In [1]:
```
# Cost of Living Index Analysis
```

In [2]:
```
# Performing EDA, Data Story Telling and Data Visualization
```

In [3]:
```
#importing important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import altair as alt
```

```
C:\Users\Sharon\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [4]:
```
df = pd.read_csv("living_cost.csv")
```

In [5]:
```
df.head()
```

Out[5]:

	Rank	Country	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index
0	1	Switzerland	101.1	46.5	74.9	109.1	97.0	158.7
1	2	Bahamas	85.0	36.7	61.8	81.6	83.3	54.6
2	3	Iceland	83.0	39.2	62.0	88.4	86.8	120.3
3	4	Singapore	76.7	67.2	72.1	74.6	50.4	111.1
4	5	Barbados	76.6	19.0	48.9	80.8	69.4	43.5

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Rank', 'Country', 'Cost of Living Index', 'Rent Index',
       'Cost of Living Plus Rent Index', 'Groceries Index',
       'Restaurant Price Index', 'Local Purchasing Power Index'],
      dtype='object')
```

In [7]:
```
1 df.isnull().sum()
```

Out[7]:
```
Rank                            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
dtype: int64
```

In [8]:
```
1 df.shape
```

Out[8]:
```
(121, 8)
```

In [9]:
```
1 df.describe()
```

Out[9]:

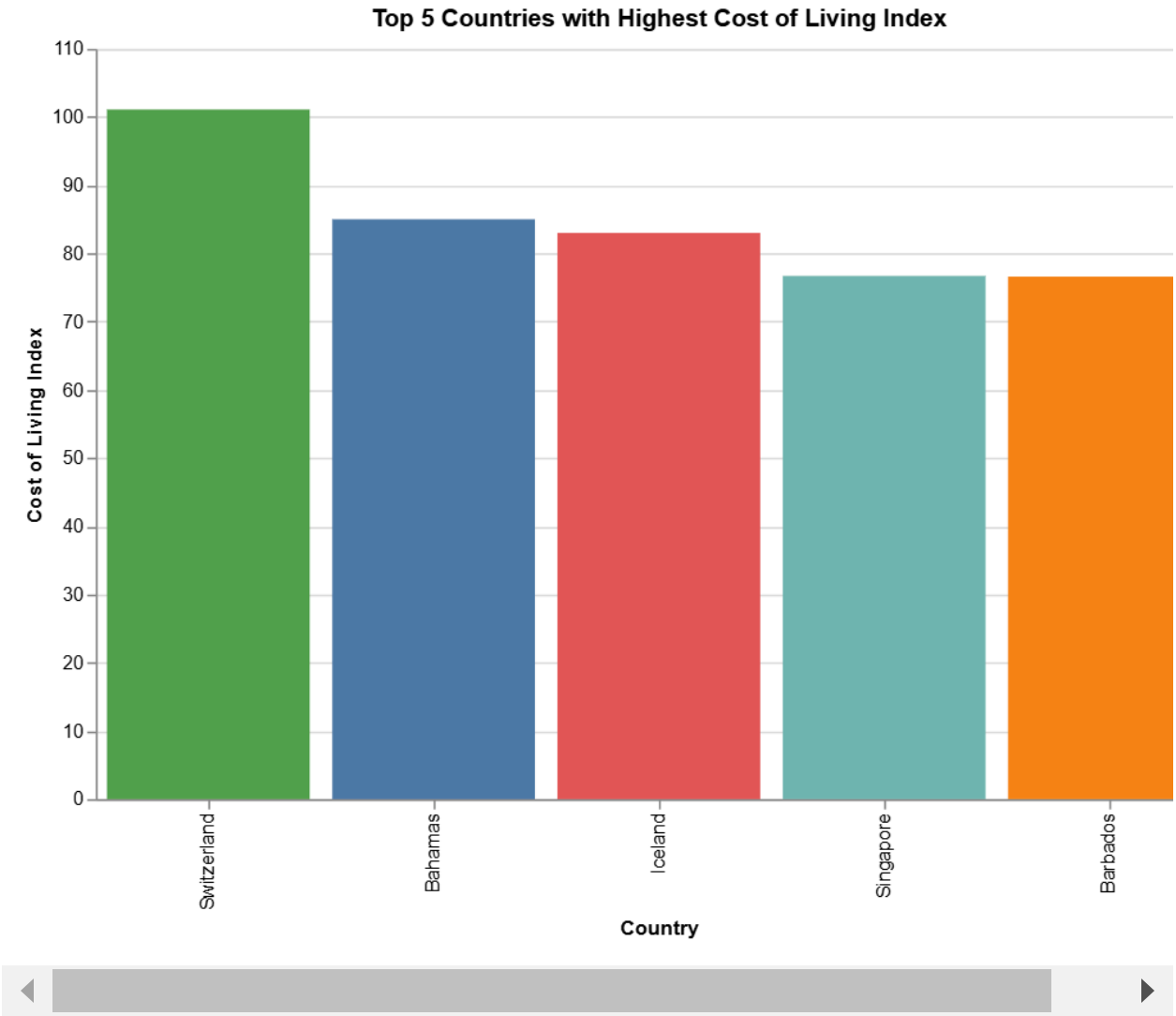
	Rank	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index
count	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000
mean	61.000000	43.555372	16.052893	30.357851	44.228926	36.471074	65.094215
std	35.073732	16.147574	11.412267	13.263721	17.055109	18.258110	39.569094
min	1.000000	18.800000	2.400000	11.100000	17.500000	12.800000	2.300000
25%	31.000000	30.200000	8.500000	19.800000	31.600000	21.600000	34.800000
50%	61.000000	39.500000	12.400000	27.000000	40.500000	33.100000	50.600000
75%	91.000000	52.800000	20.100000	37.000000	53.700000	47.200000	99.400000
max	121.000000	101.100000	67.200000	74.900000	109.100000	97.000000	182.500000

In [10]:
```
1 ### visuallization part
2 ## in this part we will see some of the basics plots and some camparisons do
3 ## find out the patterns in data
```

In [11]:

```
## Countries with the highest CLI(Cost of Living Index)

top_5_countries = df.nlargest(5, 'Cost of Living Index')

# Create the bar plot
bar_chart = alt.Chart(top_5_countries).mark_bar().encode(
    x=alt.X('Country:N', sort='-y'),  # Nominal type for country
    y=alt.Y('Cost of Living Index:Q'),  # Quantitative type for the index
    color=alt.Color('Country:N', legend=None)  # Color the bars by country,
).properties(
    title='Top 5 Countries with Highest Cost of Living Index',
    width=600,  # Adjust the width to make the chart wider
    height=400  # You can keep the height as is or adjust it
)

# Display the plot
bar_chart
```

Out[11]:

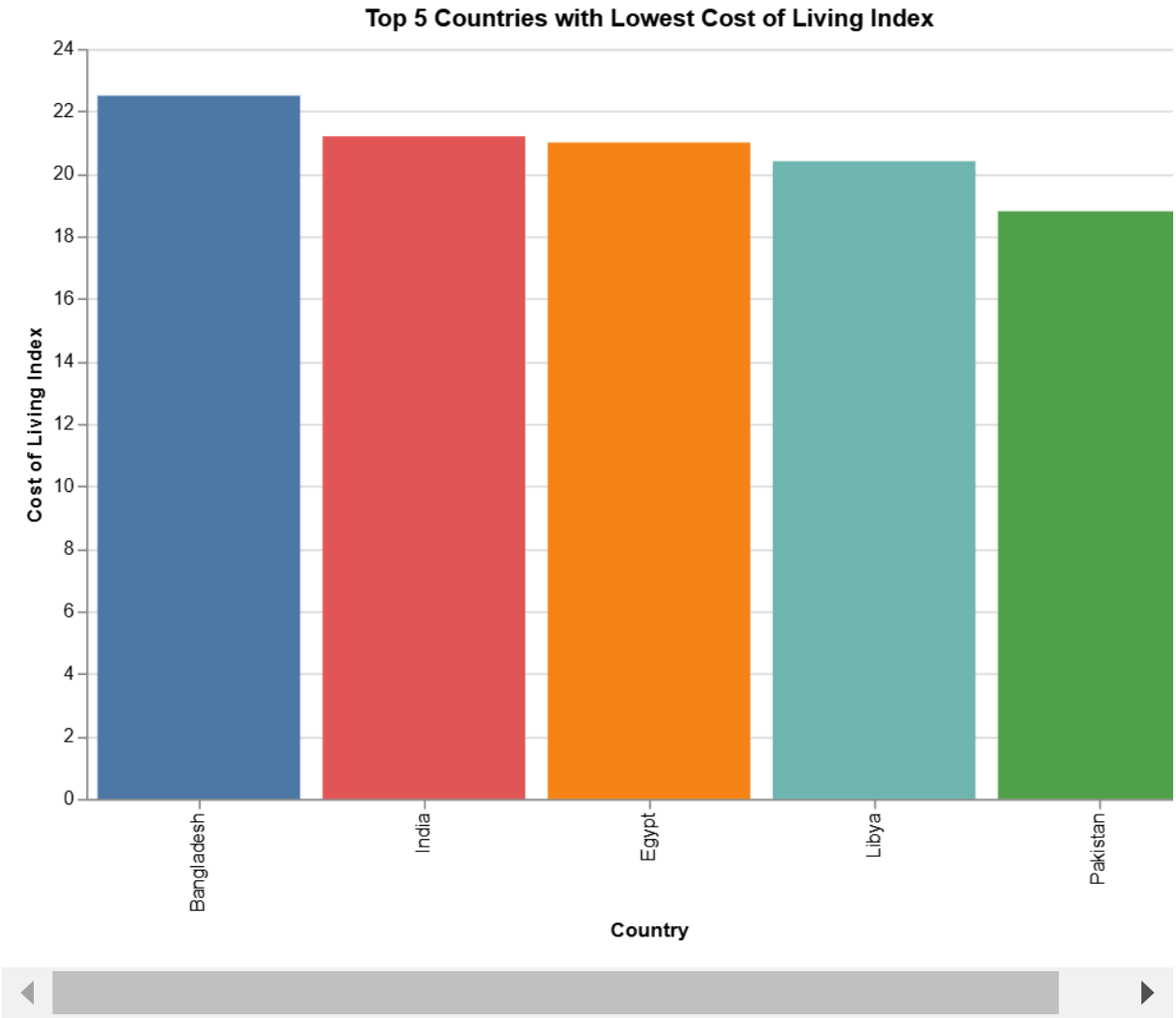


This bar chart highlights the countries with the highest cost of living index. Switzerland leads the chart, followed by Bahamas, Iceland, Singapore, and Barbados. These countries are well-known for their high living standards, luxury, and advanced infrastructure, making them more expensive to reside in.

In [12]:

```
## Countries with the lowest CLI(Cost of Living Index)

low_5_countries = df.nsmallest(5, 'Cost of Living Index')

# Create the bar plot
bar_chart = alt.Chart(low_5_countries).mark_bar().encode(
    x=alt.X('Country:N', sort='-y'),  # Nominal type for country
    y=alt.Y('Cost of Living Index:Q'),  # Quantitative type for the index
    color=alt.Color('Country:N', legend=None)  # Color the bars by country,
).properties(
    title='Top 5 Countries with Lowest Cost of Living Index',
    width=600,  # Adjust the width to make the chart wider
    height=400  # You can keep the height as is or adjust it
)

# Display the plot
bar_chart
```

Out[12]:

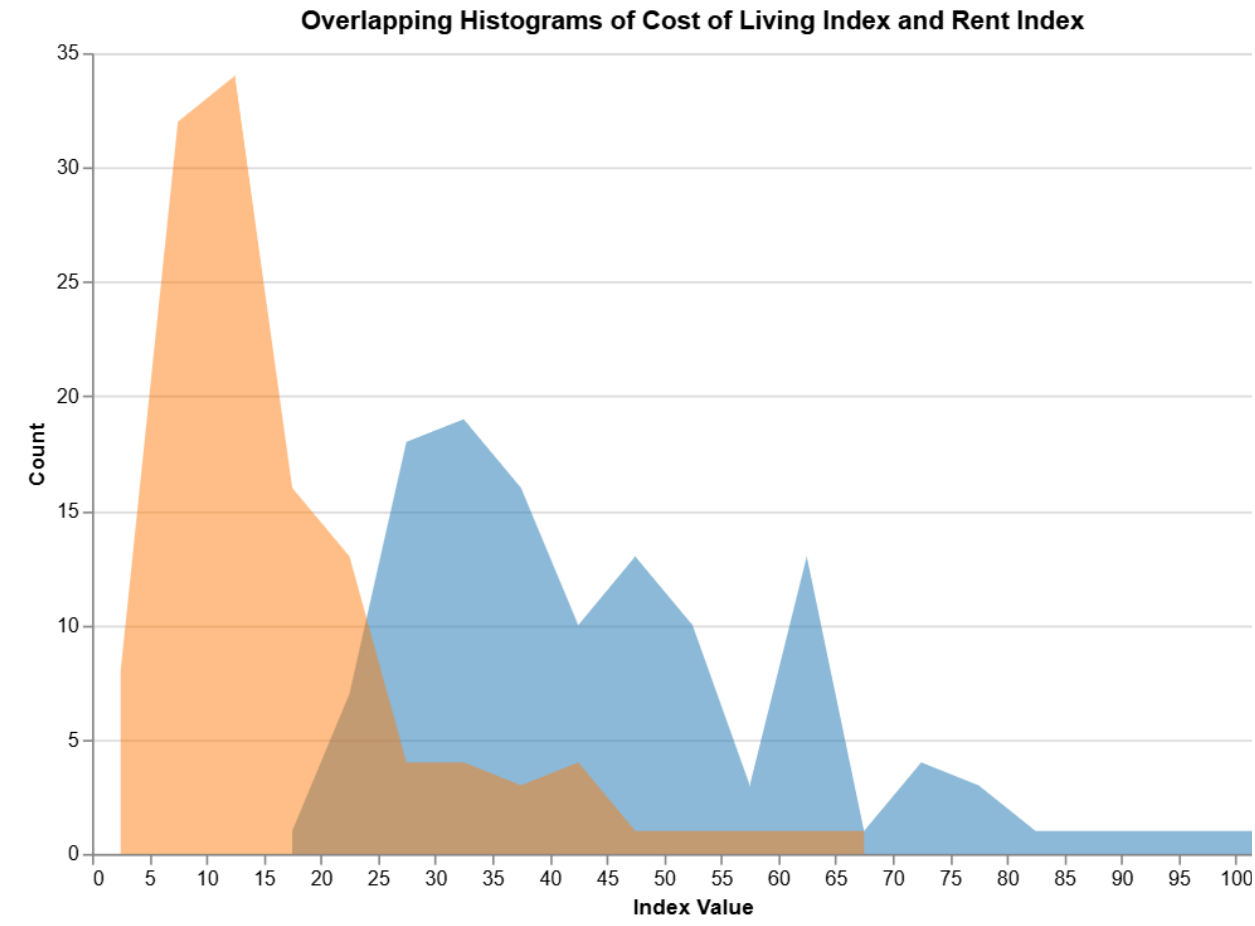


This bar chart displays the countries with the lowest cost of living index. It highlights affordability in these countries. Pakistan ranks the lowest, followed by Libya, Egypt, India, and Bangladesh. These countries are known for their low living costs, making them attractive for individuals or businesses seeking affordability.

In [13]:

```
## Cost of Living index Vs Rent Index

# Prepare data for melting
melted_df = df.melt(value_vars=['Cost of Living Index', 'Rent Index'], var_n

# Create overlapping histograms
overlapping_histogram = alt.Chart(melted_df).mark_area(opacity=0.5).encode(
    alt.X('Value:Q', bin=alt.Bin(maxbins=30), title='Index Value'),
    alt.Y('count():Q', stack=None, title='Count'),
    alt.Color('Index Type:N', scale=alt.Scale(scheme='category10')),
    tooltip=[
        alt.Tooltip('Index Type:N', title='Index Type'),
        alt.Tooltip('count():Q', title='Count')
    ]
).properties(
    title='Overlapping Histograms of Cost of Living Index and Rent Index',
    width=600,
    height=400
)

# Display the plot
overlapping_histogram
```

Out[13]:



This histogram compares the distributions of the Cost of Living Index and Rent Index. It shows that the Cost of Living Index values are generally higher than Rent Index values, indicating that overall living costs (beyond just rent) tend to dominate expenses. The overlapping areas help compare density patterns, emphasizing where both indices align and diverge.

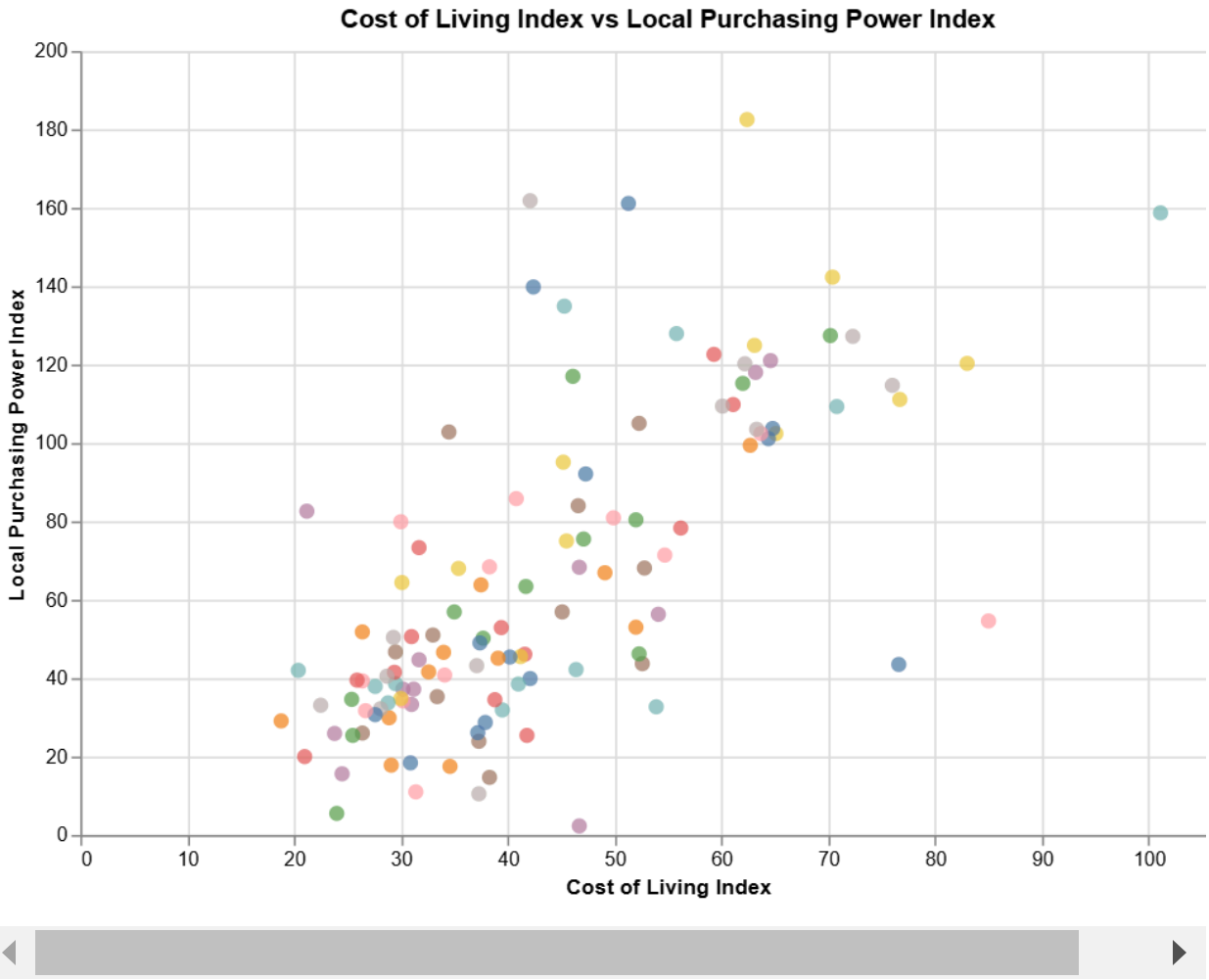
In [14]:
```
## CLI Vs CLI plus RI

# Create the scatter plot
scatter_plot = alt.Chart(df).mark_circle(size=60).encode(
    x=alt.X('Cost of Living Index:Q', title='Cost of Living Index'),
    y=alt.Y('Cost of Living Plus Rent Index:Q', title='Cost of Living Plus R
    color=alt.Color('Country:N', legend=None),  # Optional: Color points by
    tooltip=['Country', 'Cost of Living Index', 'Cost of Living Plus Rent In
).properties(
    title='Cost of Living Index vs Cost of Living Plus Rent Index',
    width=600,  # Adjust width as needed
    height=400  # Adjust height as needed
).interactive()  # Enable zoom and pan

# Display the plot
scatter_plot
```

Out[14]:

Countries with higher Cost of Living Index values are likely to have higher rent, contributing significantly to overall living costs. Further analysis could involve identifying specific countries that deviate from the trend to understand why their rent index behaves differently.

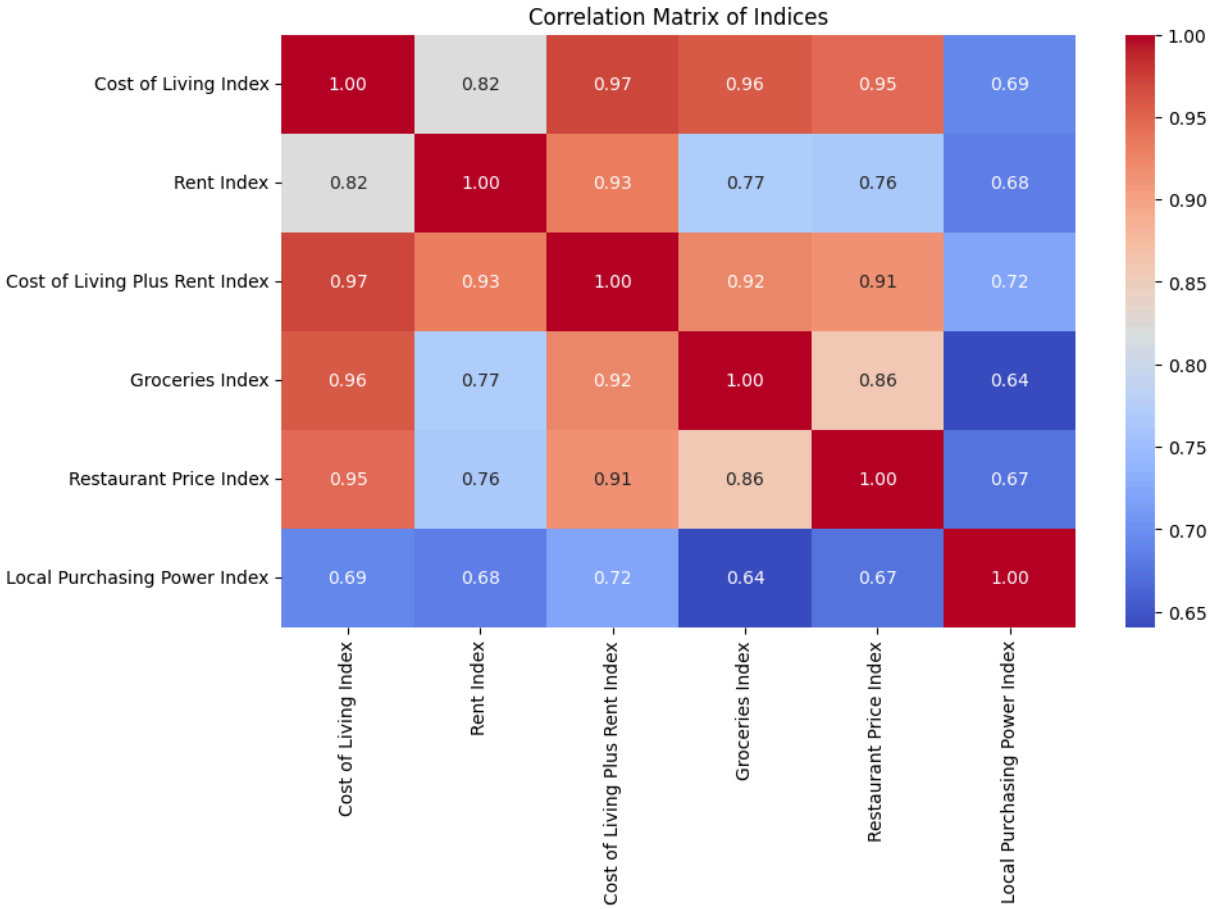
In [15]:
```
# Create the scatter plot
scatter_plot = alt.Chart(df).mark_circle(size=60).encode(
    x=alt.X('Cost of Living Index:Q', title='Cost of Living Index'),
    y=alt.Y('Local Purchasing Power Index:Q', title='Local Purchasing Power
    color=alt.Color('Country:N', legend=None),  # Optional: Color points by
    tooltip=['Country', 'Cost of Living Index', 'Local Purchasing Power Inde
).properties(
    title='Cost of Living Index vs Local Purchasing Power Index',
    width=600,  # Adjust width as needed
    height=400  # Adjust height as needed
).interactive()  # Enable zoom and pan

# Display the plot
scatter_plot
```

Out[15]:



This scatter plot shows the relationship between the cost of living plus rent index and the local purchasing power index. A negative correlation can be observed: as the cost of living and rent increase, the local purchasing power tends to decrease. This demonstrates economic stress in areas with high living expenses.

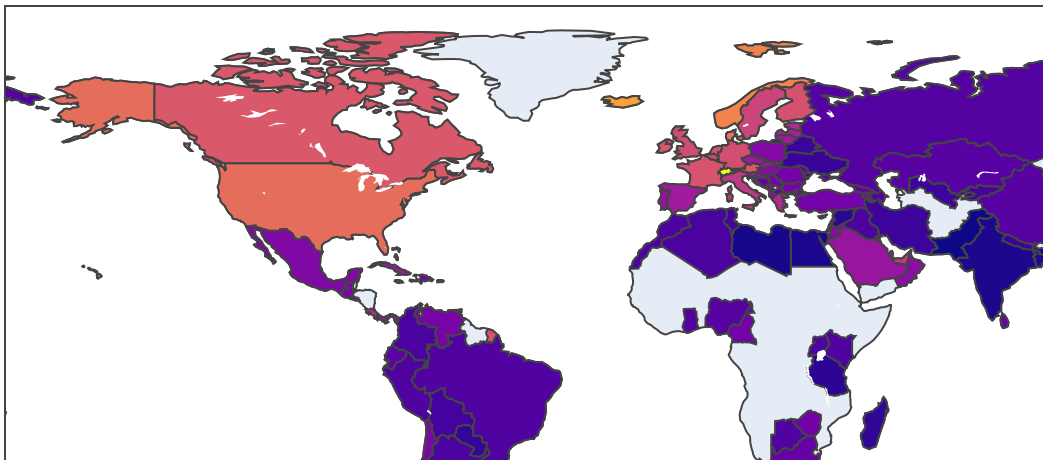
In [16]:
```
# Correlation matrix and heatmap
correlation_matrix = df.iloc[:, 2:].corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Indices')
plt.show()

```



The heatmap visualizes correlations between all numerical indices in the dataset:

- A strong positive correlation between Cost of Living Index and Rent Index highlights rent as a significant factor driving living costs.
- Weak or negative correlations between Local Purchasing Power Index and other indices suggest areas with higher living costs often have diminished purchasing power.

In [17]:
```
#Geographic Analysis
import plotly.express as px

# Map visualization for Cost of Living Index
fig = px.choropleth(df,
                    locations="Country",
                    locationmode="country names",
                    color="Cost of Living Index",
                    title="Cost of Living Index by Country",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()

```



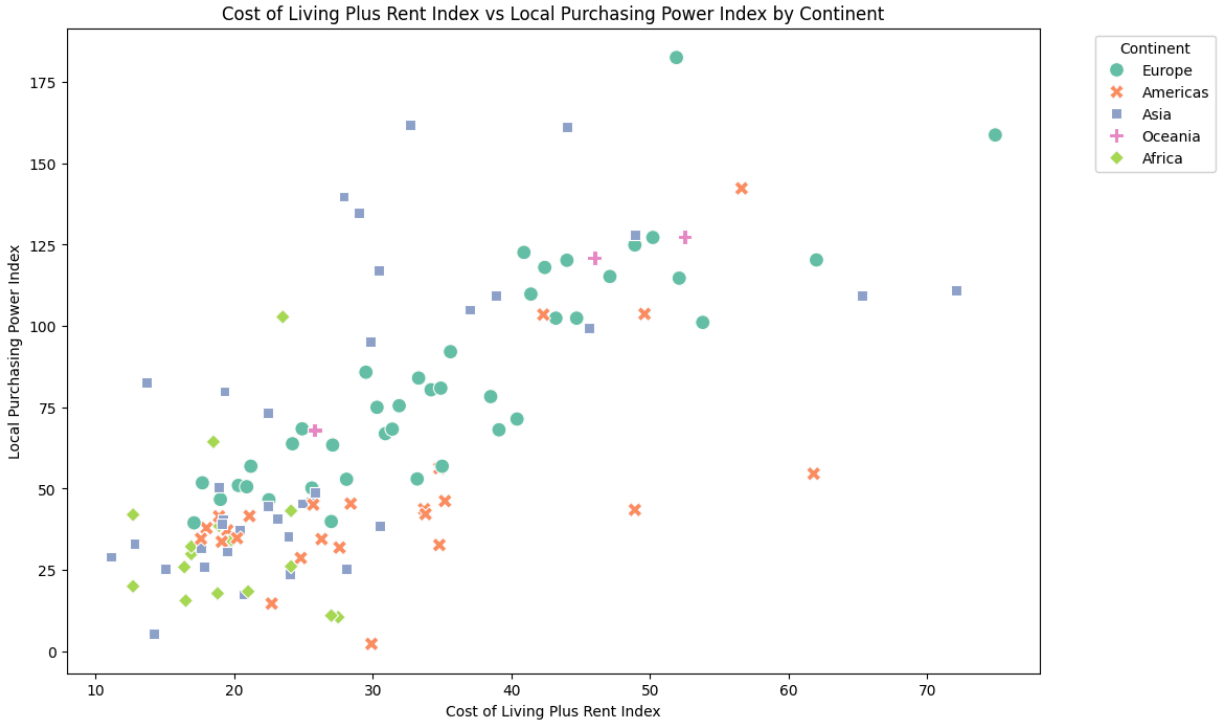
This map illustrates the Cost of Living Index across countries, using a gradient from blue (low cost of living) to yellow (high cost of living). Wealthier regions like Western Europe, North America, and Oceania stand out with lighter shades, reflecting higher living costs, while South Asia and Africa are predominantly darker, indicating lower costs. Countries such as Switzerland, Iceland, and Australia are among the most expensive, whereas nations like India, Pakistan, and many in Africa offer significantly lower living expenses. The map provides a clear visual contrast of global economic disparities, highlighting how living costs vary dramatically based on regional and economic factors.

In [18]:

```
# Mapping countries to their respective continents
continent_map = {
    'Switzerland': 'Europe', 'Bahamas': 'Americas', 'Iceland': 'Europe', 'Si
    'Barbados': 'Americas', 'Norway': 'Europe', 'Denmark': 'Europe', 'Hong K
    'United States': 'Americas', 'Australia': 'Oceania', 'Austria': 'Europe'
    'New Zealand': 'Oceania', 'Ireland': 'Europe', 'France': 'Europe', 'Puer
    'Finland': 'Europe', 'Netherlands': 'Europe', 'Israel': 'Asia', 'Luxembo
    'Germany': 'Europe', 'United Kingdom': 'Europe', 'Belgium': 'Europe', 'S
    'Sweden': 'Europe', 'Italy': 'Europe', 'United Arab Emirates': 'Asia', '
    'Uruguay': 'Americas', 'Jamaica': 'Americas', 'Malta': 'Europe', 'Trinid
    'Costa Rica': 'Americas', 'Bahrain': 'Asia', 'Greece': 'Europe', 'Estoni
    'Qatar': 'Asia', 'Slovenia': 'Europe', 'Latvia': 'Europe', 'Spain': 'Eur
    'Slovakia': 'Europe', 'Cuba': 'Americas', 'Czech Republic': 'Europe', 'P
    'Japan': 'Asia', 'Croatia': 'Europe', 'Saudi Arabia': 'Asia', 'Taiwan':
    'Portugal': 'Europe', 'Oman': 'Asia', 'Kuwait': 'Asia', 'Albania': 'Euro
    'Hungary': 'Europe', 'Palestine': 'Asia', 'Jordan': 'Asia', 'Armenia': '
    'Mexico': 'Americas', 'El Salvador': 'Americas', 'Montenegro': 'Europe',
    'Guatemala': 'Americas', 'Venezuela': 'Americas', 'Bulgaria': 'Europe',
    'Serbia': 'Europe', 'Romania': 'Europe', 'Turkey': 'Asia', 'Cambodia': '
    'Zimbabwe': 'Africa', 'Mauritius': 'Africa', 'Fiji': 'Oceania', 'Bosnia
    'Sri Lanka': 'Asia', 'South Africa': 'Africa', 'Thailand': 'Asia', 'Mold
    'North Macedonia': 'Europe', 'Ecuador': 'Americas', 'Kazakhstan': 'Asia'
    'Nigeria': 'Africa', 'Azerbaijan': 'Asia', 'Philippines': 'Asia', 'Russi
    'Brazil': 'Americas', 'Kenya': 'Africa', 'Botswana': 'Africa', 'Malaysia
    'Morocco': 'Africa', 'Kosovo (Disputed Territory)': 'Europe', 'Argentina
    'Iraq': 'Asia', 'Uganda': 'Africa', 'Algeria': 'Africa', 'Colombia': 'Am
    'Tunisia': 'Africa', 'Bolivia': 'Americas', 'Kyrgyzstan': 'Asia', 'Indon
    'Uzbekistan': 'Asia', 'Belarus': 'Europe', 'Ukraine': 'Europe', 'Nepal':
    'Madagascar': 'Africa', 'Syria': 'Asia', 'Tanzania': 'Africa', 'Banglade
    'Egypt': 'Africa', 'Libya': 'Africa', 'Pakistan': 'Asia'
}

# Add continent information to the dataset
df['Continent'] = df['Country'].map(continent_map)

# Check for unmapped countries (if any)
unmapped_countries = df[df['Continent'].isna()]
print("Unmapped countries:", unmapped_countries['Country'].tolist())

# Scatter plot grouped by continents
plt.figure(figsize=(12, 8))
sns.scatterplot(data=df,
                x="Cost of Living Plus Rent Index",
                y="Local Purchasing Power Index",
                hue="Continent",
                style="Continent",
                palette="Set2",
                s=100)  # Marker size
plt.title("Cost of Living Plus Rent Index vs Local Purchasing Power Index by
plt.xlabel("Cost of Living Plus Rent Index")
plt.ylabel("Local Purchasing Power Index")
plt.legend(title="Continent", bbox_to_anchor=(1.05, 1), loc="upper left")
plt.show()
```

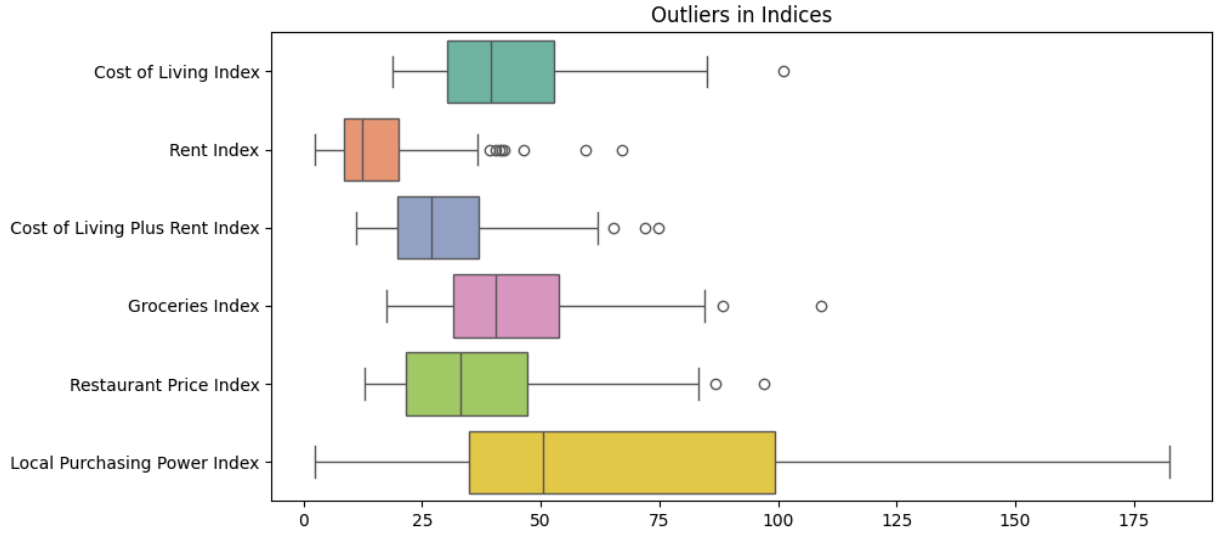
54

```
Unmapped countries: []
```



This scatter plot shows the relationship between the Cost of Living Plus Rent Index (x-axis) and the Local Purchasing Power Index (y-axis), categorized by continents. European countries (teal circles) dominate the higher purchasing power spectrum despite varying costs of living, while African nations (green diamonds) cluster in the lower ranges of both indices, reflecting lower costs and limited purchasing power. The Americas (orange crosses) display a broader spread, highlighting economic diversity within the region. Asian countries (blue squares) span from low to mid-range costs, with a few outliers showing higher purchasing power. Oceania (pink plus signs) has moderate representation, balancing living costs and purchasing power. This visualization emphasizes the economic disparities across continents and the varying relationship between costs and purchasing power.

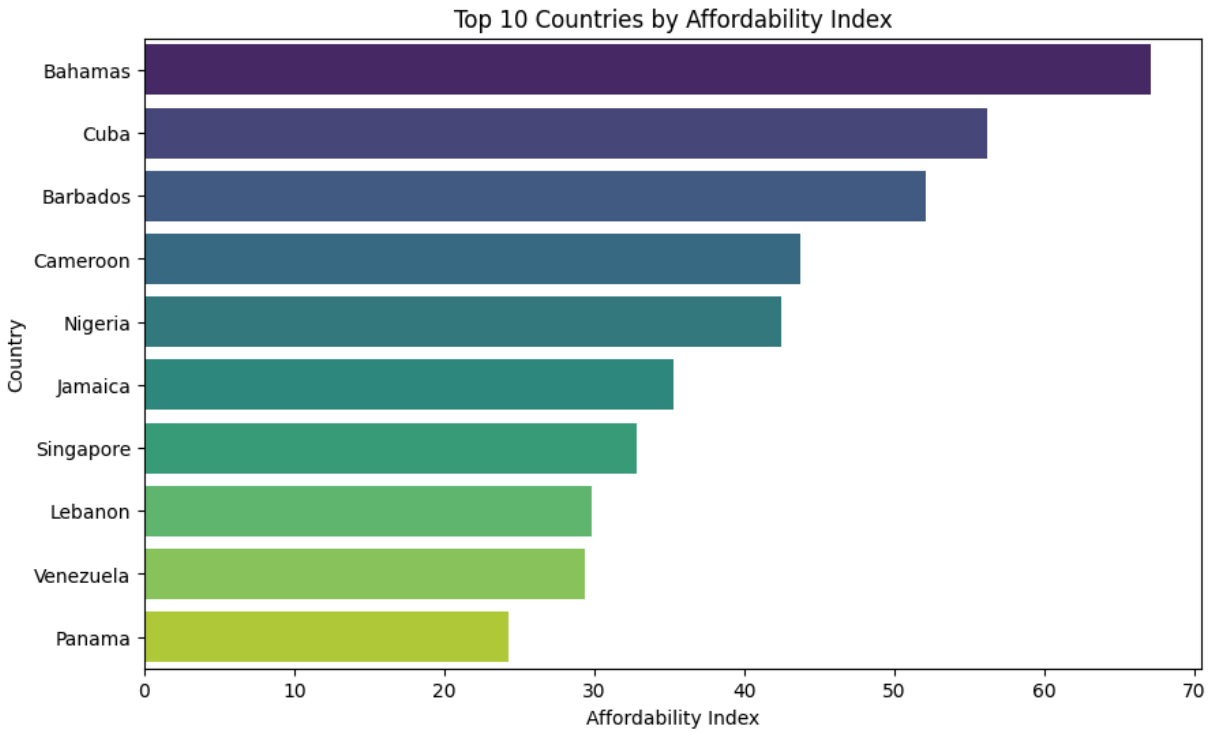
In [19]:
```
# Box plot for outlier detection
plt.figure(figsize=(10, 5))
sns.boxplot(data=df.iloc[:, 2:], orient="h", palette="Set2")
plt.title("Outliers in Indices")
plt.show()

```



This boxplot visualizes the distribution and outliers across various indices, including the Cost of Living Index, Rent Index, Cost of Living Plus Rent Index, Groceries Index, Restaurant Price Index, and Local Purchasing Power Index. The Cost of Living Index shows a compact distribution, with a few countries standing out as expensive outliers. Similarly, the Rent Index reveals that while rent costs are generally moderate across countries, there are several notable high-value outliers where rents are exceptionally high. The Cost of Living Plus Rent Index follows a similar pattern, indicating that rent significantly influences overall living costs. Both the Groceries Index and Restaurant Price Index exhibit moderate variability, with outliers suggesting regional disparities in food and dining costs. The Local Purchasing Power Index has the widest range, with significant outliers at the higher end, highlighting that while most countries have moderate or low purchasing power, a few enjoy exceptionally high purchasing power. Lastly, the cluster boxplot suggests a segmentation of countries into distinct groups, with no significant outliers within clusters. Overall, this visualization underscores the economic disparities globally, with outliers reflecting countries experiencing extreme costs or purchasing power.

In [20]:
```
# Creating a composite affordability index
df["Affordability Index"] = df["Cost of Living Index"] + df["Rent Index"] -

# Top 10 countries by Affordability Index
top_affordable_countries = df.nlargest(10, "Affordability Index")

# Bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=top_affordable_countries, x="Affordability Index", y="Count
plt.title("Top 10 Countries by Affordability Index")
plt.show()

```

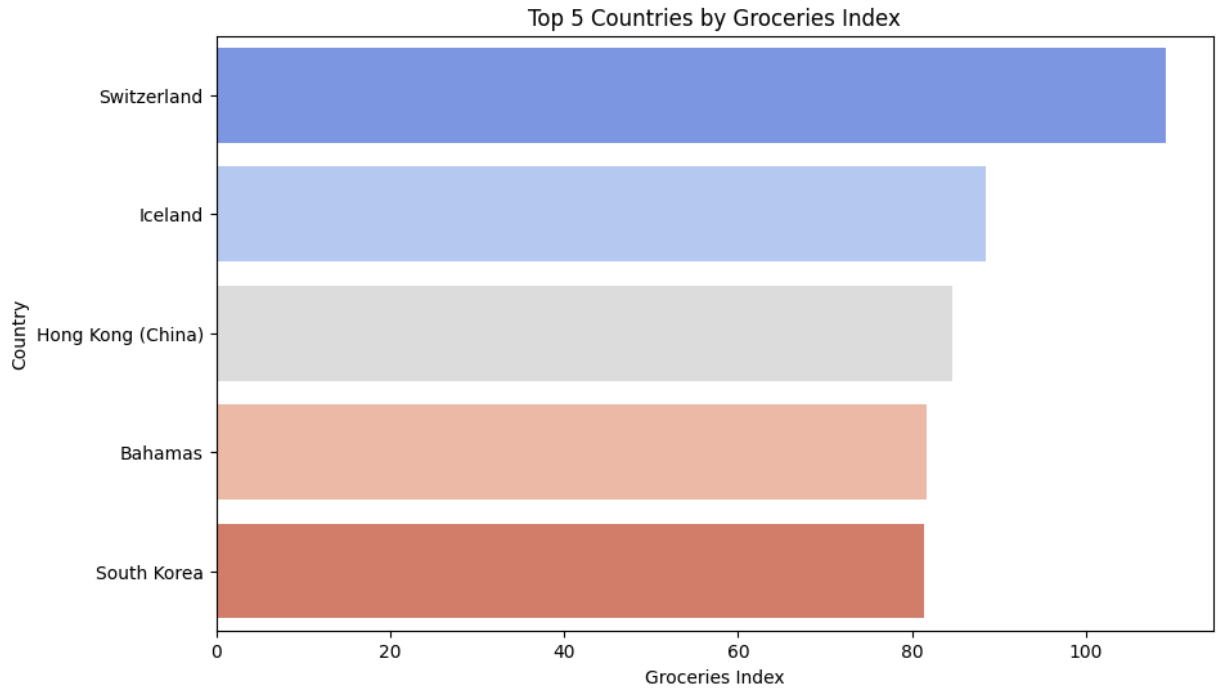
```
C:\Users\Sharon\AppData\Local\Temp\ipykernel_19060\4208509857.py:9: FutureWarni
ng:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```



This bar chart displays the Top 10 Countries by Affordability Index, with the Bahamas ranking the highest, followed by Cuba and Barbados. The affordability index, likely a composite measure of cost of living and purchasing power, highlights countries where expenses are balanced with income levels, making them more financially accessible. The chart showcases a mix of Caribbean, African, and other nations such as Singapore and Lebanon, which reflect a blend of lower living costs and relatively strong purchasing power. This visualization provides a clear insight into which countries offer the most affordable living conditions globally.

In [21]:
```
# Top 5 countries for Groceries Index
top_groceries_countries = df.nlargest(5, "Groceries Index")

# Bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=top_groceries_countries, x="Groceries Index", y="Country",
plt.title("Top 5 Countries by Groceries Index")
plt.show()

```

```
C:\Users\Sharon\AppData\Local\Temp\ipykernel_19060\743820248.py:6: FutureWarnin
g:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.
```

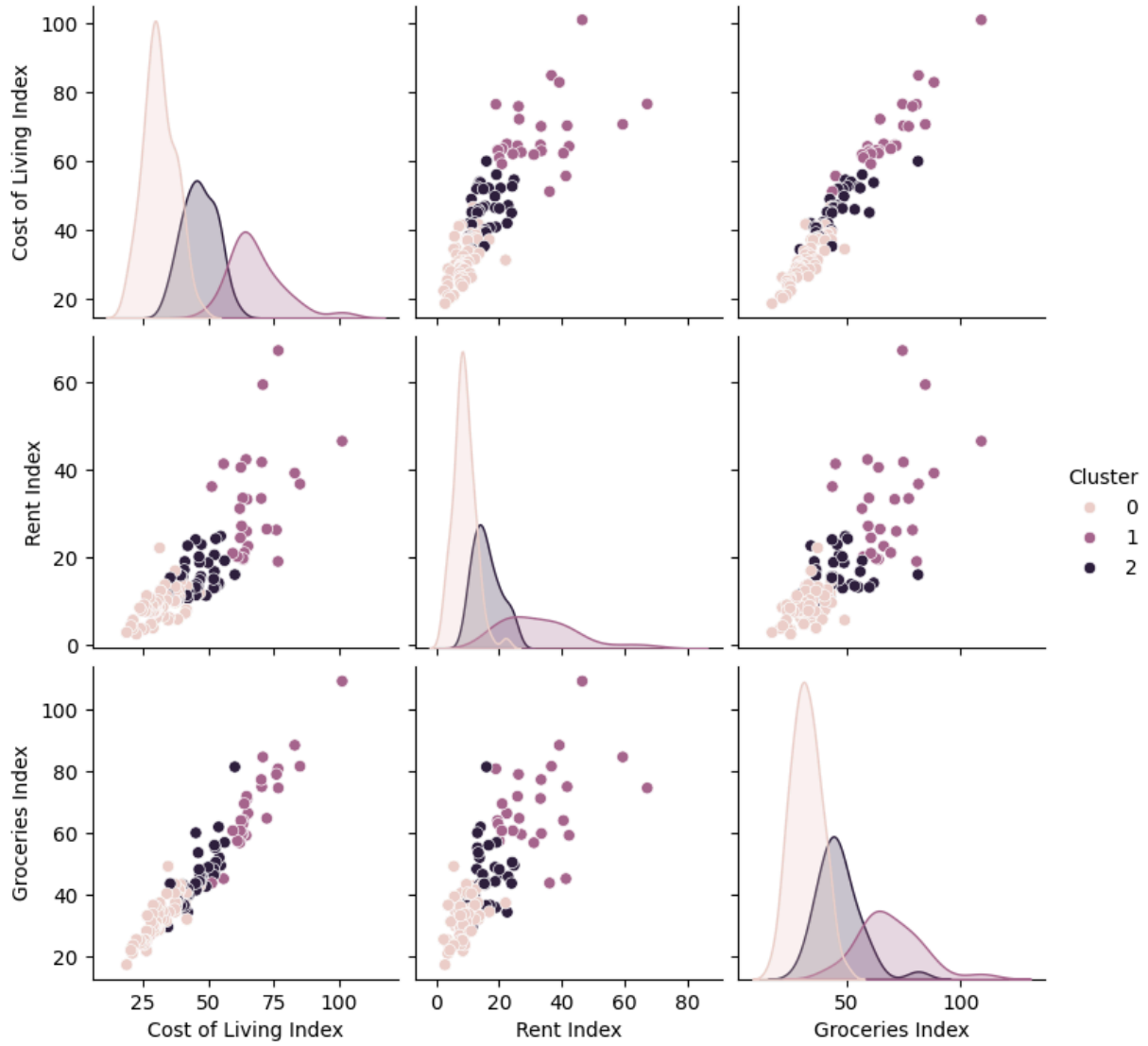


This bar chart highlights the Top 5 Countries by Groceries Index, with Switzerland leading as the most expensive country for groceries, followed by Iceland, Hong Kong (China), the Bahamas, and South Korea. The Groceries Index measures the cost of essential food items and household supplies, reflecting economic and logistical factors such as production costs, import dependency, and purchasing power. Switzerland and Iceland are known for their high living costs, which extend to grocery prices, while regions like Hong Kong face higher prices due to space constraints and reliance on imports. This visualization emphasizes the substantial variation in grocery costs worldwide, with the highlighted countries being some of the most expensive for basic goods.

In [22]:

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import matplotlib.pyplot as plt

# Select only numeric columns
numeric_indices = df.select_dtypes(include='number')

# Scale the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_indices)

# Applying KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# Add cluster labels to the original dataframe
df['Cluster'] = clusters

# Visualize the clusters using a pairplot
sns.pairplot(df, hue='Cluster', vars=["Cost of Living Index", "Rent Index",
plt.show()

```

```
C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: Fu
tureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the val
ue of `n_init` explicitly to suppress the warning

C:\Users\Sharon\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: Us
erWarning:

KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment vari
able OMP_NUM_THREADS=1.
```



The scatter plots show strong positive correlations between all three indices, indicating that countries with higher living costs tend to have higher rent and grocery costs as well. The density plots on the diagonal highlight the distribution of each index within the clusters. Cluster 0 (light pink) represents countries with lower index values, likely reflecting affordable regions. Cluster 2 (dark purple) contains countries with the highest costs across all indices, reflecting expensive regions like Switzerland or Iceland. Cluster 1 (purple) lies in the middle, representing countries with moderate costs. This plot effectively reveals groupings of countries based on living costs and provides insights into economic segmentation globally.

In [ ]:
```
1
```