ChrisRodgers.Python.Project

May 1, 2024

Python Project

Christopher Rodgers

1 May 2024

[7]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[8]:
```
ds_salaries = pd.read_csv('C:/Users/crodg/Documents/DSE5002/DSE5002/Week 8/
 ↪Python Project/ds_salaries.csv')
cost_of_living = pd.read_csv('C:/Users/crodg/Documents/DSE5002/DSE5002/Week 8/
 ↪Python Project/cost_of_living.csv')
```

[9]:
```
ds_salaries_filtered = ds_salaries[ds_salaries['job_title'] == 'Data Scientist']
```

[10]:
```
ds_salaries_filtered = ds_salaries[ds_salaries['job_title'] == 'Data Scientist'].
 ↪copy()
```

[31]:
```
mean_salary_data_scientist = ds_salaries_filtered['salary_in_usd'].mean()
```

[32]:
```
print("The mean salary of a Data Scientist is:", mean_salary_data_scientist)
```

```
The mean salary of a Data Scientist is: 108187.83216783217
```

[12]:
```
country_code_to_name = {
    'DE': 'Germany', 'HU': 'Hungary', 'FR': 'France', 'IN': 'India', 'US':␣
 ↪'United States',
    'GB': 'United Kingdom', 'ES': 'Spain', 'IT': 'Italy', 'AT': 'Austria', 'LU':␣
 ↪'Luxembourg',
    'NG': 'Nigeria', 'CA': 'Canada', 'UA': 'Ukraine', 'IL': 'Israel', 'MX':␣
 ↪'Mexico',
    'CL': 'Chile', 'BR': 'Brazil', 'VN': 'Vietnam', 'TR': 'Turkey', 'DZ':␣
 ↪'Algeria',
    'PL': 'Poland', 'MY': 'Malaysia', 'AU': 'Australia', 'CH': 'Switzerland'
}
```

[13]:
```
ds_salaries_filtered.loc[:, 'country_name'] =␣
 ↪ds_salaries_filtered['company_location'].map(country_code_to_name)
```

[39]:
```
ds_salaries_filtered['country_name'] = ds_salaries_filtered['country_name'].
↪fillna(ds_salaries_filtered['company_location'])
```

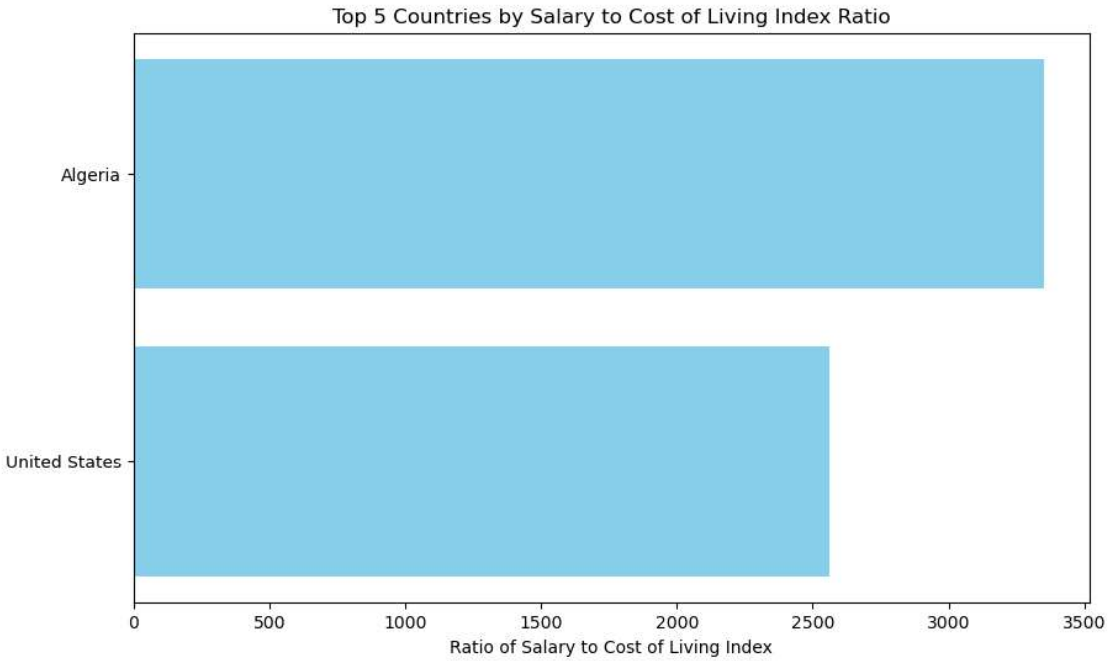
[16]:
```
average_salary_by_country_name = ds_salaries_filtered.
↪groupby('country_name')['salary_in_usd'].mean().reset_index()
```

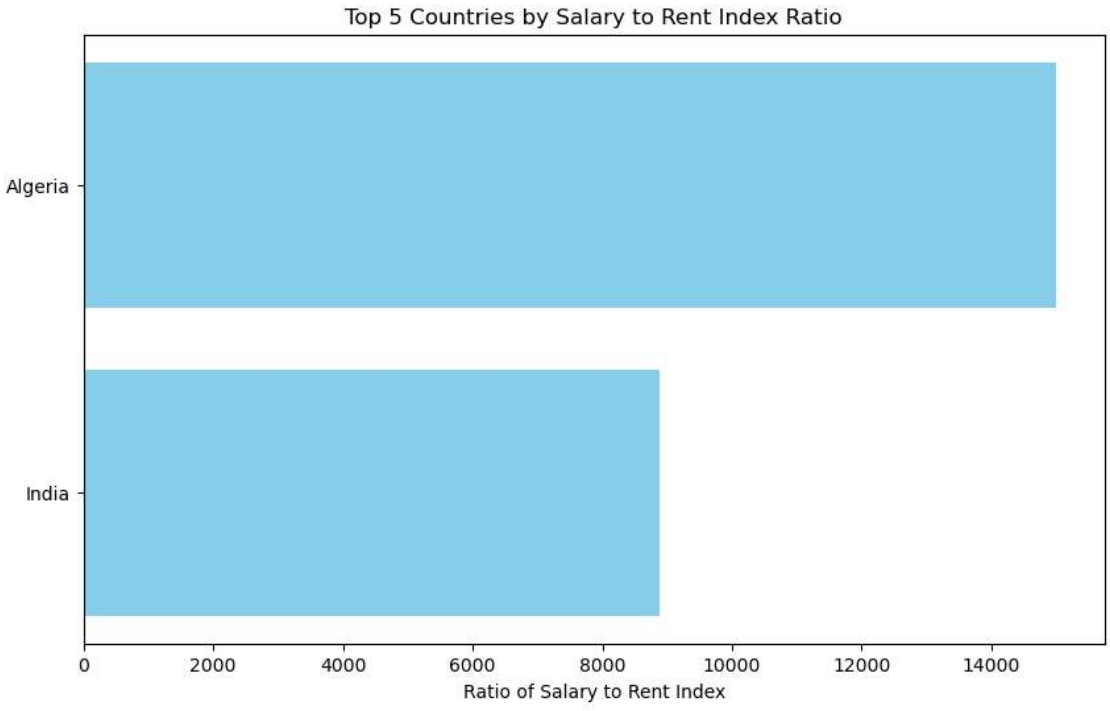
[17]:
```
cost_of_living['Country'] = cost_of_living['City'].apply(lambda x: x.split(',␣
↪')[-1] if ', ' in x else x)
numeric_cols = cost_of_living.select_dtypes(include=['number']).columns.tolist()
cost_of_living_by_country = cost_of_living.groupby('Country')[numeric_cols].
↪mean().reset_index()
```

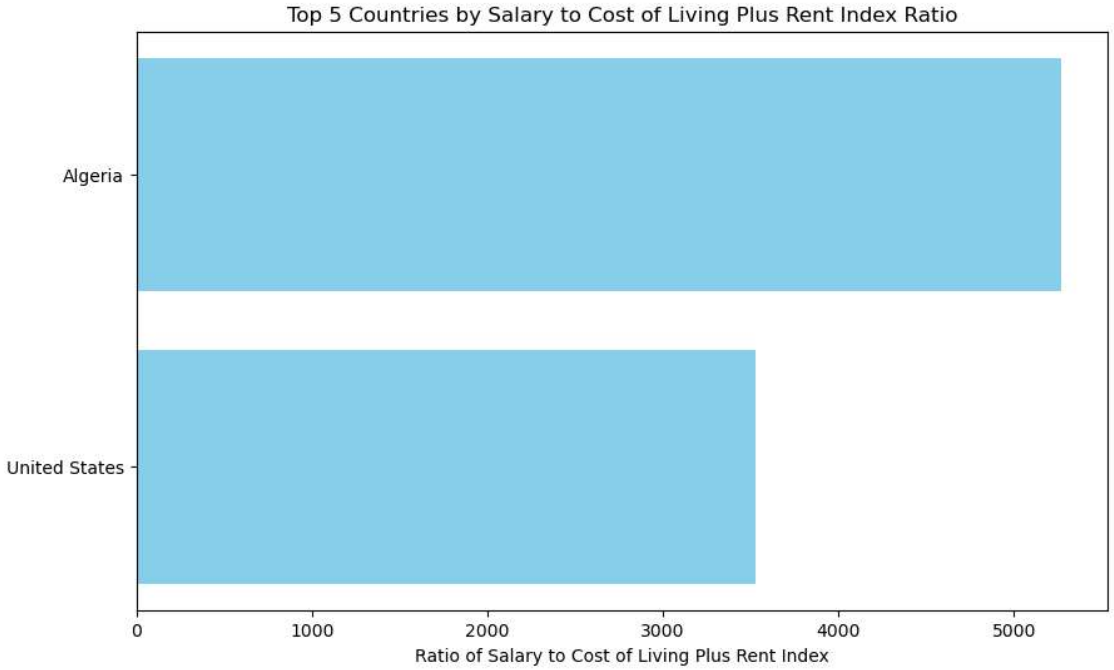
[18]:
```
merged_data = pd.merge(average_salary_by_country_name, cost_of_living,␣
↪left_on='country_name', right_on='Country', how='inner')
index_columns = ['Cost of Living Index', 'Rent Index', 'Cost of Living Plus Rent␣
↪Index', 'Groceries Index', 'Restaurant Price Index', 'Local Purchasing Power␣
↪Index']
for index in index_columns:
    merged_data[index + ' Ratio'] = merged_data['salary_in_usd'] /␣
↪merged_data[index]
top_cities_by_index = {index: merged_data.nlargest(5, index + ' Ratio')[['City',␣
↪'salary_in_usd', index, index + ' Ratio']]
                       for index in index_columns}
```

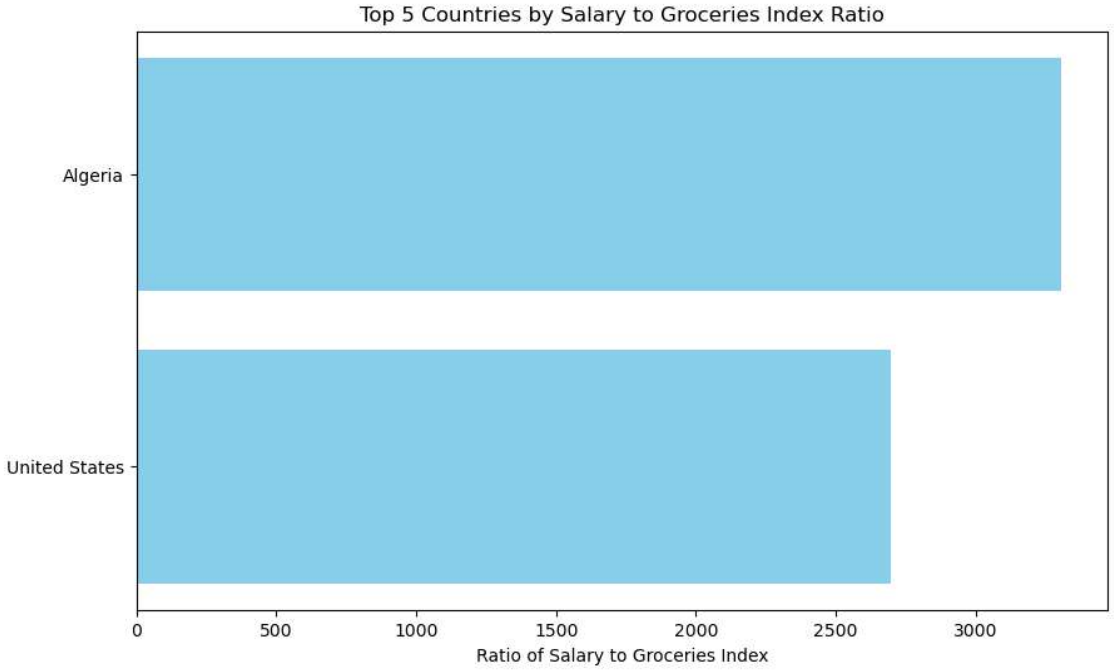
[19]:
```
top_countries_by_index = {index: merged_data.nlargest(5, index + '␣
↪Ratio')[['Country', 'salary_in_usd', index, index + ' Ratio']]
                          for index in index_columns}
```

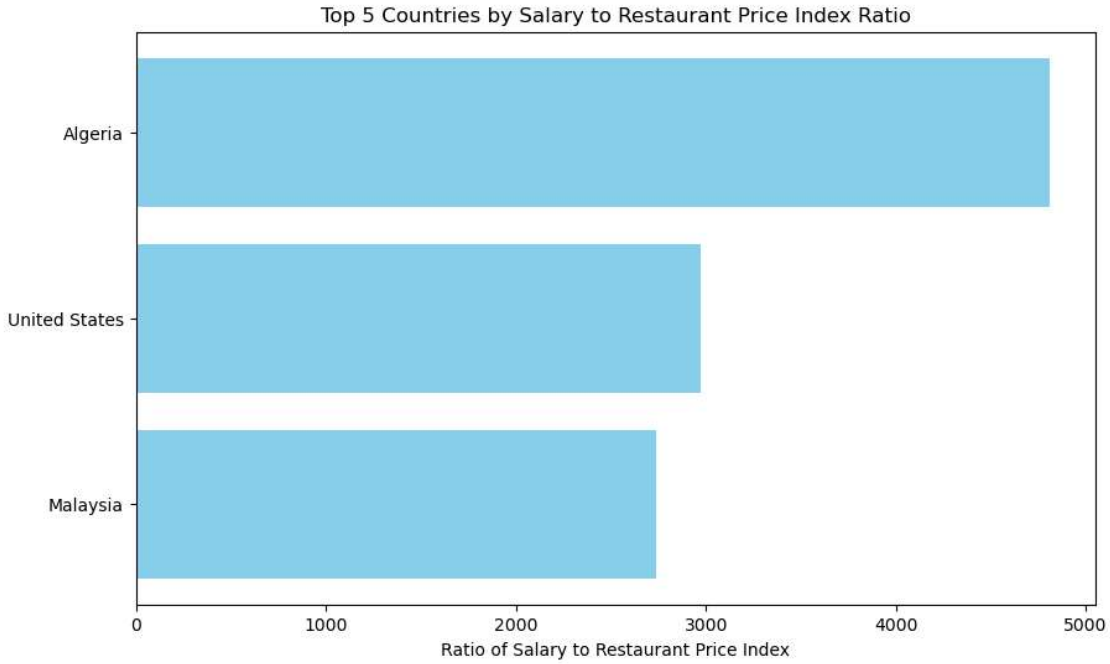
[20]:
```
def create_bar_plot(data, index):
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.barh(data['Country'], data[index + ' Ratio'], color='skyblue')
    ax.set_xlabel('Ratio of Salary to ' + index)
    ax.set_title('Top 5 Countries by Salary to ' + index + ' Ratio')
    plt.gca().invert_yaxis()
    plt.show()

for index in index_columns:
    create_bar_plot(top_countries_by_index[index], index)
```

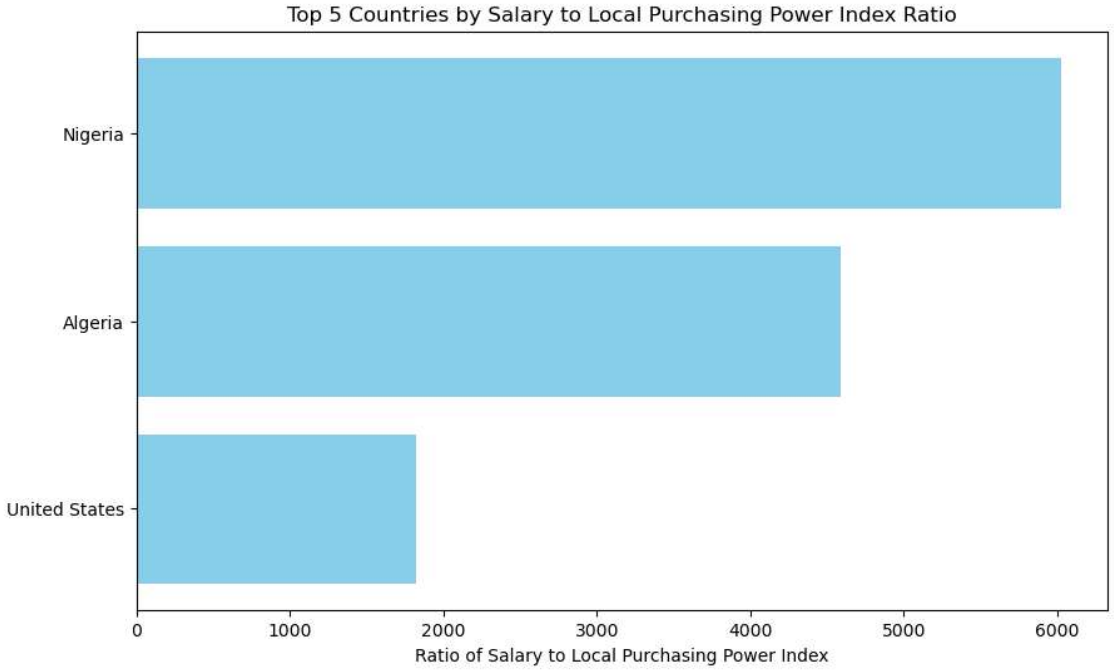

Top 5 Countries by Salary to Cost of Living Index Ratio
Algeria
United States
0
500
1000
1500
2000
2500
3000
3500
Ratio of Salary to Cost of Living Index


Top 5 Countries by Salary to Rent Index Ratio
Algeria
India
0
2000
4000
6000
8000
10000
12000
14000
Ratio of Salary to Rent Index


Top 5 Countries by Salary to Cost of Living Plus Rent Index Ratio
Algeria
United States
0
1000
2000
3000
4000
5000
Ratio of Salary to Cost of Living Plus Rent Index

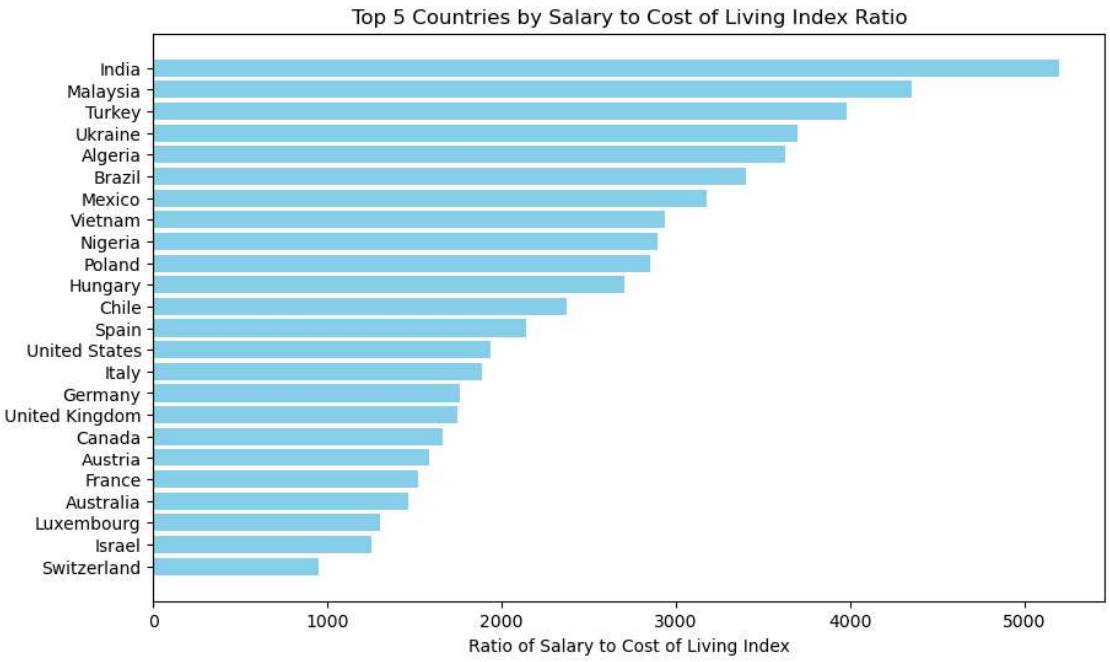

Top 5 Countries by Salary to Groceries Index Ratio
Algeria
United States
0
500
1000
1500
2000
2500
3000
Ratio of Salary to Groceries Index

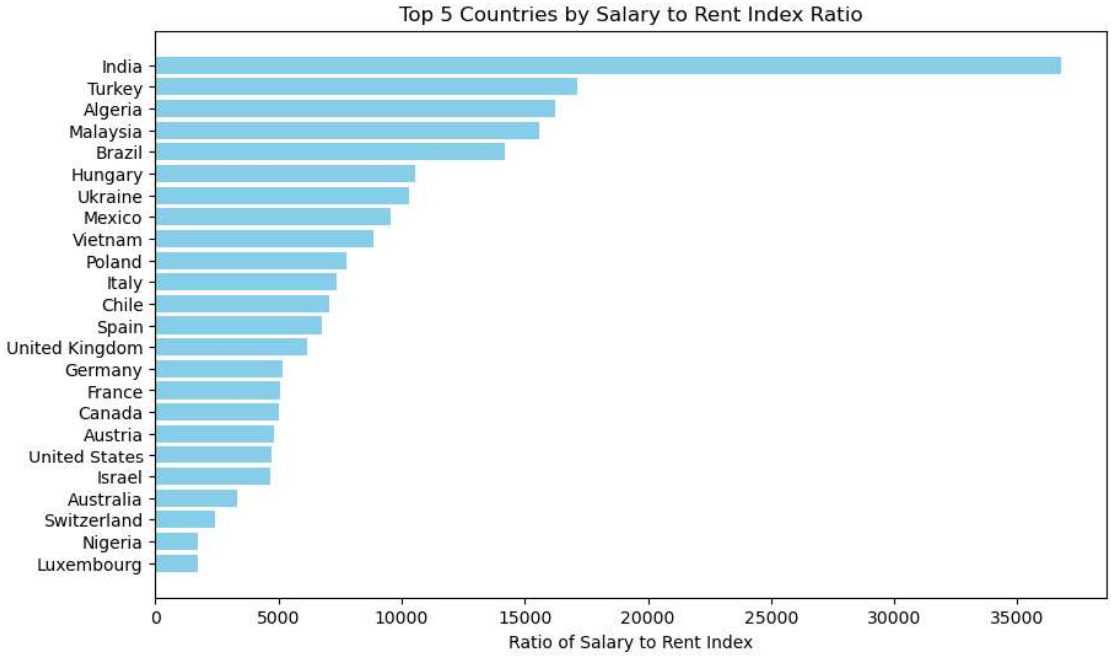


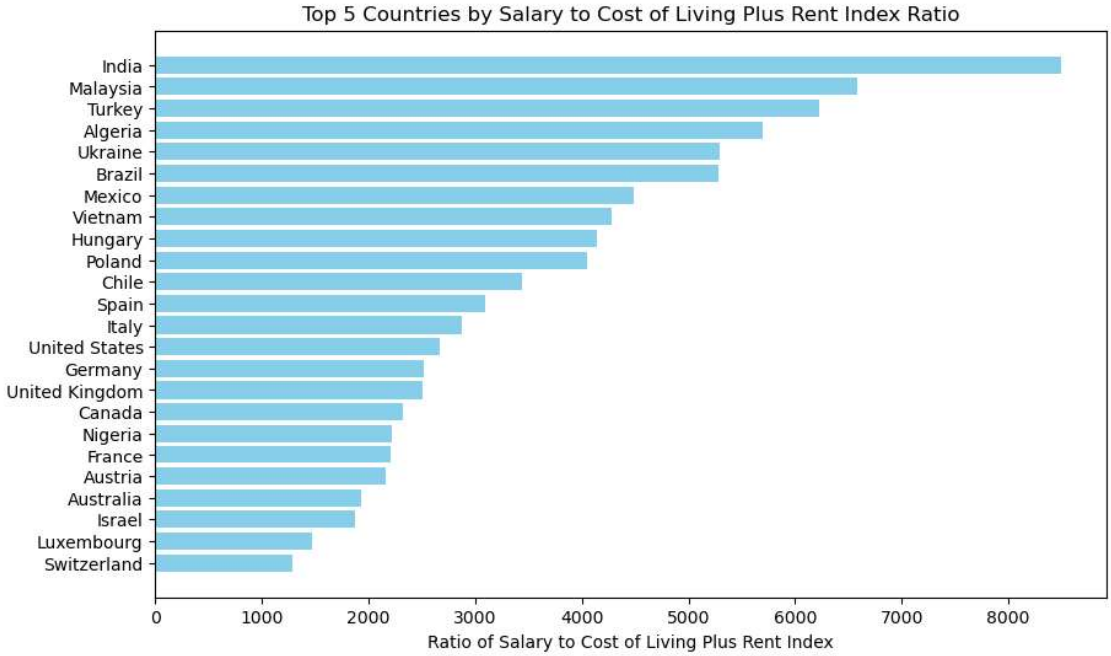


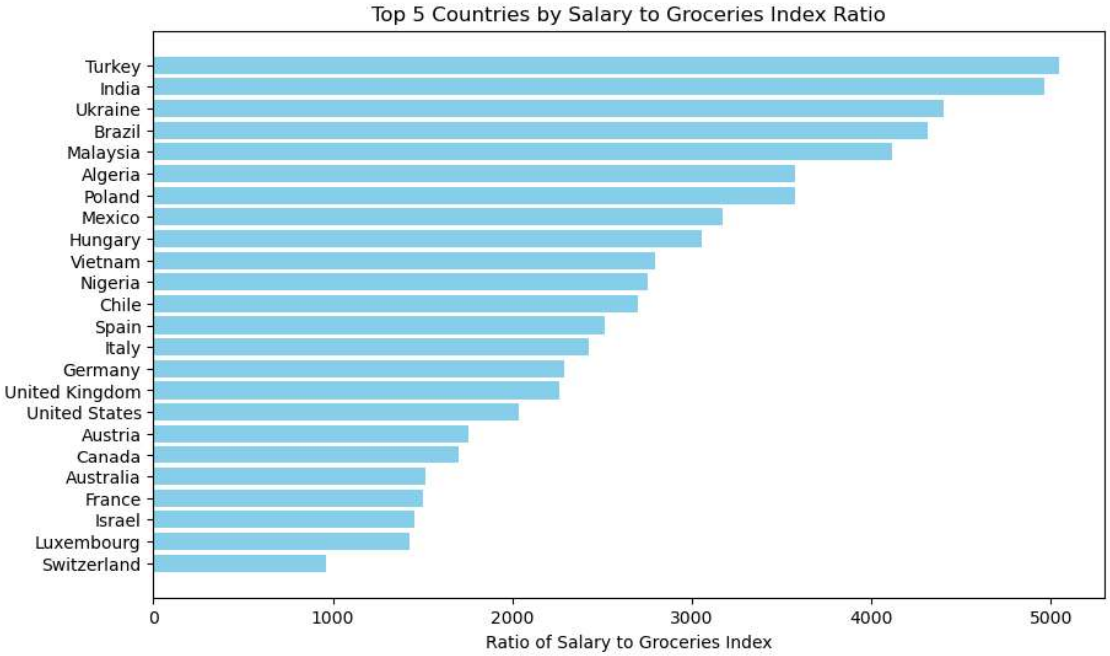
[40]:
```
def create_bar_plot(data, index):
    fig, ax = plt.subplots(figsize=(10, 6))
    unique_top_data = data.drop_duplicates(subset=[index + ' Ratio'])
```

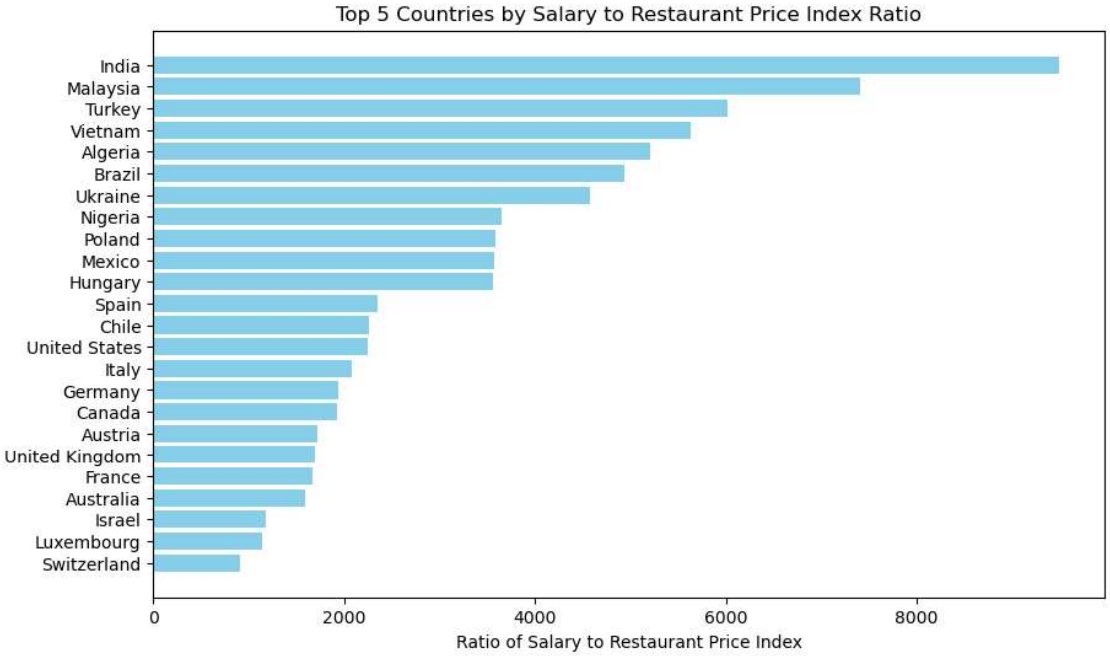
```
    top_data = unique_top_data.nlargest(max(5, unique_top_data.shape[0]), index␣
↪+ ' Ratio')
    ax.barh(top_data['Country'], top_data[index + ' Ratio'], color='skyblue')
    ax.set_xlabel('Ratio of Salary to ' + index)
    ax.set_title('Top 5 Countries by Salary to ' + index + ' Ratio')
    plt.gca().invert_yaxis()
    plt.show()

for index in index_columns:
    create_bar_plot(merged_data, index)
```

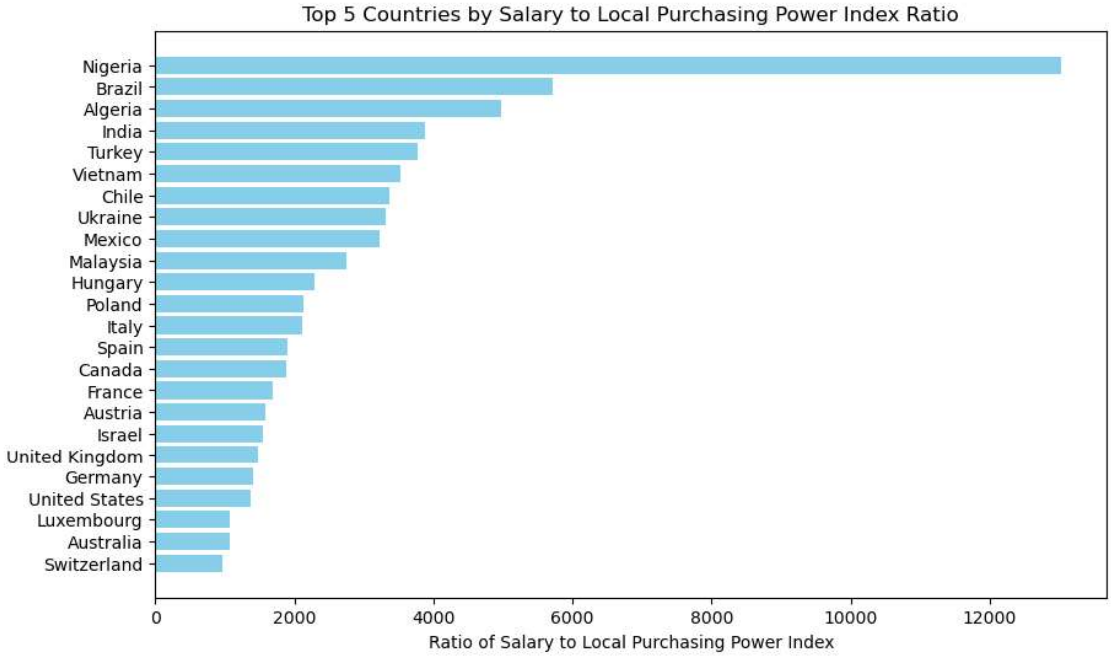



Top 5 Countries by Salary to Rent Index Ratio
India
Turkey
Algeria
Malaysia
Brazil
Hungary
Ukraine
Mexico
Vietnam
Poland
Italy
Chile
Spain
United Kingdom
Germany
France
Canada
Austria
United States
Israel
Australia
Switzerland
Nigeria
Luxembourg
0
5000
10000
15000
20000
25000
30000
35000
Ratio of Salary to Rent Index


Top 5 Countries by Salary to Cost of Living Plus Rent Index Ratio
India
Malaysia
Turkey
Algeria
Ukraine
Brazil
Mexico
Vietnam
Hungary
Poland
Chile
Spain
Italy
United States
Germany
United Kingdom
Canada
Nigeria
France
Austria
Australia
Israel
Luxembourg
Switzerland
0
1000
2000
3000
4000
5000
6000
7000
8000
Ratio of Salary to Cost of Living Plus Rent Index


Top 5 Countries by Salary to Groceries Index Ratio
Turkey
India
Ukraine
Brazil
Malaysia
Algeria
Poland
Mexico
Hungary
Vietnam
Nigeria
Chile
Spain
Italy
Germany
United Kingdom
United States
Austria
Canada
Australia
France
Israel
Luxembourg
Switzerland
0
1000
2000
3000
4000
5000
Ratio of Salary to Groceries Index


Top 5 Countries by Salary to Restaurant Price Index Ratio
India
Malaysia
Turkey
Vietnam
Algeria
Brazil
Ukraine
Nigeria
Poland
Mexico
Hungary
Spain
Chile
United States
Italy
Germany
Canada
Austria
United Kingdom
France
Australia
Israel
Luxembourg
Switzerland
0
2000
4000
6000
8000
Ratio of Salary to Restaurant Price Index



[22]:
```
import numpy as np
```

[23]:
```
for index in index_columns:
    max_value = merged_data[index + ' Ratio'].max()
    merged_data[index + ' Normalized'] = merged_data[index + ' Ratio'] /␣
 ↪max_value
```

[24]:
```
merged_data['Composite Score'] = merged_data[[index + ' Normalized' for index in␣
 ↪index_columns]].sum(axis=1)
```

[25]:
```
top_overall_countries = merged_data.nlargest(5, 'Composite Score')
```

[26]:
```
top_overall_countries[['Country', 'Composite Score']]
```

[26]:

	Country	Composite Score
0	Algeria	5.762167
338	United States	3.458059
337	United States	3.287904
335	United States	3.171545
336	United States	3.115697

[27]:
```
merged_data_sorted = merged_data.sort_values(by='Composite Score',␣
 ↪ascending=False)
```

[28]:
```
merged_data_unique = merged_data_sorted.drop_duplicates(subset='Country')
```

[29]:
```
top_overall_unique_countries = merged_data_unique.head(5)
```

[30]:
```
top_overall_unique_countries[['Country', 'Composite Score']]
```

[30]:

	Country	Composite Score
0	Algeria	5.762167
338	United States	3.458059
163	Malaysia	2.408360
170	Nigeria	2.384230
137	Israel	2.124315

[33]:
```
for index in index_columns:
    merged_data[index + ' Ratio'] = mean_salary_data_scientist /␣
 ↪merged_data[index]
```

[34]:
```
for index in index_columns:
    max_value = merged_data[index + ' Ratio'].max()
    merged_data[index + ' Normalized'] = merged_data[index + ' Ratio'] /␣
 ↪max_value
```

[35]:
```
merged_data['Composite Score'] = merged_data[[index + ' Normalized' for index in␣
 ↪index_columns]].sum(axis=1)
```

[36]:
```
merged_data_sorted = merged_data.sort_values(by='Composite Score',␣
 ↪ascending=False)
merged_data_unique = merged_data_sorted.drop_duplicates(subset='Country')
```

[37]:
```
top_overall_unique_countries = merged_data_unique.head(5)
top_overall_unique_countries[['Country', 'Composite Score']]
```

[37]:

	Country	Composite Score
133	India	4.851917
204	Turkey	3.809413
163	Malaysia	3.694605
0	Algeria	3.445574
20	Brazil	3.389254

The top 5 countries where my salary in USD would go the furthest using mean data scientist salary(USD), would be India, Turkey, Malaysia, Algeria, and Brazil.

[ ]:
```

```