

```
In [1]: import pandas as pd
```

```
In [2]: import plotly.express as px
```

```
In [3]: import plotly.io as pio
```

```
In [4]: import plotly.graph_objects as go
```

```
In [5]: pio.templates.default = "plotly_white"
```

```
In [6]: data = pd.read_csv("RFM Analysis Dataset.csv")
```

```
In [7]: print(data.head())
```

| | CustomerID | PurchaseDate | TransactionAmount | ProductInformation | OrderID \ |
|---|------------|--------------|-------------------|--------------------|-----------|
| 0 | 8814 | 11-04-2023 | 943.31 | Product C | 890075 |
| 1 | 2188 | 11-04-2023 | 463.70 | Product A | 176819 |
| 2 | 4608 | 11-04-2023 | 80.28 | Product A | 340062 |
| 3 | 2559 | 11-04-2023 | 221.29 | Product A | 239145 |
| 4 | 9482 | 11-04-2023 | 739.56 | Product A | 194545 |

| | Location |
|---|----------|
| 0 | Tokyo |
| 1 | London |
| 2 | New York |
| 3 | London |
| 4 | Paris |

```
In [8]: # Calculating the Recency , Frequency and Monetary values of the customers segmentation
```

```
In [9]: from datetime import datetime
```

```
In [10]: #Convert 'PurchaseDate' to datetime
```

```
In [86]: data['PurchaseDate'] = pd.to_datetime(data['PurchaseDate'])
```

```
In [12]: #Calculate Recency
```

```
In [13]: data['Recency'] = (datetime.now().date() - data['PurchaseDate'].dt.date).dt.days
```

```
In [14]: #Calculate Frequency
```

```
In [15]: frequency_data = data.groupby('CustomerID')['OrderID'].count().reset_index()
```

```
In [16]: frequency_data.rename(columns={'OrderID': 'Frequency'}, inplace=True)
```

```
In [17]: data = data.merge(frequency_data, on='CustomerID', how='left')
```

```
In [18]: #Calculate Monetary Value
```

```
In [19]: monetary_data = data.groupby('CustomerID')['TransactionAmount'].sum().reset_index()
```

```
In [20]: monetary_data.rename(columns={'TransactionAmount': 'MonetaryValue'}, inplace=True)
```

```
In [21]: data = data.merge(monetary_data, on='CustomerID', how='left')
```

```
In [22]: print(data.head())
```

| | CustomerID | PurchaseDate | TransactionAmount | ProductInformation | OrderID \ |
|---|------------|--------------|-------------------|--------------------|-----------|
| 0 | 8814 | 2023-11-04 | 943.31 | Product C | 890075 |
| 1 | 2188 | 2023-11-04 | 463.70 | Product A | 176819 |
| 2 | 4608 | 2023-11-04 | 80.28 | Product A | 340062 |
| 3 | 2559 | 2023-11-04 | 221.29 | Product A | 239145 |
| 4 | 9482 | 2023-11-04 | 739.56 | Product A | 194545 |

| | Location | Recency | Frequency | MonetaryValue |
|---|----------|---------|-----------|---------------|
| 0 | Tokyo | 25 | 1 | 943.31 |
| 1 | London | 25 | 1 | 463.70 |
| 2 | New York | 25 | 1 | 80.28 |
| 3 | London | 25 | 1 | 221.29 |
| 4 | Paris | 25 | 1 | 739.56 |

```
In [23]: #Calculate RFM Scores
```

```
In [24]: #defining score criteria for each RFM Value
```

```
In [25]: recency_scores = [5, 4, 3, 2, 1] #higher score for lower recency
```

```
In [26]: frequency_scores = [1, 2, 3, 4, 5] #higher score higher frequency
```

```
In [27]: monetary_scores = [1, 2, 3, 4, 5] #higher score for higher monetary value
```

```
In [28]: #RFM Scores
```

```
In [29]: data['RecencyScore'] = pd.cut(data['Recency'], bins=5, labels=recency_scores)
```

```
In [30]: data['FrequencyScore'] = pd.cut(data['Frequency'], bins=5, labels=frequency_scores)
```

```
In [31]: data['MonetaryScore'] = pd.cut(data['MonetaryValue'], bins=5, labels=monetary_scores)
```

```
In [32]: #convert rfm scores to integers
```

```
In [33]: data['RecencyScore'] = data['RecencyScore'].astype(int)
```

```
In [34]: data['FrequencyScore'] = data['FrequencyScore'].astype(int)
```

```
In [35]: data['MonetaryScore'] = data['MonetaryScore'].astype(int)
```

```
In [36]: #Calculate RFM scores by combining with the individual score
```

```
In [37]: data['RFM_score'] = data['RecencyScore'] + data['FrequencyScore'] + data['MonetaryScore']
```

```
In [38]: #create RFM segments based on the RFM score
```

```
In [39]: segment_labels = ['Low-Value', 'Mid-Value', 'High-Value']
```

```
In [40]: data['Value Segment'] = pd.qcut(data['RFM_score'], q=3, labels=segment_labels)
```

```
In [41]: print(data.head())
```

| | CustomerID | PurchaseDate | TransactionAmount | ProductInformation | OrderID | \ |
|---|------------|--------------|-------------------|--------------------|---------|---|
| 0 | 8814 | 2023-11-04 | 943.31 | Product C | 890075 | |
| 1 | 2188 | 2023-11-04 | 463.70 | Product A | 176819 | |
| 2 | 4608 | 2023-11-04 | 80.28 | Product A | 340062 | |
| 3 | 2559 | 2023-11-04 | 221.29 | Product A | 239145 | |
| 4 | 9482 | 2023-11-04 | 739.56 | Product A | 194545 | |

| | Location | Recency | Frequency | MonetaryValue | RecencyScore | FrequencyScore | \ |
|---|----------|---------|-----------|---------------|--------------|----------------|---|
| 0 | Tokyo | 25 | 1 | 943.31 | 5 | 1 | |
| 1 | London | 25 | 1 | 463.70 | 5 | 1 | |
| 2 | New York | 25 | 1 | 80.28 | 5 | 1 | |
| 3 | London | 25 | 1 | 221.29 | 5 | 1 | |
| 4 | Paris | 25 | 1 | 739.56 | 5 | 1 | |

| | MonetaryScore | RFM_score | Value | Segment |
|---|---------------|-----------|------------|---------|
| 0 | 2 | 8 | High-Value | |
| 1 | 1 | 7 | High-Value | |
| 2 | 1 | 7 | High-Value | |
| 3 | 1 | 7 | High-Value | |
| 4 | 2 | 8 | High-Value | |

```
In [42]: #RFM Segment Distribution
```

```
In [43]: segment_counts = data['Value Segment'].value_counts().reset_index()
```

```
In [44]: segment_counts.columns = ['Value Segment', 'Count']
```

```
In [45]: pastel_colors = px.colors.qualitative.Pastel
```

```
In [46]: #create bar chart
```

```
In [47]: fig_segment_dist = px.bar(segment_counts, x='Value Segment', y='Count', color='Value Segment', color_discrete_sequence= pastel_co
```

```
In [48]: #the layout
```

```
In [49]: fig_segment_dist.update_layout(xaxis_title='RFM Value Segment', yaxis_title='Count', showlegend=False)
```

RFM Value Segment Distribution



```
In [50]: #Create a new column for RFM Customer Segmentation
```

```
In [51]: data['RFM Customer Segments']= ''
```

```
In [52]: #Assign RFM Segments based on the RFM score
```

```
In [53]: data.loc[data['RFM_score'] >=9, 'RFM Customer Segments'] = 'Champions'
```

```
In [54]: data.loc[(data['RFM_score'] >= 6) & (data['RFM_score'] <9), 'RFM Customer Segments'] = 'Potential Customers'
```

```
In [55]: data.loc[(data['RFM_score'] >= 5) & (data['RFM_score'] <6), 'RFM Customer Segments'] = 'At Risk'
```

```
In [56]: data.loc[(data['RFM_score'] >= 4) & (data['RFM_score'] <5), 'RFM Customer Segments'] = 'Cannot Lose'
```

```
In [57]: data.loc[(data['RFM_score'] >= 4) & (data['RFM_score'] <3), 'RFM Customer Segments'] = 'Already Lost'
```

```
In [58]: print(data[['CustomerID', 'RFM Customer Segments']])
```

| | CustomerID | RFM Customer Segments |
|-----|------------|-----------------------|
| 0 | 8814 | Potential Customers |
| 1 | 2188 | Potential Customers |
| 2 | 4608 | Potential Customers |
| 3 | 2559 | Potential Customers |
| 4 | 9482 | Potential Customers |
| .. | ... | ... |
| 995 | 2970 | Potential Customers |
| 996 | 6669 | Potential Customers |
| 997 | 8836 | Potential Customers |
| 998 | 1440 | Potential Customers |
| 999 | 4759 | Potential Customers |

[1000 rows x 2 columns]

```
In [59]: #RFM Analysis undereach value segments
```

```
In [60]: segment_product_counts = data.groupby(['Value Segment', 'RFM Customer Segments']).size().reset_index(name='Count')
```

```
In [61]: segment_product_counts = segment_product_counts.sort_values('Count', ascending=False)
```

```
In [62]: pip install --upgrade plotly
```

Requirement already satisfied: plotly in c:\users\kalyani nayak\anaconda3\lib\site-packages (5.18.0)

Requirement already satisfied: tenacity>=6.2.0 in c:\users\kalyani nayak\anaconda3\lib\site-packages (from plotly) (8.0.1)

Requirement already satisfied: packaging in c:\users\kalyani nayak\anaconda3\lib\site-packages (from plotly) (21.3)

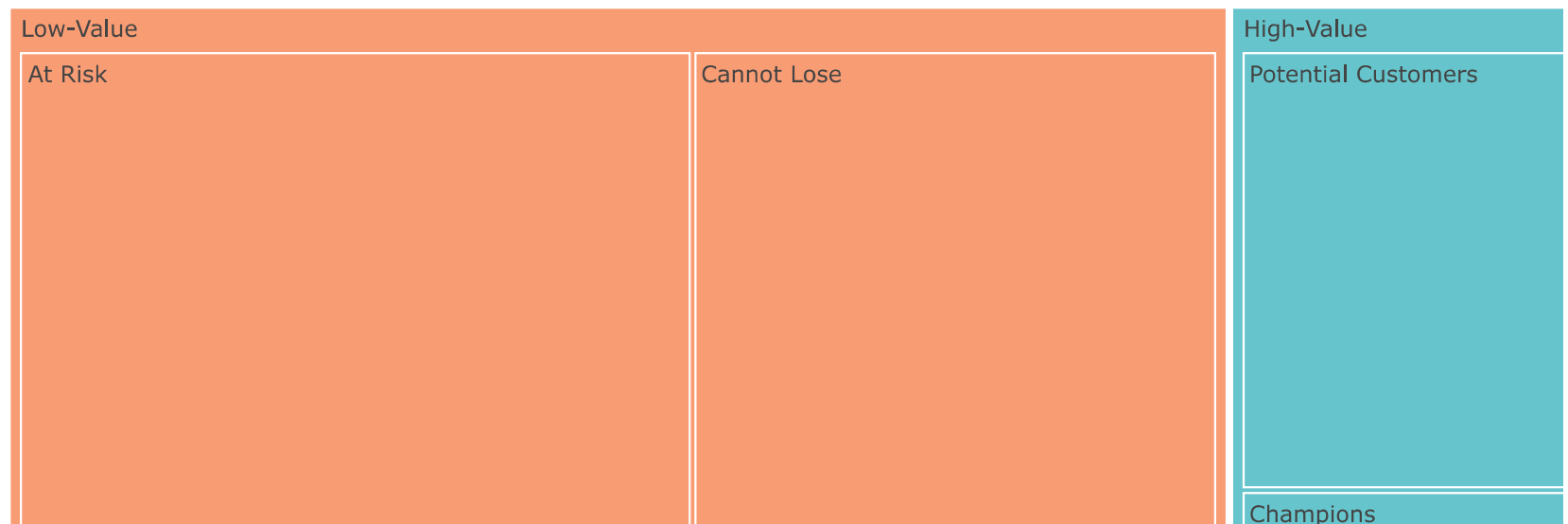
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\kalyani nayak\anaconda3\lib\site-packages (from packaging->plotly) (3.0.4)

Note: you may need to restart the kernel to use updated packages.

```
In [63]: fig_treemap_segment_product = px.treemap(segment_product_counts,
                                                path=['Value Segment', 'RFM Customer Segments'],
                                                values='Count',
                                                color='Value Segment', color_discrete_sequence=px.colors.qualitative.Pastel,
                                                title='RFM Customer Segments by Value')
```

```
In [64]: fig_treemap_segment_product.show()
```

RFM Customer Segments by Value



```
In [65]: #number of customers in all the segments
```

```
In [66]: import plotly.colors
```

```
In [67]: pastel_colors = plotly.colors.qualitative.Pastel
```

```
In [68]: segment_counts = data['RFM Customer Segments'].value_counts()
```

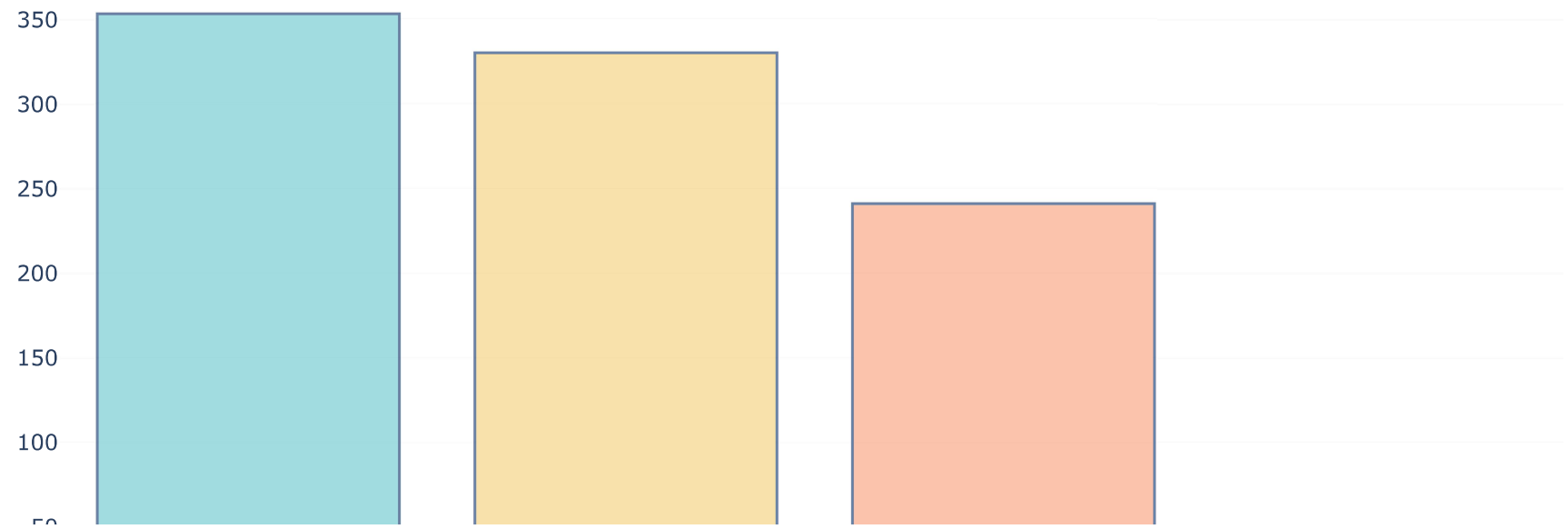
```
In [69]: # bar chart to compare segment colors
```

```
In [70]: fig = go.Figure(data=[go.Bar(x=segment_counts.index,y=segment_counts.values,marker=dict(color=pastel_colors))])
```

```
In [71]: #Set the color of the champions segment as a different color
```

```
In [72]: champions_color='rgb(158, 202, 225)'
```

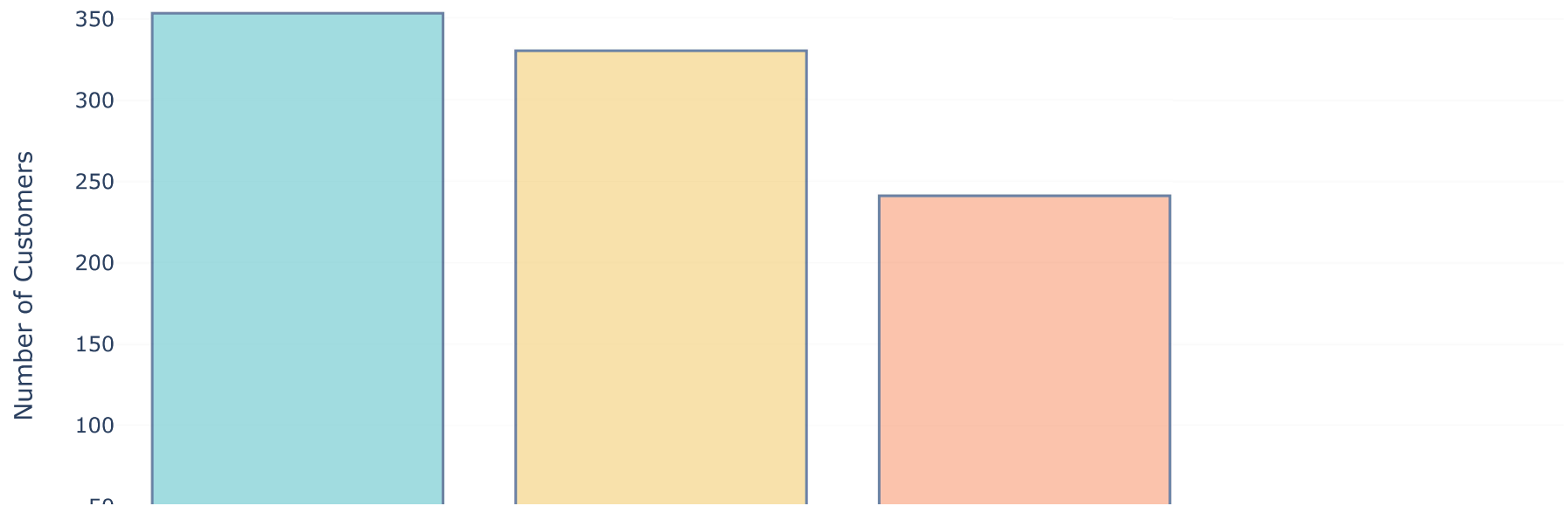
```
In [73]: fig.update_traces(marker_color=[champions_color if segment == 'Champions' else pastel_colors[i] for i, segment in enumerate(segment_counts.index)])
```

```
In [74]: #Update the Layout
```

```
In [75]: fig.update_layout(title='Comparison of RFM Segments',xaxis_title='RFM Segments',yaxis_title='Number of Customers',showlegend=False)
```

Comparison of RFM Segments



```
In [76]: #calculate the average Recency, Frequency, and Monetary scores for each segment
```

```
In [87]: segment_scores = data.groupby('RFM Customer Segments')['RecencyScore', 'FrequencyScore', 'MonetaryScore'].mean().reset_index()
```

C:\Users\KALYANI NAYAK\AppData\Local\Temp\ipykernel_9276\3036139649.py:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

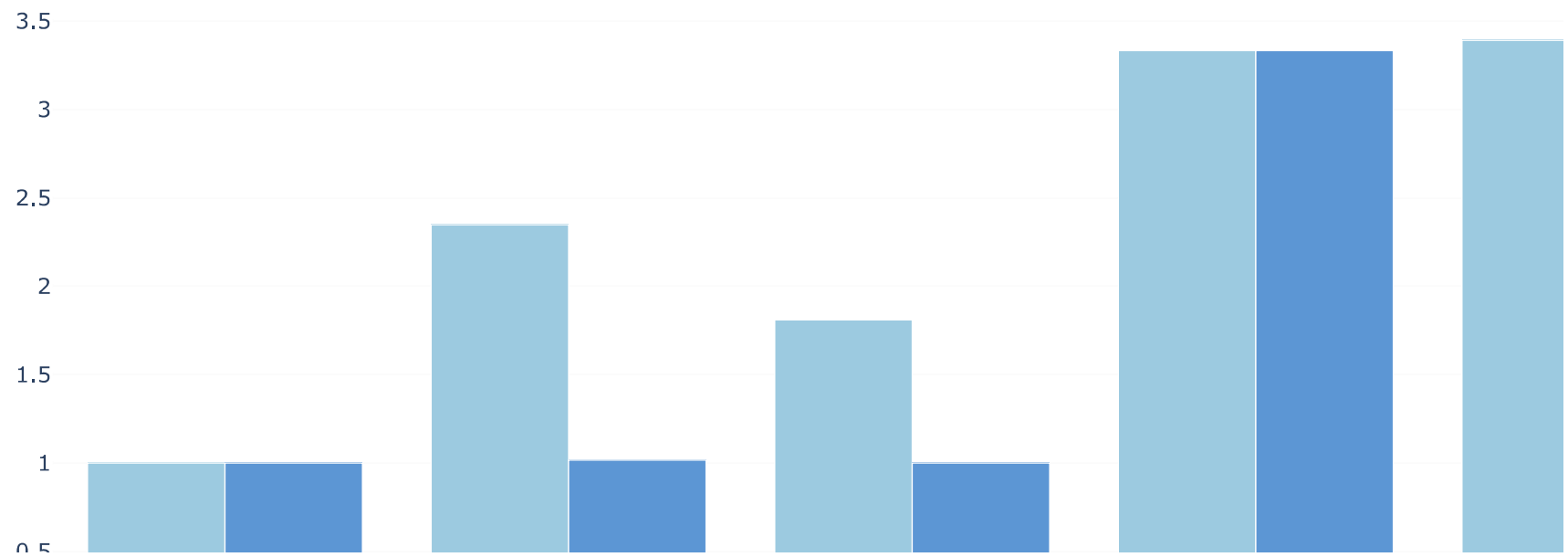
```
#grouped bar chart to compare segment scores
```

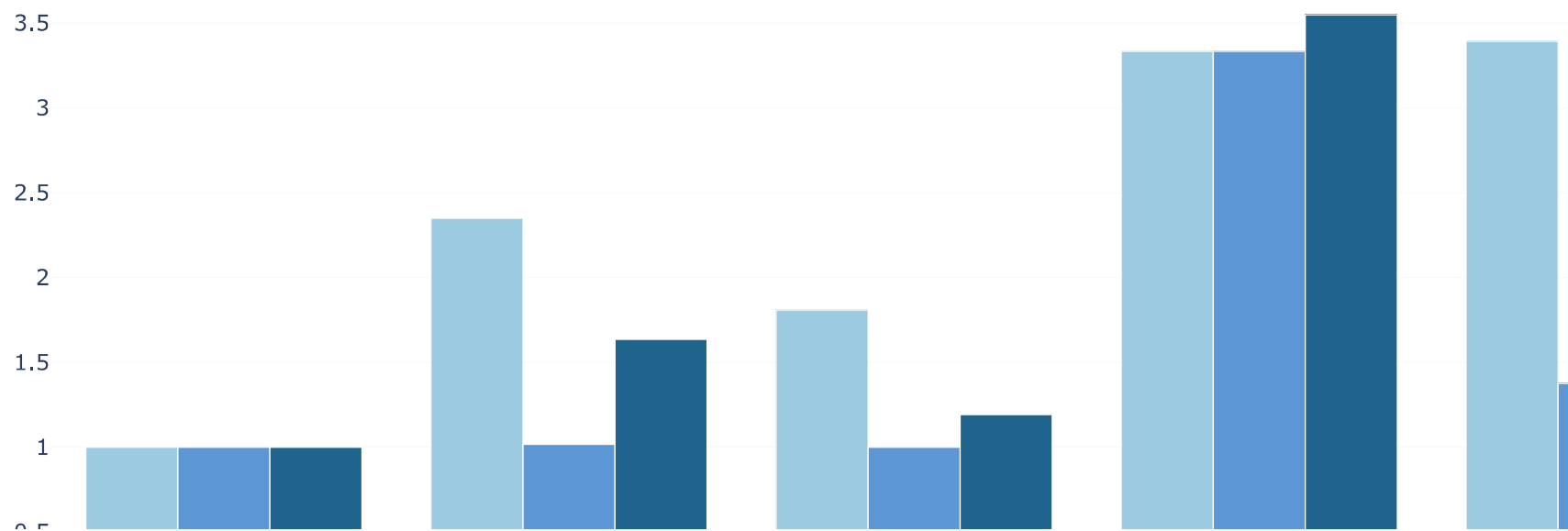
```
fig = go.Figure()
```

```
#Add bars for Recency score, Frequency score and Monetary score
```

```
fig.add_trace(go.Bar(x=segment_scores['RFM Customer Segments'],
                    y=segment_scores['RecencyScore'],
                    name='Recency Score',
                    marker_color='rgb(156,202,224)'))
```

[illegible]

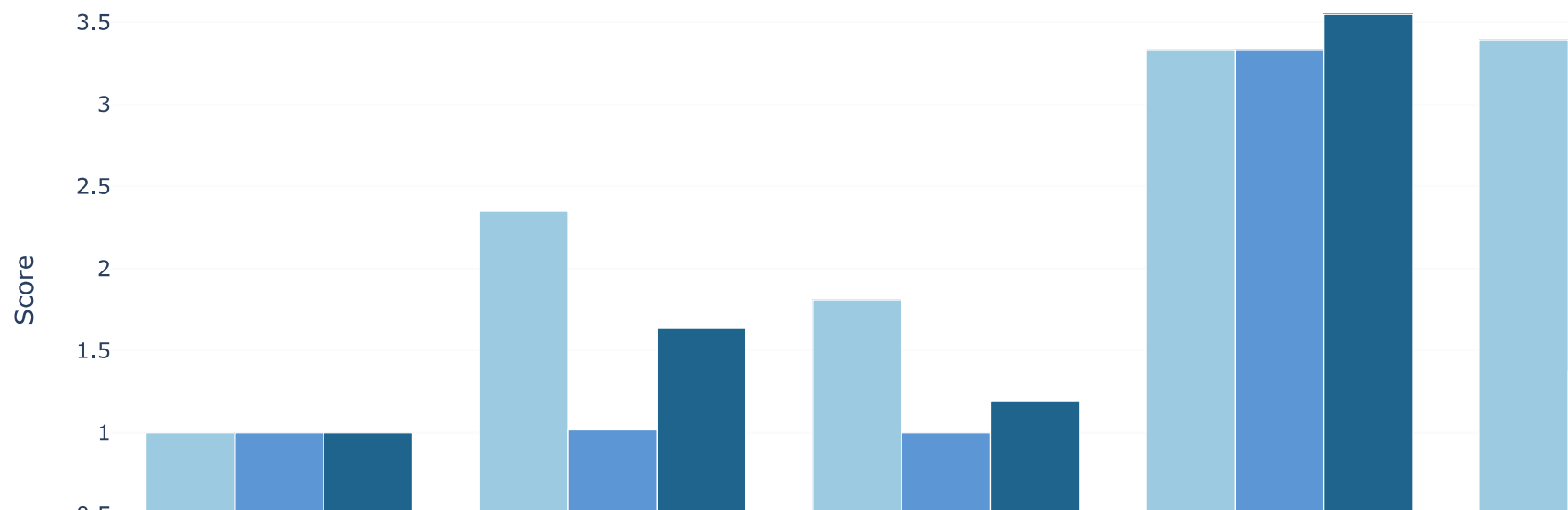
[illegible]



In [84]: *#update the layout*

```
In [85]: fig.update_layout(  
    title='Comparison of RFM Segments based on Recency, Frequency, and Monetary Scores',  
    xaxis_title='RFM Segments',  
    yaxis_title='Score',  
    barmode='group', showlegend=True)
```

Comparison of RFM Segments based on Recency, Frequency, and Monetary Scores



In []: