```
import pandas as pd
import plotly.express as px
import plotly.io as pio
import plotly.graph_objects as go
pio.templates.default ="plotly_white"
```

```
data=pd.read_csv("bounce-rate.csv")
print(data.head())
```

```
      Client ID  Sessions Avg. Session Duration Bounce Rate
0  5.778476e+08       367              00:01:35      87.19%
1  1.583822e+09       260              00:01:04      29.62%
2  1.030699e+09       237              00:00:02      99.16%
3  1.025030e+09       226              00:02:22      25.66%
4  1.469968e+09       216              00:01:23      46.76%
```

```
print(data.isnull().sum())
```

```
Client ID                0
Sessions                 0
Avg. Session Duration    0
Bounce Rate              0
dtype: int64
```

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Client ID              999 non-null    float64
 1   Sessions               999 non-null    int64
 2   Avg. Session Duration  999 non-null    object
 3   Bounce Rate            999 non-null    object
dtypes: float64(1), int64(1), object(2)
memory usage: 31.3+ KB
None
```

```
print(data.columns)
```

```
Index(['Client ID', 'Sessions', 'Avg. Session Duration', 'Bounce Rate'], dtype='object')
```

```
data['Avg. Session Duration'] = data['Avg. Session Duration'].astype(str).str[1:]
# Filter out rows where 'Avg. Session Duration' is just a dot ('.')
data = data[data['Avg. Session Duration'] != '.']
data['Avg. Session Duration'] = pd.to_timedelta(data['Avg. Session Duration'])
data['Avg. Session Duration'] = data['Avg. Session Duration'] / pd.Timedelta(minutes=1)
```

```
data['Avg. Sessiom Duration'] = data['Avg. Session Duration'] / pd.Timedelta(minutes=1)
data['Bounce Rate'] = data['Bounce Rate'].str.rstrip('%').astype('float')
print(data)
```

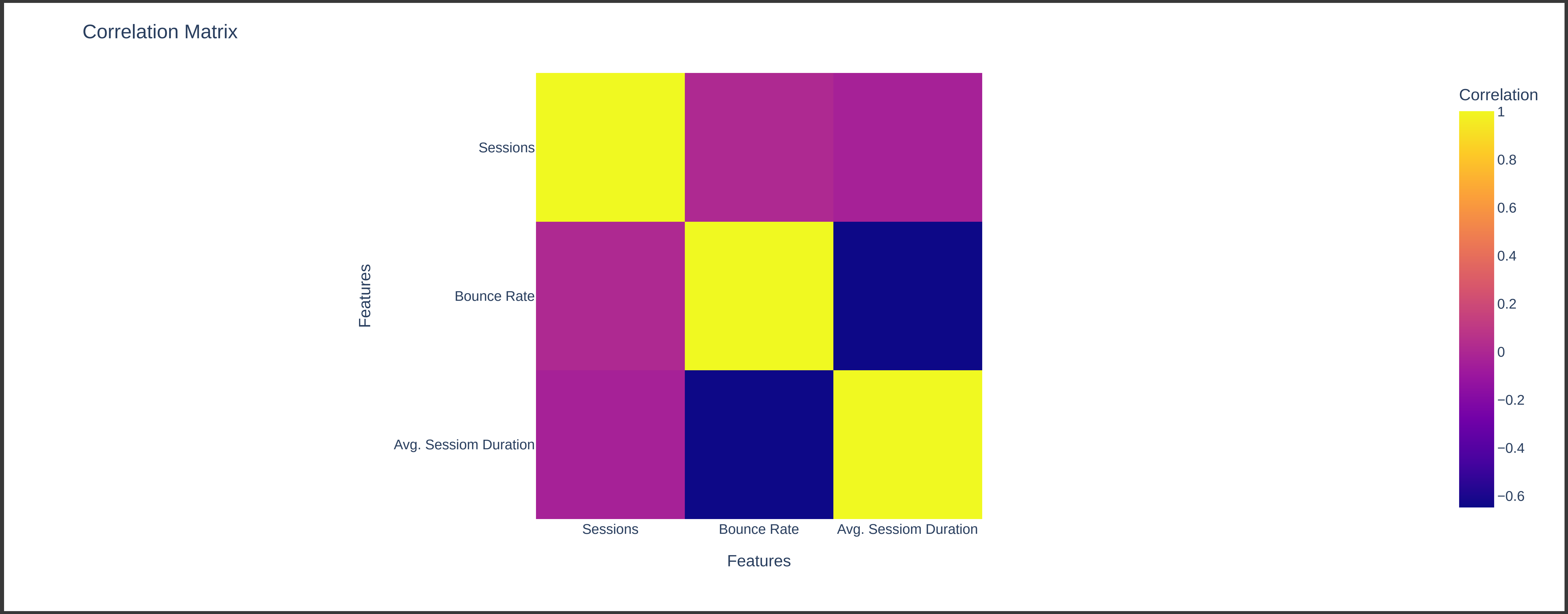
```
         Client ID  Sessions Avg. Session Duration  Bounce Rate  \
0    5.778476e+08       367       0 days 00:01:35        87.19
1    1.583822e+09       260       0 days 00:01:04        29.62
2    1.030699e+09       237       0 days 00:00:02        99.16
3    1.025030e+09       226       0 days 00:02:22        25.66
4    1.469968e+09       216       0 days 00:01:23        46.76
..            ...       ...                   ...          ...
994  1.049263e+09        17       0 days 00:07:44        41.18
995  1.145806e+09        17       0 days 00:05:37        47.06
996  1.153811e+09        17       0 days 00:00:12        94.12
997  1.182133e+09        17       0 days 00:01:13        88.24
998  1.184187e+09        17       0 days 00:02:34        64.71

     Avg. Sessiom Duration
0                 1.583333
1                 1.066667
2                 0.033333
3                 2.366667
4                 1.383333
..                     ...
994               7.733333
995               5.616667
996               0.200000
997               1.216667
998               2.566667

[999 rows x 5 columns]
```

```
print(data.describe())
```

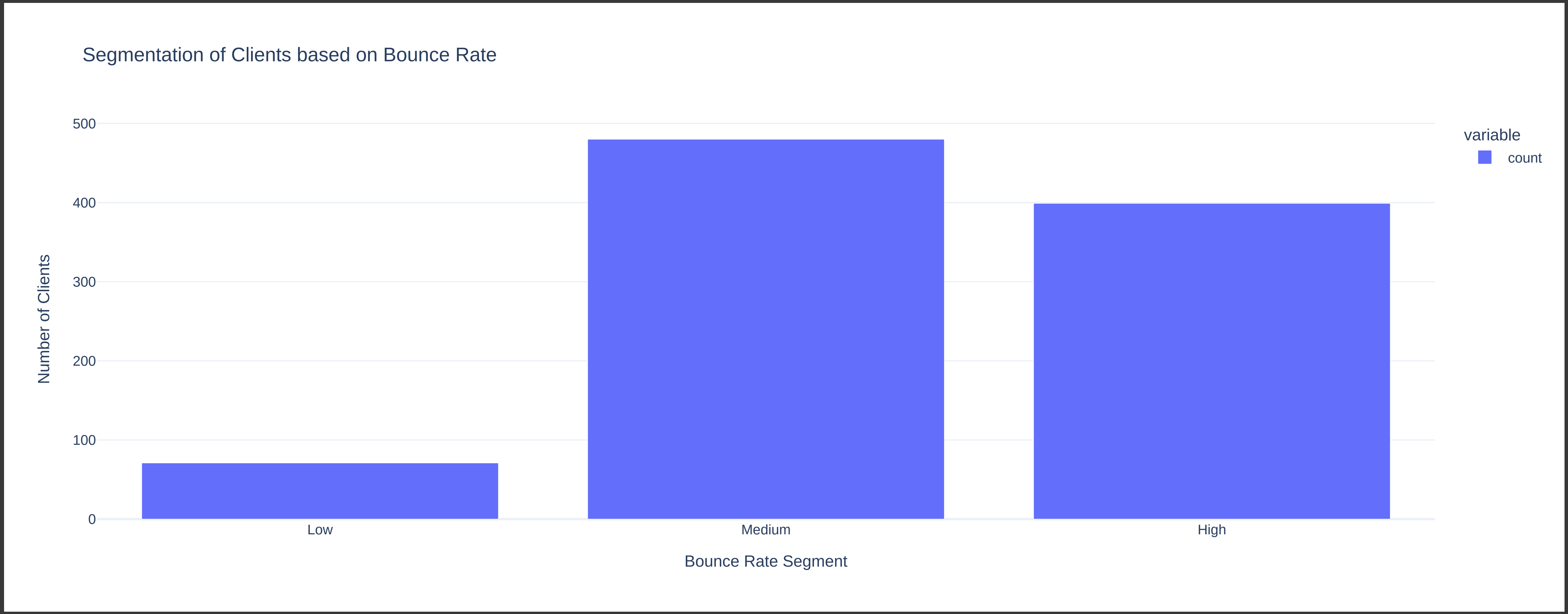
```
          Client ID    Sessions      Avg. Session Duration  Bounce Rate  \
count  9.990000e+02  999.000000                        999   999.000000
mean   1.036401e+09   32.259259  0 days 00:03:38.191191191    65.307978
std    6.151503e+08   24.658588  0 days 00:04:02.433724353    22.997270
min    1.849182e+05   17.000000            0 days 00:00:00     4.880000
25%    4.801824e+08   21.000000     0 days 00:00:53.500000    47.370000
50%    1.029507e+09   25.000000            0 days 00:02:28    66.670000
75%    1.587982e+09   35.000000            0 days 00:04:49    85.190000
max    2.063338e+09  367.000000            0 days 00:30:40   100.000000

