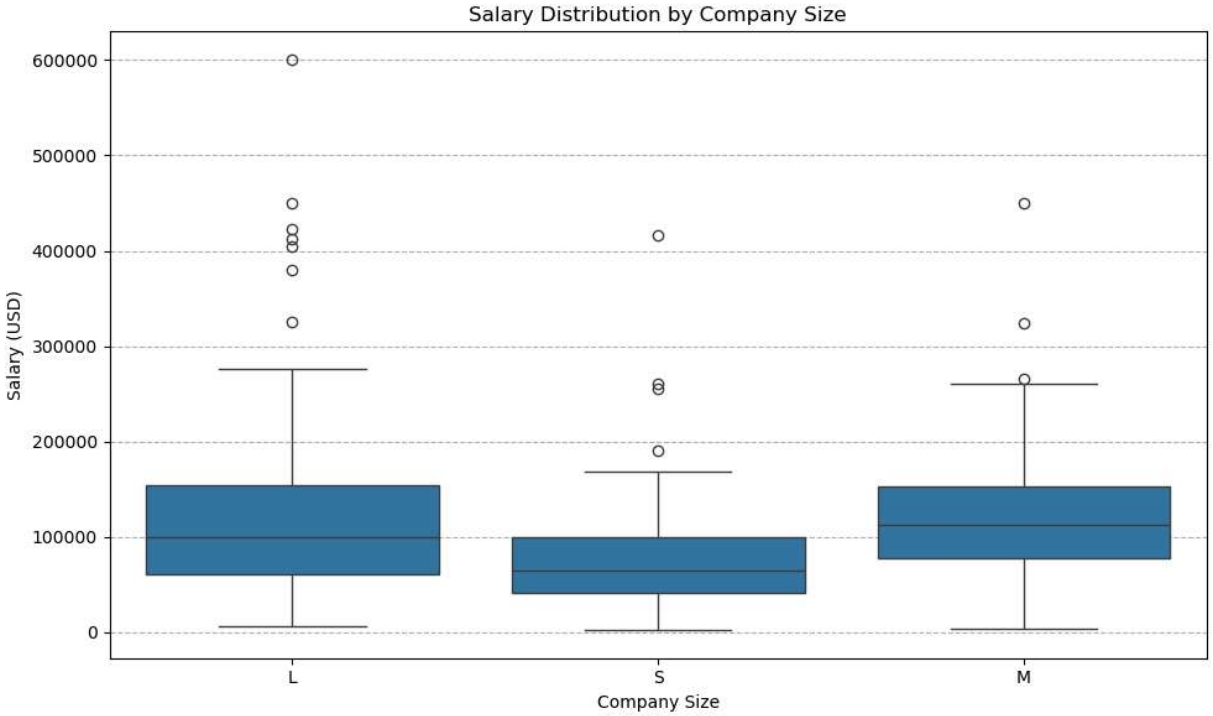
```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607

Average Salary by Company Size and Location (Sorted):
company_size company_location average_salary_usd
           L               AE           $115,000
           L               AT            $69,489
           L               AU            $87,425
           L               CA           $118,806
           L               CH            $64,114
           L               CL            $40,038
           L               CN           $100,000
           L               CZ            $69,999
           L               DE            $81,646
           L               DK            $88,654
           L               ES            $57,552
           L               FR            $75,360
           L               GB            $80,031
           L               GR            $47,899
           L               HU            $35,735
           L               IN            $38,590
           L               IT            $51,064
           L               LU            $59,102
           L               MT            $28,369
           L               MX            $60,000
           L               MY            $40,000
           L               NG            $50,000
           L               NL            $54,945
           L               PK             $8,000
           L               PL            $66,082
           L               PT            $46,998
           L               RU           $230,000
           L               SG            $89,294
           L               SI            $63,831
           L               TR            $20,171
```

L	UA	$13,400
L	US	$160,967
M	AT	$61,467
M	AU	$86,703
M	BE	$85,699
M	BR	$21,453
M	CA	$87,932
M	CN	$43,331
M	CO	$21,844
M	CZ	$31,875
M	DE	$104,273
M	DZ	$100,000
M	ES	$50,351
M	FR	$59,524
M	GB	$81,194
M	GR	$56,369
M	IL	$119,059
M	IN	$18,834
M	IR	$4,000
M	LU	$10,000
M	PK	$16,000
M	PT	$50,180
M	RO	$60,000
M	RU	$85,000
M	TR	$20,059
M	US	$141,446
M	VN	$4,000
S	AE	$92,500
S	AS	$18,053
S	AT	$91,237
S	AU	$150,000
S	BR	$12,901
S	CA	$93,641
S	DE	$53,518
S	DK	$37,252
S	EE	$32,974
S	ES	$58,511
S	FR	$50,085
S	GB	$87,404
S	GR	$20,000
S	HN	$20,000
S	HR	$45,618
S	IE	$71,444
S	IN	$13,110
S	IQ	$100,000
S	IT	$21,669
S	JP	$114,127
S	KE	$9,272
S	LU	$62,726
S	MD	$18,000
S	MX	$18,185
S	NG	$10,000
S	NZ	$125,000
S	US	$104,570

In [116…

```python
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns

def analyze_salary_by_company_size_with_iqr_details(csv_file):
    """
    Generates a box plot of salary distribution for each company size
    and prints the interquartile range (IQR), the first quartile (Q1),
    and the third quartile (Q3) for each size.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'company_size'])
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate Q1, Q3, and IQR for each company size
        quartile_info = df.groupby('company_size')['salary_in_usd'].agg(
            Q1=lambda x: x.quantile(0.25),
            Q3=lambda x: x.quantile(0.75),
            IQR=lambda x: x.quantile(0.75) - x.quantile(0.25)
        ).reset_index()

        print("\nSalary Quartile Information by Company Size:")
        for index, row in quartile_info.iterrows():
            print(f"Company Size: {row['company_size']}, Q1 (Bottom of Box): ${int(

        # Create box plot
        plt.figure(figsize=(10, 6))
        sns.boxplot(x='company_size', y='salary_in_usd', data=df)
        plt.title('Salary Distribution by Company Size')
        plt.xlabel('Company Size')
        plt.ylabel('Salary (USD)')
        plt.grid(axis='y', linestyle='--')
        plt.tight_layout()
        plt.show()

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
analyze_salary_by_company_size_with_iqr_details("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607

Salary Quartile Information by Company Size:
Company Size: L, Q1 (Bottom of Box): $61,041, Q3 (Top of Box): $154,600, IQR: $93,55
8
Company Size: M, Q1 (Bottom of Box): $78,131, Q3 (Top of Box): $152,875, IQR: $74,74
3
Company Size: S, Q1 (Bottom of Box): $41,943, Q3 (Top of Box): $100,000, IQR: $58,05
7
```



In [117…

```
import pandas as pd
import os
```

```python
import matplotlib.pyplot as plt
import seaborn as sns

def analyze_salary_by_experience_level_with_iqr_and_mean(csv_file):
    """
    Generates a box plot of salary distribution for each experience level
    and prints the interquartile range (IQR), the first quartile (Q1),
    the third quartile (Q3), and the average salary for each experience level.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'experience_level'])
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate Q1, Q3, IQR, and mean salary for each experience level
        salary_info = df.groupby('experience_level')['salary_in_usd'].agg(
            Q1=lambda x: x.quantile(0.25),
            Q3=lambda x: x.quantile(0.75),
            IQR=lambda x: x.quantile(0.75) - x.quantile(0.25),
            Average='mean'
        ).reset_index()

        print("\nSalary Information by Experience Level:")
        for index, row in salary_info.iterrows():
            print(f"Experience Level: {row['experience_level']}, "
                  f"Q1 (Bottom of Box): ${int(row['Q1']):,}, "
                  f"Q3 (Top of Box): ${int(row['Q3']):,}, "
                  f"IQR: ${int(row['IQR']):,}, "
                  f"Average Salary: ${int(row['Average']):,}")

        # Create box plot
        plt.figure(figsize=(10, 6))
        sns.boxplot(x='experience_level', y='salary_in_usd', data=df)
        plt.title('Salary Distribution by Experience Level')
        plt.xlabel('Experience Level')
        plt.ylabel('Salary (USD)')
        plt.grid(axis='y', linestyle='--')
        plt.tight_layout()
        plt.show()

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
analyze_salary_by_experience_level_with_iqr_and_mean("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607

Salary Information by Experience Level:
Experience Level: EN, Q1 (Bottom of Box): $27,505, Q3 (Top of Box): $85,425, IQR: $5
7,920, Average Salary: $61,643
Experience Level: EX, Q1 (Bottom of Box): $130,006, Q3 (Top of Box): $233,750, IQR:
$103,743, Average Salary: $199,392
Experience Level: MI, Q1 (Bottom of Box): $48,000, Q3 (Top of Box): $112,000, IQR:
$64,000, Average Salary: $87,996
Experience Level: SE, Q1 (Bottom of Box): $100,000, Q3 (Top of Box): $170,000, IQR:
$70,000, Average Salary: $138,617
```

In [118…

```python
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns

def analyze_salary_by_experience_level_with_iqr_and_mean(csv_file):
    """
    Generates a box plot of salary distribution for each experience level
    and prints the interquartile range (IQR), the first quartile (Q1),
    the third quartile (Q3), and the average salary for each experience level.
    """
    try:
        print(f"Current working directory: {os.getcwd()}")
        df = pd.read_csv(csv_file)
        print(f"DataFrame length after read: {len(df)}")
        print(df.head())
        df['salary_in_usd'] = pd.to_numeric(df['salary_in_usd'], errors='coerce')
        df = df.dropna(subset=['salary_in_usd', 'experience_level'])
        print(f"DataFrame length after dropna: {len(df)}")

        # Calculate Q1, Q3, IQR, and mean salary for each experience level
        salary_info = df.groupby('experience_level')['salary_in_usd'].agg(
            Q1=lambda x: x.quantile(0.25),
            Q3=lambda x: x.quantile(0.75),
            IQR=lambda x: x.quantile(0.75) - x.quantile(0.25),
            Average='mean'
        ).reset_index()

        print("\nSalary Information by Experience Level:")
        for index, row in salary_info.iterrows():
            print(f"Experience Level: {row['experience_level']}, "
                  f"Q1 (Bottom of Box): ${int(row['Q1']):,}, "
                  f"Q3 (Top of Box): ${int(row['Q3']):,}, "
                  f"IQR: ${int(row['IQR']):,}, "
                  f"Average Salary: ${int(row['Average']):,}")

        # Create box plot
        plt.figure(figsize=(10, 6))
        sns.boxplot(x='experience_level', y='salary_in_usd', data=df)
        plt.title('Salary Distribution by Experience Level')
        plt.xlabel('Experience Level')
        plt.ylabel('Salary (USD)')
        plt.grid(axis='y', linestyle='--')
        plt.tight_layout()
        plt.show()

    except FileNotFoundError:
        print(f"Error: File '{csv_file}' not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
analyze_salary_by_experience_level_with_iqr_and_mean("r project data-1.csv")
```

```
Current working directory: C:\Users\petea
DataFrame length after read: 607
   Unnamed: 0  work_year experience_level employment_type  \
0           0       2020               MI              FT
1           1       2020               SE              FT
2           2       2020               SE              FT
3           3       2020               MI              FT
4           4       2020               SE              FT

                    job_title  salary salary_currency  salary_in_usd  \
0              Data Scientist   70000             EUR          79833
1  Machine Learning Scientist  260000             USD         260000
2           Big Data Engineer   85000             GBP         109024
3        Product Data Analyst   20000             USD          20000
4   Machine Learning Engineer  150000             USD         150000

  employee_residence  remote_ratio company_location company_size
0                 DE             0               DE            L
1                 JP             0               JP            S
2                 GB            50               GB            M
3                 HN             0               HN            S
4                 US            50               US            L
DataFrame length after dropna: 607

Salary Information by Experience Level:
Experience Level: EN, Q1 (Bottom of Box): $27,505, Q3 (Top of Box): $85,425, IQR: $5
7,920, Average Salary: $61,643
Experience Level: EX, Q1 (Bottom of Box): $130,006, Q3 (Top of Box): $233,750, IQR:
$103,743, Average Salary: $199,392
Experience Level: MI, Q1 (Bottom of Box): $48,000, Q3 (Top of Box): $112,000, IQR:
$64,000, Average Salary: $87,996
Experience Level: SE, Q1 (Bottom of Box): $100,000, Q3 (Top of Box): $170,000, IQR:
$70,000, Average Salary: $138,617
```

In [ ]: