Python_Project_Simonsen

May 3, 2024

1 Python Project

1.0.1 Steven Simonsen

1.0.2 5/3/24

[2]:
```
#Note: Analysis findings and description at the bottom of this notebook after␣
↪code
#Import Libraries
import pandas as pd
import numpy as np
import requests
import seaborn as sns
import matplotlib.pyplot as plt
```

[3]:
```
#Import the Data
col_df = pd.read_csv("C:/Users/steve/OneDrive/Documents/School/DSE5002/Week_8/
↪Project/cost_of_living.csv")
#Note: pip install openpyxl required to resolve errror
country_codes_df = pd.read_excel("C:/Users/steve/OneDrive/Documents/School/
↪DSE5002/Week_8/Project/country_codes.xlsx")
salaries_df = pd.read_csv("C:/Users/steve/OneDrive/Documents/School/DSE5002/
↪Week_8/Project/ds_salaries.csv")
salary_detail_df = pd.read_csv("C:/Users/steve/OneDrive/Documents/School/DSE5002/
↪Week_8/Project/Levels_Fyi_Salary_Data.csv")
```

[4]:
```
#Initial Data Cleansing
"""
-col file - split "City" into 3 columns - City, State (if applies), and Country
-separate by comma
-Reorder the dataframe
"""
```

[4]:
```
'\n-col file - split "City" into 3 columns - City, State (if applies), and
Country\n-separate by comma\n-Reorder the dataframe\n'
```

[5]:
```
col_df[['City','Country']] = col_df['City'].str.rsplit(',',n=1,expand=True)
```

```
    #Need second str.rsplit to account for states/provinces
col_df[['City','State or Province']] = col_df['City'].str.
 ↪rsplit(',',n=1,expand=True)

    #Reorder dataframe, will address rank later in the code
col_df = col_df[['Rank','City','State or Province','Country','Cost of Living␣
 ↪Index','Rent Index',
                 'Cost of Living Plus Rent Index','Groceries Index',
                 'Restaurant Price Index','Local Purchasing Power Index']]
```

[6]:
```
print(col_df.head())
```

```
   Rank      City State or Province       Country  Cost of Living Index  \
0   NaN  Hamilton              None       Bermuda                149.02
1   NaN    Zurich              None   Switzerland                131.24
2   NaN     Basel              None   Switzerland                130.93
3   NaN       Zug              None   Switzerland                128.13
4   NaN    Lugano              None   Switzerland                123.99

   Rent Index  Cost of Living Plus Rent Index  Groceries Index  \
0       96.10                          124.22           157.89
1       69.26                          102.19           136.14
2       49.38                           92.70           137.07
3       72.12                          101.87           132.61
4       44.99                           86.96           129.17

   Restaurant Price Index  Local Purchasing Power Index
0                  155.22                         79.43
1                  132.52                        129.79
2                  130.95                        111.53
3                  130.93                        143.40
4                  119.80                        111.96
```

[7]:
```
"""
levels_fyi (salary detail) column
-Perform similar split to separate city from state from country
"""
```

[7]:
```
'\nlevels_fyi (salary detail) column\n-Perform similar split to separate city
from state from country\n'
```

[8]:
```
#separate city and state through split
salary_detail_df[['city','state or province']] = salary_detail_df['location'].
 ↪str.split(',',n=1,expand=True)
```

[9]:
```
#separate state from country with second rsplit
salary_detail_df[['state or province','country']] = salary_detail_df['state or␣
 ↪province'].str.rsplit(',',n=1,expand=True)
```

[10]:
```
#mask variables used as temp variable for reordering dataframe as done below
mask1 = salary_detail_df.pop('city')
salary_detail_df.insert(6, mask1.name, mask1)
```

[11]:
```
mask2 = salary_detail_df.pop('state or province')
salary_detail_df.insert(7, mask2.name, mask2)
```

[12]:
```
mask3=salary_detail_df.pop('country')
salary_detail_df.insert(8, mask3.name, mask3)
```

[13]:
```
#strip white space for easier reading and maintain data cleanliness
salary_detail_df['city']=salary_detail_df['city'].str.strip()
salary_detail_df['state or province']=salary_detail_df['state or province'].str.
 ↪strip()
salary_detail_df['country']=salary_detail_df['country'].str.strip()
```

[16]:
```
#print head to show dataframe
print(salary_detail_df.iloc[:, :10])
```

```
                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
...                   ...         ...       ...                            ...
62637   9/9/2018 11:52:32      Google        T4              Software Engineer
62638   9/13/2018 8:23:32   Microsoft        62              Software Engineer
62639  9/13/2018 14:35:59        MSFT        63              Software Engineer
62640  9/16/2018 16:10:35  Salesforce  Lead MTS              Software Engineer
62641   1/29/2019 5:12:59       apple      ict3              Software Engineer

       totalyearlycompensation           location           city  \
0                       127000   Redwood City, CA   Redwood City
1                       100000  San Francisco, CA  San Francisco
2                       310000        Seattle, WA        Seattle
3                       372000      Sunnyvale, CA      Sunnyvale
4                       157000  Mountain View, CA  Mountain View
...                        ...                ...            ...
62637                   327000        Seattle, WA        Seattle
62638                   237000        Redmond, WA        Redmond
62639                   220000        Seattle, WA        Seattle
62640                   280000  San Francisco, CA  San Francisco
62641                   200000      Sunnyvale, CA      Sunnyvale

      state or province country  yearsofexperience
0                    CA    None                1.5
1                    CA    None                5.0
```

```
2                    WA    None                8.0
3                    CA    None                7.0
4                    CA    None                5.0
...                 ...     ...                ...
62637                WA    None               10.0
62638                WA    None                2.0
62639                WA    None               14.0
62640                CA    None                8.0
62641                CA    None                0.0

[62642 rows x 10 columns]
```

[19]:
```
"""
Salaries file
-rename first column to index
"""
```

[19]:
```
'\nSalaries file\n-rename first column to index\n'
```

[20]:
```
salaries_df.rename(columns={'Unnamed: 0':'index'}, inplace=True)
print(salaries_df.iloc[:, :5])
```

```
     index  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
..     ...        ...              ...             ...
602    602       2022               SE              FT
603    603       2022               SE              FT
604    604       2022               SE              FT
605    605       2022               SE              FT
606    606       2022               MI              FT

                     job_title
0               Data Scientist
1    Machine Learning Scientist
2              Big Data Engineer
3          Product Data Analyst
4     Machine Learning Engineer
..                          ...
602               Data Engineer
603               Data Engineer
604                Data Analyst
605                Data Analyst
606                AI Scientist
```

```
[607 rows x 5 columns]
```

[21]:
```
#Joins
"""
1) Join ds_salaries to country codes to get two digit country code in table
2) join levels fyi to country codes to get two digit country code
3) join two tables above to get salary in levels fyi in USD
"""
```

[21]:
```
'\n1) Join ds_salaries to country codes to get two digit country code in
table\n2) join levels fyi to country codes to get two digit country code\n3)
join two tables above to get salary in levels fyi in USD \n'
```

[22]:
```
#Create new column in salaries_df to create common join field
salaries_df['Alpha-2 code'] = salaries_df['employee_residence'].str.
 ↪split(',',n=1,expand=True)
```

[23]:
```
#create join from salary detail to country code
left_dsal_codes = salaries_df.merge(country_codes_df, how='left', on='Alpha-2␣
 ↪code')
```

[24]:
```
sum_sal_stats_country = left_dsal_codes.groupby(['Country','Alpha-2 code']).agg(
    mean_salary_usd=('salary_in_usd', np.mean),
    median_salary_usd=('salary_in_usd', np.median),
    std_salary_usd = ('salary_in_usd', np.std)
    )

print(sum_sal_stats_country.head())
```

```
                      mean_salary_usd  median_salary_usd  std_salary_usd
Country   Alpha-2 code
Algeria   DZ              100000.000000           100000.0             NaN
Argentina AR               60000.000000            60000.0             NaN
Australia AU              108042.666667            87425.0    36337.909768
Austria   AT               76738.666667            74130.0    13386.018539
Belgium   BE               85699.000000            85699.0     4179.001077

C:\Users\steve\AppData\Local\Temp\ipykernel_29948\1597426545.py:1:
FutureWarning: The provided callable <function mean at 0x0000027DC1617380> is
currently using SeriesGroupBy.mean. In a future version of pandas, the provided
callable will be used directly. To keep current behavior pass the string "mean"
instead.
  sum_sal_stats_country = left_dsal_codes.groupby(['Country','Alpha-2
code']).agg(
C:\Users\steve\AppData\Local\Temp\ipykernel_29948\1597426545.py:1:
FutureWarning: The provided callable <function median at 0x0000027DC1844720> is
currently using SeriesGroupBy.median. In a future version of pandas, the
provided callable will be used directly. To keep current behavior pass the
string "median" instead.
```

```
  sum_sal_stats_country = left_dsal_codes.groupby(['Country','Alpha-2
code']).agg(
C:\Users\steve\AppData\Local\Temp\ipykernel_29948\1597426545.py:1:
FutureWarning: The provided callable <function std at 0x0000027DC16174C0> is
currently using SeriesGroupBy.std. In a future version of pandas, the provided
callable will be used directly. To keep current behavior pass the string "std"
instead.
  sum_sal_stats_country = left_dsal_codes.groupby(['Country','Alpha-2
code']).agg(
```

[26]:
```
#join col_df to country codes
col_df['Country']=col_df['Country'].str.strip()
col_df['Country']=col_df['Country'].str.replace('United States','United States␣
↪of America (the)')
left_col_codes = col_df.merge(country_codes_df, how='left', on='Country')
```

[27]:
```
#Join to cost_of_Living to summary statistics grouped table above
left_col_sumstats = left_col_codes.merge(sum_sal_stats_country, how='left',␣
↪on='Alpha-2 code')
```

[28]:
```
#Add columnn that takes mean of sumstats index ratings
left_col_sumstats['Average Index Rating']=left_col_sumstats.iloc[:, 4:10].
↪mean(axis=1)
```

[29]:
```
#Divide mean_salary_usd by mean index ratings - Call this column salary_score
left_col_sumstats['salary_score']=left_col_sumstats['mean_salary_usd'] /␣
↪left_col_sumstats['Average Index Rating']
```

[30]:
```
#populate rank based on salary_score
left_col_sumstats['Rank'] = (left_col_sumstats['salary_score']
                 .rank(method='dense', ascending=False)
                )
```

[31]:
```
#group summary stats by country to get avg. salary score by country
left_col_sumstats_grouped = left_col_sumstats.groupby(['Country']).agg(
    average_salary_score=('salary_score', np.mean))
```

```
C:\Users\steve\AppData\Local\Temp\ipykernel_29948\3065518909.py:2:
FutureWarning: The provided callable <function mean at 0x0000027DC1617380> is
currently using SeriesGroupBy.mean. In a future version of pandas, the provided
callable will be used directly. To keep current behavior pass the string "mean"
instead.
  left_col_sumstats_grouped = left_col_sumstats.groupby(['Country']).agg(
```

[39]:
```
#top 5 countries with city included to plot based on salary score for countries
top_5_by_city = left_col_sumstats[left_col_sumstats['Country'].isin(
    ['Malaysia', 'Algeria', 'Iraq', 'Puerto Rico', 'Bulgaria'])]
print(top_5_by_city.iloc[:, :4])
```

	Rank	City	State or Province	Country
192	9.0	San Juan	None	Puerto Rico
393	40.0	Sofia	None	Bulgaria
414	3.0	Petaling Jaya	None	Malaysia
418	1.0	Kota Kinabalu	None	Malaysia
419	4.0	Kuala Lumpur	None	Malaysia
420	17.0	Varna	None	Bulgaria
429	12.0	Plovdiv	None	Bulgaria
432	8.0	Erbil (Irbil)	None	Iraq
438	2.0	Penang	None	Malaysia
449	10.0	Burgas	None	Bulgaria
471	7.0	Baghdad	None	Iraq
512	6.0	Algiers	None	Algeria
547	5.0	Cyberjaya	Selangor	Malaysia

[36]:
```
#Salary Detail - Filter to top 5 countries in detail dataset to examine top 5␣
↪country in additional viz below
top_5_det = salary_detail_df[salary_detail_df['country'].isin(
    ['Malaysia', 'Algeria', 'Iraq', 'Puerto Rico', 'Bulgaria'])]
print(top_5_det.iloc[:, :9])
```

	timestamp	company	level	\
3839	1/21/2019 23:23:56	VMware	MTS 2	
5684	4/11/2019 6:39:20	Microsoft	62	
8616	7/12/2019 5:47:20	Uber	Software Engineer II	
15373	1/22/2020 0:32:49	Automattic	NaN	
19406	4/11/2020 12:53:55	Uber	L6	
20226	4/30/2020 0:20:06	SAP	t2	
22461	6/12/2020 15:31:52	HPE	M26	
28229	8/30/2020 18:33:21	Huawei	19	
29354	9/9/2020 8:06:13	Guidewire Software	L1	
38391	12/31/2020 8:51:33	Schlumberger	E3	
38554	1/3/2021 7:22:11	Accenture	Analyst	
39815	1/19/2021 23:03:50	test	SE1	
45909	3/24/2021 9:08:59	Dell Technologies	Senior Principal Engineer	
48086	4/13/2021 7:58:13	Uber	Senior Software Engineer II	
53374	6/4/2021 4:13:10	VMware	MTS 3	
54162	6/11/2021 4:14:20	VMware	P3	
59952	8/3/2021 11:21:02	SAP	L2	
60933	8/10/2021 11:53:05	Facebook	L3	

	title	totalyearlycompensation	\
3839	Software Engineer	50000	
5684	Software Engineer	104000	
8616	Software Engineer	67000	
15373	Software Engineer	121000	
19406	Software Engineering Manager	320000	
20226	Software Engineer	32000	

```
22461            Hardware Engineer                  89000
28229           Solution Architect                 300000
29354            Software Engineer                  20000
38391          Mechanical Engineer                  16000
38554             Business Analyst                  17000
39815            Software Engineer                  62000
45909            Software Engineer                  79000
48086            Software Engineer                 201000
53374            Software Engineer                  52000
54162                    Recruiter                  22000
59952            Product Designer                   30000
60933                    Marketing                 150000
```

```
                      location          city state or province      country
3839         Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
5684   San Juan, PR, Puerto Rico      San Juan                PR  Puerto Rico
8616         Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
15373        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
19406        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
20226        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
22461  Aguadilla, PR, Puerto Rico     Aguadilla                PR  Puerto Rico
28229  Kuala Lumpur, KL, Malaysia  Kuala Lumpur                KL     Malaysia
29354  Kuala Lumpur, KL, Malaysia  Kuala Lumpur                KL     Malaysia
38391  Johor Baharu, JH, Malaysia  Johor Baharu                JH     Malaysia
38554  Kuala Lumpur, KL, Malaysia  Kuala Lumpur                KL     Malaysia
39815       Penang, PG, Malaysia        Penang                PG     Malaysia
45909        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
48086        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
53374        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
54162        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
59952        Sofia, SF, Bulgaria         Sofia                SF     Bulgaria
60933          Nineveh, NI, Iraq       Nineveh                NI         Iraq
```

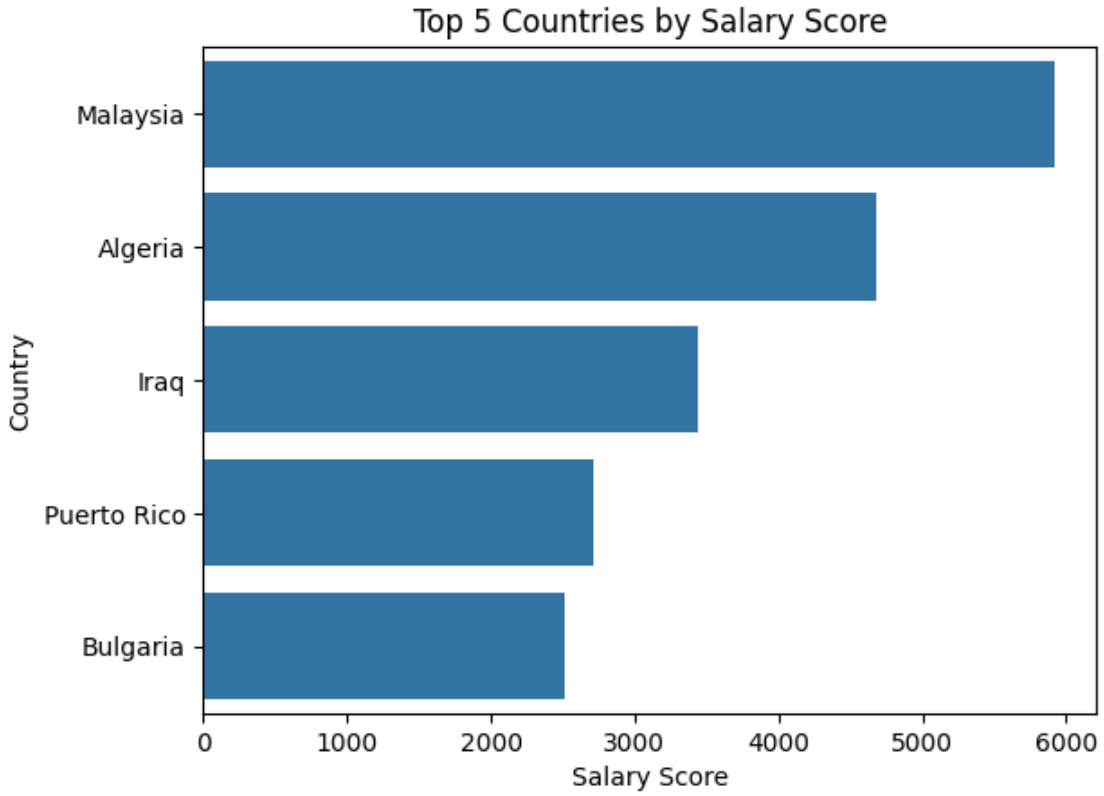
[34]:
```
#Group salary detail by title to show and count most common jobs available
title_counts = top_5_det.groupby(['country', 'title']).size().
 ↪reset_index(name='counts')
print(title_counts.head())
```

```
    country                         title  counts
0  Bulgaria              Product Designer       1
1  Bulgaria                     Recruiter       1
2  Bulgaria             Software Engineer       7
3  Bulgaria  Software Engineering Manager       1
4      Iraq                     Marketing       1
```

[ ]:
```
#Visuals
```

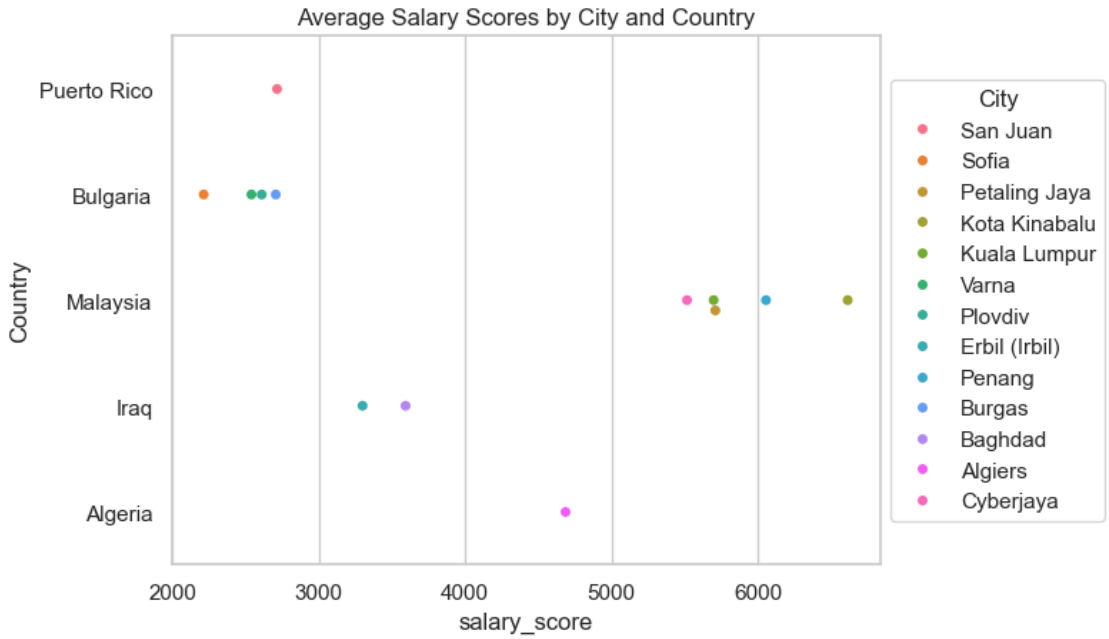
[40]:
```
top_5_countries = left_col_sumstats_grouped.nlargest(5, 'average_salary_score')

sns.barplot(x='average_salary_score', y='Country', data=top_5_countries)
plt.xlabel('Salary Score')
plt.ylabel('Country')
plt.title('Top 5 Countries by Salary Score')
plt.show()
```



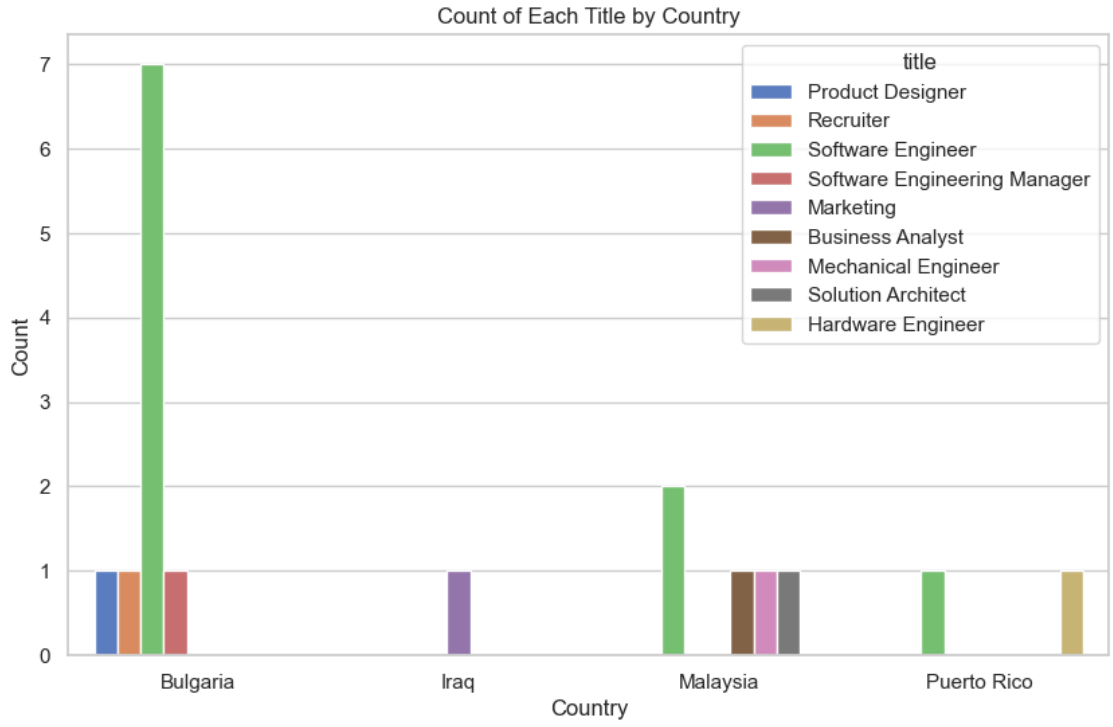
[42]:
```
#Citites in top 5
sns.set_theme(style="whitegrid", palette="muted")

ax = sns.swarmplot(data=top_5_by_city, x="salary_score", y="Country", hue="City")
ax.set(ylabel="Country")
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5), title="City")
plt.title("Average Salary Scores by City and Country")
plt.show()
```



[43]:
```
#Title Counts in Detail Dataset
plt.figure(figsize=(10, 6))
sns.barplot(x='country', y='counts', hue='title', data=title_counts)

plt.title('Count of Each Title by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.show()
```



[45]:
```
"""
As can be seen in the visuals above, the five top countries where salary would␣
 ↪go the farthest
in USD are as follows (1=salary goes the furthest):
1) Malaysia
2) Algeria
3) Iraq
4) Puerto Rico
5) Bulgaria


To produce this list, I took an average of all indexes in the cost of living␣
 ↪file across the columns.
Then, I pulled in the mean salary by country based on the ds_salaries file in␣
 ↪USD.
Finally, I created a salary_score by dividing the mean salary for the country by␣
 ↪the
average quality of life index. Taking the average again and grouping across␣
 ↪country,
I was able to find the top 5 countries where salary would go the farthest in USD.
```

```
I was then able to provide some additional visuals around cities and types of␣
 ↪jobs.
As can be seen within the graphs, there is a large gap in salary_score between
the top cities in Malaysia vs the bottom cities in Bulgaria. Even though they're␣
 ↪both
in the top 5, I imagine life to be pretty different between these two places␣
 ↪based on the
variance of the salary score.

In the third graph derived from the salary detail file, software developers
seem to have the best chance across the board of landing a position in one of␣
 ↪these countries.
However, if given the chance, I would like to explore this data further to␣
 ↪gather more information,
as Algeria is not included in the detail salary dataset.
"""
```

[45]: "\nAs can be seen in the visuals above, the five top countries where salary would go the farthest\nin USD are as follows (1=salary goes the furthest):\n1) Malaysia\n2) Algeria\n3) Iraq\n4) Puerto Rico\n5) Bulgaria\n\n\nTo produce this list, I took an average of all indexes in the cost of living file across the columns.\nThen, I pulled in the mean salary by country based on the ds_salaries file in USD.\nFinally, I created a salary_score by dividing the mean salary for the country by the\naverage quality of life index. Taking the average again and grouping across country,\nI was able to find the top 5 countries where salary would go the farthest in USD.\n\n\nI was then able to provide some additional visuals around cities and types of jobs.\nAs can be seen within the graphs, there is a large gap in salary_score between\nthe top cities in Malaysia vs the bottom cities in Bulgaria. Even though they're both\nin the top 5, I imagine life to be pretty different between these two places based on the\nvariance of the salary score.\n\nIn the third graph derived from the salary detail file, software developers \nseem to have the best chance across the board of landing a position in one of these countries. \nHowever, if given the chance, I would like to explore this data further to gather more information, \nas Algeria is not included in the detail salary dataset.\n"