After cleaning the 3 datasets in different tab (cost_of_living_df, ds_salaries_df, levels_salary_DS) I load those cleaned version of the datasets into this segment, including country codes. I will now start doing the analysing the datas but first I will merge 3 of these datasets

In [86]:
```
import pandas as pd
```

In [88]:
```
cost_of_living_df = pd.read_csv("/Users/ertuboston/Documents/Data_Science_Merrimack/DSE5002/P

ds_salaries_df = pd.read_csv("/Users/ertuboston/Documents/Data_Science_Merrimack/DSE5002/PROJ

levels_salary_DS = pd.read_csv("/Users/ertuboston/Documents/Data_Science_Merrimack/DSE5002/PR

country_codes_df = pd.read_csv("/Users/ertuboston/Documents/Data_Science_Merrimack/DSE5002/PR
```

In [90]:
```
cost_of_living_df.head(5)
```

Out[90]:

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index
0	Hamilton	Bermuda	149.02	96.10	124.22	157.89	155.22	79.43
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79
2	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53
3	Zug	Switzerland	128.13	72.12	101.87	132.61	130.93	143.40
4	Lugano	Switzerland	123.99	44.99	86.96	129.17	119.80	111.96

In [92]:
```
ds_salaries_df.head(5)
```

Out[92]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	company
0	2020	MI	FT	Data Scientist	79833	DE	
1	2020	SE	FT	Lead Data Scientist	190000	US	
2	2020	MI	FT	Data Scientist	35735	HU	
3	2020	EN	FT	Data Scientist	51321	FR	
4	2020	MI	FT	Data Scientist	40481	IN	

In [94]:
```
levels_salary_DS.head(5)
```

Out[94]:

	company	title	totalyearlycompensation	basesalary	stockgrantvalue	bonus	cityid	dmaid	
0	Google	Data Scientist	170000	170000.0	0.0	0.0	7419	807.0	Fra
1	Facebook	Data Scientist	205000	150000.0	40000.0	15000.0	7300	807.0	
2	Microsoft	Data Scientist	220000	150000.0	60000.0	10000.0	11470	819.0	Be
3	PayPal	Data Scientist	216000	160000.0	40000.0	16000.0	7422	807.0	Sa
4	Amazon	Data Scientist	185000	185000.0	5000.0	0.0	8821	506.0	Cam

In [96]:
```
country_codes_df.head(5)
```

Out[96]:

	Country	Alpha-2 code	Alpha-3 code	Numeric
0	Afghanistan	AF	AFG	4
1	Albania	AL	ALB	8
2	Algeria	DZ	DZA	12
3	American Samoa	AS	ASM	16
4	Andorra	AD	AND	20

In [98]:
```
### I will merge two datasets first, country_codes_df and cost_of_living_df.
### I assume I would need this merging to be able to merge one of the other
### datasets to calculate the spending.
```

```
merged_data = pd.merge( cost_of_living_df, country_codes_df, on = 'Country', how = 'left')
merged_data.head(5)
```

Out[98]:

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	Alpha-3 code	Numer
0	Hamilton	Bermuda	149.02	96.10	124.22	157.89	155.22	79.43	BM	BMU	60
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	CHE	756
2	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	CHE	756
3	Zug	Switzerland	128.13	72.12	101.87	132.61	130.93	143.40	CH	CHE	756
4	Lugano	Switzerland	123.99	44.99	86.96	129.17	119.80	111.96	CH	CHE	756

In [100…

```
### Now we can merge the merged data with levels_salary_DS to keep salaries and cost indexes

merged_salary_and_cost = pd.merge(merged_data, levels_salary_DS, left_on='Alpha-2 code',
                                  right_on='country', how = 'left')
merged_salary_and_cost.head(5)
```

Out [100…

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	Alpha-3 code	…	
0	Hamilton	Bermuda	149.02	96.10	124.22	157.89	155.22	79.43	BM	BMU	…	
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	CHE	…	Sc
2	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	CHE	…	Sc
3	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	CHE	…	Sc
4	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	CHE	…	Sc

5 rows × 22 columns

In [102…

```
print(merged_salary_and_cost.columns)
```

```
Index(['City', 'Country', 'Cost of Living Index', 'Rent Index',
       'Cost of Living Plus Rent Index', 'Groceries Index',
       'Restaurant Price Index', 'Local Purchasing Power Index',
       'Alpha-2 code', 'Alpha-3 code', 'Numeric', 'company', 'title',
       'totalyearlycompensation', 'basesalary', 'stockgrantvalue', 'bonus',
       'cityid', 'dmaid', 'city', 'state', 'country'],
      dtype='object')
```

In [104…

```
### There are some columns that we won't need it for our research,
### I will drop those columns first to keep the data more readable.
```

```
merged_salary_and_cost.drop(columns = ['Alpha-3 code', 'Numeric', 'cityid', 'dmaid','city',
                                       'state','country'], inplace=True)
merged_salary_and_cost
```

Out [104...

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	company
0	Hamilton	Bermuda	149.02	96.10	124.22	157.89	155.22	79.43	BM	NaN
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Google
2	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Roche
3	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Google
4	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Roche
...	...	...	...	...	...	...	...	...	...	...
5574	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Accenture
5575	Karachi	Pakistan	20.75	4.84	13.29	18.48	15.21	29.16	PK	NaN
5576	Rawalpindi	Pakistan	20.52	4.78	13.14	18.51	16.18	22.91	PK	NaN
5577	Multan	Pakistan	18.68	2.94	11.30	18.37	11.80	25.09	PK	NaN
5578	Peshawar	Pakistan	18.55	2.37	10.97	16.62	14.39	26.00	PK	NaN

5579 rows × 15 columns

In [106…
```
### Lets see the missing values
```

In [108…
```
merged_salary_and_cost.isna().sum()
```

Out[108…
```
City                                0
Country                             0
Cost of Living Index                0
Rent Index                          0
Cost of Living Plus Rent Index      0
Groceries Index                     0
Restaurant Price Index              0
Local Purchasing Power Index        0
Alpha-2 code                     2090
company                           231
title                             231
totalyearlycompensation           231
basesalary                        231
stockgrantvalue                   231
bonus                             231
dtype: int64
```

In [110…
```
### I would like to see it as percentage.

missing_values_perc = merged_salary_and_cost.isna().sum()/ len(merged_salary_and_cost) * 100
missing_values_perc
```

Out[110…
```
City                                0.000000
Country                             0.000000
Cost of Living Index                0.000000
Rent Index                          0.000000
Cost of Living Plus Rent Index      0.000000
Groceries Index                     0.000000
Restaurant Price Index              0.000000
Local Purchasing Power Index        0.000000
Alpha-2 code                       37.461911
company                             4.140527
title                               4.140527
totalyearlycompensation             4.140527
basesalary                          4.140527
stockgrantvalue                     4.140527
bonus                               4.140527
dtype: float64
```

In [112…
```
merged_salary_and_cost['Alpha-2 code'].unique()
```

Out[112…
```
array(['BM', 'CH', 'LB', 'NO', nan, 'IS', 'JE', 'IL', 'DK', 'JP', 'FR',
       'SG', 'AU', 'LU', 'FI', 'HK', 'NZ', 'IE', 'SE', 'DE', 'AT', 'CA',
       'BE', 'IT', 'MT', 'PR', 'MO', 'CY', 'ES', 'QA', 'GR', 'MV', 'SI',
       'CU', 'EE', 'PA', 'BH', 'CN', 'SA', 'JO', 'UY', 'PT', 'HR', 'JM',
       'LV', 'OM', 'SN', 'ET', 'TH', 'KH', 'SK', 'SR', 'KW', 'CR', 'LT',
       'HU', 'ZW', 'CL', 'SV', 'ZA', 'GT', 'PL', 'ID', 'BW', 'BG', 'EC',
       'RO', 'RS', 'MY', 'MA', 'ME', 'FJ', 'MX', 'GH', 'AL', 'IQ', 'BR',
       'NG', 'UG', 'KE', 'AR', 'BD', 'MN', 'PE', 'UA', 'IN', 'AM', 'LK',
       'ZM', 'BY', 'EG', 'RW', 'AZ', 'TR', 'GE', 'PY', 'KZ', 'TN', 'NP',
       'DZ', 'UZ', 'CO', 'KG', 'PK', 'AF'], dtype=object)
```

In [114…
```
### As we see that there nan values in Alpha-2 code column.
### I will drop those columns since it is only the %38 of the entire data.
### We will be working on the rest of the data which will be enough for calculations
```

```
merged_salary_and_cost.dropna(inplace=True)
```

In [116…
```
merged_salary_and_cost.isna().sum()
```

Out[116…
```
City                              0
Country                           0
Cost of Living Index              0
Rent Index                        0
Cost of Living Plus Rent Index    0
Groceries Index                   0
Restaurant Price Index            0
Local Purchasing Power Index      0
Alpha-2 code                      0
company                           0
title                             0
totalyearlycompensation           0
basesalary                        0
stockgrantvalue                   0
bonus                             0
dtype: int64
```

In [118…
```
merged_salary_and_cost

### Now we have no NA values.
```

Out [118...

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	company
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Google
2	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Roche
3	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Google
4	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Roche
5	Zug	Switzerland	128.13	72.12	101.87	132.61	130.93	143.40	CH	Google
...	...	...	...	...	...	...	...	...	...	...
5570	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Dream11
5571	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Amazon
5572	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	IQVIA
5573	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Fidelity Investments

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	company
5574	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Accenture

3258 rows × 15 columns

==========

Now we can start coding for the question that was asked.

Which city or country is the best place to live based on the cost of living index and etc...

In [121…

```
### Let's see what we would spend on rent in a month.
### My datasets name is merged_salary_and_cost

totalyearlycompensation = merged_salary_and_cost['totalyearlycompensation']

### Calculate monthly salary
monthly_salary = totalyearlycompensation / 12

### Typically how much percentage of your monthly salary would a person spend
### for the cost of living, rent, restaurant, groceries, and purchasing power

rent_percentage = 0.30  # Typically 30% of income
cost_of_living_percentage = 0.20  # Typically 20% of income
groceries_percentage = 0.15  # Typically 15% of income
restaurant_percentage = 0.10  # Typically 10% of income
```

```
purchasing_power_percentage = 0.10 # Typically 10% of income

### Calculate monthly spending based on indices
monthly_spending_for_rent = (merged_salary_and_cost['Rent Index'] / 100) * (rent_percentage *
merged_salary_and_cost['monthly_spending_for_rent'] = monthly_spending_for_rent.round(2)
```

In [123…

```
merged_salary_and_cost
```

Out [123...

