Data Translation Challenge

## Libraries and Dataset

library(rio)  
library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(ggplot2)  
library(maps)  
library(zipcodeR)  
library(vtable)

Loading required package: kableExtra

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':  
  
 group\_rows

library(scales)  
sales <- import('sales\_data.Rdata')  
zip\_info <- import('zip\_info.csv')

#convert Quantity column from strings to numeric values  
sales$Quantity <- as.numeric(sales$Quantity)  
  
#convert PriceEach column from strings to numeric values  
sales$PriceEach <- as.numeric(sales$PriceEach)  
  
#working with Date variable  
sales$Date <- as.Date(sales$Date)

## Data Exploration

vtable(sales, lush = TRUE, factor.limit = 9, char.values = TRUE)

sales

| Name | Class | Values | Missing | Summary |
| --- | --- | --- | --- | --- |
| Product | character | '20in Monitor' '27in 4K Gaming Monitor' '27in FHD Monitor' '34in Ultrawide Monitor' 'AA Batteries (4-pack)' 'AAA Batteries (4-pack)' 'Apple Airpods Headphones' 'Bose SoundSport Headphones' 'Flatscreen TV' and more | 0 | nuniq: 19 |
| Quantity | numeric | Num: 1 to 9 | 0 | mean: 1.124, sd: 0.443, nuniq: 9 |
| PriceEach | numeric | Num: 2.99 to 1700 | 0 | mean: 184.4, sd: 332.731, nuniq: 17 |
| DateTime | POSIXct | Time: 2019-01-01 03:07:00 to 2020-01-01 05:13:00 | 0 | median: 2019-07-17 20:40:30, nuniq: 142395 |
| Date | Date | Time: 2019-01-01 to 2020-01-01 | 0 | median: 2019-07-17, nuniq: 366 |
| ZIP | character | '02215' '04101' '10001' '30301' '73301' '75001' '90001' '94016' '97035' and more | 0 | nuniq: 10 |
| State | character | 'CA' 'GA' 'MA' 'ME' 'NY' 'OR' 'TX' 'WA' | 0 | nuniq: 8 |
| City | character | 'Atlanta' 'Austin' 'Boston' 'Dallas' 'Los Angeles' 'New York City' 'Portland' 'San Francisco' 'Seattle' | 0 | nuniq: 9 |

vtable(zip\_info, lush = TRUE)

zip\_info

| Name | Class | Values | Missing | Summary |
| --- | --- | --- | --- | --- |
| ZIP | integer | Num: 2215 to 98101 | 0 | mean: 57407.3, sd: 40907.601, nuniq: 10 |
| TotalPopulation | integer | Num: 12792 to 58975 | 0 | mean: 26051.6, sd: 12885.755, nuniq: 10 |
| MedianHHIncome | integer | Num: 46309 to 119370 | 0 | mean: 81151, sd: 25319.33, nuniq: 10 |
| PCIncome | integer | Num: 14814 to 100364 | 0 | mean: 57085.4, sd: 28034.787, nuniq: 10 |
| MedianAge | numeric | Num: 21.6 to 44.3 | 0 | mean: 34.02, sd: 6.243, nuniq: 10 |
| Race\_White | integer | Num: 9231 to 22921 | 0 | mean: 16140.6, sd: 5012.111, nuniq: 10 |
| Race\_Black | integer | Num: 459 to 5483 | 0 | mean: 2179.9, sd: 1366.307, nuniq: 10 |
| Race\_American\_Indian | integer | Num: 148 to 802 | 0 | mean: 426.3, sd: 229.788, nuniq: 10 |
| Race\_Asian | integer | Num: 173 to 10134 | 0 | mean: 3238.9, sd: 3245.489, nuniq: 10 |
| Race\_Pacific\_Islander | integer | Num: 0 to 237 | 0 | mean: 71.3, sd: 81.024, nuniq: 9 |
| Race\_Other | integer | Num: 181 to 30491 | 0 | mean: 5003.8, sd: 9745.687, nuniq: 10 |
| Ethnicity\_Hispanic | integer | Num: 609 to 53085 | 0 | mean: 9223, sd: 16630.268, nuniq: 10 |
| Citizens | integer | Num: 10432 to 24069 | 0 | mean: 17171.8, sd: 4595.578, nuniq: 10 |

## Story & Visualizations

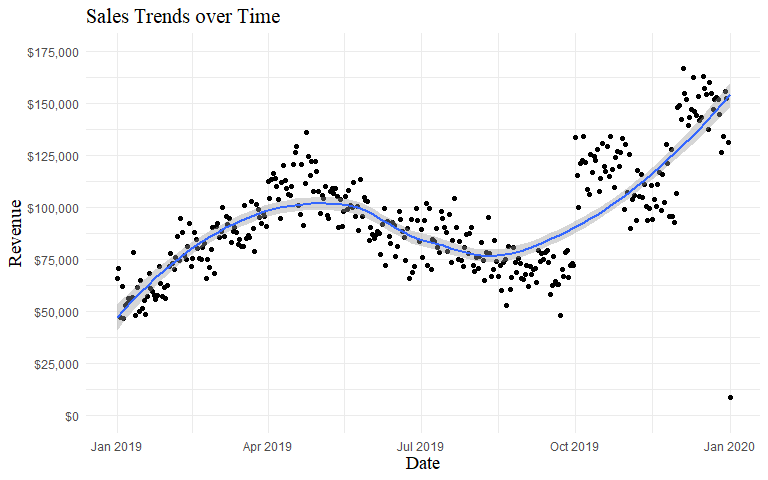
Through my analysis, Amazon’s technology products division can identify the top-selling products and understand the key factors that contribute to generating revenue. They can use these insights to help inform future business decisions and improve Amazon’s tech division’s growth.

##### Visualization 1:

First, let’s get an idea of the sales trends over 2019 of this data. We can see that the sales peaked around mid-year, followed by a dip towards the end of the year. This graph helps us better understand the overall sales patterns of the products and provide a more complete picture of our sales story before diving any deeper into the data.

TotalSales <- sales %>%   
 group\_by(Date) %>%   
 summarise(TS = sum(Quantity \* PriceEach))  
ggplot(TotalSales, aes(x= Date, y= TS)) +  
 geom\_point() +  
 geom\_smooth() +  
 labs(title = "Sales Trends over Time", x = "Date", y = "Revenue") +  
 scale\_y\_continuous(labels = dollar\_format(prefix = "$"), limits = c(0, 175000), breaks = seq(0, 175000, by = 25000)) +  
 theme\_minimal() +  
 theme(axis.title.x = element\_text(size = 14, family = "serif"),  
 axis.title.y = element\_text(size = 14, family = "serif"),  
 plot.title = element\_text(size = 16, family = "serif"))

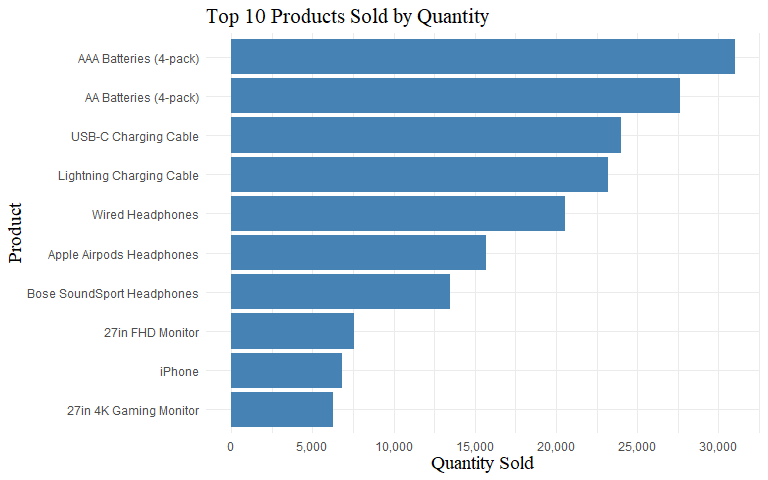
`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



##### Visualization 2:

Now we want to see which of the products we are selling the most of. We can do this by looking at the top 10 most products sold from the data. In the following table, we can see that the batteries and charging cables sell the most in quantity. This information can be used to prioritize inventory management and ensure that we have enough batteries and charging cable units in stock to meet demand.

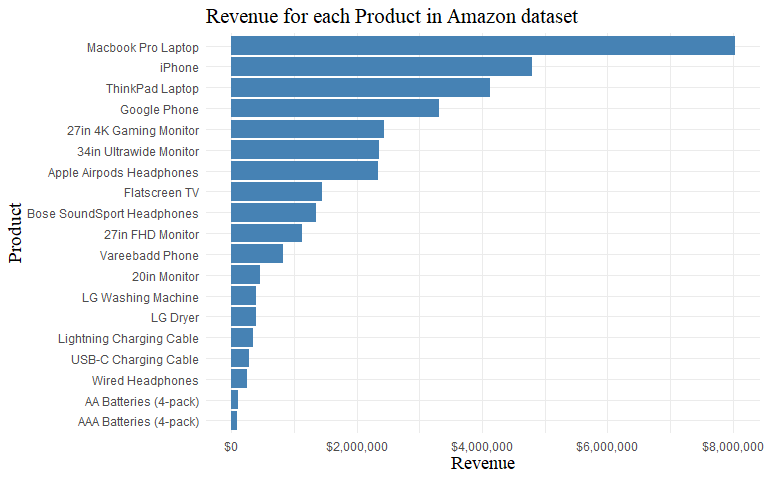
#top products sold by quantity  
topProducts <- sales %>%   
 group\_by(Product) %>%   
 summarize(totalQuantity = sum(Quantity)) %>%  
 arrange(desc(totalQuantity)) %>%   
 head(10)  
  
ggplot(topProducts, aes(x = reorder(Product, totalQuantity), y = totalQuantity)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 scale\_y\_continuous(labels = comma\_format(big.mark = ","), breaks = seq(0, 30000, by = 5000)) +  
 labs(x = "Product", y = "Quantity Sold", title = "Top 10 Products Sold by Quantity") +  
 theme\_minimal() +  
 theme(axis.title.x = element\_text(size = 14, family = "serif"),  
 axis.title.y = element\_text(size = 14, family = "serif"),  
 plot.title = element\_text(size = 16, family = "serif"))



##### Visualization 3:

Next we would want to identify which products from the data are generating the most money. As we can see in the bar chart below, laptops are by far the highest money making product in the data, followed by smartphones and monitors. We can increase our profits by prioritizing these products in inventory and marketing.

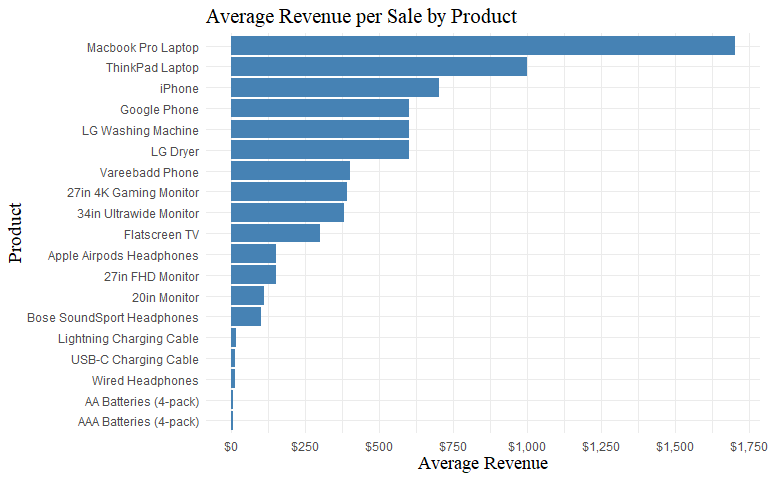
#total revenue by product  
totalrev <- sales %>%  
 group\_by(Product) %>%   
 summarize(TRproduct = sum(Quantity \* PriceEach))  
  
ggplot(totalrev, aes(x = reorder(Product, TRproduct), y = TRproduct)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 scale\_y\_continuous(labels = dollar\_format(prefix = "$")) +  
 labs(x = "Product", y = "Revenue", title = "Revenue for each Product in Amazon dataset") +  
 theme\_minimal() +  
 theme(axis.title.x = element\_text(size = 14, family = "serif"),  
 axis.title.y = element\_text(size = 14, family = "serif"),  
 plot.title = element\_text(size = 16, family = "serif"))



##### Visualization 4:

However, revenue alone may not tell the full story. It’s possible that some products are actually more expensive than others, leading to higher revenue per sale. To account for this, let’s look at the average revenue per sale by product. From this we can see dryers and washing machines move higher. We can use this to help determine the effectiveness of our pricing strategies.

#average revenue per sale  
avgRevperSale <- sales %>%  
 group\_by(Product) %>%  
 summarize(avgRev = mean(Quantity \* PriceEach)) %>%  
 arrange(desc(avgRev))  
  
ggplot(avgRevperSale, aes(x = reorder(Product, avgRev), y = avgRev)) +  
 geom\_col(fill = "steelblue") +  
 coord\_flip() +  
 labs(x = "Product", y = "Average Revenue", title = "Average Revenue per Sale by Product") +  
 scale\_y\_continuous(labels = dollar\_format(prefix = "$"), breaks = seq(0, 1750, by = 250)) +  
 theme\_minimal() +  
 theme(axis.title.x = element\_text(size = 14, family = "serif"),  
 axis.title.y = element\_text(size = 14, family = "serif"),  
 plot.title = element\_text(size = 16, family = "serif"))



##### Visualization 5:

Finally, we want to see the total sales for each zip code in the dataset. It can help to identify areas with lower sales and potentially target those areas for marketing efforts or promotions. This graph shows zip code 04101 has the least amount of revenue from the subset of Amazon’s technology products, so we could focus marketing for these products in this zip code.

# Calculate total sales by ZIP code  
totalSalesZIP <- sales %>%   
 group\_by(ZIP) %>%   
 summarize(TSzip = sum(Quantity \* PriceEach))  
  
# Create bar chart  
ggplot(totalSalesZIP, aes(x=reorder(ZIP,TSzip), y=TSzip)) +  
 geom\_bar(stat="identity", fill="steelblue") +  
 coord\_flip() +  
 scale\_y\_continuous(labels = dollar\_format(prefix = "$")) +  
 labs(title="Revenue by ZIP Code", x="ZIP Code", y="Revenue") +  
 theme\_minimal() +  
 theme(axis.title.x = element\_text(size = 14, family = "serif"),  
 axis.title.y = element\_text(size = 14, family = "serif"),  
 plot.title = element\_text(size = 16, family = "serif"))