       Avg. Sessiom Duration
count             999.000000
mean                3.636520
std                 4.040562
min                 0.000000
25%                 0.891667
50%                 2.466667
75%                 4.816667
max                30.666667
```

Double-click (or enter) to edit

```
data_without_id = data.drop(['Client ID'],axis=1)

# Calculate the correlation matrix
correlation_matrix = data_without_id.corr(numeric_only=True)

# Visualize the correlation matrix
correlation_fig = px.imshow(correlation_matrix,
                            labels=dict(x='Features',
                                        y='Features',
                                        color='Correlation'))
correlation_fig.update_layout(title='Correlation Matrix')
correlation_fig.show()
```



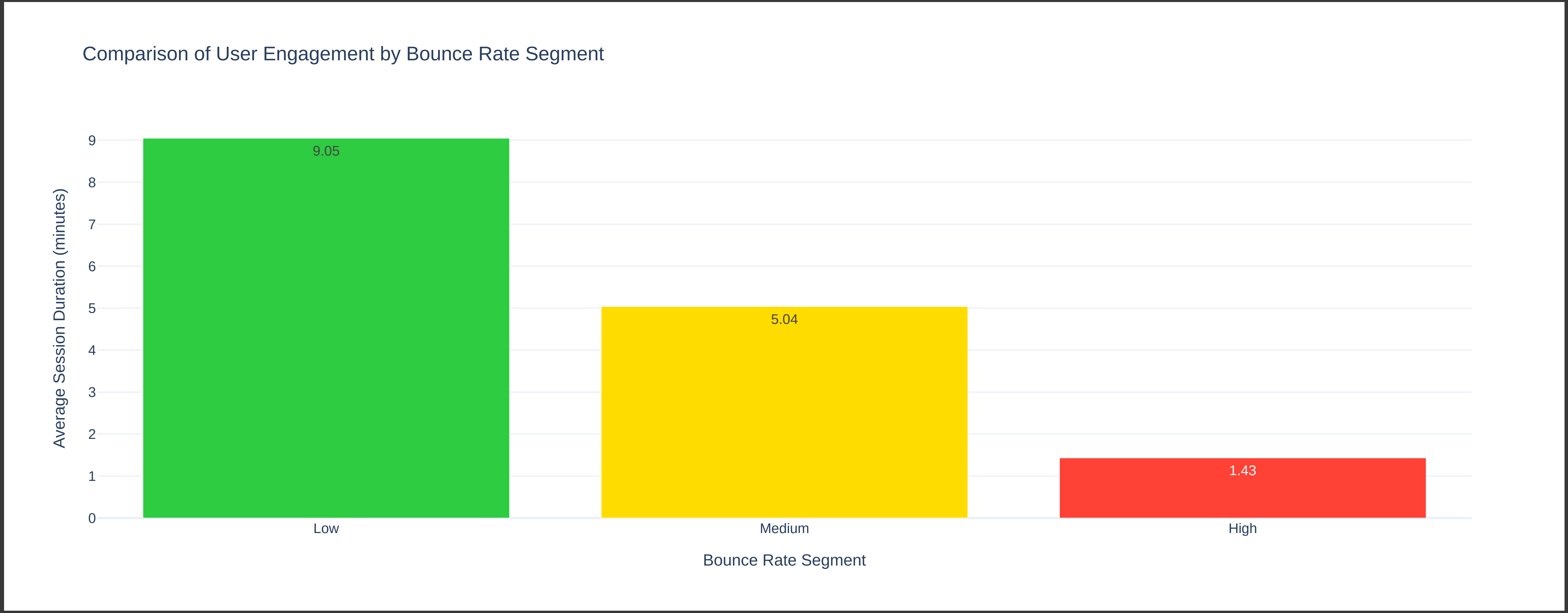
```
high_bounce_rate=70
low_bounce_rate=30

data['Bounce Rate Segment'] = pd.cut(data['Bounce Rate'], bins=[0, low_bounce_rate,
                                                              high_bounce_rate, 100],
                                   labels=['Low', 'Medium', 'High'], right=False)
segments_count = data['Bounce Rate Segment'].value_counts().sort_index()
segments_count = data['Bounce Rate Segment'].value_counts().sort_index()

segments_fig = px.bar(segments_count, labels={'index': 'Bounce Rate Segment',
                                              'value': 'Number of Clients'},
                      title='Segmentation of Clients based on Bounce Rate')

segments_fig.show()
```



```
segment_avg_duration = data.groupby('Bounce Rate Segment')['Avg. Sessiom Duration'].mean()

# Create a bar chart to compare user engagement
engagement_fig = go.Figure(data=go.Bar(
    x=segment_avg_duration.index,
    y=segment_avg_duration,  # Use Avg. Sessiom Duration
    text=segment_avg_duration.round(2),
    textposition='auto',
    marker=dict(color=['#2ECC40', '#FFDC00', '#FF4136'])
))
```

```
engagement_fig.update_layout(
    title='Comparison of User Engagement by Bounce Rate Segment',
    xaxis=dict(title='Bounce Rate Segment'),
    yaxis=dict(title='Average Session Duration (minutes)'),
)

engagement_fig.show()
```

```
<ipython-input-25-1047fd887afb>:1: FutureWarning:

The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to
```



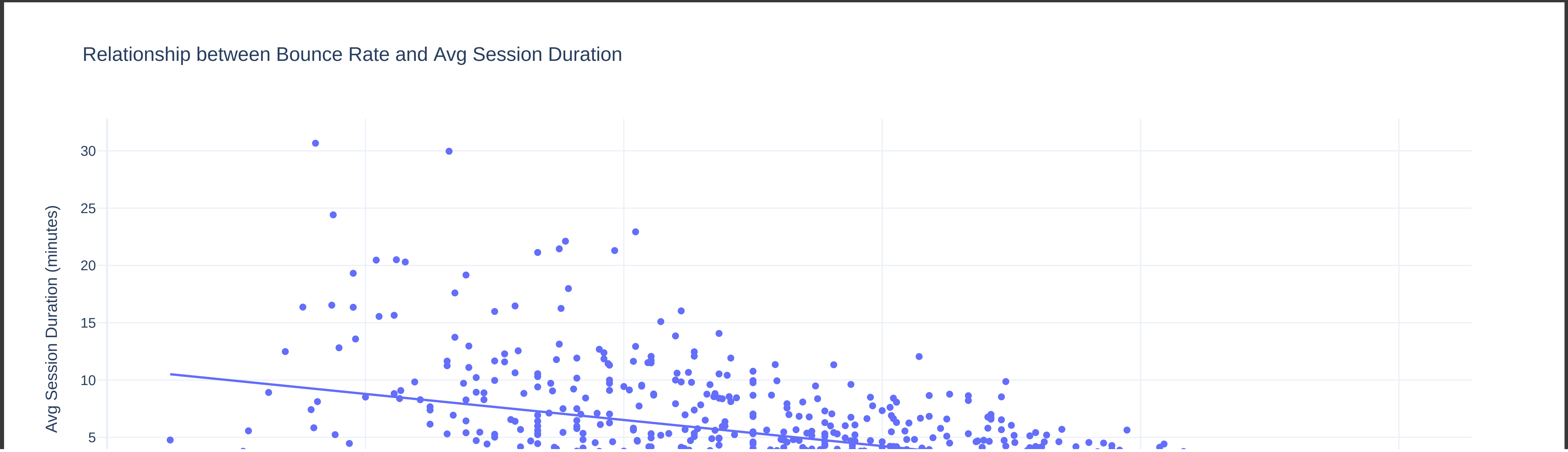
```
data['Total Session Duration']=data['Sessions']*data['Avg. Session Duration']

df_sorted=data.sort_values('Total Session Duration',ascending=False)

df_sorted.head(10)
```

| Client ID | Sessions | Avg. Session Duration | Bounce Rate | Avg. Session Duration | Bounce Rate Segment | Total Session Duration |
|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **20** | 1.884620e+09 | 93 | 0 days 00:30:40 | 16.13 | 30.666667 | Low | 1 days 23:32:00 |
| **54** | 1.041722e+09 | 67 | 0 days 00:20:30 | 22.39 | 20.500000 | Low | 0 days 22:53:30 |
| **262** | 8.756557e+08 | 34 | 0 days 00:29:58 | 26.47 | 29.966667 | Low | 0 days 16:58:52 |
| **10** | 1.461865e+09 | 117 | 0 days 00:08:27 | 48.72 | 8.450000 | Medium | 0 days 16:28:39 |
| **173** | 1.849182e+05 | 40 | 0 days 00:24:25 | 17.50 | 24.416667 | Low | 0 days 16:16:40 |
| **15** | 1.049234e+09 | 99 | 0 days 00:09:43 | 34.34 | 9.716667 | Medium | 0 days 16:01:57 |
| **310** | 2.026953e+09 | 31 | 0 days 00:22:07 | 35.48 | 22.116667 | Medium | 0 days 11:25:37 |
| **24** | 1.903206e+09 | 90 | 0 days 00:07:01 | 36.67 | 7.016667 | Medium | 0 days 10:31:30 |
| **211** | 2.054569e+09 | 37 | 0 days 00:16:15 | 35.14 | 16.250000 | Medium | 0 days 10:01:15 |
| **402** | 6.220935e+08 | 28 | 0 days 00:21:18 | 39.29 | 21.300000 | Medium | 0 days 09:56:24 |

Next steps: Generate code with df_sorted | View recommended plots | New interactive sheet

```
data['Avg. Session Duration_minutes'] = data['Avg. Session Duration'] / pd.Timedelta(minutes=1)

scatter_fig = px.scatter(data, x='Bounce Rate', y='Avg. Session Duration_minutes',  # Changed y-axis column
                         title='Relationship between Bounce Rate and Avg Session Duration', trendline='ols')
scatter_fig.update_layout(
    xaxis=dict(title='Bounce Rate'),
    yaxis=dict(title='Avg Session Duration (minutes)')  # Updated y-axis title
)

scatter_fig.show()
```

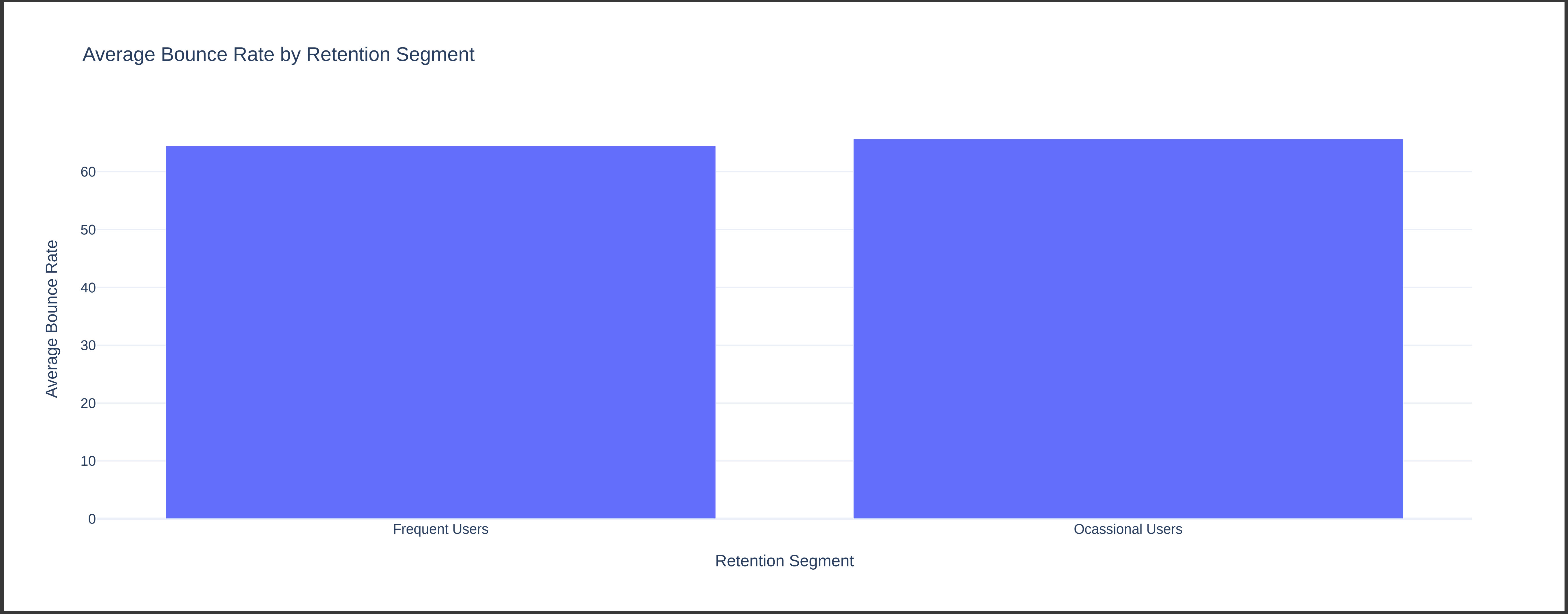


```
def get_retention_segment(row):
  if row['Sessions']>=32:
    return 'Frequent Users'
  else:
    return 'Ocassional Users'

data['Retention Segment']= data.apply(get_retention_segment,axis=1)