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	company
1	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Google
2	Zurich	Switzerland	131.24	69.26	102.19	136.14	132.52	129.79	CH	Roche
3	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Google
4	Basel	Switzerland	130.93	49.38	92.70	137.07	130.95	111.53	CH	Roche
5	Zug	Switzerland	128.13	72.12	101.87	132.61	130.93	143.40	CH	Google
...	...	...	...	...	...	...	...	...	...	...
5570	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Dream11
5571	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Amazon
5572	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	IQVIA
5573	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Fidelity Investments

	City	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index	Alpha-2 code	company
5574	Kanpur	India	20.79	3.60	12.73	22.19	13.31	38.83	IN	Accenture

3258 rows × 16 columns

In [125…
```
print(merged_salary_and_cost.columns)
```

```
Index(['City', 'Country', 'Cost of Living Index', 'Rent Index',
       'Cost of Living Plus Rent Index', 'Groceries Index',
       'Restaurant Price Index', 'Local Purchasing Power Index',
       'Alpha-2 code', 'company', 'title', 'totalyearlycompensation',
       'basesalary', 'stockgrantvalue', 'bonus', 'monthly_spending_for_rent'],
      dtype='object')
```

In [127…
```
### The dataset "merged_salary_and_cost" is getting very crowded and difficult to follow.
### I will create a new data frame and add all these findings
### with City and Country and ALpha-2 code into that dataframe
### We will call it 'yearly_spendings_df'

columns_to_include = ['City','Country','Alpha-2 code', 'title',
                      'company','totalyearlycompensation']

yearly_spendings_df = merged_salary_and_cost[columns_to_include].copy()

yearly_spendings_df
```

Out [127…

	City	Country	Alpha-2 code	title	company	totalyearlycompensation
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0
...	...	...	...	...	...	...
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0

3258 rows × 6 columns

In [131…

```
### Now that we have tested our code on the rent index, we can apply the same formula to othe
monthly_spending_for_rent = (merged_salary_and_cost['Rent Index'] / 100) * (rent_percentage *
monthly_cost_of_living = (merged_salary_and_cost['Cost of Living Index'] / 100) * (cost_of_li
monthly_groceries = (merged_salary_and_cost['Groceries Index'] / 100) * (groceries_percentage
monthly_restaurant_price = (merged_salary_and_cost['Restaurant Price Index'] / 100) * (restau
monthly_local_purchasing_power = (merged_salary_and_cost['Local Purchasing Power Index'] / 10

yearly_spending_for_rent = monthly_spending_for_rent * 12
yearly_cost_of_living = monthly_cost_of_living * 12
```

```
yearly_groceries = monthly_groceries * 12
yearly_local_purchasing_power = monthly_local_purchasing_power * 12
yearly_restaurant_price = monthly_restaurant_price * 12
```

In [133…

```
### Now let's add all these data into new data frame 'yearly_spendings_df'

yearly_spendings_df['yearly_spending_for_rent'] = yearly_spending_for_rent.round(2)
yearly_spendings_df['yearly_cost_of_living'] = yearly_cost_of_living.round(2)
yearly_spendings_df['yearly_groceries'] = yearly_groceries.round(2)
yearly_spendings_df['yearly_restaurant_price'] = yearly_restaurant_price.round(2)
yearly_spendings_df['yearly_local_purchasing_power'] = yearly_local_purchasing_power.round(2)
```

In [135…

```
yearly_spendings_df.head()
```

Out[135…

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	yearly_spending_for_rent	yea
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	71684.10	
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	19531.32	
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	51108.30	
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	13925.16	
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0	74644.20	

In [137…

```
### Now we can calculate the total spending based on the calculations.
```

```
yearly_total_spending = sum([yearly_cost_of_living,yearly_groceries,yearly_local_purchasing_p
```

In [139…
```
yearly_total_spending
```

Out[139…
```
1       323189.100
2        88057.320
3       296039.325
4        80659.990
5       326323.425
           ...
5570     38585.400
5571     18328.065
5572      2893.905
5573      3582.930
5574      2756.100
Length: 3258, dtype: float64
```

In [141…
```
### Let's add these to the dataset

yearly_spendings_df.insert(6,'yearly_total_spending', yearly_total_spending.round(2))
yearly_spendings_df
```

Out [141...

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	yearly_total_spending
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	323189.10
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	88057.32
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	296039.32
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	80659.99
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0	326323.43
...	...	...	...	...	...	...	...
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0	38585.40
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0	18328.07
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0	2893.91
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0	3582.93
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0	2756.10

3258 rows × 12 columns

In [143…
```
### I just want to check if we have any duplicates after all calculations.

yearly_spendings_df.duplicated().sum()
```

Out[143… 93

In [145…
```
### As we see we have 93 duplicates, so now let's remove the duplicates.

yearly_spendings_df.drop_duplicates(inplace=True)

yearly_spendings_df    ###(should be 3258 - 93 = 3165 rows)
```

Out [145...

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	yearly_total_spending
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	323189.10
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	88057.32
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	296039.32
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	80659.99
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0	326323.43
...	...	...	...	...	...	...	...
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0	38585.40
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0	18328.07
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0	2893.91
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0	3582.93
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0	2756.10

3165 rows × 12 columns

In [147…

```
### Now let's calculate the savings and add that into the data frame in index 7.

yearly_saving = totalyearlycompensation - yearly_total_spending

yearly_spendings_df.insert(7,'yearly_saving', yearly_saving.round(2))
```

In [149…

```
yearly_spendings_df
```

Out [149...

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	yearly_total_spending
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	323189.10
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	88057.32
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	296039.32
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	80659.99
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0	326323.43
...	...	...	...	...	...	...	...
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0	38585.40
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0	18328.07
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0	2893.91
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0	3582.93
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0	2756.10

3165 rows × 13 columns

In [151…
```
#### Since we are looking into the top 5 cities for each rent, groceries, cost of living,
### and restaurant indexes based on our salary
### I will create variables showing how many percent of our salary goes to rent,
### restaurants, groceries, and cost of living.
### Then we can put them to find the top 5 cities.

salary_to_cost_of_living_perc = ((yearly_cost_of_living / totalyearlycompensation) * 100).rou
salary_to_rent_perc = ((yearly_spending_for_rent / totalyearlycompensation) * 100).round(2)
salary_to_groceries_perc = ((yearly_groceries / totalyearlycompensation) * 100).round(2)
salary_to_restaurant_perc = ((yearly_restaurant_price / totalyearlycompensation) * 100).round
salary_to_purchase_power_perc = ((yearly_local_purchasing_power / totalyearlycompensation) *
```

