

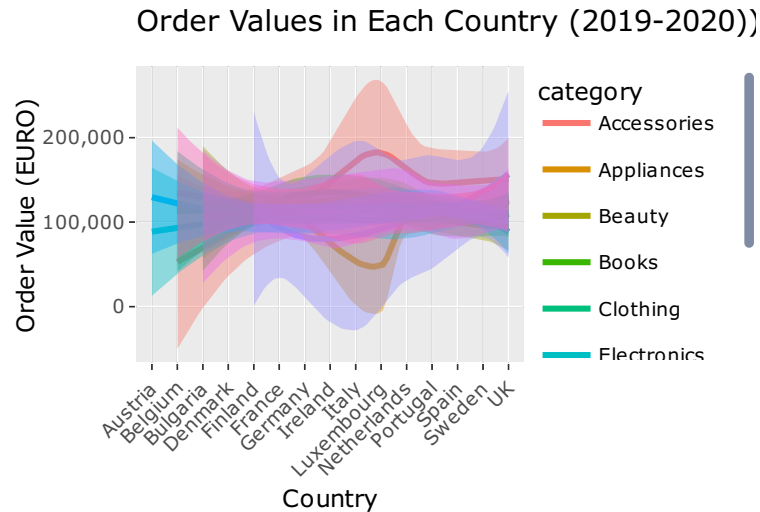
# Visualization

In my earlier analysis, I delved into the Sales\_Export\_2019\_2020 dataset, conducting data exploration and cleaning. In this phase, I will generate visualizations aimed at gaining insights into various aspects:

1. I intend to identify the countries with the highest cost and order values, shedding light on their respective standings.
2. Additionally, I plan to investigate the discrepancies in cost and order values across different categories within each country, providing a comprehensive view of category-specific performance.
3. Furthermore, I aim to examine the cumulative profit earned within each country and category, providing a holistic perspective on profitability.
4. By creating these visualizations, I hope to derive meaningful patterns and trends that will contribute to a more profound understanding of the dataset and its underlying dynamics.

## Sales By Country and Category

```
#Smoothed scatter plot of order values in Euros against countries using the #Sales_Export_2019_2020 data  
#  
orderval <- ggplot(data = Sales_Export_2019_2020, aes(x = country, y = order_value_EUR, color = category))  
  
#converting ggplot graph into a interactive Plotly graph, with a customized tooltip.  
p <- ggplotly(orderval, tooltip = c("x", "y", "color"))  
  
#Displaying Plotly graph.  
p
```



I utilize the ggplot function with our Sales\_Export\_2019\_2020 data to create a visualization. To establish visual characteristics, I make use of the aes function, with the x-axis displaying the country variable and the y-axis representing “Order\_value\_EUR.” Furthermore, I associate the “category” variable with color and fill aesthetics to differentiate between visual categories.

I create a smoothed line on the plot using the geom\_smooth function with the “loess” method. The group aesthetic is set to “category” to ensure each category has its own smoothed line, and the smoothing equation is outlined in the formula parameter.

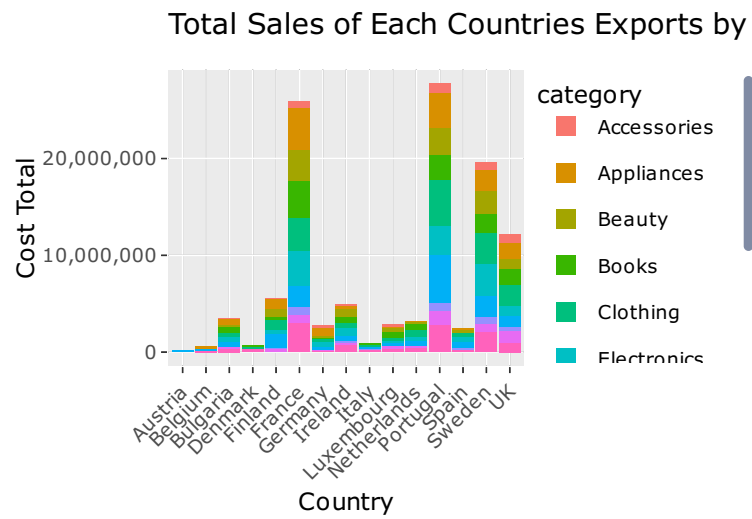
To enhance the legibility of x-axis text labels, I use the theme function to adjust the angle to 45 degrees and align the labels to the right using the hjust = 1 setting. I also format the y-axis labels using the scale\_y\_continuous function with the scales::comma function to add commas for better readability.

To add a title and axis labels, I use the labs function. Lastly, I convert the ggplot visualization (orderval) into an interactive Plotly graph with the ggplotly function, and customize the tooltip parameter to display the relevant information when hovering over data points.

The resulting interactive Plotly graph (p) provides an insightful visualization of order values across various countries, enriched with interactivity and tailor-made tooltips.

```
#Grouped column chart to visualize the total sales of each country's exports by category
#using the CountryTotalSaleExport dataset.
#Each country is represented on the x-axis, and the total sale values are shown as grouped #columns with
TCCSales <- ggplot(data = CountryTotalSaleExport, aes(x = country, y = saleCategoryTotal, fill = category))
#Converting ggplot graph into an interactive plotly graph.
tcc <- ggplotly(TCCSales)
```

```
#Displaying Plotly graph.
tcc
```



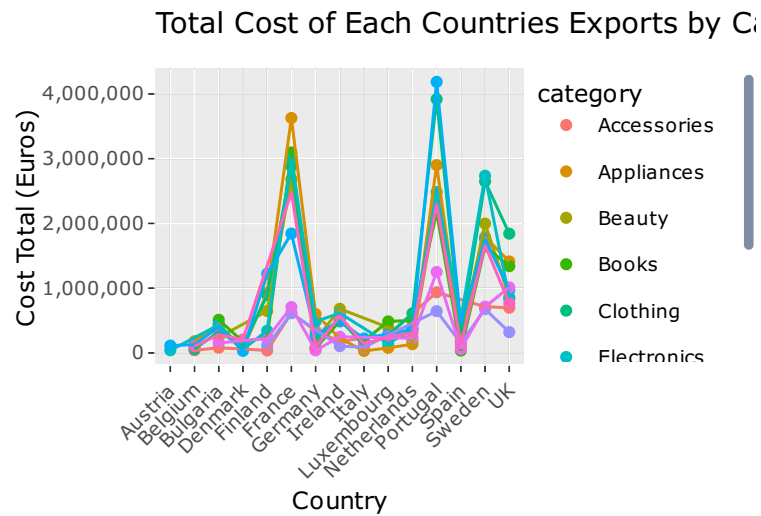
### Cost Per Category for Each Country

```
#Scatter plot with connected lines to visualize the total cost of each country's exports by #category u
#Each country is represented on the x-axis, and the total cost values are shown as scatter #points with

TCCCost <- ggplot(data = CountryTotalcostExport, aes(x = country, y = costCategoryTotal, colour = categ

#Converting ggplot graph into interactive Plotly graph.
TCCC <- ggplotly(TCCCost)

#Displaying Plotly Graph.
TCCC
```



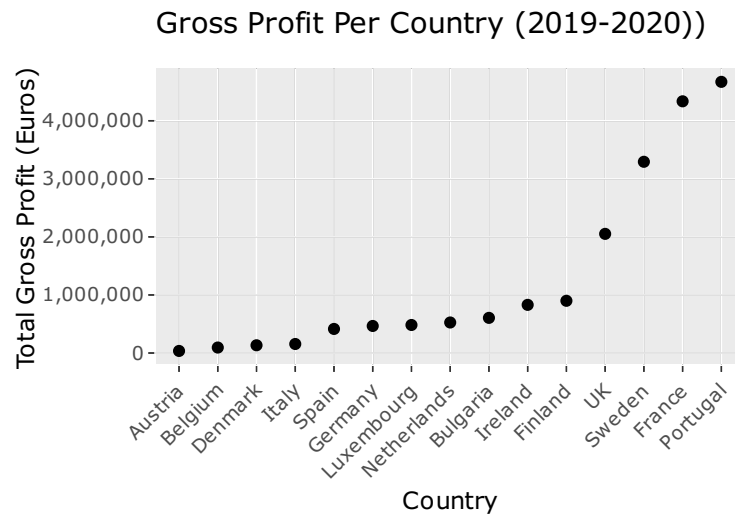
### Gross Profit for Each Country

```
#Scatter plot to visualize the cross profit per country using the CountryProfit dataset.
#Each country is represented on the x-axis, and the total gross profit values are shown as scatter points.
#The x-axis labels are reordered based on the total gross profit values, and the plot includes axis labels.

CGProfit <- ggplot(data = CountryProfit, aes(x = reorder(country, GrossProfit), y = GrossProfit)) + geom_point()

#Converting ggplot graph into interactive Plotly graph, with customized tooltip.
CG <- ggplotly(CGProfit, tooltip = c("x", "y"))

#Displaying Plotly Graph
CG
```



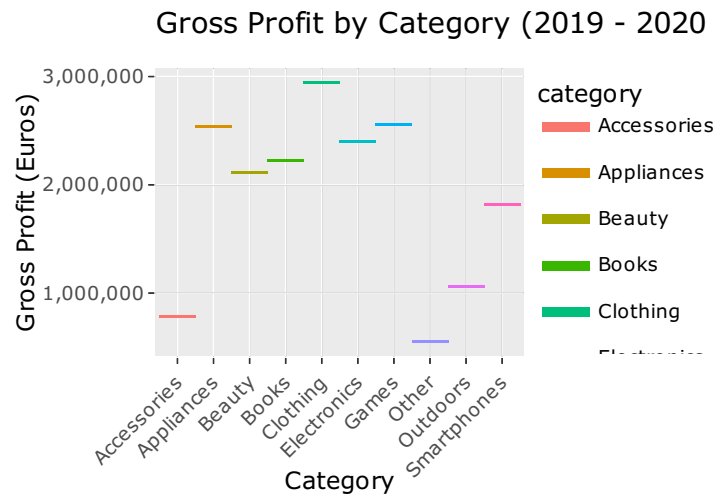
### Gross Profit based on Category

```
#Heatmap to visualize gross profit by category using the CategoryProfit dataset.
#Each category is represented on the x-axis, and the gross profit values are shown as #colored tiles.

CProfit <- ggplot(data = CategoryProfit, aes(x = category, y = GrossProfit, fill = category)) + geom_tile()

#Convert ggplot graph into interactive Plotly graph, with customized tooltip.
CP <- ggplotly(CProfit, tooltip = c("x", "y"))

#Display Plotly graph.
CP
```



```
# Loop through unique countries in the CoProfit dataset.
for (country in unique(CoProfit$country)) {
  country_data <- CoProfit %>%
    filter(country == country) # Filter data for the current country.

  # Extract category data for country_data.
  labels <- country_data$category
  # Extract Gross Profit from country_data.
  values <- country_data$GProfit

  # Pie chart to visualize total gross profit based on category contribution.
  pie_chart <- plot_ly(type = 'pie', labels = labels, values = values,
    textinfo = 'label+percent',
    insidetextorientation = 'radial')

  # Customize the layout of the pie chart for the current country.
  pie_chart <- pie_chart %>%
    layout(title = paste("Total Gross Profit based on Category Contribution -", country),
      xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
      yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

  print(pie_chart)
}
```