Group 5 Final Project

2023-09-12

Cheromaine Nisa Ornella Smith, Jordan Epstein, Thomas Lento, Vic Millar

STEP 1: LIBRARY NECESSARY PACKAGES

STEP 2: READ DATA INTO R STUDIO

```
# you can download the amazon.csv data here:
# https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset?resource=download

# setwd('/Users/victormillar/downloads')
AmazonData <- read_csv("amazon.csv")

# this line of code will need to be updated for your own directory,
# wherever amazon.csv is stored for you
```

STEP 3: CLEAN DATA

```
sum(is.na(AmazonData))
```

```
## [1] 3
```

```
# str(AmazonData)
glimpse(AmazonData)
```

```
## Rows: 1,465
## Columns: 16
## $ product_id          <chr> "B07JW9H4J1", "B098NS6PVG", "B096MSW6CT", "B08HDJ8~
## $ product_name        <chr> "Wayona Nylon Braided USB to Lightning Fast Chargi~
## $ category            <chr> "Computers&Accessories|Accessories&Peripherals|Cab~
## $ discounted_price    <chr> "₹399", "₹199", "₹199", "₹329", "₹154", "₹149", "~
## $ actual_price        <chr> "₹1,099", "₹349", "₹1,899", "₹699", "₹399", "₹1,00~
## $ discount_percentage <chr> "64%", "43%", "90%", "53%", "61%", "85%", "65%", "~
## $ rating              <dbl> 4.2, 4.0, 3.9, 4.2, 4.2, 3.9, 4.1, 4.3, 4.2, 4.0, ~
## $ rating_count        <dbl> 24269, 43994, 7928, 94363, 16905, 24871, 15188, 30~
## $ about_product       <chr> "High Compatibility : Compatible With iPhone 12, 1~
## $ user_id             <chr> "AG3D6O4STAQKAY2UVGEUV46KN35Q,AHMY5CWJMMK5BJRBBSNL~
## $ user_name           <chr> "Manav,Adarsh gupta,Sundeep,S.Sayeed Ahmed,jaspree~
## $ review_id           <chr> "R3HXWT0LRP0NMF,R2AJM3LFTLZHFO,R6AQJGUP6P86,R1KD19~
## $ review_title        <chr> "Satisfied,Charging is really fast,Value for money~
## $ review_content      <chr> "Looks durable Charging is fine tooNo complains,Ch~
## $ img_link            <chr> "https://m.media-amazon.com/images/W/WEBP_402378-T~
## $ product_link        <chr> "https://www.amazon.in/Wayona-Braided-WN3LG1-Synci~
```

```r
# To Do: remove NA values, convert key metrics to numeric instead of
# character

na_counts <- colSums(is.na(AmazonData))
na_counts
```

```
##          product_id        product_name            category    discounted_price
##                   0                   0                   0                   0
##        actual_price discount_percentage              rating        rating_count
##                   0                   0                   1                   2
##       about_product             user_id           user_name           review_id
##                   0                   0                   0                   0
##        review_title      review_content            img_link        product_link
##                   0                   0                   0                   0
```

```r
# only 3 NA values - one in Rating, 2 in Rating_count

AmazonData <- na.omit(AmazonData)
na_counts <- colSums(is.na(AmazonData))
na_counts
```

```
##          product_id        product_name            category    discounted_price
##                   0                   0                   0                   0
##        actual_price discount_percentage              rating        rating_count
##                   0                   0                   0                   0
##       about_product             user_id           user_name           review_id
##                   0                   0                   0                   0
##        review_title      review_content            img_link        product_link
##                   0                   0                   0                   0
```

```r
# 3 rows removed, no more NAs

# Need to remove Indian Rupee currency symbol before converting to
# Numeric
AmazonData$discounted_price <- substring(AmazonData$discounted_price, 2)
AmazonData$actual_price <- substring(AmazonData$actual_price, 2)
# symbols are removed, will try numeric again now

# now need to remove the commas from the prices
AmazonData$actual_price <- as.numeric(gsub("[^0-9.]", "", AmazonData$actual_price))
AmazonData$discounted_price <- as.numeric(gsub("[^0-9.]", "", AmazonData$discounted_price))
# prices are now numeric and have no comma

AmazonData$rating <- as.numeric(AmazonData$rating)
# Rating is now also numeric
```

STEP 4: CONVERT CURRENCY VIA FUNCTION

```r
AmazonData2 <- AmazonData
# Creating a Save Point so I don't mess up previous work
```

```r
# I am going to write a function that converts Indian Rupee to USD

convert_usd <- function(rupee_price) {
    exchange_rate <- 0.012  # Replace this if exchange rate changes
    usd_price <- rupee_price * exchange_rate
    return(usd_price)
}

# Will now test the function to confirm it works

test_data1 <- data.frame(rupee_test = c(100, 200, 300, 400))
test_data1$USD_price <- convert_usd(test_data1$rupee_test)
test_data1  # The test was successful
```

```
##   rupee_test USD_price
## 1        100       1.2
## 2        200       2.4
## 3        300       3.6
## 4        400       4.8
```

```r
AmazonData2 <- AmazonData2 %>%
    mutate(discounted_price_usd = round(convert_usd(discounted_price),
        2), actual_price_usd = round(convert_usd(actual_price), 2))

# Now have two new columns with prices in USD
```

STEP 5: SEPARATING CATEGORIES

```r
AmazonData3 <- AmazonData2
# Creating another Save Point

# I want to separate out the Category column into 7 separate columns,
# one for each tier of the category.

categories <- strsplit(AmazonData3$category, "\\|")
AmazonData3$Cat1 <- sapply(categories, `[`, 1)
AmazonData3$Cat2 <- sapply(categories, `[`, 2)
AmazonData3$Cat3 <- sapply(categories, `[`, 3)
AmazonData3$Cat4 <- sapply(categories, `[`, 4)
AmazonData3$Cat5 <- sapply(categories, `[`, 5)
AmazonData3$Cat6 <- sapply(categories, `[`, 6)
AmazonData3$Cat7 <- sapply(categories, `[`, 7)

AmazonData3 <- subset(AmazonData3, select = -category)  #delete original column

# Final look at clean data
summary(AmazonData3)
```

```
##   product_id        product_name       discounted_price  actual_price
##  Length:1462        Length:1462        Min.   :   39    Min.   :    39
```

```
##  Class :character   Class :character   1st Qu.:  325    1st Qu.:   800
##  Mode  :character   Mode  :character   Median :  799    Median :  1670
##                                        Mean   : 3130    Mean   :  5453
##                                        3rd Qu.: 1999    3rd Qu.:  4321
##                                        Max.   :77990    Max.   :139900
##  discount_percentage     rating       rating_count    about_product
##  Length:1462         Min.   :2.000   Min.   :     2   Length:1462
##  Class :character    1st Qu.:4.000   1st Qu.:  1192   Class :character
##  Mode  :character    Median :4.100   Median :  5179   Mode  :character
##                      Mean   :4.097   Mean   : 18307
##                      3rd Qu.:4.300   3rd Qu.: 17342
##                      Max.   :5.000   Max.   :426973
##    user_id           user_name          review_id         review_title
##  Length:1462        Length:1462        Length:1462        Length:1462
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  review_content       img_link         product_link       discounted_price_usd
##  Length:1462        Length:1462        Length:1462        Min.   :  0.47
##  Class :character   Class :character   Class :character   1st Qu.:  3.90
##  Mode  :character   Mode  :character   Mode  :character   Median :  9.59
##                                                          Mean   : 37.56
##                                                          3rd Qu.: 23.99
##                                                          Max.   :935.88
##  actual_price_usd      Cat1               Cat2              Cat3
##  Min.   :   0.47   Length:1462        Length:1462        Length:1462
##  1st Qu.:   9.60   Class :character   Class :character   Class :character
##  Median :  20.04   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  65.44
##  3rd Qu.:  51.85
##  Max.   :1678.80
##      Cat4               Cat5               Cat6              Cat7
##  Length:1462        Length:1462        Length:1462        Length:1462
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

STEP 6: ANSWERING BUSINESS QUESTIONS

```
AmazonData4 <- AmazonData3  # New Save Point

# Now that the data is cleaned up, I can start working through the
# business questions.
```

```
# Group the data by the Cat1 column and calculate the total revenue
# for each category
```

```
category_revenue <- AmazonData4 %>%
    mutate(total_revenue = discounted_price_usd * rating_count) %>%
    group_by(Cat1) %>%
    summarize(total_revenue = sum(total_revenue, na.rm = TRUE))

# Sort results to find top category
top_category <- category_revenue %>%
    arrange(desc(total_revenue)) %>%
    head(1)

top_category
```

QUESTION 1: Which top level category brought in the most revenue for Amazon?

```
## # A tibble: 1 x 2
##   Cat1        total_revenue
##   <chr>               <dbl>
## 1 Electronics    710185543.
```

```
# Based on a rough estimate using discounted price x rating count, we
# believe teh top selling category in this dataset is Electronics.
# This is based on the assumption that everyone who gave a product a
# rating also purchased a product, but it is only useful for trends
# and not an exact revenue count, since there will be customers who
# bought a product but did not give it a rating.
```

QUESTION 2: What is the average rating by broad category (Tier 1, ex. Electronics)? What is the average rating by price? Can both results be visualized?

```
average_rating_by_cat <- AmazonData4 %>%
    group_by(Cat1) %>%
    summarize(average_rating = mean(rating)) %>%
    arrange(desc(average_rating))

average_rating_by_cat
```

Q 2.1: Average Rating by Category

```
## # A tibble: 9 x 2
##   Cat1                  average_rating
##   <chr>                          <dbl>
## 1 OfficeProducts                  4.31
## 2 Toys&Games                      4.3
## 3 HomeImprovement                 4.25
## 4 Computers&Accessories           4.16
## 5 Electronics                     4.08
## 6 Home&Kitchen                    4.04
## 7 Health&PersonalCare             4
## 8 MusicalInstruments              3.9
## 9 Car&Motorbike                   3.8
```

```r
# Office Products has the highest average rating with 4.31/5 stars.
# Car & Motorbike has the worst average rating with 3.8
```

```r
price_bins <- c(0, 25, 50, 100, Inf)  # Creating price bins

AmazonData4 <- AmazonData4 %>%
    mutate(price_group = cut(discounted_price_usd, breaks = price_bins,
        labels = c("0-25", "25-50", "50-100", "100+")))

# New column 'price_group' now exists

average_rating_by_price <- AmazonData4 %>%
    group_by(price_group) %>%
    summarize(average_rating = mean(rating)) %>%
    arrange(desc(average_rating))

average_rating_by_price
```

Q 2.2: Average Rating by Discounted Price

```
## # A tibble: 4 x 2
##   price_group average_rating
##   <fct>                <dbl>
## 1 100+                  4.18
## 2 50-100                4.16
## 3 0-25                  4.08
## 4 25-50                 4.08
```

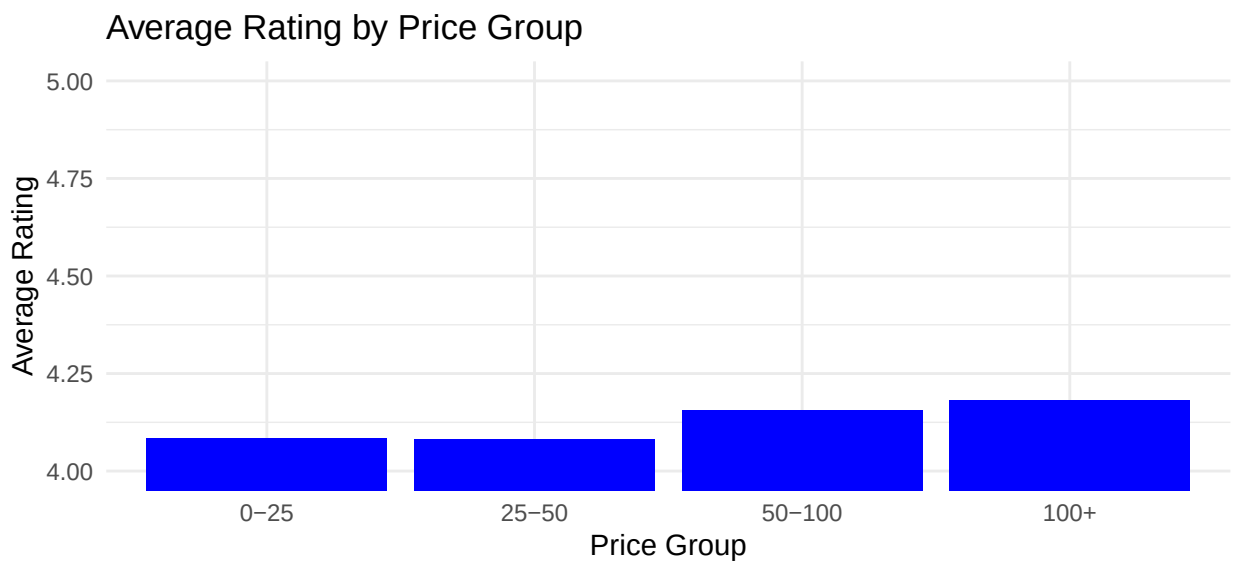
```r
# Rating seems to go up as the price goes up
cor(AmazonData4[, c("discounted_price_usd", "rating")], use = "complete")
```

```
##                     discounted_price_usd    rating
## discounted_price_usd            1.0000000 0.1211309
## rating                          0.1211309 1.0000000
```

```r
# there is a positive, but weak, correlation between
# discount_price_usd and rating
```

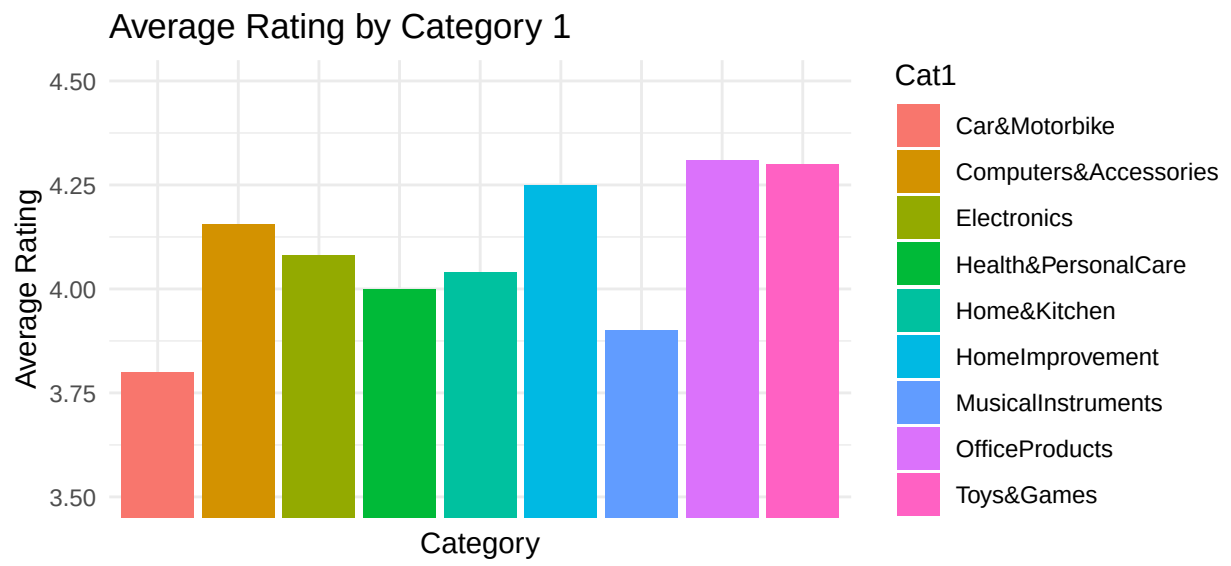
Q 2.3: Visualizations Average Rating by Price Group

```r
library(ggplot2)

# Create a bar chart for Average Rating by Price Group
ggplot(average_rating_by_price, aes(x = price_group, y = average_rating)) +
    geom_bar(stat = "identity", fill = "blue") + labs(title = "Average Rating by Price Group",
    x = "Price Group", y = "Average Rating") + theme_minimal() + coord_cartesian(ylim = c(4,
    5))
```



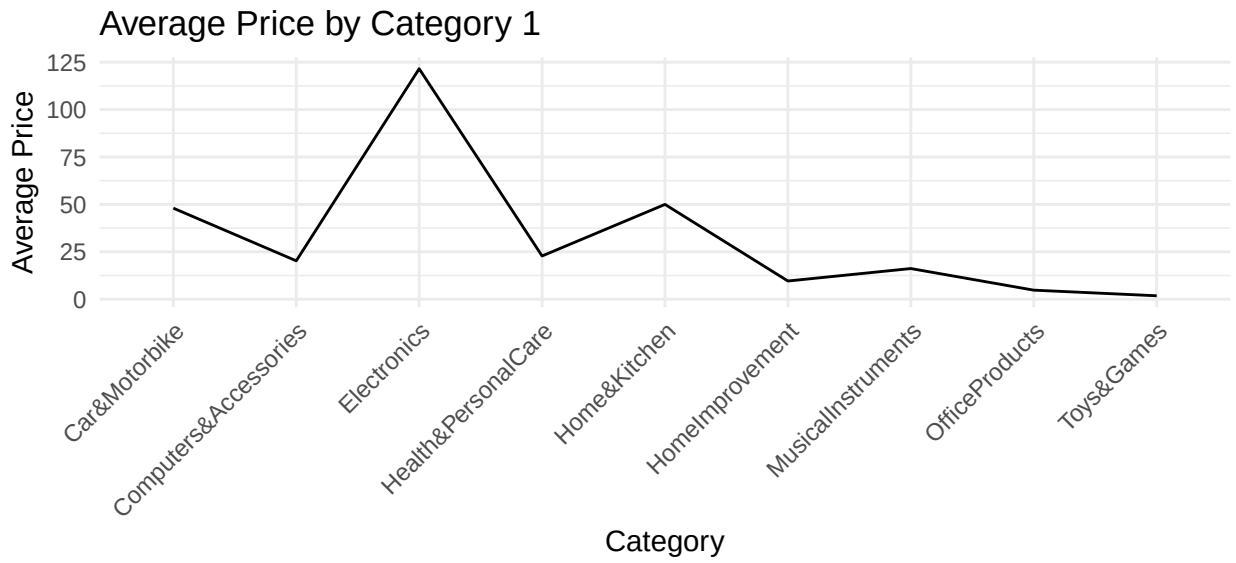
Average Rating by Tier 1 Category

```
# Create a bar chart for Average Rating by Category 1
ggplot(average_rating_by_cat, aes(x = Cat1, y = average_rating, fill = Cat1)) +
    geom_bar(stat = "identity") + labs(title = "Average Rating by Category 1",
    x = "Category", y = "Average Rating") + theme_minimal() + coord_cartesian(ylim = c(3.5,
    4.5)) + theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```



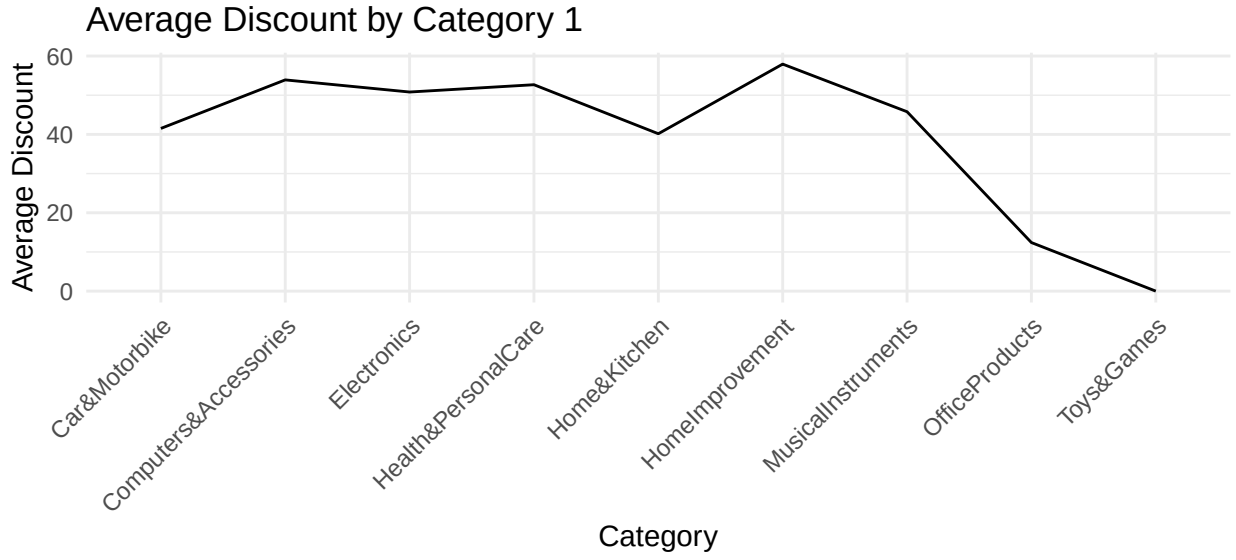

```
# figure out average price by category
average_price_by_cat <- AmazonData4 %>%
    group_by(Cat1) %>%
    summarize(average_price = mean(actual_price_usd))
```

```
ggplot(average_price_by_cat, aes(x = Cat1, y = average_price, group = 1)) +
    geom_line() + labs(title = "Average Price by Category 1", x = "Category",
    y = "Average Price") + theme_minimal() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1))
```




```
# figure out average price by category
average_discount_by_cat1 <- AmazonData4 %>%
    group_by(Cat1) %>%
    summarize(average_discount = mean(100 * (1 - (discounted_price_usd/actual_price_usd))))
```

```
ggplot(average_discount_by_cat1, aes(x = Cat1, y = average_discount, group = 1)) +
    geom_line() + labs(title = "Average Discount by Category 1", x = "Category",
    y = "Average Discount") + theme_minimal() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1))
```



QUESTION 3: What are the top 5 selling products (based on number of ratings), and what are the top 5 products based on revenue (rating count x discounted price)? How many products overlap of each set of 5? What is the top selling product in each category?

```r
top_rated_products <- AmazonData4 %>%
    arrange(desc(rating_count)) %>%
    head(5) %>%
    select(product_name, rating_count)

top_rated_products
```

Q 3.1: Top 5 selling products based on number of ratings

```
## # A tibble: 5 x 2
##   product_name                                                  rating_count
##   <chr>                                                            <dbl>
## 1 AmazonBasics Flexible Premium HDMI Cable (Black, 4K@60Hz, 18Gbps~       426973
## 2 Amazon Basics High-Speed HDMI Cable, 6 Feet - Supports Ethernet,~       426973
## 3 Amazon Basics High-Speed HDMI Cable, 6 Feet (2-Pack),Black              426973
## 4 AmazonBasics Flexible Premium HDMI Cable (Black, 4K@60Hz, 18Gbps~       426972
## 5 boAt Bassheads 100 in Ear Wired Earphones with Mic(Taffy Pink)          363713
```

```r
# The 4 products with highest rating counts are all HDMI cables
# Number 5 is a set of wired headphones
```

```r
top_revenue_products <- AmazonData4 %>%
    mutate(revenue = rating_count * discounted_price_usd) %>%
    arrange(desc(revenue)) %>%
    head(5) %>%
    select(product_name, revenue)

top_revenue_products
```

Q 3.2: Top 5 products based on revenue (rating count x discounted price)

```
## # A tibble: 5 x 2
##   product_name                                                    revenue
##   <chr>                                                             <dbl>
## 1 Redmi 9 Activ (Carbon Black, 4GB RAM, 64GB Storage) | Octa-core Helio~  3.20e7
## 2 Redmi 9A Sport (Coral Green, 3GB RAM, 32GB Storage) | 2GHz Octa-core ~  2.82e7
## 3 Redmi 9A Sport (Coral Green, 2GB RAM, 32GB Storage) | 2GHz Octa-core ~  2.45e7
## 4 Redmi 9A Sport (Carbon Black, 2GB RAM, 32GB Storage) | 2GHz Octa-core~  2.45e7
## 5 Redmi 126 cm (50 inches) 4K Ultra HD Android Smart LED TV X50 | L50M6~  1.79e7
```

```r
# The products with the most revenue are cellphones and tvs There is
# NO OVERLAP between these two groups of 5 products.
```

```r
categories <- AmazonData4 %>%
    select("Cat1") %>%
```

```
    distinct()

conflicts_prefer(dplyr::filter)
```

Q 3.2: Top Product in each Tier 1 Category

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
top_revenue_products_by_cat <- AmazonData4 %>%
    filter(Cat1 %in% categories$Cat1) %>%
    mutate(revenue = rating_count * discounted_price_usd) %>%
    group_by(Cat1) %>%
    arrange(desc(revenue)) %>%
    top_n(1, wt = revenue) %>%
    ungroup() %>%
    select(product_name, revenue, Cat1)

top_revenue_products_by_cat
```

```
## # A tibble: 9 x 3
##   product_name                                                revenue Cat1
##   <chr>                                                         <dbl> <chr>
## 1 Redmi 9 Activ (Carbon Black, 4GB RAM, 64GB Storage) | Octa-core~  3.20e7 Elec~
## 2 SanDisk 1TB Extreme Portable SSD 1050MB/s R, 1000MB/s W,Upto 2 ~  5.16e6 Comp~
## 3 Aquaguard Aura RO+UV+UF+Taste Adjuster(MTDS) with Active Copper~  2.15e6 Home~
## 4 Boya ByM1 Auxiliary Omnidirectional Lavalier Condenser Micropho~  6.58e5 Musi~
## 5 Casio FX-991ES Plus-2nd Edition Scientific Calculator, Black      8.95e4 Offi~
## 6 Dr Trust Electronic Kitchen Digital Scale Weighing Machine (Blu~  3.95e4 Heal~
## 7 Reffair AX30 [MAX] Portable Air Purifier for Car, Home & Office~  3.14e4 Car&~
## 8 Faber-Castell Connector Pen Set - Pack of 25 (Assorted)           2.86e4 Toys~
## 9 Gizga Essentials Cable Organiser, Cord Management System for PC~  1.79e4 Home~
```

```
# These are the top selling products in each category
```

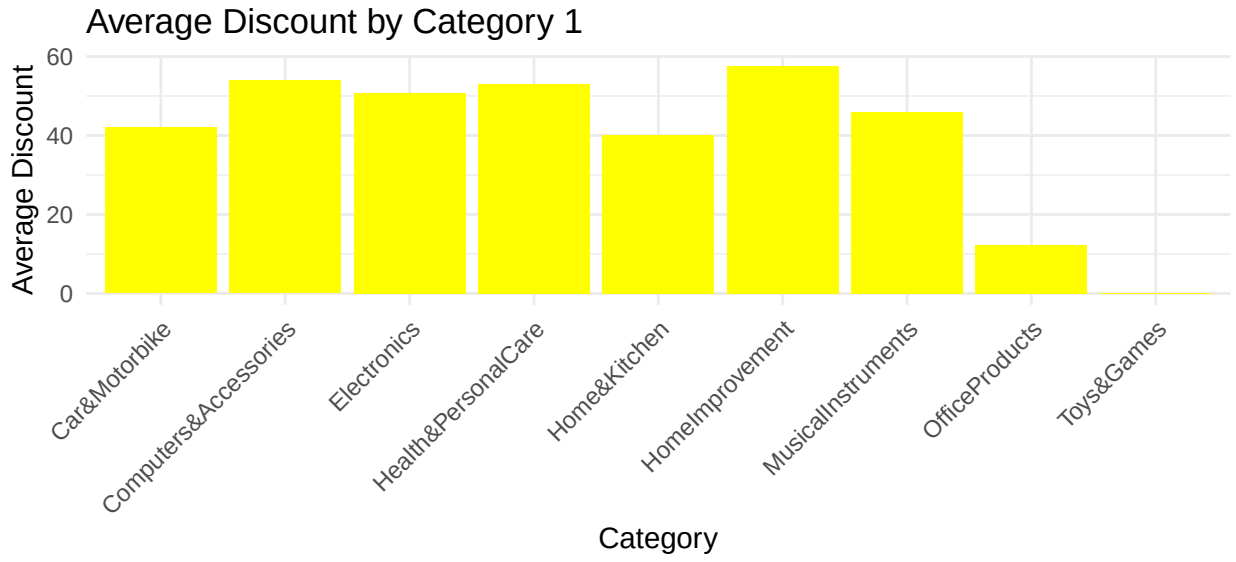
```
# Remove the % symbol and convert 'discount_percentage' to numeric
AmazonData4$discount_percentage <- as.numeric(sub("%", "", AmazonData4$discount_percentage))

# Calculate the average price discount by category
average_discount_by_category <- AmazonData4 %>%
    group_by(Cat1) %>%
    summarize(average_discount_percentage = mean(discount_percentage, na.rm = TRUE)) %>%
    arrange(desc(average_discount_percentage))

average_discount_by_category
```

QUESTION 4: What is the average price discount by Tier 1 category?

```
## # A tibble: 9 x 2
##   Cat1                  average_discount_percentage
```

```
##   <chr>                                <dbl>
## 1 HomeImprovement                       57.5
## 2 Computers&Accessories                 53.9
## 3 Health&PersonalCare                   53
## 4 Electronics                           50.8
## 5 MusicalInstruments                    46
## 6 Car&Motorbike                         42
## 7 Home&Kitchen                          40.2
## 8 OfficeProducts                        12.4
## 9 Toys&Games                             0
```

```
# Home Improvement has the highest discount percentage with 57.5%

# Q 4.2 Create a bar chart for Average Discount % by Category 1
ggplot(average_discount_by_category, aes(x = Cat1, y = average_discount_percentage)) +
    geom_bar(stat = "identity", fill = "yellow") + labs(title = "Average Discount by Category 1",
    x = "Category", y = "Average Discount") + theme_minimal() + theme(axis.text.x = element_text(angle =
    hjust = 1))
```




```
# PDF sidenote - angle = 45, couldn't figure out why this kept
# getting cut off
```

```
library(quanteda)
library(quanteda.textplots)
library(tm)
library(wordcloud)
library(NLP)
conflicts_prefer(quanteda::stopwords)
```

QUESTION 5: Key Word Analysis: What words appear the most frequently in the Review Title and User Reviews section? Visualize this with a wordcloud.

```
## [conflicted] Will prefer quanteda::stopwords over any other package.
```

```
AmazonData4$doc_id <- 1:nrow(AmazonData4)  #Adding unique id for every row

review_title_corpus <- corpus(AmazonData4$review_title, docnames = AmazonData4$doc_id)
review_title_dfm <- dfm(review_title_corpus, remove_punct = TRUE, remove = stopwords("english"),
    )
textplot_wordcloud(review_title_dfm, min_count = 2)
```

Q 5.1: Review Title Wordcloud

```
# Wordcloud based on Review Title, words used at least twice Top
# words: good product, nice, quality, money, price
```

```
review_content_corpus <- corpus(AmazonData4$review_content, docnames = AmazonData4$doc_id)
review_content_dfm <- dfm(review_content_corpus, remove_punct = TRUE, remove = stopwords("english"),
    )
textplot_wordcloud(review_content_dfm, min_count = 3)
```

Q 5.2: Review Content Wordcloud

```
# Wordcloud based on Review Content, words used at least 3 times
# Larger word cloud, but similar top words: good, product, quality,
# price, easy, phone, batter
```

```r
library(tm)
library(quanteda)
library(quanteda.textplots)
library(tm)
library(wordcloud)
library(magrittr)
library(dplyr)
library(knitr)
library(tidyverse)
library(tibble)
library(conflicted)
library(devtools)
library(readr)

# Read in Amazon.csv
azm <- read_csv("amazon.csv")
```

QUESTION 6: Are we able to accurately predict user rating based on key words and price discount percentage?

```
## Rows: 1465 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (14): product_id, product_name, category, discounted_price, actual_price...
## dbl  (1): rating
## num  (1): rating_count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# function to create word cloud from a dataframe
wordCloudFromDataFrame <- function(df_, max_words = 50) {
    df1_ <- df_[df_$review_content != "", ]

    review_content <- as.vector(df1_$review_content)
    review_content <- iconv(review_content, from = "UTF-8", to = "UTF-8",
        sub = "")
    words.vec <- VectorSource(review_content)
    words.corpus <- Corpus(words.vec)
    words.corpus <- tm_map(words.corpus, removePunctuation)
    words.corpus <- tm_map(words.corpus, removeNumbers)
    words.corpus <- tm_map(words.corpus, content_transformer(tolower))
    conflicts_prefer(tm::stopwords)
    words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))

    tdm <- TermDocumentMatrix(words.corpus)
    tdm

    m <- as.matrix(tdm)

    wordCounts <- rowSums(m)
```

```
    totalWords <- sum(wordCounts)
    totalWords
    words <- names(wordCounts)
    head(words)

    wordCounts <- sort(wordCounts, decreasing = TRUE)
    length(wordCounts)
    wordCounts <- head(wordCounts, max_words)
    length(wordCounts)
    head(wordCounts)

    cloudFrame <- data.frame(word = names(wordCounts), freq = wordCounts)
    suppressWarnings(wordcloud(cloudFrame$word, cloudFrame$freq))
}

wordCloudFromDataFrame(azm, 200)
```

```
## [conflicted] Removing existing preference.
## [conflicted] Will prefer tm::stopwords over any other package.
```


```
# function to get word counts
wordCounts <- function(df_, max_words = 50) {
    df1_ <- df_[df_$review_content != "", ]

    review_content <- as.vector(df1_$review_content)
    review_content <- iconv(review_content, from = "UTF-8", to = "UTF-8",
        sub = "")
    words.vec <- VectorSource(review_content)
    words.corpus <- Corpus(words.vec)
    words.corpus <- tm_map(words.corpus, removePunctuation)
    words.corpus <- tm_map(words.corpus, removeNumbers)
    words.corpus <- tm_map(words.corpus, content_transformer(tolower))
    conflicts_prefer(tm::stopwords)
    words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
```

```
    tdm <- TermDocumentMatrix(words.corpus)
    tdm

    m <- as.matrix(tdm)

    wordCounts <- rowSums(m)

    totalWords <- sum(wordCounts)
    totalWords
    words <- names(wordCounts)
    head(words)

    wordCounts <- sort(wordCounts, decreasing = TRUE)
    length(wordCounts)
    wordCounts <- head(wordCounts, max_words)
    return(wordCounts)
}

# display total occurrences of each word in the corpus
wc <- wordCounts(azm)
```

```
## [conflicted] Removing existing preference.
## [conflicted] Will prefer tm::stopwords over any other package.
```

```
wc
```

```
##     good  product  quality      use      can      one    cable     like
##     4485     2800     2080     1492     1426     1269     1233     1155
##    price     will     also    using    phone charging  battery     easy
##     1147     1141     1138      951      948      871      772      759
##     time     just     well  working      buy    watch    sound      get
##      748      747      737      728      683      669      668      649
##     used     even   better    works    great   really     dont     best
##      637      635      593      591      567      565      560      559
##      now     fast      got     much    water     nice   camera     need
##      515      507      488      474      470      451      450      449
##   amazon    money    power  overall     fine   screen     work      bit
##      448      440      431      427      426      426      419      417
##   little     long
##      412      407
```

```
# count total reviews above 3
azm_pos <- azm[azm$rating > 3, ]
nrow(azm_pos)
```

```
## [1] 1455
```

```
# count total reviews below 3
azm_neg <- azm[azm$rating < 3, ]
nrow(azm_neg)
```

```
## [1] 7
```

```
# The data Amazon provides in this is strongly biased. There are only
# 7 reviews below 3, but 1455 reviews above 3
```

```
# display word cloud of positive reviews
wordCloudFromDataFrame(azm_pos, 50)
```

```
## [conflicted] Removing existing preference.
## [conflicted] Will prefer tm::stopwords over any other package.
```


```
# display word cloud of negative reviews
wordCloudFromDataFrame(azm_neg, 20)
```

```
## [conflicted] Removing existing preference.
## [conflicted] Will prefer tm::stopwords over any other package.
```


```
library(dplyr)
library(stringr)

# function to check for correlation between word and rating
correlate <- function(word) {
```

```
    azm1 <- azm %>%
        mutate(word_present = as.numeric(str_detect(review_content, fixed(word))))

    azm1 <- azm1 %>%
        filter(!is.na(rating), !is.na(word_present))

    correlation <- cor(azm1$rating, azm1$word_present)
    return(correlation)
}

# find the word with the strongest correlation for rating
max(c(correlate("good"), correlate("product"), correlate("quality"), correlate("use"),
    correlate("can"), correlate("one"), correlate("cable"), correlate("like"),
    correlate("price"), correlate("will"), correlate("also"), correlate("using"),
    correlate("phone"), correlate("charging"), correlate("battery"), correlate("easy"),
    correlate("time"), correlate("just"), correlate("well"), correlate("working"),
    correlate("buy"), correlate("watch")))
```

```
## [1] 0.1292382
```

```
# the word 'good' has the strongest correlation with a value of 0.129
```

```
cor(AmazonData4$rating, AmazonData4$rating_count)
```

QUESTION 7: Are good or bad user feelings about a product more likely to generate a high volume or ratings and reviews? Are users more motivated to write a good product review or a bad product review?

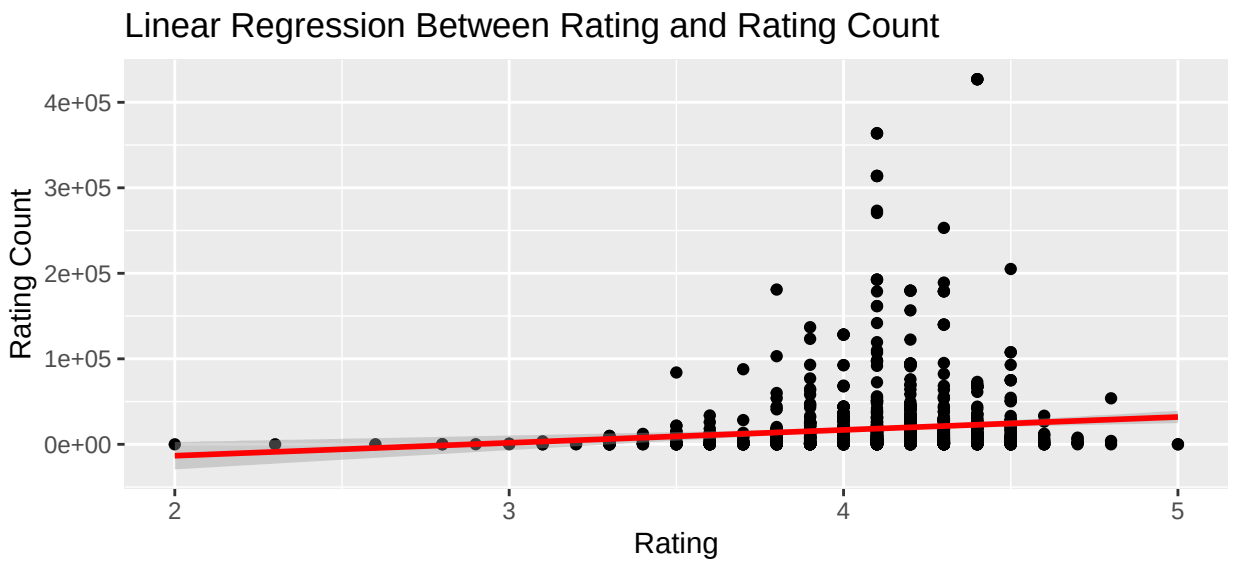
```
## [1] 0.1022348
```

```
# 0.1022348

# Yes, there is a weak positive correlation between rating and
# rating_count of 0.1022348.  The higher the product rating, the more
# likely the buyer is to rate that product, which means a higher
# rating count for that product.

lm1 <- lm(rating_count ~ rating, data = AmazonData4)
lm1
```

```
##
## Call:
## lm(formula = rating_count ~ rating, data = AmazonData4)
##
## Coefficients:
## (Intercept)       rating
##      -43564        15103
```

```r
# Linear regression between rating and rating_count
ggplot(data = AmazonData4, aes(x = rating, y = rating_count)) + geom_point() +
    geom_smooth(method = "lm", color = "red") + labs(title = "Linear Regression Between Rating and Rati
    x = "Rating", y = "Rating Count")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
summary(lm1)
```

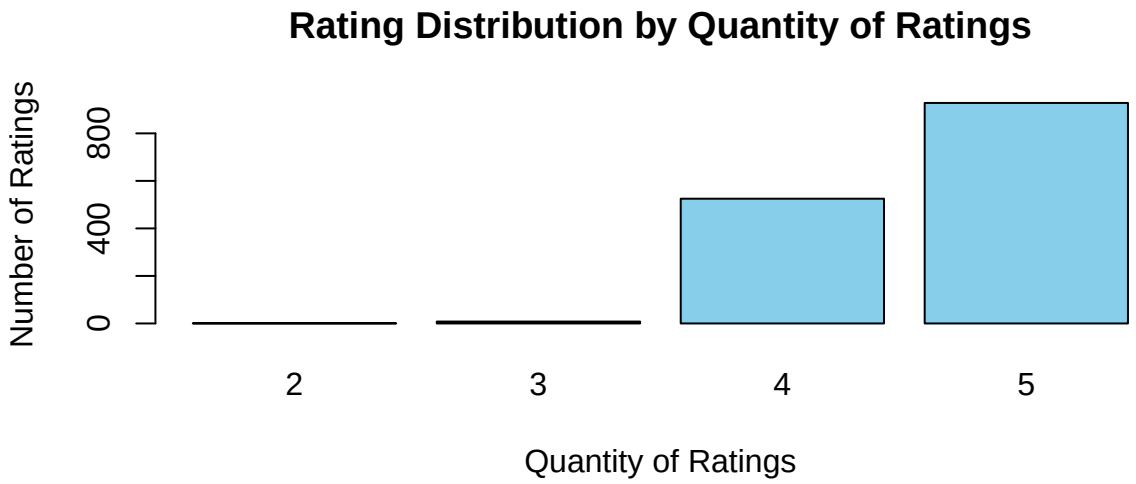
```
##
## Call:
## lm(formula = rating_count ~ rating, data = AmazonData4)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -31944 -16641 -11609   -294 404085
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -43564      15795  -2.758  0.00589 **
## rating         15103       3846   3.927    9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42560 on 1460 degrees of freedom
## Multiple R-squared:  0.01045,    Adjusted R-squared:  0.009774
## F-statistic: 15.42 on 1 and 1460 DF,  p-value: 9.004e-05
```

```r
# Correlation Coefficient: 0.1022348 Adjusted R Squared = 0.009774
# p-value: 9.004e-05, therefore the relationship is statistically
# significant

# The p-value suggests there is a statistically significant
```

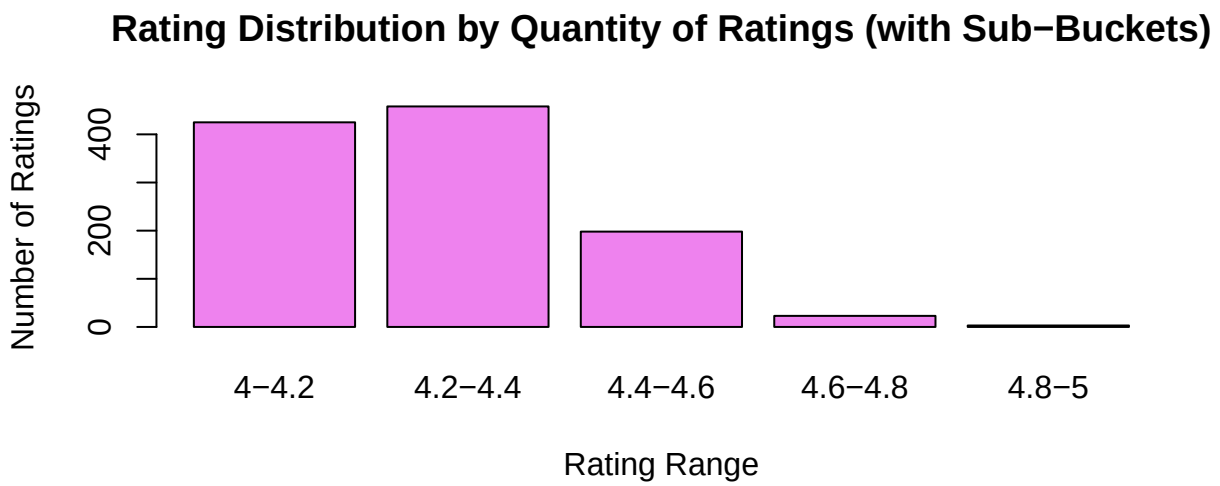
```r
# relationship between rating and rating count The correlation
# coefficient suggests that rating is a weak predictor for rating
# count.  The Adjusted R-squared value suggests that rating accounts
# for a small amount of the variability in rating count
```

```r
# Step 1: Create Rating Buckets
AmazonData4$rating_bucket <- cut(AmazonData4$rating, breaks = seq(0, 5,
    by = 1), labels = FALSE)

# Step 2: Count the Number of Ratings in Each Bucket
rating_counts <- table(AmazonData4$rating_bucket)

# Step 3: Visualize the Distribution
barplot(rating_counts, main = "Rating Distribution by Quantity of Ratings",
    xlab = "Quantity of Ratings", ylab = "Number of Ratings", col = "skyblue")
```

QUESTION 8: How are ratings distributed based on quantity of ratings? Bucket all ratings (0 - 1, 1 - 2, 2 - 3, etc) and visualize this distribution.




```r
# Step 4: Deeper dive into ratings 4 - 5

# Step 4.1: Create Rating Buckets with Sub-Buckets
breaks <- seq(4, 5, by = 0.2)
AmazonData4$rating_bucket <- cut(AmazonData4$rating, breaks = breaks, labels = FALSE,
    right = FALSE)

# Step 4.2: Count the Number of Ratings in Each Bucket
rating_counts <- table(AmazonData4$rating_bucket)

# Step 4.3: Visualize the Distribution
barplot(rating_counts, main = "Rating Distribution by Quantity of Ratings (with Sub-Buckets)",
    xlab = "Rating Range", ylab = "Number of Ratings", col = "violet",
    names.arg = paste(breaks[-length(breaks)], breaks[-1], sep = "-"))
```




```
# Most ratings fall between 4.2 and 4.4.
```

QUESTION 9: Is there a way to estimate the actual number of sales based on the available data here? We found that the best we could do with this data was to use the ratings count * discounted_price_usd formula which we've used in previous questions. While there are lightly positive correlations present in the data, there isn't enough to make a confident guess into the actual number of sales based on the data available in this dataset.

```
cor(AmazonData4$discount_percentage, AmazonData4$rating_count)
```

```
## [1] 0.01129439
```

```
# 0.01129439
cor(AmazonData4$discount_percentage, AmazonData4$rating)
```

```
## [1] -0.155679
```

```
#-0.155679

# We expected that the greater the discount percentage, the higher
# the rating would be and the higher the rating count would be, ie we
# expected a positive and stronger correlation between discount
# percentage and rating, as well as discount percentage and rating
# count. Contrary to what we expected, the resulting correlation was
# actually very weak and negative


cor(AmazonData4$rating, AmazonData4$rating_count)
```

```
## [1] 0.1022348
```

```
# 0.1022348

# We though that the higher ratings would be conducive to higher
```

```
# rating counts, that is to say we expected a positive and stronger
# correlation between rating and rating_count We were correct that
# there was a positive correlation, but the correlation was much
# weaker than we expected


cor(AmazonData4$discounted_price_usd, AmazonData4$rating_count)
```

```
## [1] -0.02730249
```

```
#-0.02730249
cor(AmazonData4$actual_price_usd, AmazonData4$rating_count)
```

```
## [1] -0.03621571
```

```
#-0.03621571

# The correlation between discounted_price_usd and rating_count, as
# well as actual_price_usd and rating_count were both what we
# expected. We thought online shoppers would expect different prices
# for different items, so We did not expect price alone be a
# significant factor in rating or rating_count. If there would be a
# correlation at all, it would probably be negative, because nobody
# wants to pay more.


cor(AmazonData4$discounted_price_usd, AmazonData4$rating)
```

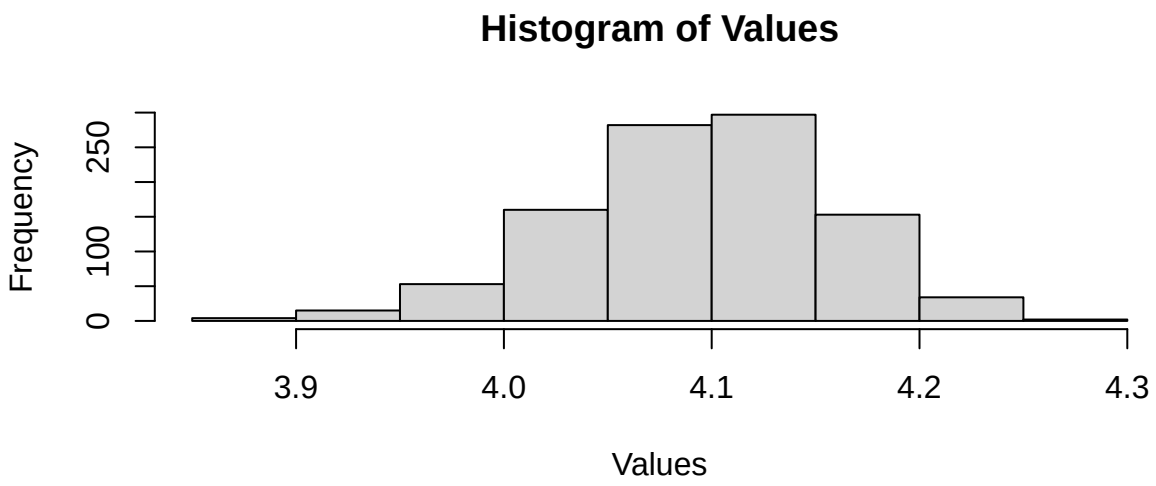
```
## [1] 0.1211309
```

```
# 0.1211309
cor(AmazonData4$actual_price_usd, AmazonData4$rating)
```

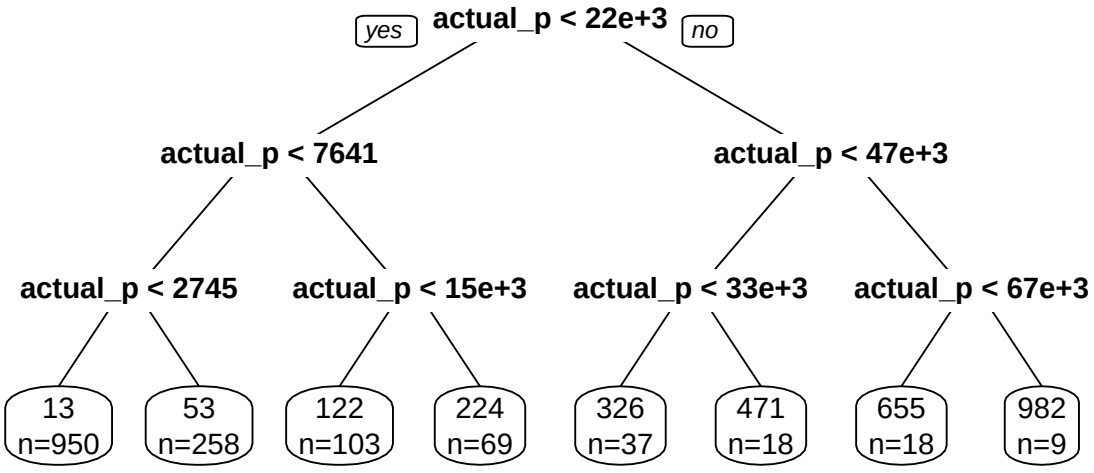
```
## [1] 0.1224666
```

```
# 0.1224666

# We expected these results to be the same as the above, but there
# was actually a positive correlation between discounted_price_usd
# and rating as well as actual_price_usd and
# ratingvgyukfdgmccccccccccccccccccccccvh the correlation was weak
# as we expected, but we didn't expect it to be positive for the same
# reason mentioned above.


# The last 4 results are very weak, but surprisingly consistent.
```

```
# Inferential Statistics#

# Generate sample means from rating column and plot histogram
Values <- replicate(1000, mean(sample(AmazonData3$rating, 22, TRUE)))
hist(Values)
```




```
# Support Vector Machine#
library(tidyverse)
library(caret)
library(rpart)
library(rpart.plot)
library(kernlab)
library(Metrics)


# Now to create a data frame with all the numeric variables#
AmazonData5 <- data.frame(discounted_price = AmazonData3$discounted_price,
    actual_price = AmazonData3$actual_price, rating = AmazonData3$rating,
    rating_count = AmazonData3$rating_count, discount_price_usd = AmazonData3$discounted_price_usd,
    actual_price_usd = AmazonData3$actual_price_usd)

# Decision tree to predict actual_price_usd using all other
# attributes
cartTree <- rpart(actual_price_usd ~ ., data = AmazonData5)
prp(cartTree, extra = 1)
```




```
# Calculate the importance of each variable
t <- varImp(cartTree)
```

```
t %>%
    arrange(desc(Overall)) %>%
    slice(1:5)
```

```
##                       Overall
## actual_price       4.8406624
## discount_price_usd 3.7262290
## discounted_price   3.7262290
## rating_count       0.3816159
## rating             0.1507718
```

```
t
```

```
##                       Overall
## actual_price       4.8406624
## discount_price_usd 3.7262290
## discounted_price   3.7262290
## rating             0.1507718
## rating_count       0.3816159
```

```
# Actual price highest importance score of 4.8406624\t discounted
# price and discount_price_usd had equal scores of 3.7262290\t
# rating_count: 0.3816159 rating: 0.1507718\t


# separate training and testing data
trainList <- createDataPartition(y = AmazonData5$actual_price_usd, p = 0.6,
    list = FALSE)
training <- AmazonData5[trainList, ]
testing <- AmazonData5[-trainList, ]

# train the decision tree model to predict rating
model.rpart <- train(rating ~ ., data = training, method = "rpart", preProc = c("center",
    "scale"))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
model.rpart
```

```
## CART
##
## 878 samples
##   5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 878, 878, 878, 878, 878, 878, ...
## Resampling results across tuning parameters:
##
##   cp          RMSE       Rsquared    MAE
```

```
##   0.01320751  0.3062290  0.06470787  0.2249845
##   0.02241070  0.3023149  0.06561912  0.2211282
##   0.08820741  0.3068114  0.03669775  0.2210875
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.0224107.
```

```
# test the accuracy of the model
predicted <- predict(model.rpart, newdata = testing)

# measure the average difference between the predicted values and the
# testing values
mae <- mae(testing$rating, predicted)
mae
```

```
## [1] 0.2084136
```

```
# 0.2150884
```

QUESTION 10: Based on the answers to questions 1, 2 and 3, pick the Tier 1 category with the most user activity. Now continue that analysis down from every subcategory, tier 2 - tier 6. What new takeaways are there from this detailed analysis? Are there any outliers in the data that can be identified?

Q 10.1: Electronics is the Tier 1 category we have selected

```
electronics <- AmazonData4 %>%
    filter(Cat1 == "Electronics")

electronics_revenue <- electronics %>%
    group_by(Cat2) %>%
    summarize(total_revenue = sum(discounted_price_usd, na.rm = TRUE)) %>%
    arrange(-total_revenue)

electronics_revenue
```

Q 10.2: Finding subcategories

```
## # A tibble: 9 x 2
##   Cat2                                    total_revenue
##   <chr>                                           <dbl>
## 1 HomeTheater,TV&Video                           20232.
## 2 Mobiles&Accessories                            13783.
## 3 WearableTechnology                              2134.
## 4 Headphones,Earbuds&Accessories                   751.
## 5 HomeAudio                                        297.
## 6 Cameras&Photography                              244.
## 7 Accessories                                      136.
## 8 GeneralPurposeBatteries&BatteryChargers           64.5
## 9 PowerAccessories                                  15.5
```

```r
# Home Theater, TV and Video is the Electronics subcategory that
# earns the most revenue

homeTheater <- electronics %>%
    filter(Cat2 == "HomeTheater,TV&Video")

homeTheater_revenue <- homeTheater %>%
    group_by(Cat3) %>%
    summarize(total_revenue = sum(discounted_price_usd, na.rm = TRUE)) %>%
    arrange(-total_revenue)

homeTheater_revenue
```

```
## # A tibble: 5 x 2
##   Cat3                   total_revenue
##   <chr>                          <dbl>
## 1 Televisions                  19296.
## 2 Accessories                    510.
## 3 Projectors                     360.
## 4 SatelliteEquipment              41.6
## 5 AVReceivers&Amplifiers          23.9
```

```r
# Televisions is the Home Theater category that earns the most
# revenue

televisions <- homeTheater %>%
    filter(Cat3 == "Televisions")

tv_revenue <- televisions %>%
    group_by(Cat4) %>%
    summarize(total_revenue = sum(discounted_price_usd, na.rm = TRUE)) %>%
    arrange(-total_revenue)

tv_revenue
```

```
## # A tibble: 2 x 2
##   Cat4                total_revenue
##   <chr>                       <dbl>
## 1 SmartTelevisions          18779.
## 2 StandardTelevisions         517.
```

```r
# Smart TV is the Televisions category that earns the most revenue
# This is the end of the category tier for this line of products
```

```r
unique(AmazonData4$Cat1)
```

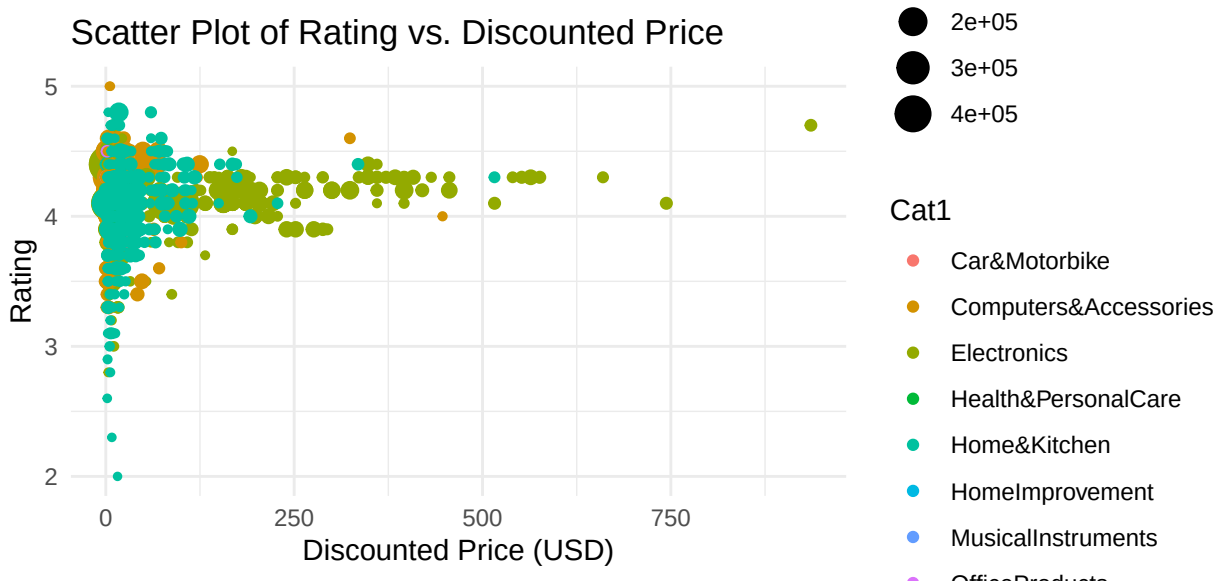
10.3: All Subcategories

```
## [1] "Computers&Accessories" "Electronics"           "MusicalInstruments"
## [4] "OfficeProducts"        "Home&Kitchen"          "HomeImprovement"
## [7] "Toys&Games"            "Car&Motorbike"         "Health&PersonalCare"
```

```
ComputersAccessories <- AmazonData4 %>%
    filter(Cat1 == "Computers&Accessories")

MusicalInstruments <- AmazonData4 %>%
    filter(Cat1 == "MusicalInstruments")

OfficeProducts <- AmazonData4 %>%
    filter(Cat1 == "OfficeProducts")

HomeKitchen <- AmazonData4 %>%
    filter(Cat1 == "Home&Kitchen")

HomeImprovement <- AmazonData4 %>%
    filter(Cat1 == "HomeImprovement")

ToysGames <- AmazonData4 %>%
    filter(Cat1 == "Toys&Games")

CarMotorbike <- AmazonData4 %>%
    filter(Cat1 == "Car&Motorbike")

HealthPersonalCare <- AmazonData4 %>%
    filter(Cat1 == "Health&PersonalCare")
```

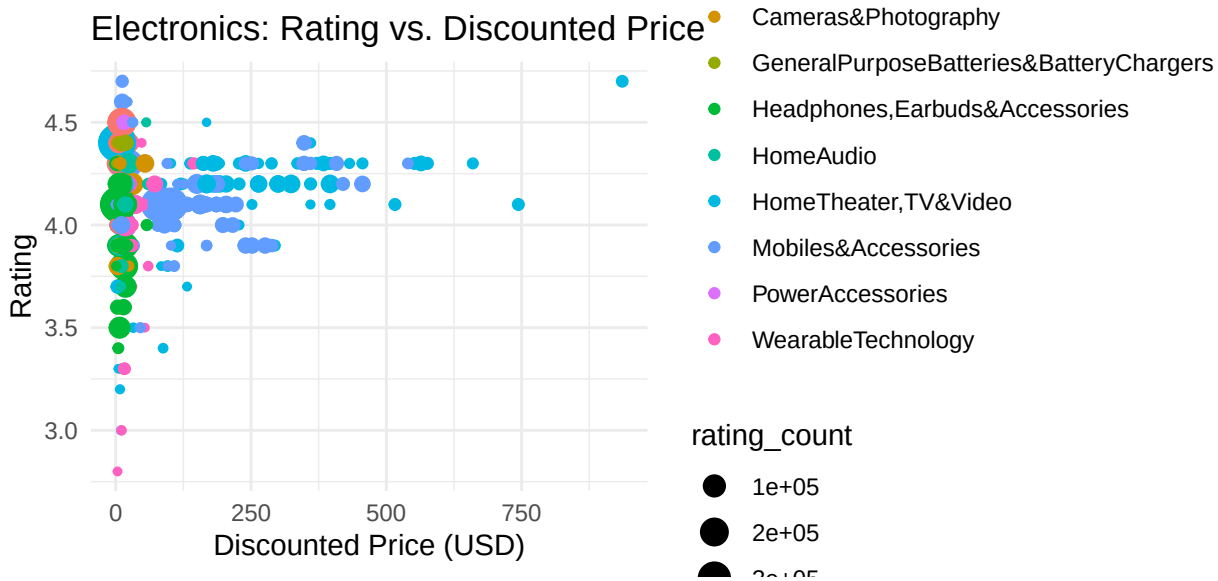
```
ggplot(data = AmazonData4, aes(x = discounted_price_usd, y = rating, color = Cat1,
    size = rating_count)) + geom_point() + labs(title = "Scatter Plot of Rating vs. Discounted Price",
    x = "Discounted Price (USD)", y = "Rating") + theme_minimal()
```

10.4: Further Analysis



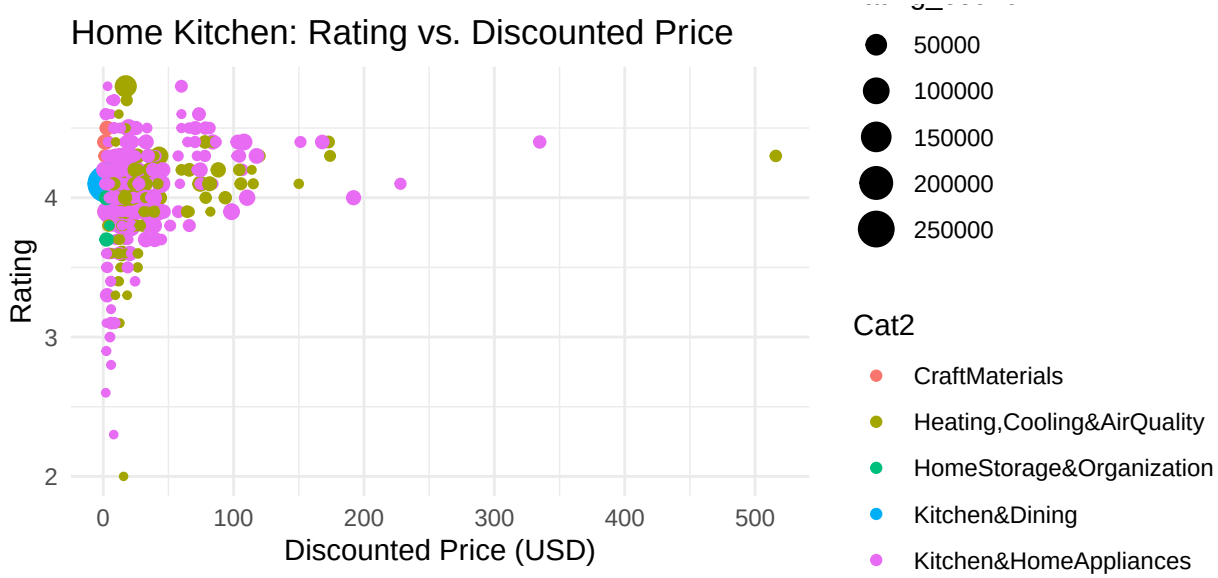

```
# Takeaways: Electronics has the highest priced products. The cheaper
# the product, the higer the ratings count. The average rating for
# most products falls between 4 and 4.5.
```

```
ggplot(data = electronics, aes(x = discounted_price_usd, y = rating, color = Cat2,
    size = rating_count)) + geom_point() + labs(title = "Electronics: Rating vs. Discounted Price",
    x = "Discounted Price (USD)", y = "Rating") + theme_minimal()
```

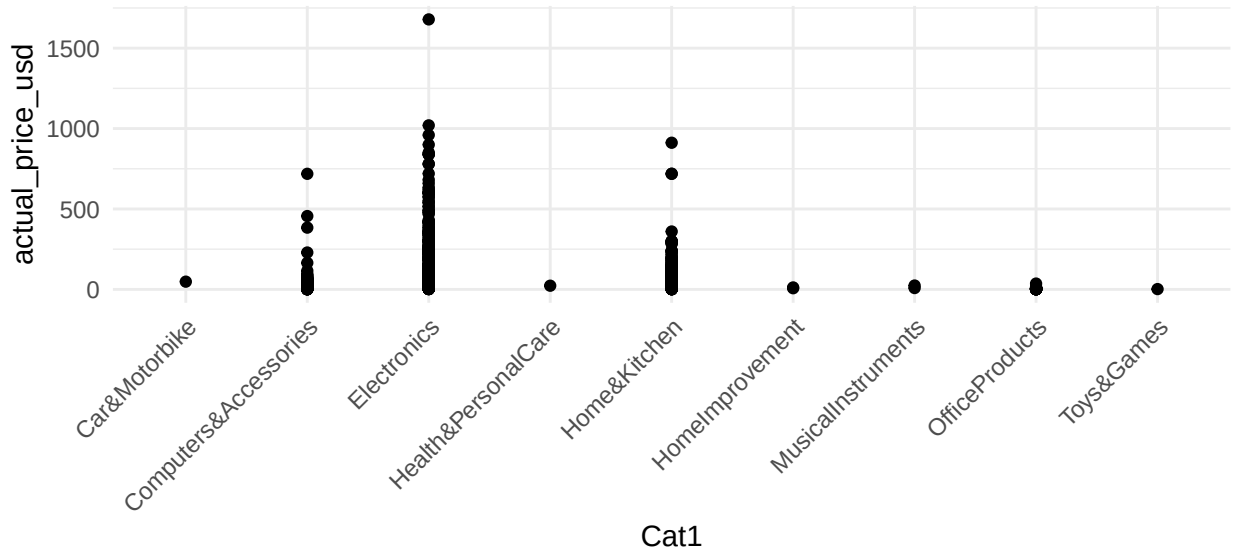



```
# Takeaways: Home Theater has the most expensive products. Mobiles
# get the highest volume of ratings. Most ratings fall between 4 and
# 4.4.
```

```
ggplot(data = HomeKitchen, aes(x = discounted_price_usd, y = rating, color = Cat2,
    size = rating_count)) + geom_point() + labs(title = "Home Kitchen: Rating vs. Discounted Price",
    x = "Discounted Price (USD)", y = "Rating") + theme_minimal()
```




```
# Price and Category Simple Linear Regression
ggplot(AmazonData4, aes(x = Cat1, y = actual_price_usd)) + geom_point() +
    theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



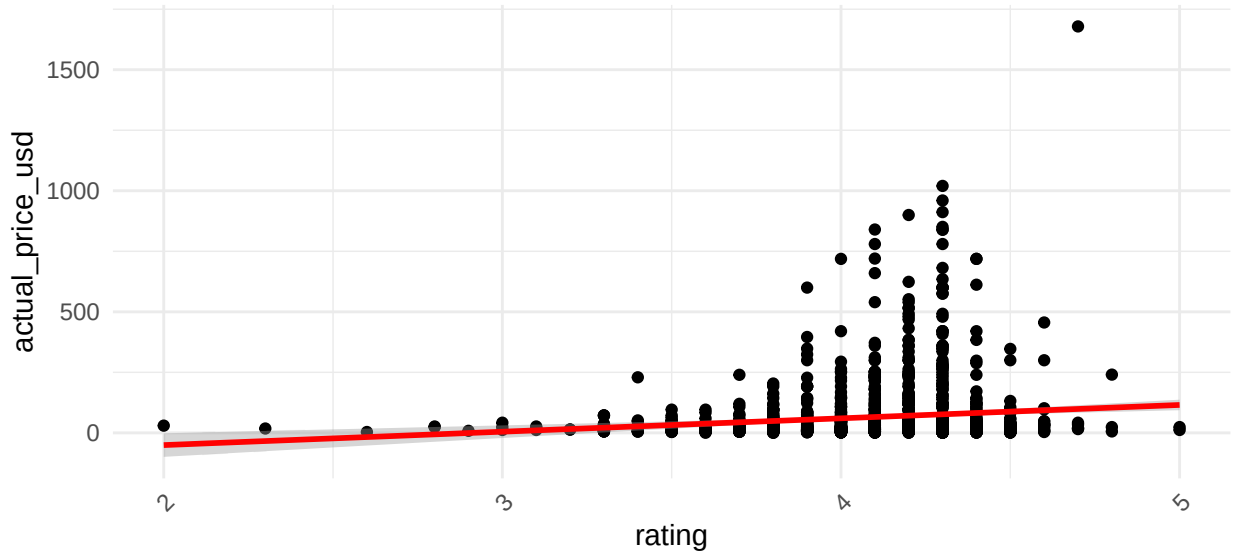

```
Ama_Cat = lm(actual_price_usd ~ Cat1, AmazonData4)

summary(Ama_Cat)
```

```
##
## Call:
## lm(formula = actual_price_usd ~ Cat1, data = AmazonData4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -119.48  -44.00  -14.25    1.79 1557.27
```

```
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    48.00     123.36   0.389    0.697
## Cat1Computers&Accessories     -27.76     123.49  -0.225    0.822
## Cat1Electronics                73.53     123.47   0.596    0.552
## Cat1Health&PersonalCare       -25.20     174.45  -0.144    0.885
## Cat1Home&Kitchen                1.99     123.50   0.016    0.987
## Cat1HomeImprovement           -38.41     151.08  -0.254    0.799
## Cat1MusicalInstruments        -31.84     151.08  -0.211    0.833
## Cat1OfficeProducts            -43.23     125.33  -0.345    0.730
## Cat1Toys&Games                -46.20     174.45  -0.265    0.791
##
## Residual standard error: 123.4 on 1453 degrees of freedom
## Multiple R-squared:  0.1129, Adjusted R-squared:  0.108
## F-statistic: 23.12 on 8 and 1453 DF,  p-value: < 2.2e-16
```

```
# Price and Rating Simple Linear Regression

ggplot(AmazonData4, aes(x = rating, y = actual_price_usd)) + geom_point() +
    stat_smooth(method = "lm", col = "red") + theme_minimal() + theme(axis.text.x = element_text(angle =
    hjust = 1))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
Ama_Rat = lm(actual_price_usd ~ rating, AmazonData4)

summary(Ama_Rat)
```

```
##
## Call:
## lm(formula = actual_price_usd ~ rating, data = AmazonData4)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -103.35  -58.43  -42.58   -1.65 1580.03
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -160.92      48.13  -3.344 0.000848 ***
## rating         55.25      11.72   4.715 2.65e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 129.7 on 1460 degrees of freedom
## Multiple R-squared:  0.015,  Adjusted R-squared:  0.01432
## F-statistic: 22.23 on 1 and 1460 DF,  p-value: 2.649e-06
```

```
model1 <- lm(rating ~ discount_percentage + actual_price_usd + discounted_price_usd,
    data = AmazonData4)

summary(model1)
```

```
##
## Call:
## lm(formula = rating ~ discount_percentage + actual_price_usd +
##     discounted_price_usd, data = AmazonData4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.08810 -0.13923  0.02957  0.18094  0.98280
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.1928934  0.0205688 203.847  < 2e-16 ***
## discount_percentage  -0.0023766  0.0003935  -6.040 1.95e-09 ***
## actual_price_usd      0.0007828  0.0002311   3.388 0.000724 ***
## discounted_price_usd -0.0009079  0.0003704  -2.451 0.014348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2841 on 1458 degrees of freedom
## Multiple R-squared:  0.03919,    Adjusted R-squared:  0.03721
## F-statistic: 19.82 on 3 and 1458 DF,  p-value: 1.351e-12
```

```
# Adjusted R Squared = 0.037, suggests that the independent variables
# (discount percentage, actual price and discounted price) are not
# doing a good job explaining the variability in the rating.

# Looking at the P Values, we can assume that: 1. As the discount
# percentage INCREASES, the rating tends to DECREASE 2. As the actual
# price INCREASES, the rating tends to also INCREASE 3. As the
# discounted price INCREASES, the rating tends to DECREASE Overall,
# based on the coefficients and p-values, it seems that the discount
# percentage and the actual price have the strongest impact on an
# Amazon product's rating compared to the discounted price.
```