In [153…
```
print(yearly_spendings_df.columns)
```

```
Index(['City', 'Country', 'Alpha-2 code', 'title', 'company',
       'totalyearlycompensation', 'yearly_total_spending', 'yearly_saving',
       'yearly_spending_for_rent', 'yearly_cost_of_living', 'yearly_groceries',
       'yearly_restaurant_price', 'yearly_local_purchasing_power'],
      dtype='object')
```

In [155…
```
### we can create another data frame to keep everything clean

columns_to_include_to_perc_df = ['City','Country','Alpha-2 code','title',
                                 'company','totalyearlycompensation']

salary_to_spendings_perc_df = yearly_spendings_df[columns_to_include_to_perc_df].copy()

salary_to_spendings_perc_df
```

Out [155…

	City	Country	Alpha-2 code	title	company	totalyearlycompensation
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0
...	...	...	...	...	...	...
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0

3165 rows × 6 columns

In [157…

```
### Now we can add the percentages for each indexes into new data frame

salary_to_spendings_perc_df.insert(6,'salary_to_cost_perc', salary_to_cost_of_living_perc)
salary_to_spendings_perc_df.insert(7,'salary_to_rent_perc', salary_to_rent_perc)
salary_to_spendings_perc_df.insert(8,'salary_to_groceries_perc', salary_to_groceries_perc)
salary_to_spendings_perc_df.insert(9,'salary_to_restaurant_perc', salary_to_restaurant_perc)
salary_to_spendings_perc_df.insert(9,'salary_to_purchase_power_perc', salary_to_purchase_powe
```

In [159…

```
salary_to_spendings_perc_df
```

Out [159...

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	salary_to_cost_perc	sa
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	26.25	
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	26.25	
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	26.19	
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	26.19	
5	Zug	Switzerland	CH	Data Scientist	Google	345000.0	25.63	
...	...	...	...	...	...	...	...	
5570	Kanpur	India	IN	Data Scientist	Dream11	280000.0	4.16	
5571	Kanpur	India	IN	Data Scientist	Amazon	133000.0	4.16	
5572	Kanpur	India	IN	Data Scientist	IQVIA	21000.0	4.16	
5573	Kanpur	India	IN	Data Scientist	Fidelity Investments	26000.0	4.16	
5574	Kanpur	India	IN	Data Scientist	Accenture	20000.0	4.16	

3165 rows × 11 columns

In [70]:
```
#salary_to_spendings_perc_df.to_csv('salary_to_spendings_perc_df.csv', index = False)
```

In [161…
```
salary_to_spendings_perc_df.duplicated().sum()
```

Out[161… 0

In [163…
```
### Sort the DataFrame based on each percentage column in ascending order
### if we sort the dataframe in ascending order(default)
### we will have the lowest percentage is on the top.
### We need the lowest percentages because lowest percentages mean
### that we spend that percent of our salary for the rent or grocery or restaurant etc...

sorted_df = salary_to_spendings_perc_df.sort_values(by=[
    'salary_to_cost_perc',
    'salary_to_rent_perc',
    'salary_to_groceries_perc',
    'salary_to_purchase_power_perc',
    'salary_to_restaurant_perc'
])
```

In [165…
```
sorted_df
```

Out [165...

	City	Country	Alpha-2 code	title	company	totalyearlycompensation	salary_to_cost_perc	sala
5521	Kanpur	India	IN	Data Scientist	Amazon	40000.0	4.16	
5522	Kanpur	India	IN	Data Scientist	Capgemini	10000.0	4.16	
5523	Kanpur	India	IN	Data Scientist	Verizon	32000.0	4.16	
5524	Kanpur	India	IN	Data Scientist	Societe Generale	26000.0	4.16	
5525	Kanpur	India	IN	Data Scientist	Fractal Analytics	85000.0	4.16	
...	...	...	...	...	...	...	...	
6	Zug	Switzerland	CH	Data Scientist	Roche	94000.0	25.63	
3	Basel	Switzerland	CH	Data Scientist	Google	345000.0	26.19	
4	Basel	Switzerland	CH	Data Scientist	Roche	94000.0	26.19	
1	Zurich	Switzerland	CH	Data Scientist	Google	345000.0	26.25	
2	Zurich	Switzerland	CH	Data Scientist	Roche	94000.0	26.25	

3165 rows × 11 columns

In [167…
```
### group by country and sort it.

cost_perc_sorted = salary_to_spendings_perc_df.groupby('Country')['salary_to_cost_perc'].mean
rent_perc_sorted = salary_to_spendings_perc_df.groupby('Country')['salary_to_rent_perc'].mean
groceries_perc_sorted = salary_to_spendings_perc_df.groupby('Country')['salary_to_groceries_p
purchase_power_perc_sorted = salary_to_spendings_perc_df.groupby('Country')['salary_to_purcha
restaurent_perc_sorted = salary_to_spendings_perc_df.groupby('Country')['salary_to_restaurant
```

In [169…
```
### I added a column name to sorted values.

cost_perc_sorted = cost_perc_sorted.reset_index(name='cost_perc')
rent_perc_sorted = rent_perc_sorted.reset_index(name='rent_perc')
groceries_perc_sorted = groceries_perc_sorted.reset_index(name='groceries_perc')
purchase_power_perc_sorted = purchase_power_perc_sorted.reset_index(name='purchase_power_perc
restaurent_perc_sorted = restaurent_perc_sorted.reset_index(name='restaurant_perc')
```

In [171…
```
### since I need only the top, I will use the .head(5)
cost_perc_sorted.head(5)
```

Out[171…

	Country	cost_perc
0	India	5.08
1	Ukraine	6.22
2	Poland	8.14
3	China	9.42
4	Germany	13.46

