DataTranslationChallenge

```r
sales <- import('sales_data.Rdata')
head(sales)
```

```
                   Product Quantity PriceEach            DateTime       Date
1       USB-C Charging Cable        2     11.95 2019-04-19 08:46:00 2019-04-19
2 Bose SoundSport Headphones        1     99.99 2019-04-07 22:30:00 2019-04-07
3               Google Phone        1       600 2019-04-12 14:38:00 2019-04-12
4           Wired Headphones        1     11.99 2019-04-12 14:38:00 2019-04-12
5           Wired Headphones        1     11.99 2019-04-30 09:27:00 2019-04-30
6       USB-C Charging Cable        1     11.95 2019-04-29 13:03:00 2019-04-29
    ZIP State          City
1 75001    TX        Dallas
2 02215    MA        Boston
3 90001    CA   Los Angeles
4 90001    CA   Los Angeles
5 90001    CA   Los Angeles
6 94016    CA San Francisco
```

The dataset consists of Amazon sales of technology products placed over several months in 2019 in a select number of urban ZIP codes. Keeping my audience in mind i.e; **Data Analyst interns at Amazon**, I would present a series of graphs that would tell a story about the product sales at Amazon. This would help you going forward as your primary responsibility is to extract insights from data to drive informed decision-making and optimize business strategies.

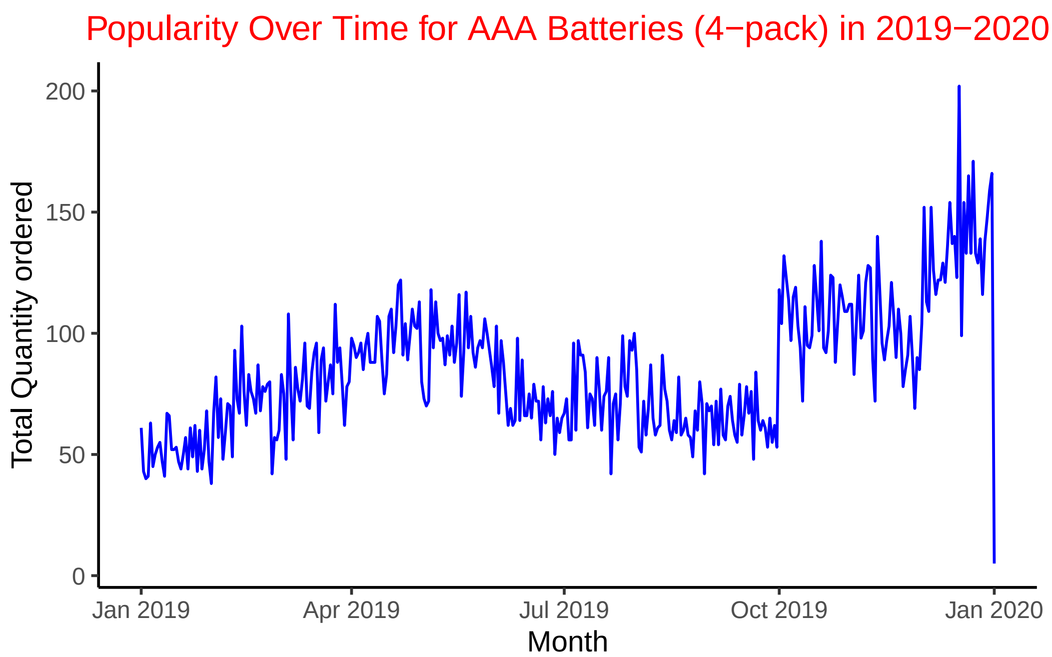
1. Popularity over time for the most popular product

To find the most popular product, we can aggregate the data by Product and sum up the Quantity for each product.

```r
# Convert Quantity to numeric
sales$Quantity <- as.numeric(sales$Quantity)


# Group by Product and Date, calculate total quantity ordered
product_popularity <- sales %>%
  group_by(Product, Date) %>%
  summarise(Total_Quantity = sum(Quantity)) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'Product'. You can override using the
`.groups` argument.
```

```r
# Find the most popular product
most_popular_product <- product_popularity %>%
  group_by(Product) %>%
  summarise(Total_Quantity = sum(Total_Quantity)) %>%
  top_n(1, Total_Quantity)

# Filter the product popularity data for the most popular product
most_popular_product_data <- product_popularity %>%
  filter(Product == most_popular_product$Product)

# Plotting
ggplot(most_popular_product_data, aes(x = Date, y = Total_Quantity)) +
  geom_line(color = 'blue') +
  labs(title = paste("Popularity Over Time for", most_popular_product$Product,"in 2019-202
       x = "Month",
       y = "Total Quantity ordered")+
  theme(
    plot.title = element_text(hjust = 0.5)
)+
  theme_classic()+
      theme(
    plot.title = element_text(hjust = 0.5, color = "red"))
```



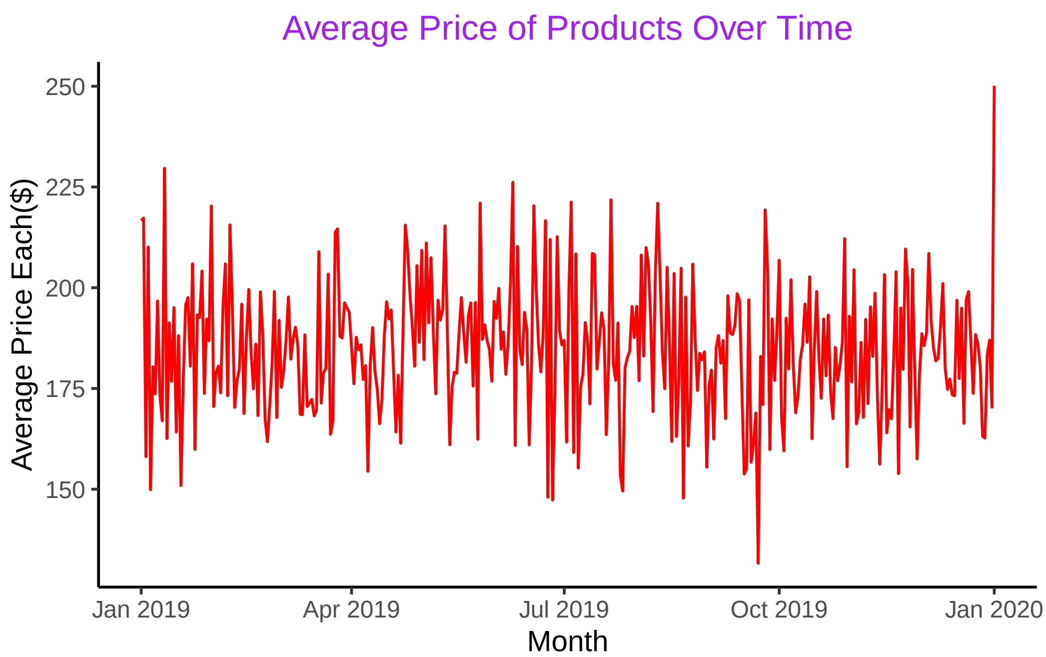
The first graph depicts the popularity over time for the most popular product. By aggregating the data and summing up the quantity for each product, we identified the product with the highest total quantity ordered. In this case, it's the AAA Batteries (4 pack) with its popularity visualized over the course of 2019-2020.

2. Average price of products over time

Moving on to the second graph, it illustrates the average price of products over time. We calculated the average price for all products and plotted it against the timeline. Despite fluctuations, it gives us an idea of the overall pricing trend across the observed period.

```
# Convert PriceEach to numeric
sales$PriceEach <- as.numeric(sales$PriceEach)


# Group by Date, calculate average price of products
average_price_over_time <- sales %>%
  group_by(Date) %>%
  summarise(Average_PriceEach = mean(PriceEach, na.rm = TRUE)) %>%
  ungroup()
```

```
# Plotting
ggplot(average_price_over_time, aes(x = Date, y = Average_PriceEach)) +
  geom_line(color = 'red') +
  labs(title = "Average Price of Products Over Time",
       x = "Month",
       y = "Average Price Each($)")+
      theme_classic()+
      theme(
    plot.title = element_text(hjust = 0.5, color = "purple"))
```



It can be observed that September shows a notable decrease in the average price of products.

3. Seasonal trends in product orders

The following graph explores seasonal trends in product orders. By extracting the month from the order date, we analyzed how the total quantity of orders varied throughout the year. This helps in understanding if there are any recurring patterns or seasonality in sales.

```
# Extract month from Date
sales$Month <- month(sales$Date, label = TRUE)
```

```r
# Group by Month, calculate total quantity ordered
seasonal_trends <- sales %>%
  group_by(Month) %>%
  summarise(Total_Quantity = sum(Quantity)) %>%
  ungroup()

seasonal_trends
```

```
# A tibble: 12 x 2
   Month Total_Quantity
   <ord>          <dbl>
 1 Jan            10903
 2 Feb            13449
 3 Mar            17005
 4 Apr            20558
 5 May            18667
 6 Jun            15253
 7 Jul            16072
 8 Aug            13448
 9 Sep            13109
10 Oct            22703
11 Nov            19798
12 Dec            28114
```

```r
# Plotting
ggplot(seasonal_trends, aes(x = Month, y = Total_Quantity, group=1)) +
  geom_line(color='green') +
  labs(title = "Seasonal Trends in Product Orders",
       x = "Month",
       y = "Total Quantity ordered")+
    theme_minimal()+
      theme(
    plot.title = element_text(hjust = 0.5, color = "red"))
```

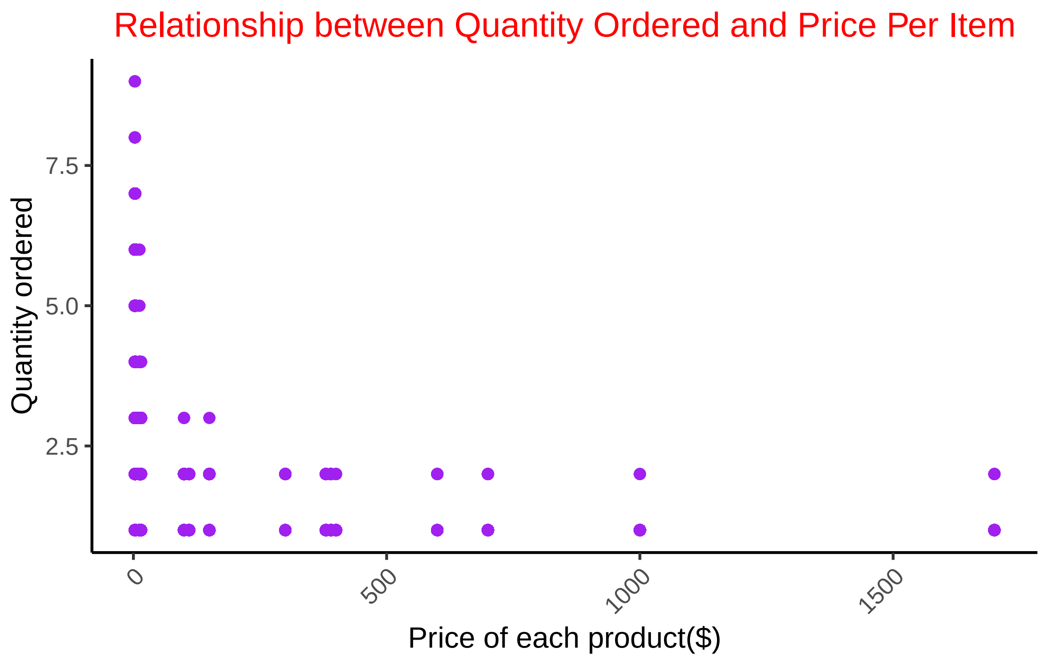
The graph reveals a decline in total quantity ordered from April to September 2019, followed by a steady increase in orders until January 2020.

4. Relationship between Quantity Ordered and Price Per Item

Next, the below graph examines the relationship between quantity ordered and the price per item. It showcases whether there's any correlation between the quantity of products ordered and their respective prices. This insight can be valuable for pricing strategies and inventory management.

```
# Plotting
ggplot(sales, aes(x = PriceEach, y = Quantity)) +
  geom_point(color = 'purple') +
  labs(title = "Relationship between Quantity Ordered and Price Per Item",
       x = "Price of each product($)",
       y = "Quantity ordered")+
        theme(
    plot.title = element_text(hjust = 0.5)
)+
  theme_classic()+
     theme(axis.text.x = element_text(angle = 45, hjust = 1),
```

```
        plot.title = element_text(hjust = 0.5, color = "red"))
```



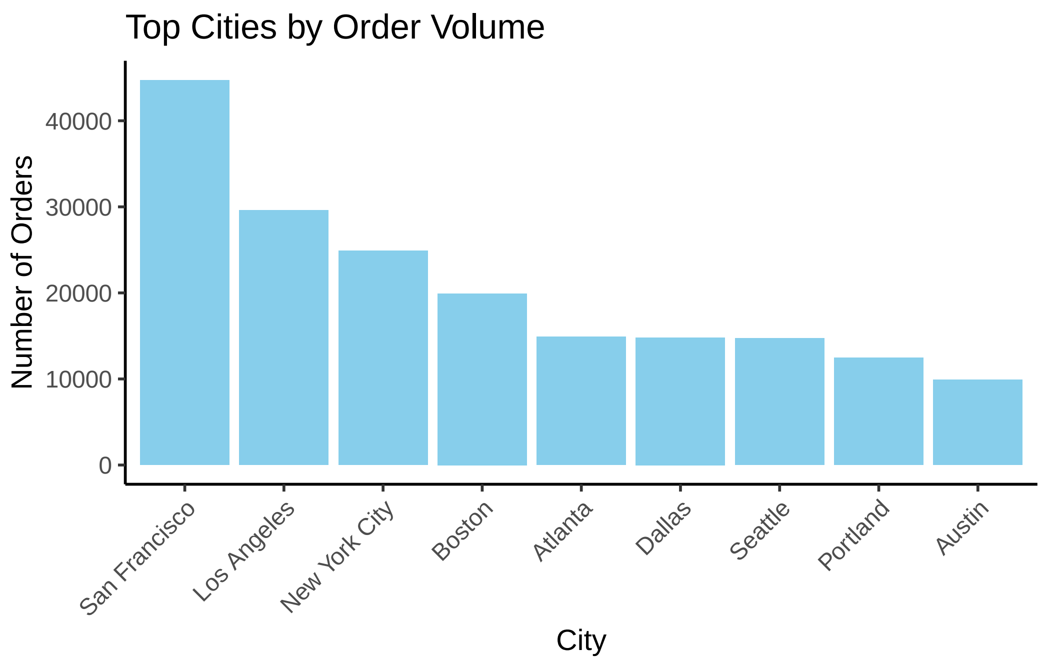
As prices rise, there is a corresponding decrease in the number of orders placed for those products.

5. Top cities by order volume

Finally, we delve into the top cities by order volume. By summing up the quantity of orders for each city, we generated a bar chart illustrating the quantity of orders by city. The analysis of top cities by order volume offers valuable insights into the distribution of sales within our dataset, shedding light on the geographical regions that contribute significantly to our business. Despite the limited scope of data encompassing a select number of ZIP codes and cities, this analysis provides a glimpse into the areas where our products are most popular. By identifying the cities with the highest order volumes, we can prioritize marketing efforts, tailor inventory management strategies, and optimize logistical operations to meet the demand effectively.

```
# Group by City, calculate number of orders
city_order_volume <- sales %>%
  group_by(City) %>%
  summarise(Orders = n()) %>%
```

```
  ungroup() %>%
  arrange(desc(Orders)) # Arrange cities by order volume

# Plotting
ggplot(city_order_volume, aes(x = reorder(City, -Orders), y = Orders)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Top Cities by Order Volume",
       x = "City",
       y = "Number of Orders") +
  theme_classic()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for bett
```



SanFrancisco stands out with the highest number of orders, while Austin exhibits the lowest in the top 10 cities. However, it's important to exercise caution when drawing conclusions from this data, given its limited coverage of zip codes, cities, and states overall.