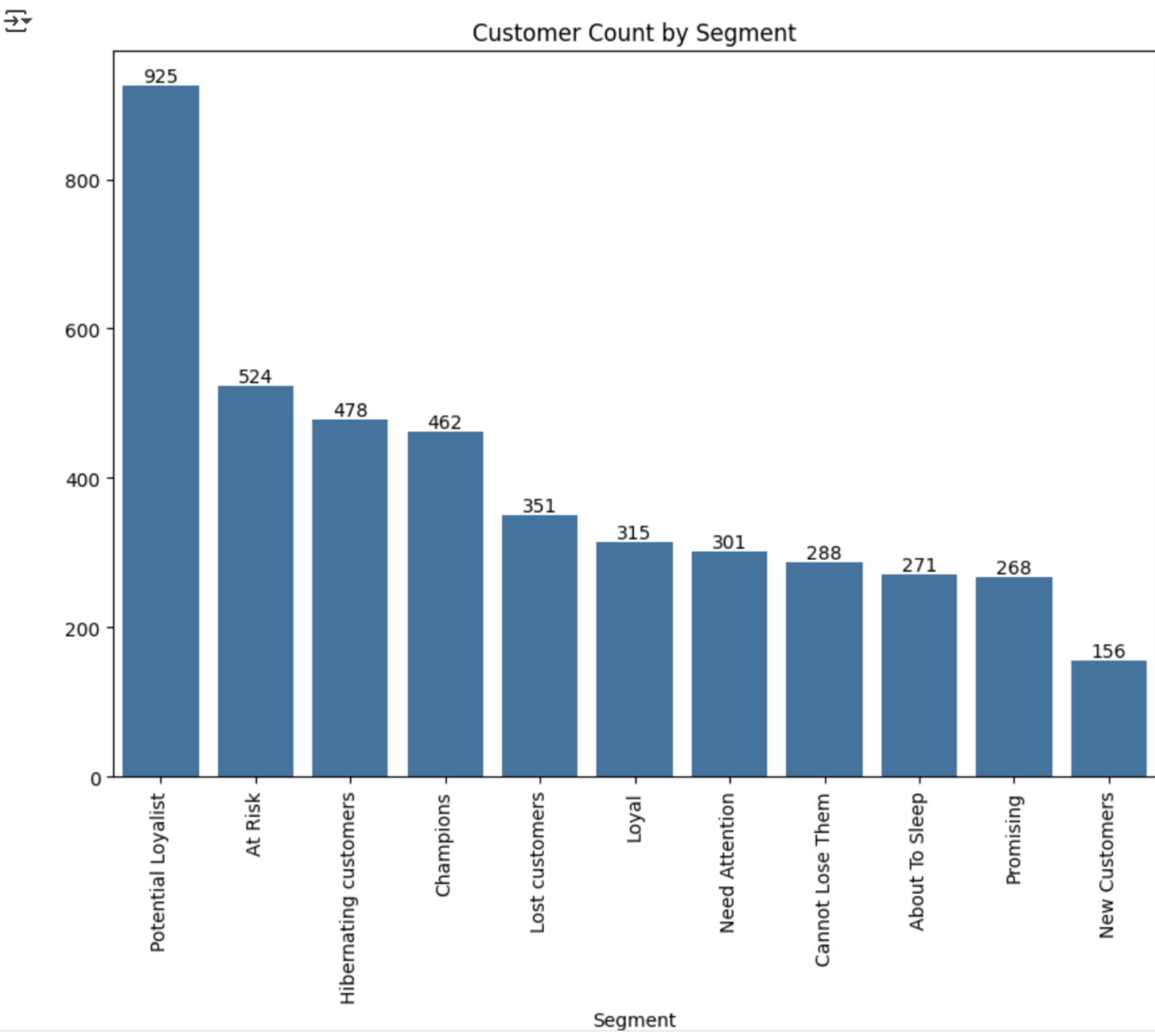


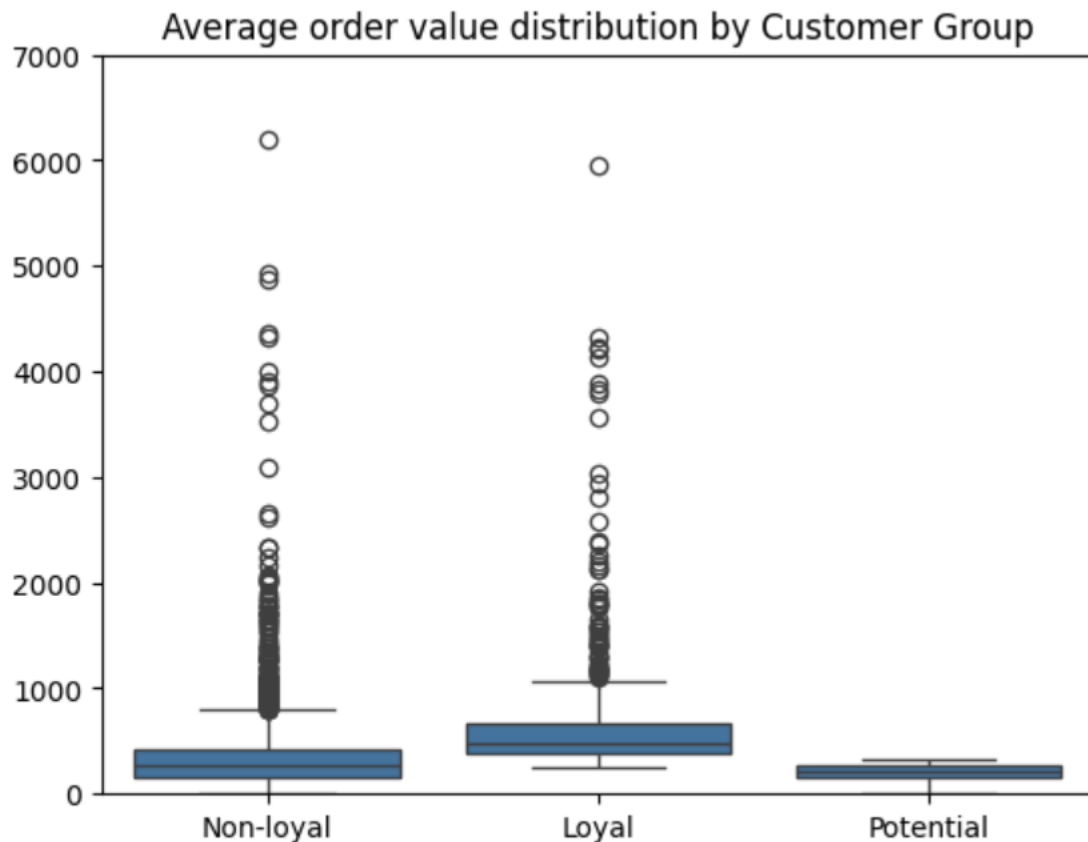
```
#customer count by segment
plt.figure(figsize=(10, 7))
ax = sns.barplot(data=df_bar, x='Segment', y='Count')
ax.set_ylabel('')
plt.xticks(rotation=90)
ax.bar_label(ax.containers[0], fontsize=10);|
plt.title('Customer Count by Segment')
plt.show()
```



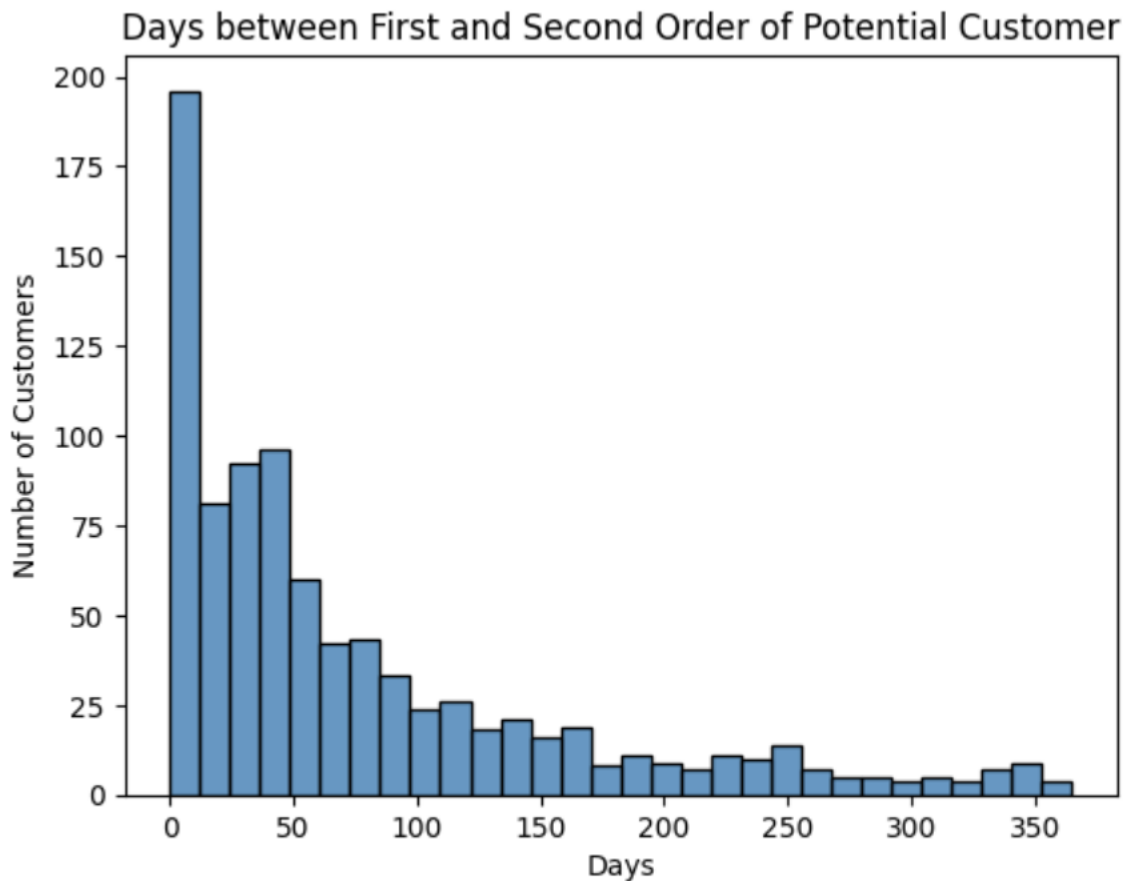


#Average order value distribution by Customer Group

```
rfm['CustomerGroup'] = rfm['Segment'].apply(  
    lambda x: 'Loyal' if x in ['Loyal', 'Champions']  
    else ('Potential' if x == 'Potential Loyalist'  
          else 'Non-loyal')  
)  
  
sns.boxplot(data=rfm, x='CustomerGroup', y='Monetary')  
plt.ylim(0, 7000)  
plt.title('Average order value distribution by Customer Group')  
plt.ylabel('')  
plt.xlabel('')  
plt.show()
```

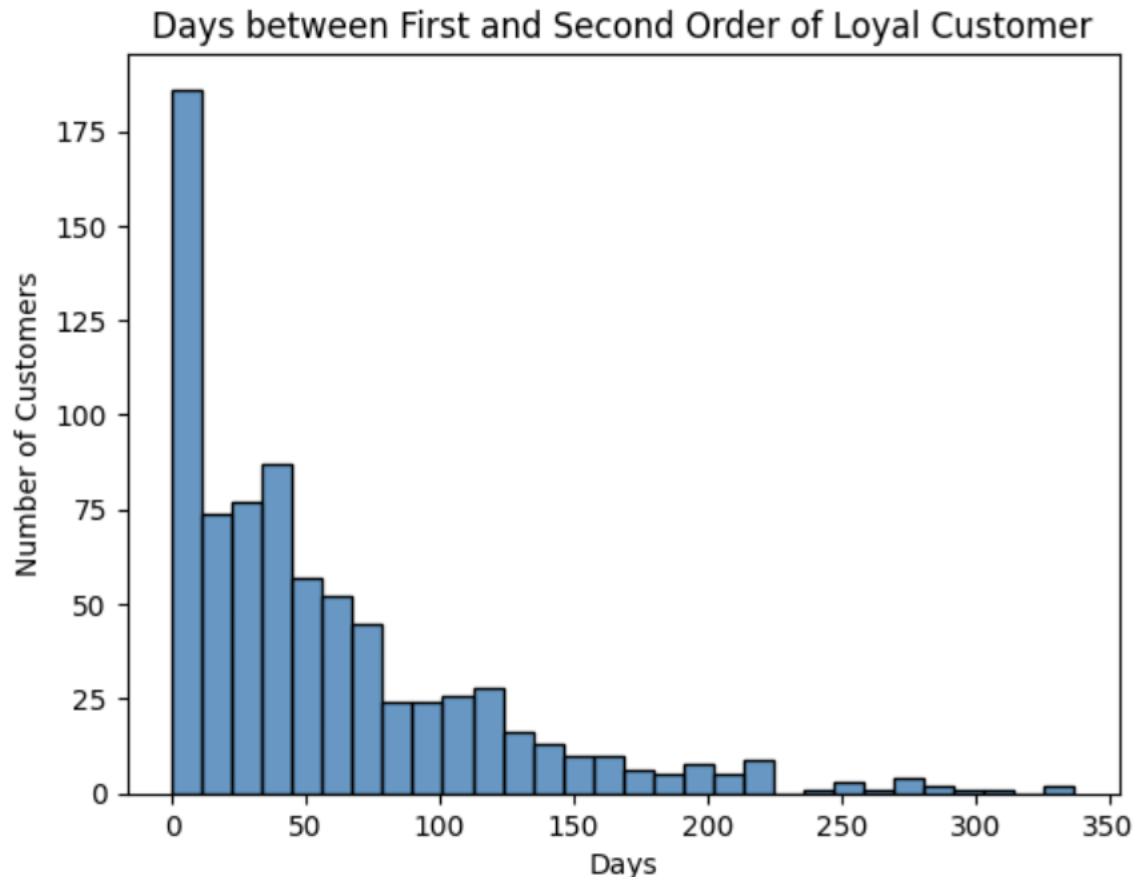


```
[ ] #Days Between First and Second Order of Potential Customer
sns.histplot(data=potential_order_dates, x='DaysBetweenOrders', bins=30)
plt.title('Days between First and Second Order of Potential Customer')
plt.xlabel('Days')
plt.ylabel('Number of Customers')
plt.show()
```





```
#Days Between First and Second Order of Loyal Customer  
sns.histplot(data= loyal_order_dates, x='DaysBetweenOrders', bins=30)  
plt.title('Days between First and Second Order of Loyal Customer')  
plt.xlabel('Days')  
plt.ylabel('Number of Customers')  
plt.show()
```



```

#create the FacetGrid
g = sns.FacetGrid(data=country_dis,
                  col='CustomerGroup',
                  col_order=['Loyal', 'Potential', 'Non-loyal'],
                  sharey=True,
                  height=5, aspect=1.2)

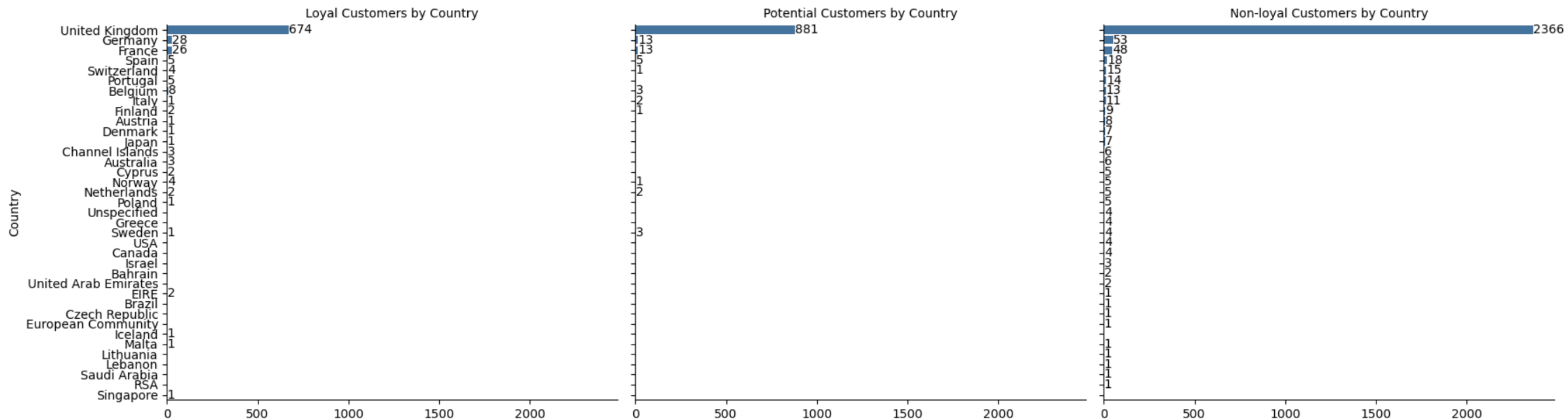
#create barplots in each facet
g.map_dataframe(sns.barplot, x='CustomerCount', y='Country', order=country_dis['Country'].unique())

#set titles and labels
g.set_titles('{col_name} Customers by Country')
g.set_axis_labels('Customer Count', 'Country')
g.set(xlabel='')

#iterate through the axes and add bar labels
for axes in g.axes.flat:
    axes.bar_label(axes.containers[0], fontsize=10)

plt.tight_layout()
plt.show()

```



```
[ ] #use stacked column chart to normalize
pivot_data = country_dis.pivot(index='Country', columns='CustomerGroup', values='CustomerCount').fillna(0)

pivot_pct = pivot_data.div(pivot_data.sum(axis=1), axis=0)

pivot_pct.plot(kind='bar', stacked=True, figsize=(14,6), colormap='Set2')
plt.title('Proportion of Customer Groups by Country')
plt.ylabel('Percentage')
plt.xticks(rotation=90)
plt.legend(title='Customer Group', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