In [173…
```
rent_perc_sorted.head(5)
```

Out[173…

	Country	rent_perc
0	India	1.73
1	Ukraine	3.68
2	Poland	4.84
3	China	7.47
4	Germany	8.68

In [175…

```
groceries_perc_sorted.head(5)
```

Out[175…

	Country	groceries_perc
0	Ukraine	3.98
1	India	4.10
2	Poland	5.01
3	China	7.66
4	Germany	7.94

In [177…

```
purchase_power_perc_sorted.head(5)
```

Out[177…

	Country	purchase_power_perc
0	Ukraine	3.72
1	India	4.95
2	Poland	5.99
3	China	6.22
4	Japan	7.59

In [179…

```
restaurent_perc_sorted.head(5)
```

Out[179…

	Country	restaurant_perc
0	India	1.86
1	Ukraine	2.67
2	China	3.39
3	Poland	3.51
4	Japan	4.57

==========

After we created our tables for the top 5 countries for each index,

we can put these information to our graph by using matplotlib.pyplot

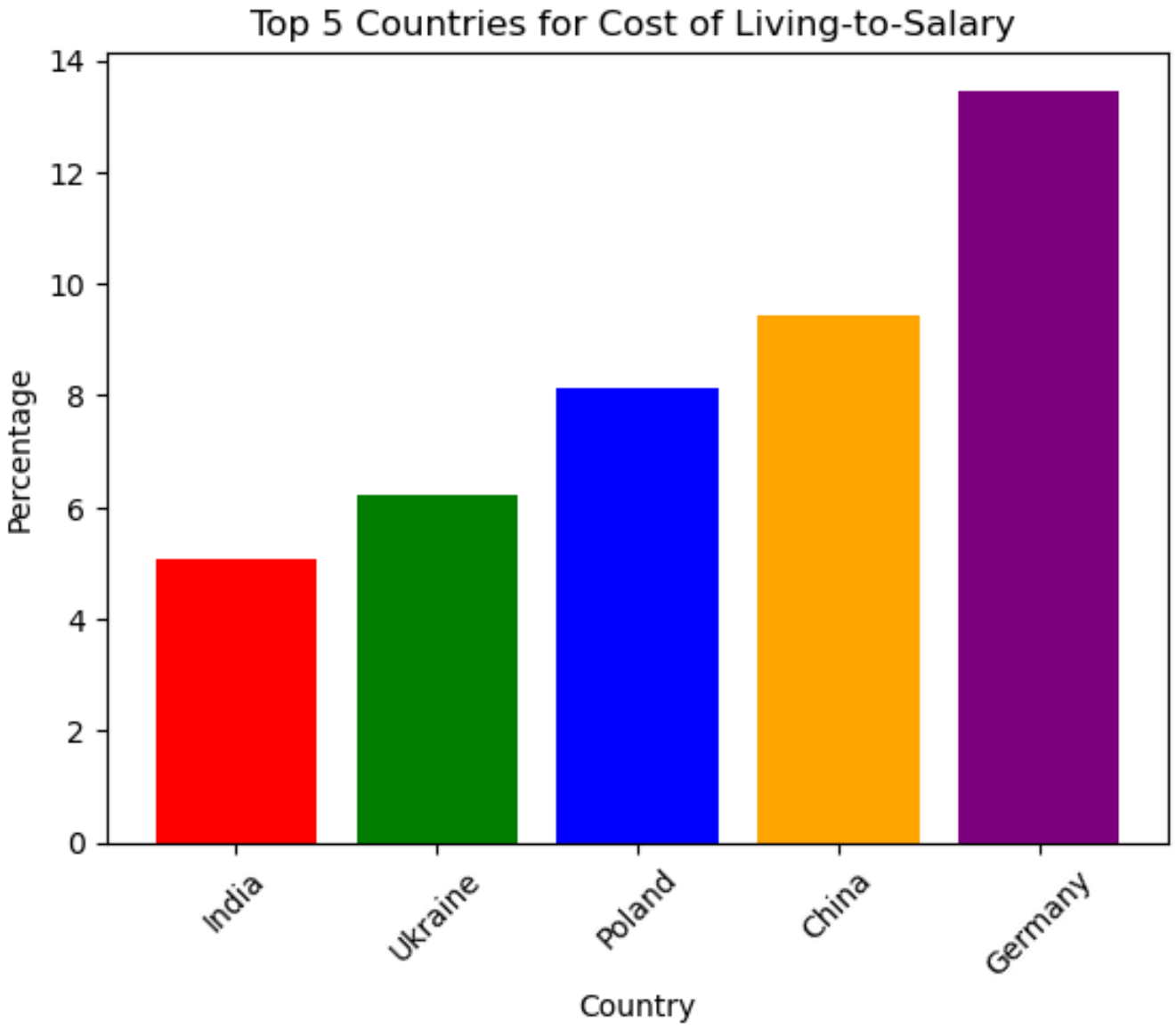
In [193…

```
### Now we can graph it simply for the cost of living.
### The smallest the percentage is the better place to live
### based on the salary amount and the spending

import matplotlib.pyplot as plt

### first we need to write the column names on plt.bar as values.
colors = ['red', 'green', 'blue', 'orange', 'purple']

plt.bar(cost_perc_sorted['Country'][:5],cost_perc_sorted['cost_perc'][:5], color = colors)
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Percentage')
plt.title('Top 5 Countries for Cost of Living-to-Salary ')
plt.show()
```


Top 5 Countries for Cost of Living-to-Salary
Percentage
0
2
4
6
8
10
12
14
India
Ukraine
Poland
China
Germany
Country

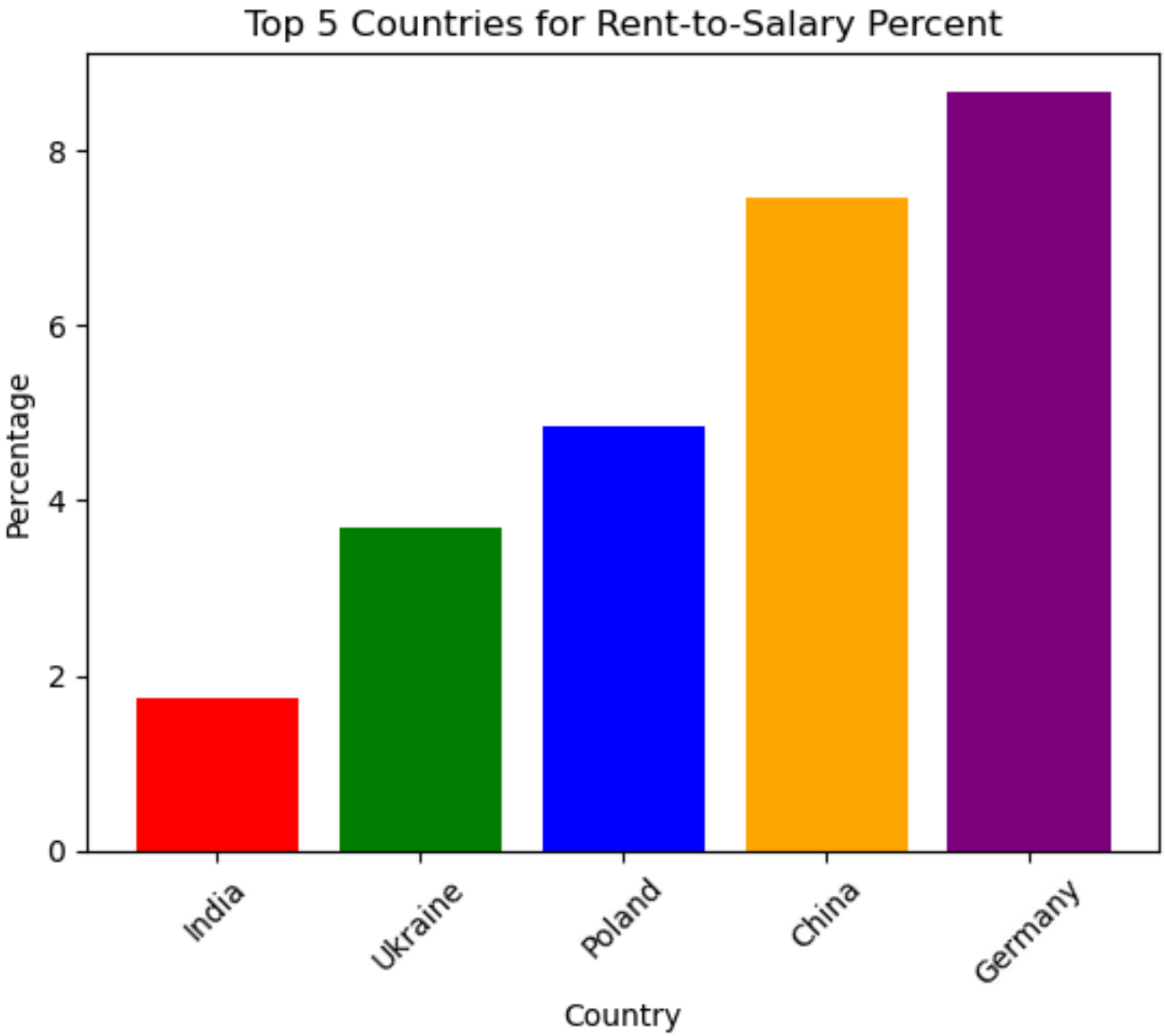
==========

In [204…

```
### Now we can graph it simply for the rent.
### The smallest the percentage is the better place to live
### based on the salary amount and the spending

### First we need to write the column names on plt.bar as values.
colors = ['red', 'green', 'blue', 'orange', 'purple']

plt.bar(rent_perc_sorted['Country'][:5],rent_perc_sorted['rent_perc'][:5], color = colors)
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Percentage')
plt.title('Top 5 Countries for Rent-to-Salary Percent')
plt.show()
```

