Susman_Python_Project_2

July 9, 2023

1 Python Project 2

1.1 By: Sue Susman, MEd, BSN, RN

[38]:
```
!pip install pandas numpy matplotlib openpyxl
```

```
Requirement already satisfied: pandas in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (1.4.4)
Requirement already satisfied: numpy in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (1.21.5)
Requirement already satisfied: matplotlib in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (3.5.2)
Requirement already satisfied: openpyxl in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (3.0.10)
Requirement already satisfied: python-dateutil>=2.8.1 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from pandas)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from pandas)
(2022.1)
Requirement already satisfied: cycler>=0.10 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(1.4.2)
Requirement already satisfied: packaging>=20.0 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(21.3)
Requirement already satisfied: pillow>=6.2.0 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from matplotlib)
(3.0.9)
```

```
Requirement already satisfied: et_xmlfile in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from openpyxl)
(1.1.0)
Requirement already satisfied: six>=1.5 in
/Users/sue_susman/opt/anaconda3/lib/python3.9/site-packages (from python-
dateutil>=2.8.1->pandas) (1.16.0)
```

This code loads and displays the first few rows of each dataset to verify if they were loaded correctly.

[39]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the datasets
cost_of_living = pd.read_csv('cost_of_living.csv')
country_codes = pd.read_excel('country_codes.xlsx')
ds_salaries = pd.read_csv('ds_salaries.csv')
levels_fyi = pd.read_csv('levels_fyi_salary_data.csv')

# Explore the datasets
print("Cost of Living Dataset:")
print(cost_of_living.head())

print("\nCountry Codes Dataset:")
print(country_codes.head())

print("\nData Scientist Salaries Dataset:")
print(ds_salaries.head())

print("\nLevels FYI Dataset:")
print(levels_fyi.head())
```

```
Cost of Living Dataset:
   Rank          City State       Country  Unnamed: 4  Unnamed: 5  Unnamed: 6  \
0   NaN         Kabul   NaN   Afghanistan         NaN         NaN         NaN
1   NaN        Tirana   NaN       Albania         NaN         NaN         NaN
2   NaN       Algiers   NaN       Algeria         NaN         NaN         NaN
3   NaN  Buenos Aires   NaN     Argentina         NaN         NaN         NaN
4   NaN       Yerevan   NaN       Armenia         NaN         NaN         NaN

   Cost of Living Index  Rent Index  Cost of Living Plus Rent Index  \
0                 21.35        3.17                            12.83
1                 38.68       11.33                            25.86
2                 29.84        6.67                            18.98
3                 35.25       10.73                            23.75
4                 34.01       11.89                            23.64

   Groceries Index  Restaurant Price Index  Local Purchasing Power Index
```

```
0            15.22                    14.85                         22.79
1            30.99                    29.86                         31.15
2            30.25                    20.79                         21.78
3            28.54                    34.35                         26.89
4            27.81                    31.01                         29.73

Country Codes Dataset:
          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

Data Scientist Salaries Dataset:
   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

Levels FYI Dataset:
            timestamp    company level                        title  \
0   6/7/2017 11:33:27     Oracle    L3                Product Manager
1  6/10/2017 17:11:29       eBay  SE 2              Software Engineer
2  6/11/2017 14:53:57     Amazon    L7                Product Manager
3   6/17/2017 0:23:14      Apple    M1  Software Engineering Manager
4  6/20/2017 10:58:51  Microsoft    60              Software Engineer

   totalyearlycompensation           location  yearsofexperience  \
0                   127000   Redwood City, CA                1.5
1                   100000  San Francisco, CA                5.0
2                   310000        Seattle, WA                8.0
```

```
3                  372000      Sunnyvale, CA                7.0
4                  157000  Mountain View, CA                5.0

   yearsatcompany  tag  basesalary  ...  Doctorate_Degree  Highschool  \
0             1.5  NaN    107000.0  ...                 0           0
1             3.0  NaN         0.0  ...                 0           0
2             0.0  NaN    155000.0  ...                 0           0
3             5.0  NaN    157000.0  ...                 0           0
4             3.0  NaN         0.0  ...                 0           0

  Some_College Race_Asian  Race_White  Race_Two_Or_More  Race_Black  \
0            0          0           0                 0           0
1            0          0           0                 0           0
2            0          0           0                 0           0
3            0          0           0                 0           0
4            0          0           0                 0           0

   Race_Hispanic  Race  Education
0              0   NaN        NaN
1              0   NaN        NaN
2              0   NaN        NaN
3              0   NaN        NaN
4              0   NaN        NaN

[5 rows x 29 columns]
```

After loading the datasets, I merged them based on the relevant columns.

[40]:
```
# Merge cost_of_living with country_codes
cost_of_living = cost_of_living.merge(country_codes, on='Country', how='left')
```

[41]:
```
print(cost_of_living.columns)
```

```
Index(['Rank', 'City', 'State', 'Country', 'Unnamed: 4', 'Unnamed: 5',
       'Unnamed: 6', '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'],
      dtype='object')
```

[42]:
```
print(ds_salaries.columns)
```

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size'],
      dtype='object')
```

[43]:
```
# Merge ds_salaries with cost_of_living
ds_salaries = ds_salaries.merge(cost_of_living[['Alpha-2 code', 'Cost of Living␣
↪Index']], left_on='employee_residence', right_on='Alpha-2 code', how='left')
```

[44]:
```
print(ds_salaries.columns)
```

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size', 'Alpha-2 code', 'Cost of Living Index'],
      dtype='object')
```

Then, I calculated the normalized salary based on the cost of living index. I used the formula normalized_salary = salary / cost_of_living_index to determine the salary's purchasing power in each location.

[45]:
```
# Calculate normalized salary
ds_salaries['Normalized Salary'] = ds_salaries['salary'] / ds_salaries['Cost of␣
↪Living Index']
```

Next, I calculated the index scores for each location by normalizing the cost of living indices.

[46]:
```
# Calculate index scores
cost_of_living['Normalized Cost of Living'] = cost_of_living['Cost of Living␣
↪Index'] / cost_of_living['Cost of Living Index'].max()
```

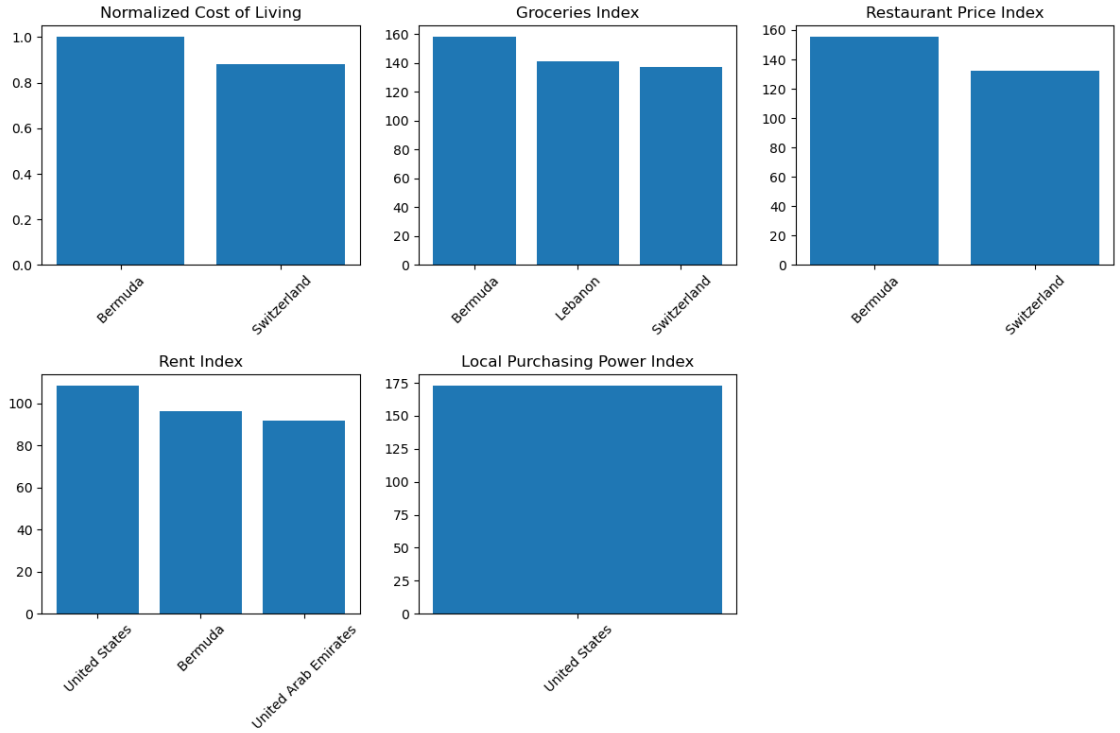
To determine the top 5 locations for each index, I sorted the dataset based on each index column and selected the top 5 rows.

[47]:
```
# Determine top 5 locations for each index
top_5_indices = {}
index_columns = ['Normalized Cost of Living', 'Groceries Index', 'Restaurant␣
↪Price Index', 'Rent Index', 'Local Purchasing Power Index']

for column in index_columns:
    top_5_indices[column] = cost_of_living[['Country', column]].
↪sort_values(by=column, ascending=False).head(5)
```

Finally, I can visualize the results using bar plots for each index.

[48]:
```
# Visualize the results
plt.figure(figsize=(12, 8))

for i, column in enumerate(index_columns):
    plt.subplot(2, 3, i+1)
    plt.bar(top_5_indices[column]['Country'], top_5_indices[column][column])
    plt.title(column)
    plt.xticks(rotation=45)
```

```
plt.tight_layout()
plt.show()
```



The code I used above, created a figure with six subplots, each displaying a bar plot for one of the index columns. The x-axis represents the top 5 locations, and the y-axis represents the index values. This project gave me a statistical analysis with visualizations showcasing the top 5 places in the world where my data scientist salary will go the farthest with respect to each individual index within the cost_of_living.csv file.