print(data)
```

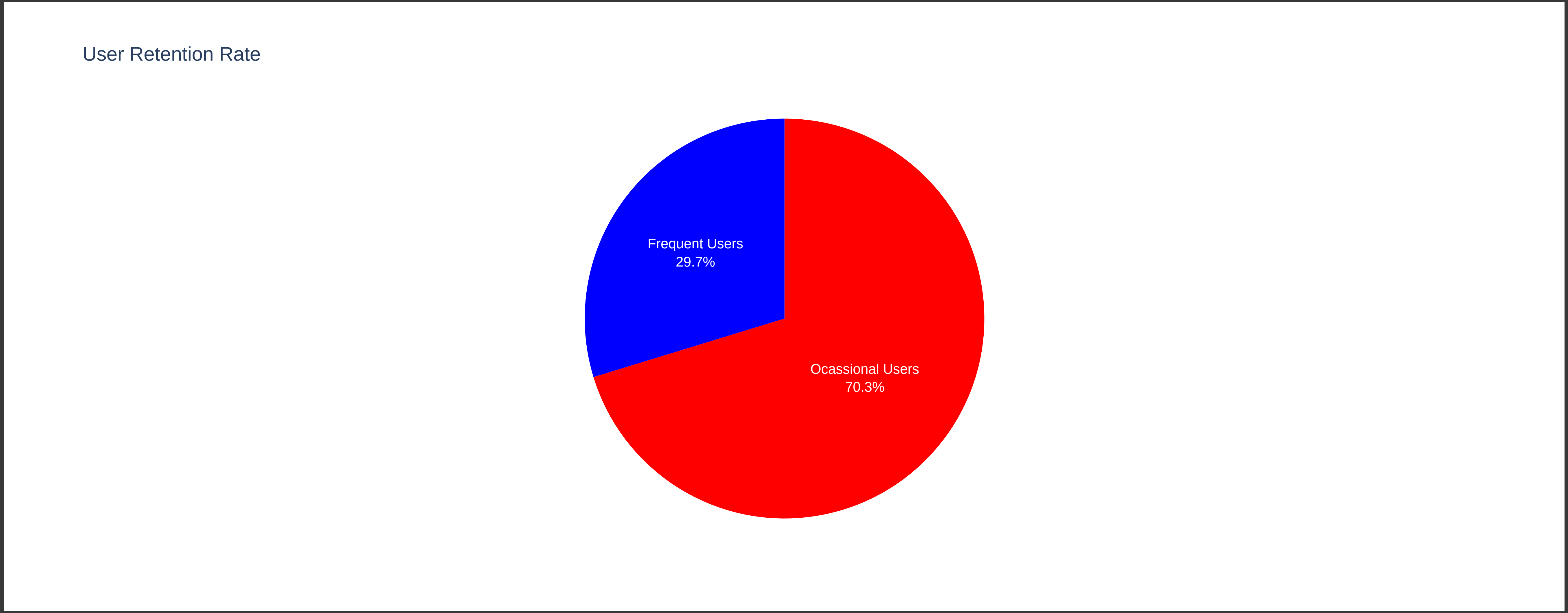
```
        Client ID  Sessions Avg. Session Duration  Bounce Rate  \
0    5.778476e+08       367       0 days 00:01:35        87.19
1    1.583822e+09       260       0 days 00:01:04        29.62
2    1.030699e+09       237       0 days 00:00:02        99.16
3    1.025030e+09       226       0 days 00:02:22        25.66
4    1.469968e+09       216       0 days 00:01:23        46.76
..            ...       ...                   ...          ...
994  1.049263e+09        17       0 days 00:07:44        41.18
995  1.145806e+09        17       0 days 00:05:37        47.06
996  1.153811e+09        17       0 days 00:00:12        94.12
997  1.182133e+09        17       0 days 00:01:13        88.24
998  1.184187e+09        17       0 days 00:02:34        64.71

     Avg. Session Duration Bounce Rate Segment Total Session Duration  \
0                 1.583333                High        0 days 09:41:05
1                 1.066667                 Low        0 days 04:37:20
2                 0.033333                High        0 days 00:07:54
3                 2.366667                 Low        0 days 08:54:52
4                 1.383333              Medium        0 days 04:58:48
..                     ...                 ...                    ...
994               7.733333              Medium        0 days 02:11:28
995               5.616667              Medium        0 days 01:35:29
996               0.200000                High        0 days 00:03:24
997               1.216667                High        0 days 00:20:41
998               2.566667              Medium        0 days 00:43:38

     Avg. Session Duration_minutes Retention Segment
0                         1.583333    Frequent Users
1                         1.066667    Frequent Users
2                         0.033333    Frequent Users
3                         2.366667    Frequent Users
4                         1.383333    Frequent Users
..                             ...               ...
994                       7.733333  Ocassional Users
995                       5.616667  Ocassional Users
996                       0.200000  Ocassional Users
```

```
997                    1.216667  Ocassional Users
998                    2.566667  Ocassional Users

[999 rows x 9 columns]
```

```
segment_bounce_rates=data.groupby('Retention Segment')['Bounce Rate'].mean().reset_index()

bar_fig=px.bar(segment_bounce_rates,x='Retention Segment', y='Bounce Rate',
                title='Average Bounce Rate by Retention Segment',
               labels={'Retention Segment':'Retention Segment','Bounce Rate':'Average Bounce Rate'})


bar_fig.show()
```



```
segment_counts = data['Retention Segment'].value_counts()

colour = ['red', 'blue']

fig = px.pie(data_frame=data,
           values=segment_counts.values,
           names=segment_counts.index,
           color=segment_counts.index,
           color_discrete_sequence=colour,
           title='User Retention Rate')

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(showlegend=False)
fig.show()
```


User Retention Rate
Frequent Users 29.7%
Ocassional Users 70.3%