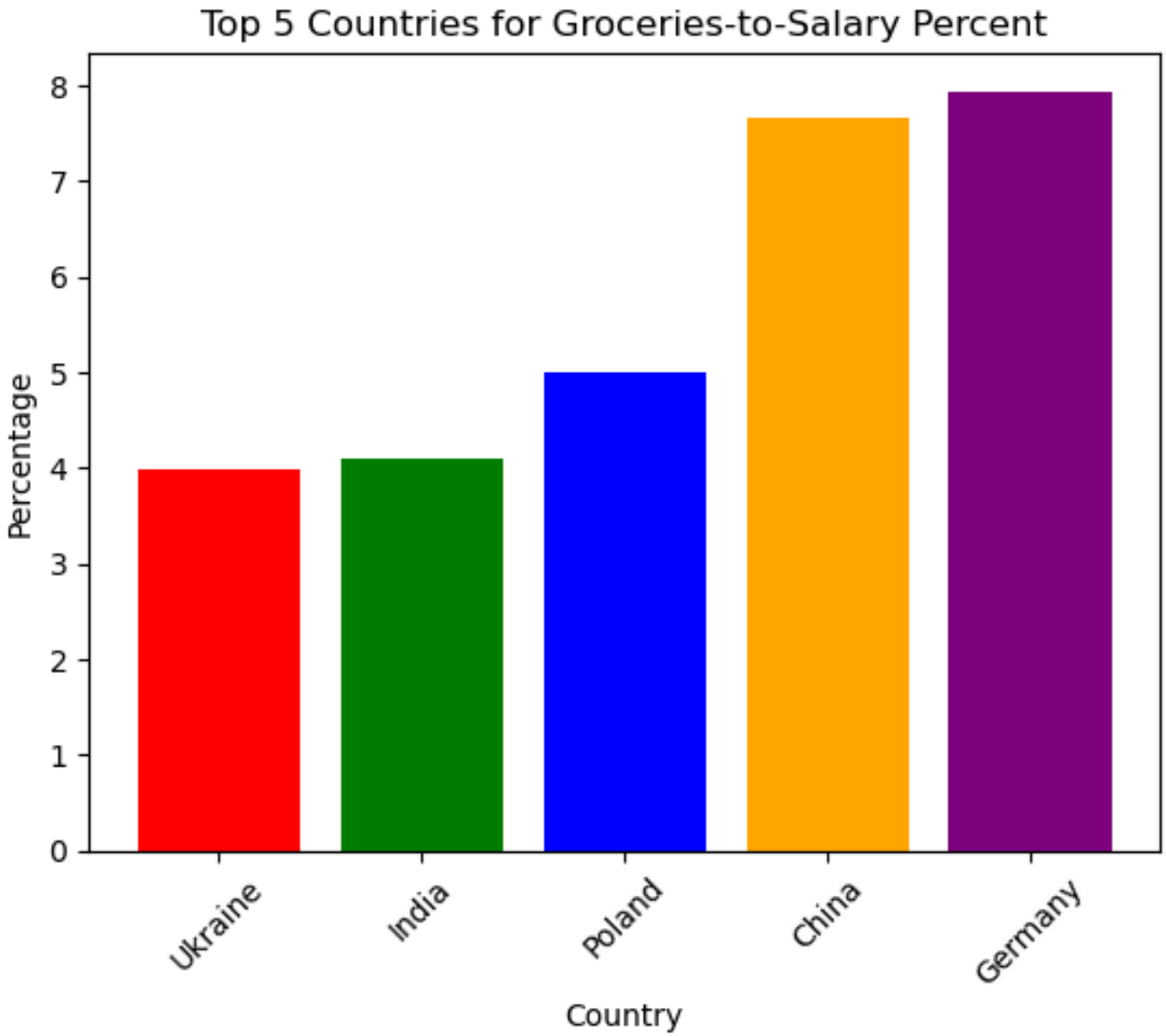

Top 5 Countries for Rent-to-Salary Percent
Percentage
0
2
4
6
8
India
Ukraine
Poland
China
Germany
Country

==========

In [207…

```
colors = ['red', 'green', 'blue', 'orange', 'purple']

plt.bar(groceries_perc_sorted['Country'][:5],groceries_perc_sorted['groceries_perc'][:5], col
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Percentage')
plt.title('Top 5 Countries for Groceries-to-Salary Percent')
plt.show()
```

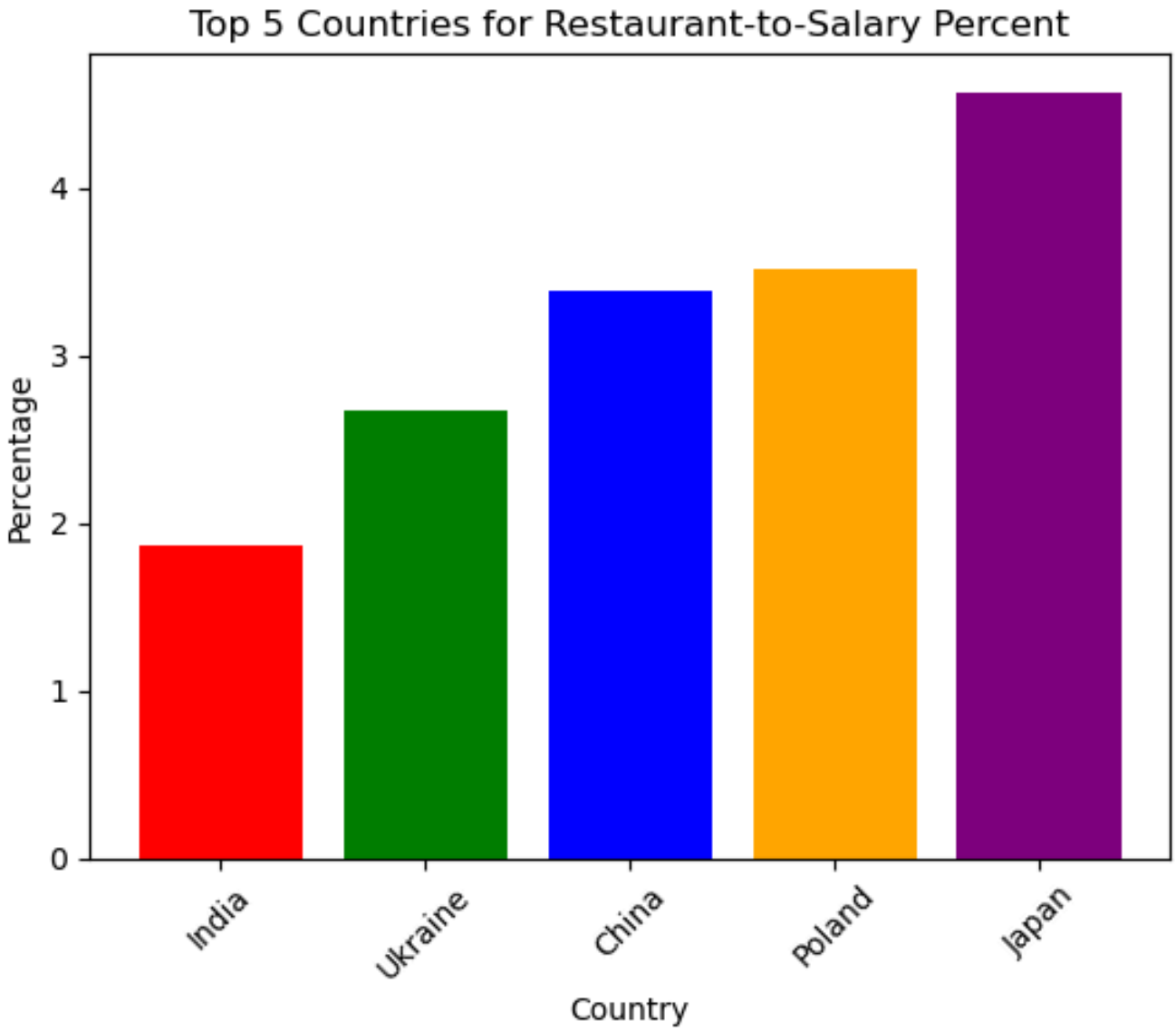

Top 5 Countries for Groceries-to-Salary Percent
Percentage
0
1
2
3
4
5
6
7
8
Ukraine
India
Poland
China
Germany
Country

==========

In [210…

```
colors = ['red', 'green', 'blue', 'orange', 'purple']

plt.bar(restaurent_perc_sorted['Country'][:5],restaurent_perc_sorted['restaurant_perc'][:5],
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Percentage')
plt.title('Top 5 Countries for Restaurant-to-Salary Percent')
plt.show()
```

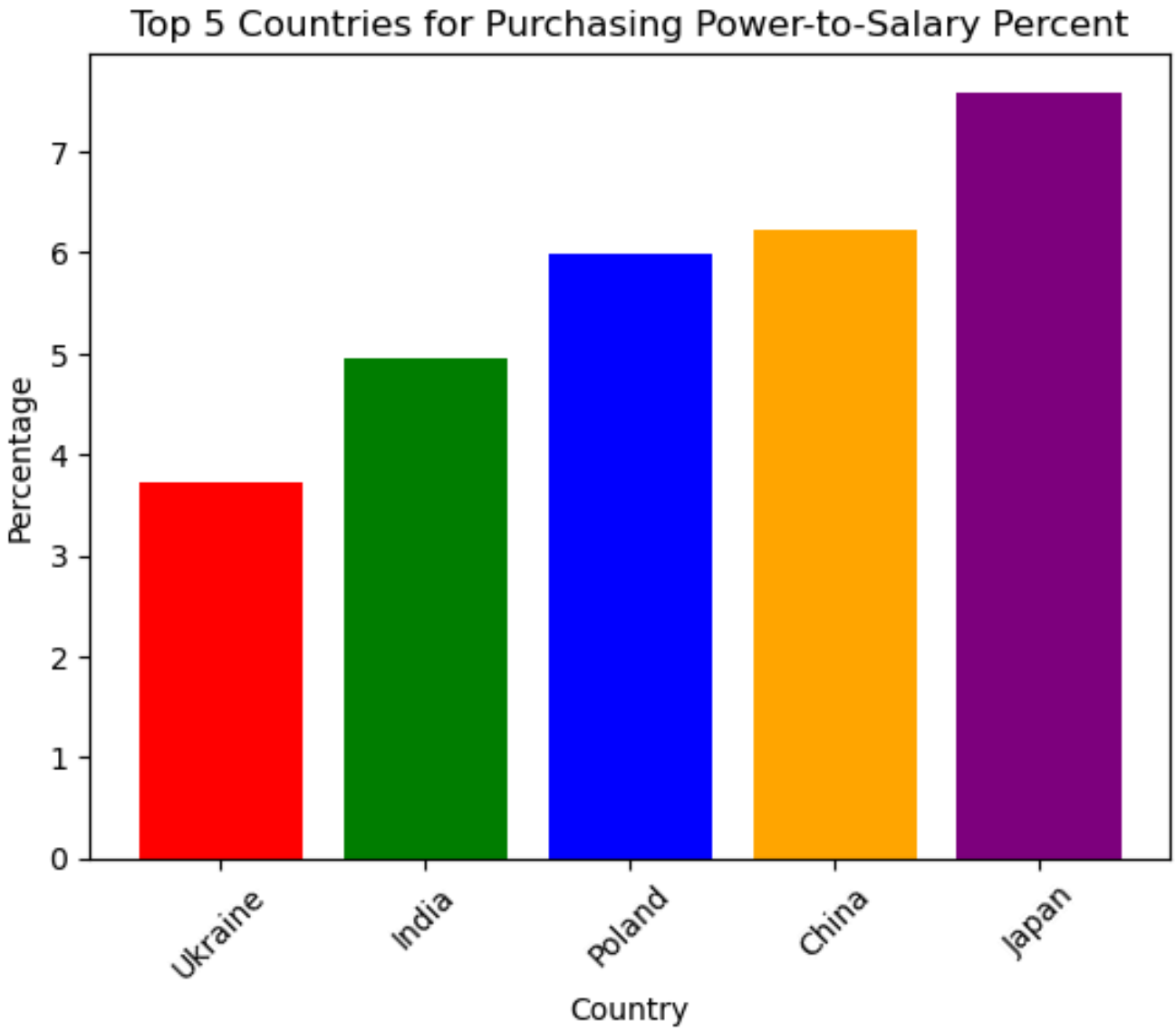


==========

In [213…

```
colors = ['red', 'green', 'blue', 'orange', 'purple']
```

```
plt.bar(purchase_power_perc_sorted['Country'][:5],purchase_power_perc_sorted['purchase_power_
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Percentage')
plt.title('Top 5 Countries for Purchasing Power-to-Salary Percent')
plt.show()
```


Top 5 Countries for Purchasing Power-to-Salary Percent
Percentage
0
1
2
3
4
5
6
7
Ukraine
India
Poland
China
Japan
Country

In [ ]: