In [11]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [12]:
```
dssalariesdata = '/Users/katarinadouglas-blake/Desktop/DSE5002/Project 02/ds
costoflivingdata = '/Users/katarinadouglas-blake/Desktop/DSE5002/Project 02/
levelssalarydata = '/Users/katarinadouglas-blake/Desktop/DSE5002/Project 02/


dssalaries=pd.read_csv(dssalariesdata)
costofliving=pd.read_csv(costoflivingdata)
levelssalary=pd.read_csv(levelssalarydata)
```

In [16]:
```
print(dssalaries.info())
print(dssalaries.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          607 non-null    int64 
 1   work_year           607 non-null    int64 
 2   experience_level    607 non-null    object
 3   employment_type     607 non-null    object
 4   job_title           607 non-null    object
 5   salary              607 non-null    int64 
 6   salary_currency     607 non-null    object
 7   salary_in_usd       607 non-null    int64 
 8   employee_residence  607 non-null    object
 9   remote_ratio        607 non-null    int64 
 10  company_location    607 non-null    object
 11  company_size        607 non-null    object
dtypes: int64(5), object(7)
memory usage: 57.0+ KB
None
       Unnamed: 0    work_year        salary  salary_in_usd  remote_ratio
count  607.000000   607.000000  6.070000e+02     607.000000     607.00000
mean   303.000000  2021.405272  3.240001e+05  112297.869852      70.92257
std    175.370085     0.692133  1.544357e+06   70957.259411      40.70913
min      0.000000  2020.000000  4.000000e+03    2859.000000       0.00000
25%    151.500000  2021.000000  7.000000e+04   62726.000000      50.00000
50%    303.000000  2022.000000  1.150000e+05  101570.000000     100.00000
75%    454.500000  2022.000000  1.650000e+05  150000.000000     100.00000
max    606.000000  2022.000000  3.040000e+07  600000.000000     100.00000
```

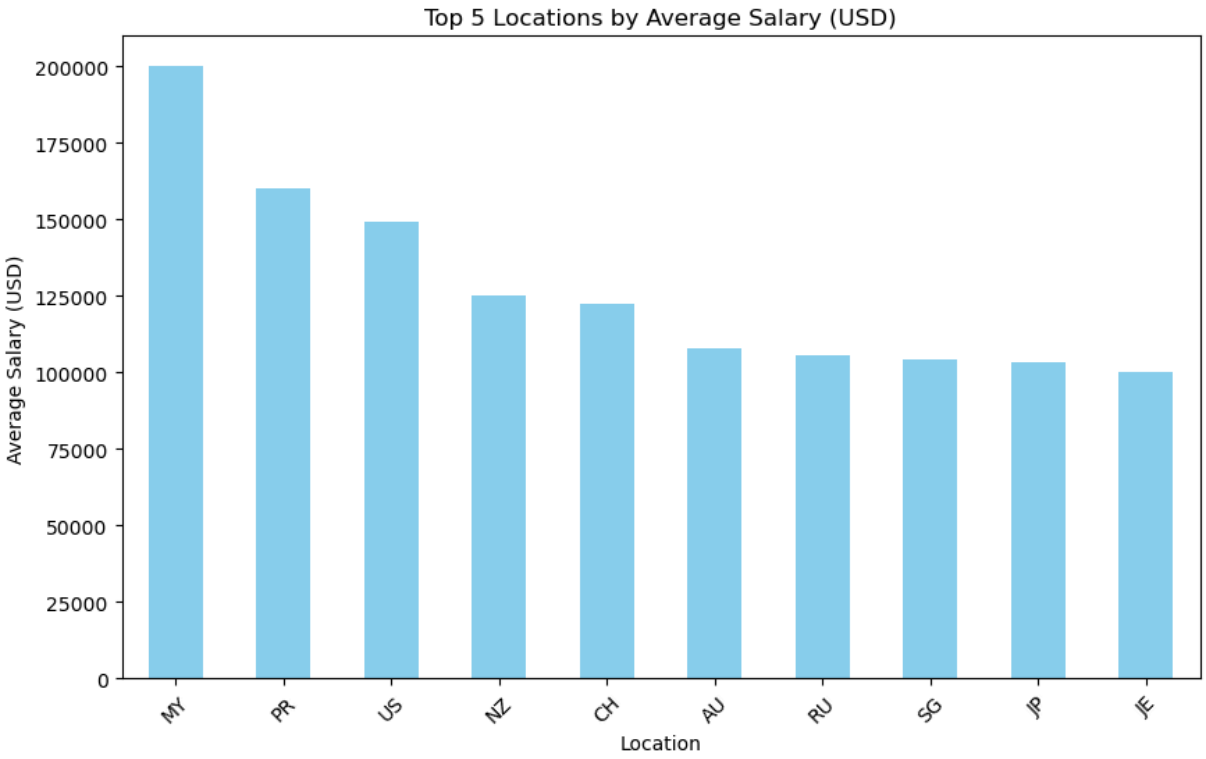
In [19]:
```
print(costofliving.info())
print(costofliving.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 578 entries, 0 to 577
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Rank                            0 non-null      float64
 1   City                            578 non-null    object 
 2   Cost of Living Index            578 non-null    float64
 3   Rent Index                      578 non-null    float64
 4   Cost of Living Plus Rent Index  578 non-null    float64
 5   Groceries Index                 578 non-null    float64
 6   Restaurant Price Index          578 non-null    float64
 7   Local Purchasing Power Index    578 non-null    float64
dtypes: float64(7), object(1)
memory usage: 36.3+ KB
None
       Rank  Cost of Living Index  Rent Index  Cost of Living Plus Rent Inde
x  \
count   0.0            578.000000  578.000000                      578.00000
0
mean    NaN             57.541349   26.650657                       43.06122
8
std     NaN             21.656441   17.841544                       18.90375
1
min     NaN             18.550000    2.370000                       10.97000
0
25%     NaN             38.015000   12.265000                       26.14500
0
50%     NaN             62.405000   23.280000                       44.99000
0
75%     NaN             73.030000   36.615000                       55.72000
0
max     NaN            149.020000  108.420000                      124.22000
0

       Groceries Index  Restaurant Price Index  Local Purchasing Power Index
count       578.000000              578.000000                    578.000000
mean         53.566782               54.354360                     71.504481
std          22.125102               25.863557                     34.206184
min          15.220000               11.390000                      1.620000
25%          34.025000               30.447500                     42.762500
50%          52.735000               59.135000                     70.935000
75%          68.942500               73.545000                     95.682500
max         157.890000              155.220000                    172.980000
```

In [21]:
```
data_scientists = dssalaries[dssalaries['job_title']=='Data Scientist']

# Groupby employee residence and find the mean salary in USD
avg_sal_by_empres = dssalaries.groupby('employee_residence')['salary_in_usd'

# Sort the results in descending order to see the top locations by average s
avg_sal_by_empres = avg_sal_by_empres.sort_values(ascending=False)
top_10_avg_sal=avg_sal_by_empres.nlargest(10)
# Display the results
print(avg_sal_by_empres.head(10))
```

```
employee_residence
MY    200000.000000
PR    160000.000000
US    149194.117470
NZ    125000.000000
CH    122346.000000
AU    108042.666667
RU    105750.000000
SG    104176.500000
JP    103537.714286
JE    100000.000000
Name: salary_in_usd, dtype: float64
```

In [23]:
```
# visual of the average salary in usd by employee residence

# Plot the data
plt.figure(figsize=(10, 6))
top_10_avg_sal.plot(kind='bar', color='skyblue')
plt.title('Top 5 Locations by Average Salary (USD)')
plt.xlabel('Location')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=45)
plt.show()
```



In [33]:
```
#filter so that the only roles are for data scientists.
data_scientists = dssalaries[dssalaries['job_title'] == 'Data Scientist']

#finding the average Local Puchasing Power Index (LPP)
average_lpp=costofliving['Local Purchasing Power Index'].mean()
print("The average local purchasing power is:", average_lpp.round(3))

#finding the 'best salary' based on the local purchasing power index (LPP)
print("local purchasing power index helps measure how far a salary will stre
```

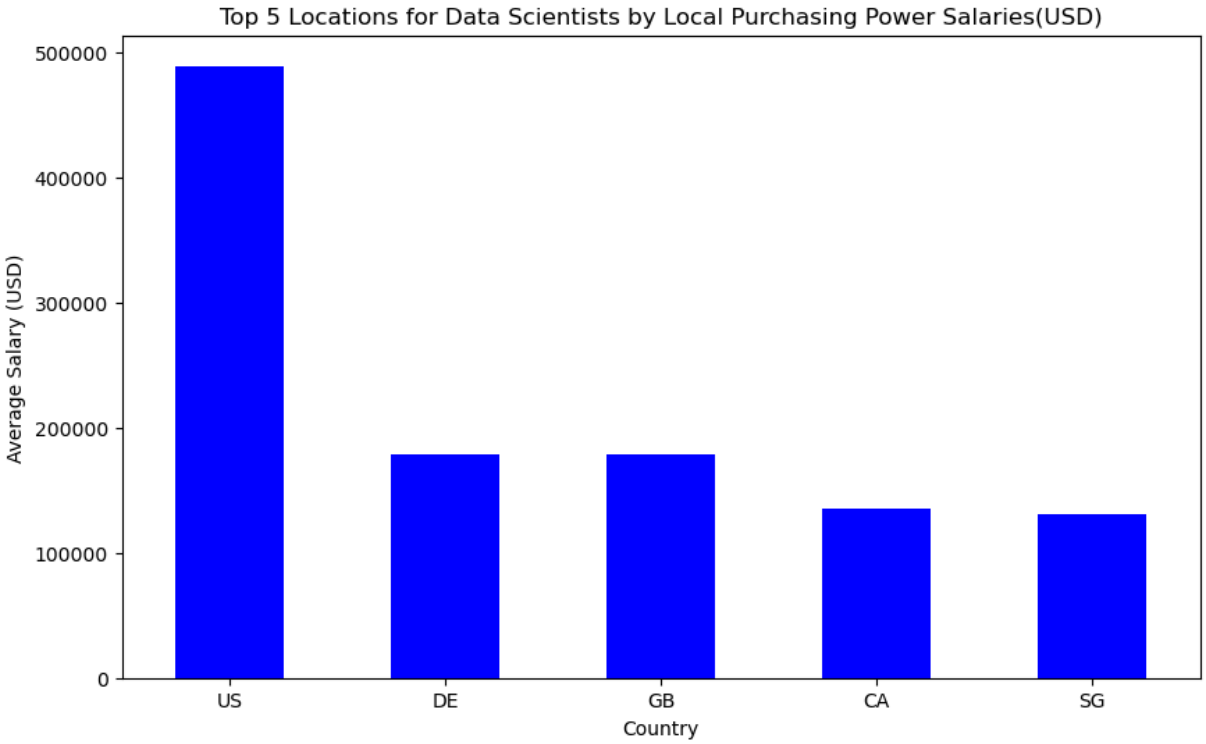
```
dssalaries['best_salary']=dssalaries['salary_in_usd']*costofliving['Local Pu
print(dssalaries[['best_salary','employee_residence']].round(2))
```

```
The average local purchasing power is: 71.504
local purchasing power index helps measure how far a salary will stretch in
a specific location.
     best_salary employee_residence
0       88687.21                 DE
1      471963.64                 JP
2      170062.19                 GB
3       40111.89                 HN
4      234881.12                 US
..           ...                ...
602          NaN                 US
603          NaN                 US
604          NaN                 US
605          NaN                 US
606          NaN                 IN

[607 rows x 2 columns]
```

In [35]:
```
# group by employee residence and find the maximum best salary for Data Scie
top_sal_by_empres = data_scientists.groupby('employee_residence')['best_sala
print(top_sal_by_empres.sort_values(ascending=False).head(10))
# Find the top 5 highest salaries based on the adjusted best salary
top_5_bestsalaries = top_sal_by_empres.nlargest(5)
print(top_5_bestsalaries.round(2))
```

```
employee_residence
US    489386.853147
DE    178696.706014
GB    178265.234685
CA    134969.267413
SG    130615.216224
AT    129428.935804
FR     90928.307133
DZ     80797.202797
ES     80035.583916
PH     77971.200000
Name: best_salary, dtype: float64
employee_residence
US    489386.85
DE    178696.71
GB    178265.23
CA    134969.27
SG    130615.22
Name: best_salary, dtype: float64
```

In [37]:
```
# visual of best salaries
plt.figure(figsize=(10, 6))
top_5_bestsalaries.plot(kind='bar', color='blue')
plt.title('Top 5 Locations for Data Scientists by Local Purchasing Power Sal
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```



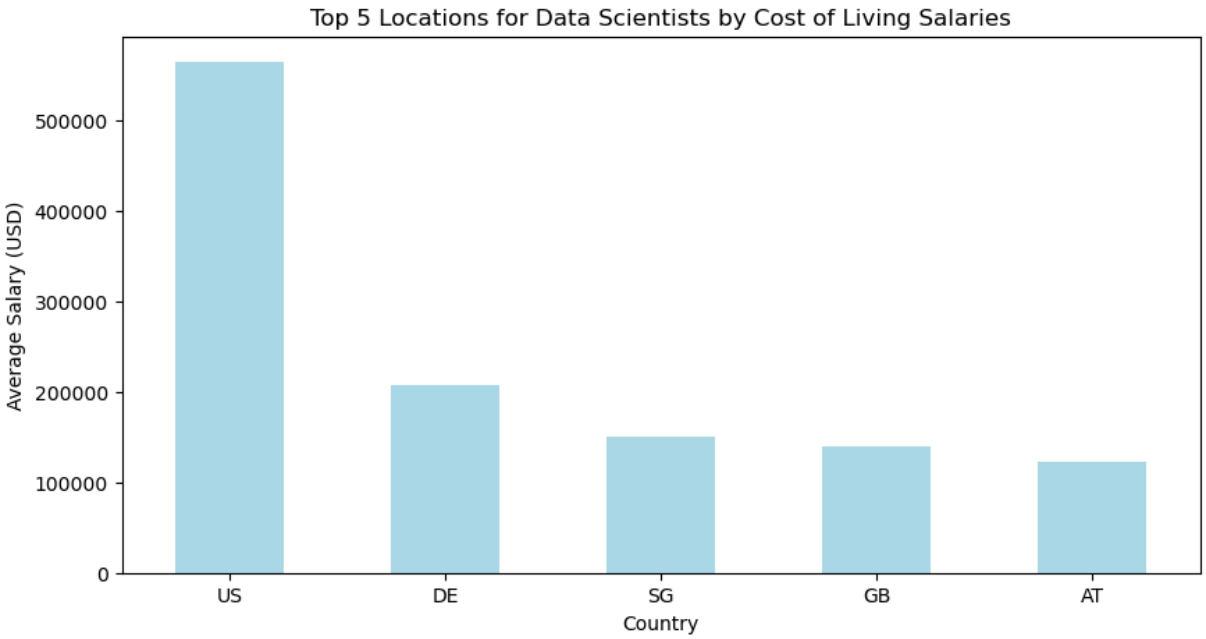
In [39]:
```
# average  Cost of Living Index (COL)
data_scientists=dssalaries[dssalaries['job_title']=='Data Scientist']

average_COL=costofliving['Cost of Living Index'].mean()
print("The average Cost of living index is:", average_COL.round(2))
```

```
The average Cost of living index is: 57.54
```

In [45]:
```
# Adjust salary based on the Cost of Living Index
dssalaries.loc[:, 'best_salary_COL']= data_scientists['salary_in_usd'] * (cc

# Group by employee residence and find the maximum salary based on the Cost
top_sal_by_COL = data_scientists.groupby('employee_residence')['best_salary_
print(top_sal_by_COL.sort_values(ascending=False).head())

# top 5 salaries by Cost of Living
top5_sal_by_COL=top_sal_by_COL.nlargest(5)
print(top5_sal_by_COL.round(2))
```

```
employee_residence
US    564440.00
DE    206767.47
SG    150014.34
GB    139127.66
AT    123169.95
Name: best_salary_COL, dtype: float64
employee_residence
US    564440.00
DE    206767.47
SG    150014.34
GB    139127.66
AT    123169.95
Name: best_salary_COL, dtype: float64
```

In [47]:
```
# visual for Cost of Living salaries for Data Scientists
plt.figure(figsize=(10,5))
top5_sal_by_COL.plot(kind='bar', color='lightblue')
plt.title('Top 5 Locations for Data Scientists by Cost of Living Salaries')
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```



In [49]:
```
# average Rent Index
data_scientists=dssalaries[dssalaries['job_title']=='Data Scientist']

average_rentindex=costofliving['Rent Index'].mean()
print("The average Rent index is:",average_rentindex.round(2))
```

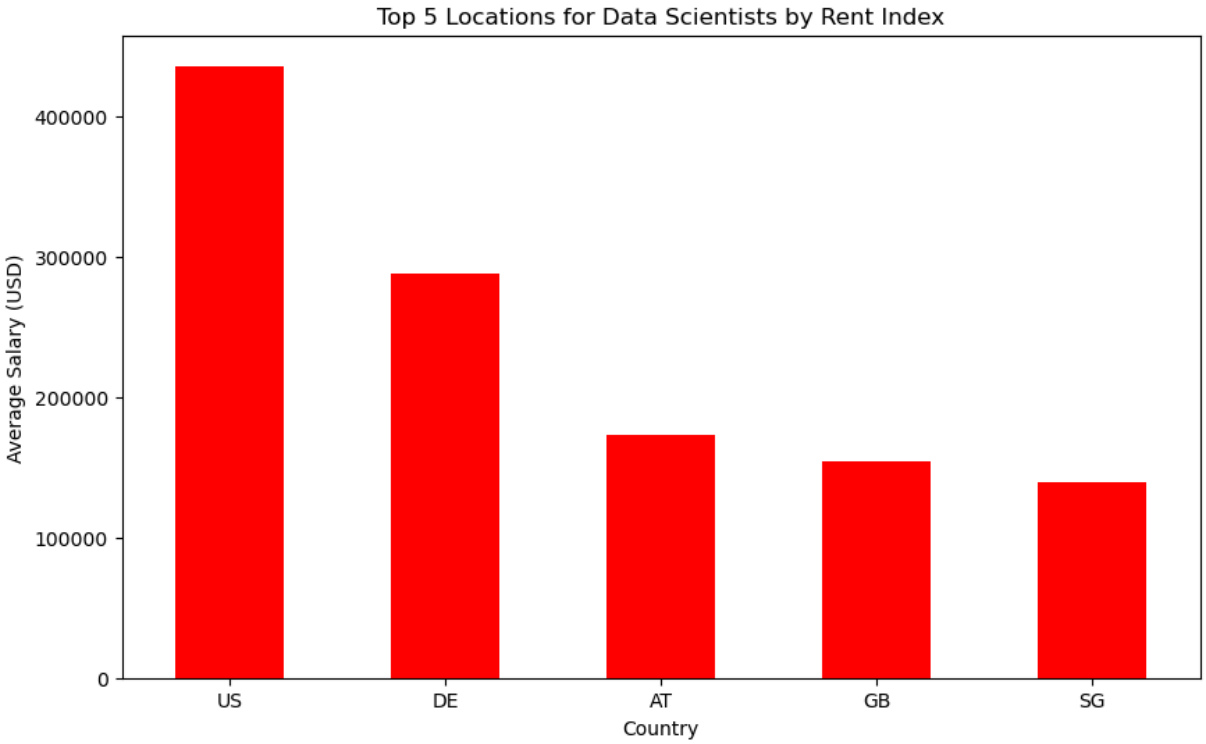
```
The average Rent index is: 26.65
```

In [55]:
```
# Adjust salary based on the Rent Index
dssalaries.loc[:, 'best_rent_salary']= data_scientists['salary_in_usd'] * (c

# group by employee residence and find max based on rent index
top_sal_by_rent=data_scientists.groupby('employee_residence')['best_rent_sal
```

```
print(top_sal_by_rent.sort_values(ascending=False).head())

#top 5 salaries by Rent Index
top5_sal_by_rent=top_sal_by_rent.nlargest(5)
print(top5_sal_by_rent.round(2))
```

```
employee_residence
US    435807.879925
DE    287878.097561
AT    173059.300188
GB    154730.085553
SG    139832.896811
Name: best_rent_salary, dtype: float64
employee_residence
US    435807.88
DE    287878.10
AT    173059.30
GB    154730.09
SG    139832.90
Name: best_rent_salary, dtype: float64
```

In [57]:
```
# Visual for Rent Index Salaries for Data Scientist
plt.figure(figsize=(10,6))
top5_sal_by_rent.plot(kind='bar', color='red')
plt.title('Top 5 Locations for Data Scientists by Rent Index')
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```



In [59]:
```
# average cost of living plus rent index
data_scientists=dssalaries[dssalaries['job_title']=='Data Scientist']
```

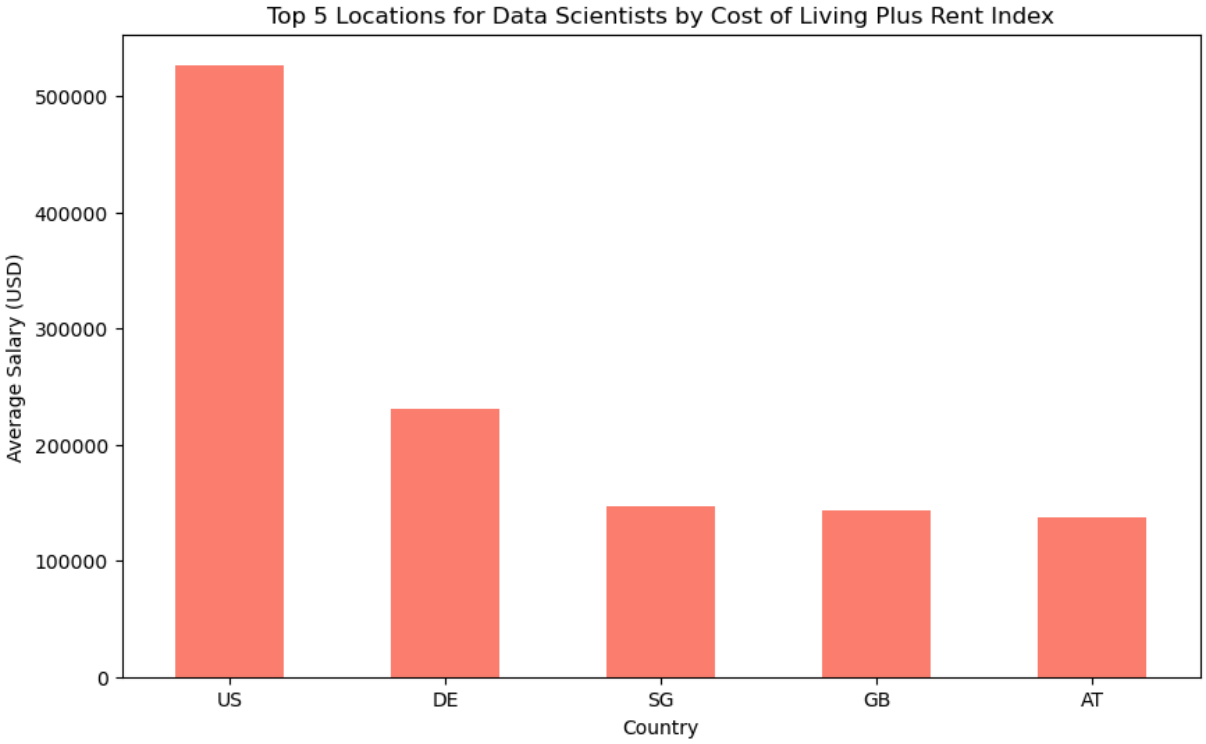
```
avg_COLrent_index=costofliving['Cost of Living Plus Rent Index'].mean()
print("The average cost of living plus rent index is:", avg_COLrent_index.rc
```

```
The average cost of living plus rent index is: 43.06
```

In [65]:
```
# adjust salary based on cost of living plus rent index
dssalaries.loc[:, 'best_COLrent_salary']=data_scientists['salary_in_usd'] *

# group by employee residence and find max based on cost of living plus rent
top_sal_by_COLrent=data_scientists.groupby('employee_residence')['best_COLre
print(top_sal_by_COLrent.sort_values(ascending=False).head())

# top 5 salaries by Cost of living plus rent index
top5_sal_by_COLrent=top_sal_by_COLrent.nlargest(5)
print(top5_sal_by_COLrent.round(2))
```

```
employee_residence
US    527103.576405
DE    230303.187645
SG    147095.652578
GB    143522.388295
AT    137872.540409
Name: best_COLrent_salary, dtype: float64
employee_residence
US    527103.58
DE    230303.19
SG    147095.65
GB    143522.39
AT    137872.54
Name: best_COLrent_salary, dtype: float64
```

In [67]:
```
# visual for Cost of living plus rent index
plt.figure(figsize=(10,6))
top5_sal_by_COLrent.plot(kind='bar',color='salmon')
plt.title('Top 5 Locations for Data Scientists by Cost of Living Plus Rent I
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```

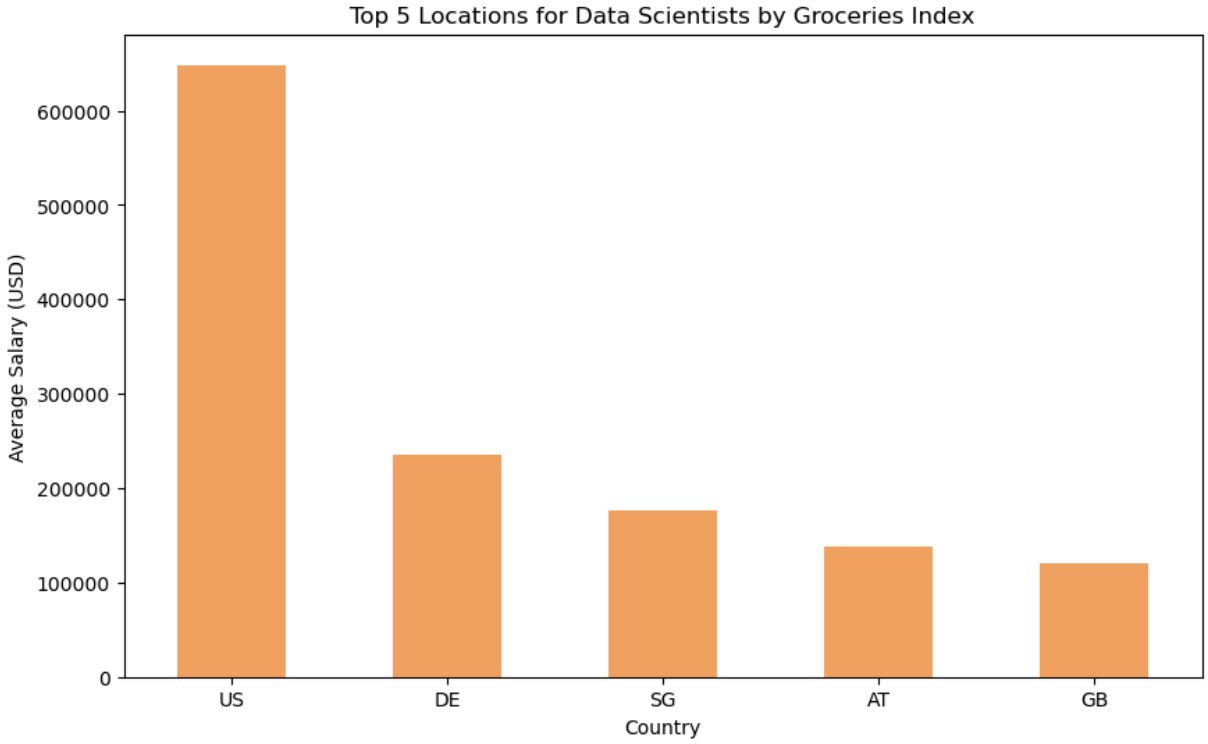


In [73]:
```
# average Grocery index
avg_grocery_index=costofliving['Groceries Index'].mean()
print("The average Grocery Index is:", avg_grocery_index)

#adjust salary based on grocery index
dssalaries.loc[:, 'best_grocery_sal']=data_scientists['salary_in_usd'] * (cc

# group by employee residence and find max based on grocery index
top_sal_by_groceries=data_scientists.groupby('employee_residence')['best_grc
print(top_sal_by_groceries.sort_values(ascending=False).head())

#top 5
top5_sal_by_groceries=top_sal_by_groceries.nlargest(5)
print(top5_sal_by_groceries.round(2))
```

```
The average Grocery Index is: 53.56678200692042
employee_residence
US    648340.489080
DE    235296.478813
SG    176043.744447
AT    137681.521001
GB    121126.134030
Name: best_grocery_sal, dtype: float64
employee_residence
US    648340.49
DE    235296.48
SG    176043.74
AT    137681.52
GB    121126.13
Name: best_grocery_sal, dtype: float64
```

In [75]:
```
# visual for grocery index
plt.figure(figsize=(10,6))
```

```
top5_sal_by_groceries.plot(kind='bar',color='sandybrown')
plt.title('Top 5 Locations for Data Scientists by Groceries Index')
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```



In [81]:
```
# average restaurant price index
avg_restprice_index=costofliving['Restaurant Price Index'].mean()
print("The average Restaurant Price Index is:", avg_restprice_index.round(2)

# adjust salary by Restaurant Price index
dssalaries.loc[:, 'best_restprice_sal']=data_scientists['salary_in_usd']*(cc

# group by employee residence and find max based on restaurant price index
top_sal_by_restprice=data_scientists.groupby('employee_residence')['best_res
print(top_sal_by_restprice.sort_values(ascending=False).head())

#top 5
top5_sal_by_restprice=top_sal_by_restprice.nlargest(5)
print(top5_sal_by_restprice.round(2))
```

```
The average Restaurant Price Index is: 54.35
employee_residence
US    572175.896964
DE    227997.760074
GB    159421.505796
SG    142432.680405
CA    131469.122539
Name: best_restprice_sal, dtype: float64
employee_residence
US    572175.90
DE    227997.76
GB    159421.51
SG    142432.68
CA    131469.12
Name: best_restprice_sal, dtype: float64
```

In [83]:
```
# visual of countries with top 5 restaurant index
plt.figure(figsize=(10,6))
top5_sal_by_restprice.plot(kind='bar', color='firebrick')
plt.title('Top 5 Locations for Data Scientists by Restaurant Price Index')
plt.xlabel('Country')
plt.ylabel('Average Salary (USD)')
plt.xticks(rotation=0)
plt.show()
```

In [85]:
```
avg_yrs_at_comp=levelssalary['yearsatcompany'].mean()
print(avg_yrs_at_comp.round(2))
```

```
2.7
```

In [87]:
```
top_salaries_df=pd.DataFrame({
    'Local Purchasing Power':top_5_bestsalaries,
    'Cost of Living':top5_sal_by_COL,
```

```
    'Rent':top5_sal_by_rent,
    'COL plus Rent':top5_sal_by_COLrent,
    'Groceries':top5_sal_by_groceries,
    'Restaurant Price':top5_sal_by_restprice}).reset_index()

top_salaries_df.rename(columns={'index':'Employee Residence'}, inplace=True)

plt.figure(figsize=(10,6))
for index in ['Local Purchasing Power','Cost of Living','Rent','COL plus Rer

plt.title('Top 5 Places for Data Scientist Salaries Adjusted by Various Cost
plt.xlabel('Country')
plt.ylabel('Average Adjusted Salary')
plt.xticks(rotation=0)
plt.legend(title="Indexes")
plt.grid(alpha=0.3)
plt.show()
```

In [91]:
```
top_5_countries={
    "LPP Index":top_5_bestsalaries,
    "Cost of Living Index":top5_sal_by_COL,
    "Rent Index":top5_sal_by_rent,
    "COL PLus Rent":top5_sal_by_COLrent,
    "Groceries Index":top5_sal_by_groceries,
    "Restaurant Price Index":top5_sal_by_restprice}
top_5_countries_df = pd.DataFrame(top_5_countries)
print(top_5_countries_df.round(2))
```

employee_residence	LPP Index	Cost of Living Index	Rent Index
AT	NaN	123169.95	173059.30
CA	134969.27	NaN	NaN
DE	178696.71	206767.47	287878.10
GB	178265.23	139127.66	154730.09
SG	130615.22	150014.34	139832.90
US	489386.85	564440.00	435807.88

employee_residence	COL PLus Rent	Groceries Index	Restaurant Price Index
AT	137872.54	137681.52	NaN
CA	NaN	NaN	131469.12
DE	230303.19	235296.48	227997.76
GB	143522.39	121126.13	159421.51
SG	147095.65	176043.74	142432.68
US	527103.58	648340.49	572175.90

In [ ]:
```
## Story Summary
# This analysis illustrates countries where data scientist salaries will
# provide financial stability based on various living costs.
# The US came consistently came first across all indices proving that
# it is the top place where data scientists would
# get the most value out of their salaries(USD). Germany, Great Britain,
# and Singapore are also countries where
# USD salaries are most powerful. Austria and Canada also offer
# favorable living conditions with their data science
# salaries but not as consistent across all iindices.
```