Project Report

Project Report on Sales of summer clothes on E-commerce Wish in July 2020 in United Kingdom (UK)

Prepared by Daksh Patel and Zishan Visram for DATA-602 project - Winter 2023

Introduction

At one point in history, a pair of good, clean clothes were considered a necessity, but times have changed. Shopping and especially buying new clothes is a leisure activity for many people. This is the domain we are tackling in this project. We have analyzed the e-commerce sale for July 2020 of summer clothes on the Wish.com website. Wish is an e-commerce company that sells varieties of clothing as well as accessories and is based in the UK. We will answer a few guiding questions that made us curious at the beginning of the project and will summarize them using statistics.

Guiding Questions

1. **Is a product sold more just because it is priced competitively?**

It is not wrong to assume that pricing a product low can lead to good sales. However, since economic conditions of people improve, priorities that govern a person's buying choices change. We find out here that is price a factor in the number of units sold for products? We perform proportion test for the same.

2. **Does the color of the clothes affect the sales?**

Choosing a color is a subjective thing for most but certain colors might be preferred by people more than others. Since we are looking at summer data, we will be dividing the colors into "warm" and "cool" categories and figure out if selling clothes from a certain category of colors can bring in more sales.

3. **What factors affect product ratings?**

Ratings are an important marker for a customer to see if a product is good or not. Hence, to know from where a good rating count can be achieved is crucial for a company. We compare ratings for products based on their shipping prices, number of shipping countries, product has been given an ad-boost or not, if it has special badges that signify if that product is of good quality, availability of fast shipping or products being made locally. We performed AOVA, linear regression and permutation test to answer this question.

Data Source and Preparation

The dataset used is taken from Kaggle. It is a public dataset and has been permitted for general use. Essentially, this data finds its origin from Wish Uk's e-commerce platform. It is a subset of total sales data showing the summer sales for July 2020, and contains 43 columns and 1574 rows.

As per the requirements of the individual guiding questions, we perform data cleaning and wrangling differently.

Question 1: Proportion Test

Is a product sold more just because it is priced competitively?

Price is an important factor a customer considers before buying a product. Through a prop test, we will see how important actual price is towards the sales of a product.

Reading data:\ (data:\)

```
library(ggplot2)
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected
by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(dplyr)
```

```
sales_data_df <- read.csv(file = '~/Desktop/MDSA/602/Project/summer-products-with-rating
-and-performance_2020-08.csv')
```

```
head(sales_data_df)
```

1
2
3
4
5
6

6 rows | 1-1 of 44 columns

```
unique(sales_data_df[c("units_sold")])
```

	units_sold <int>
1	100
2	20000
4	5000
6	10
7	50000
8	1000
17	10000
18	100000
61	50
127	1

1-10 of 15 rows Previous **1** 2 Next

We find that the units_sold column has the following unique values: 1, 2, 3, 6, 7, 8, 10, 50, 100, 1000, 5000, 10000, 20000, 50000, 100000.

We divide this data into two categories: products that are sold less than 100 and products that are sold more than 5000. This will make a good distinction between products' proportions based on their sales value.

```
sales_data_df %>% group_by(units_sold) %>%tally()
```

units_sold <int>	n <int>
1	3
2	2
3	2
6	1
7	2
8	4
10	49
50	76
100	509
1000	405

1-10 of 15 rows Previous **1** 2 Next

We will compare prices of products that have been sold less than 100 units with products that have been sold more than 5000 units. We perform a proportion test for these two categories.

Based on these, our hypothesis will be:

H0: Products that were sold less than 100 units are priced equally as products that were sold more than 5000 units.

p100sold = p5000sold

HA: Products that are sold more than 5000 units are priced differently than products that are sold less than 100.

p100sold != p5000sold

Carrying out proportion test

1. Data Preparation

Filtering out required columns.

```
q1_df = data.frame(price = sales_data_df$price, retail_price = sales_data_df$retail_pric
e, units_sold = sales_data_df$units_sold)
```

```
head(q1_df)
```

	price <dbl>	retail_price <int>	units_sold <int>
1	16.00	14	100
2	8.00	22	20000
3	8.00	43	100
4	8.00	8	5000
5	2.72	3	100
6	3.92	9	10

6 rows

Combining “units_sold” with values less than 100 into the category of 100 product_sold.

```
q1_df$units_sold <- factor(q1_df$units_sold)
```

```
levels(q1_df$units_sold)
```

```
##  [1] "1"      "2"      "3"      "6"      "7"      "8"      "10"     "50"
##  [9] "100"    "1000"   "5000"   "10000"  "20000"  "50000"  "100000"
```

```
levels(q1_df$units_sold) <- list("100"=c("1", "2", "3", "6", "7", "8", "10", "50", "10
0"),"5000"=c("5000", "10000", "20000", "50000", "100000"))
```

```
q1_df %>% group_by(units_sold) %>%tally()
```

units_sold <fct>	n <int>
100	648
5000	520
NA	405

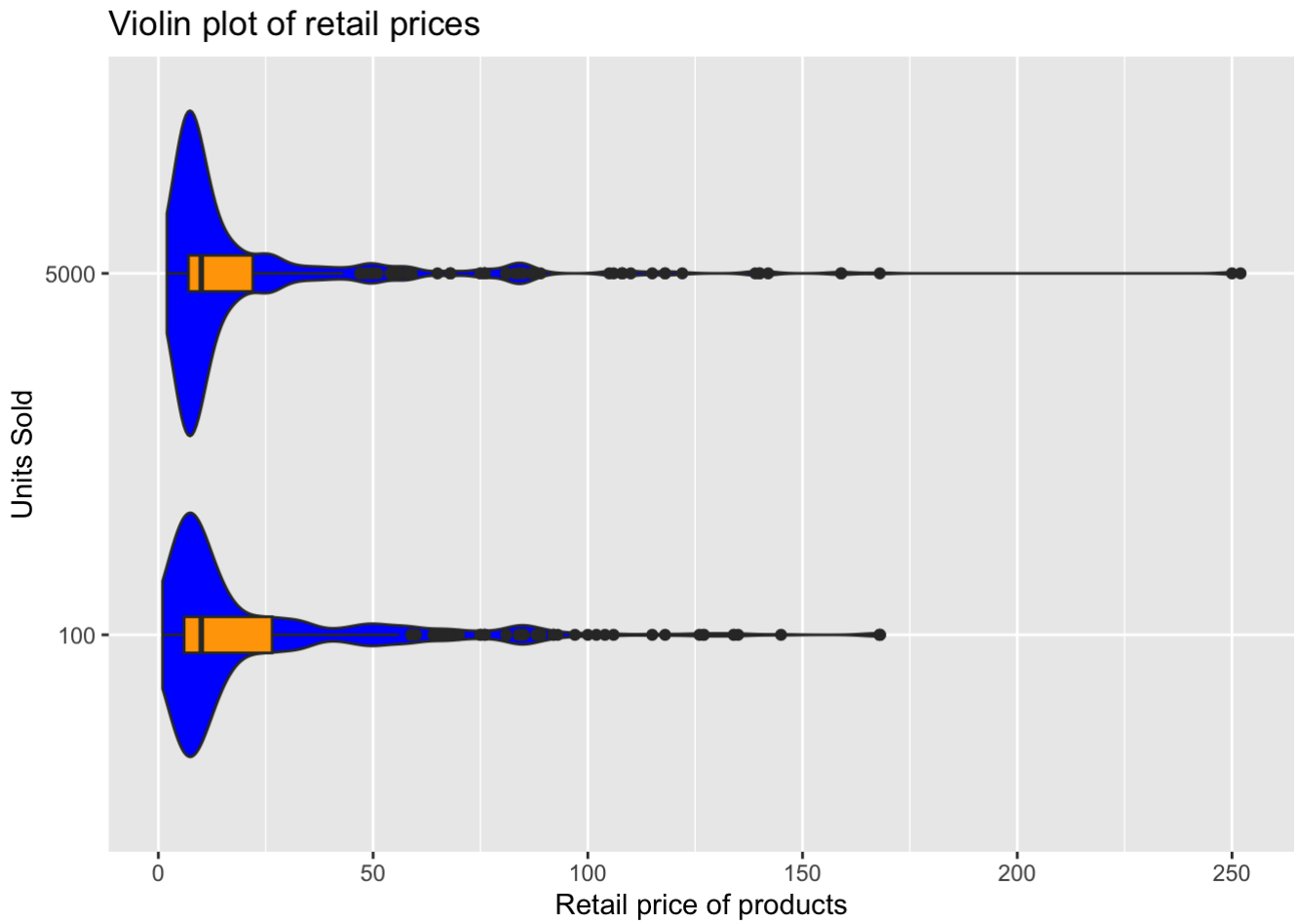
3 rows

Removing NA values.

```
q1_df <- na.omit(q1_df)
```

- Plotting a violin plot for this distribution

```
ggplot(data=q1_df, aes(x = units_sold, y = retail_price)) + geom_violin(fill="blue") + g
eom_boxplot(width = 0.1, fill="orange") + xlab("Units Sold") + ylab("Retail price of pro
ducts") + ggtitle("Violin plot of retail prices") + coord_flip()
```



Creating distribution of bootstrap statistics for the difference in means.

```
ntrials = 1000
n.sold_100 = favstats(~retail_price|units_sold, data=q1_df)$n[1]
n.sold_5000 = favstats(~retail_price|units_sold, data=q1_df)$n[2]
n.sold_100
```

```
## [1] 648
```

```
n.sold_5000
```

```
## [1] 520
```

```
mean.sold_100 = numeric(ntrials)
mean.sold_5000 = numeric(ntrials)
diffmeanssold = numeric(ntrials)
units_sold_100 = filter(q1_df, units_sold=="100")
units_sold_5000 = filter(q1_df, units_sold=="5000")
```

```
head(units_sold_100)
```

	price	retail_price	units_sold
	<dbl>	<int>	<fct>
1	16.00	14	100
2	8.00	43	100
3	2.72	3	100
4	3.92	9	100
5	11.00	84	100
6	6.00	8	100

6 rows

```
head(units_sold_5000)
```

	price	retail_price	units_sold
	<dbl>	<int>	<fct>
1	8.00	22	5000
2	8.00	8	5000
3	7.00	6	5000
4	5.78	22	5000
5	2.00	2	5000
6	11.00	10	5000

6 rows

```
for(i in 1:ntrials)
{   mean.sold_100[i] = mean(sample(units_sold_100$retail_price, n.sold_100, replace=TRU
E))
    mean.sold_5000[i] = mean(sample(units_sold_5000$retail_price, n.sold_5000, replace=T
RUE))
    diffmeanssold[i] = mean.sold_100[i] - mean.sold_5000[i]
}
boot_diffmeanssold = data.frame(mean.sold_100, mean.sold_5000, diffmeanssold)
```

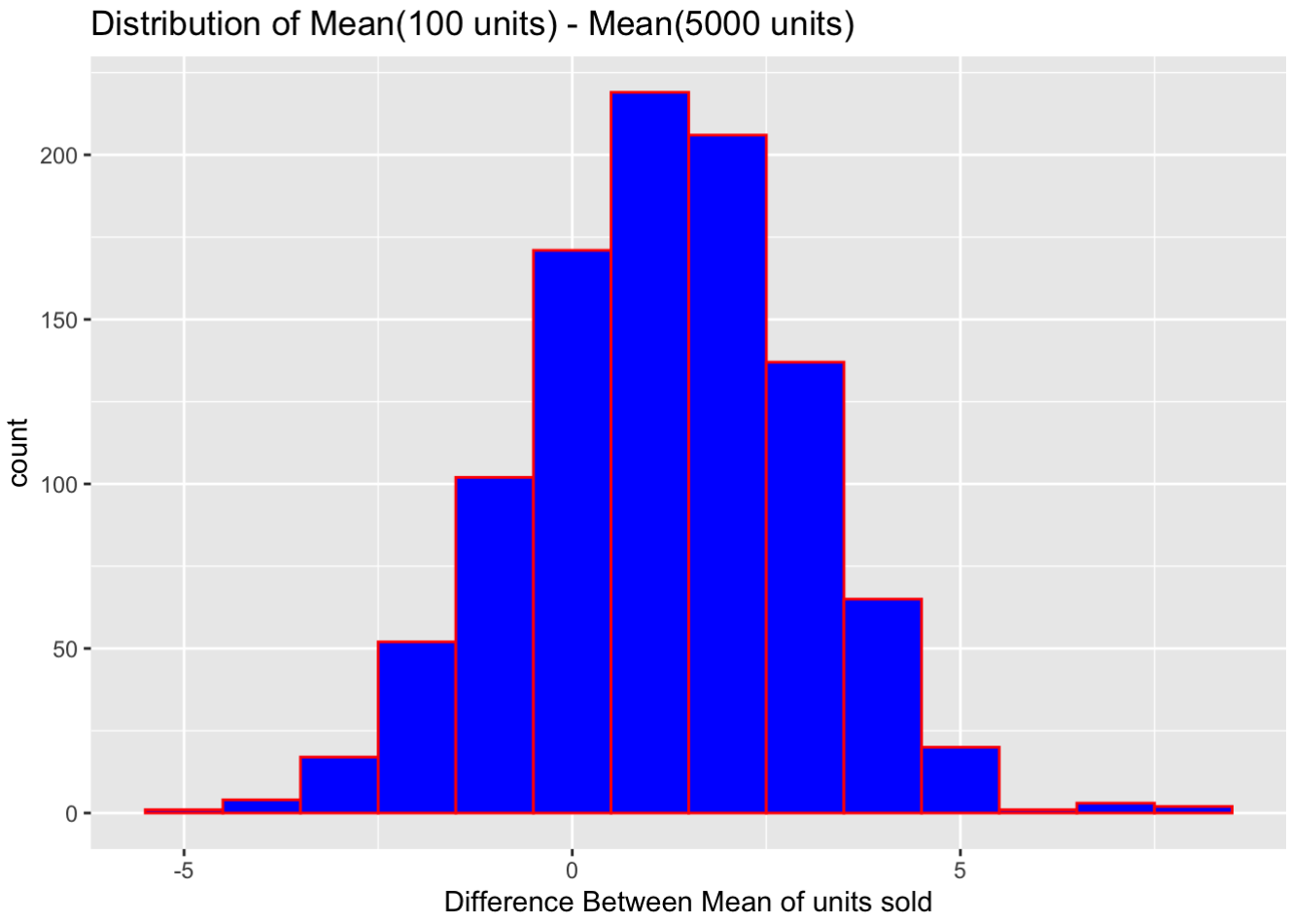
```
head(boot_diffmeanssold)
```

	mean.sold_100	mean.sold_5000	diffmeanssold
	<dbl>	<dbl>	<dbl>
1	23.35185	20.74038	2.61146724
2	23.92901	20.65577	3.27324311

	mean.sold_100 <dbl>	mean.sold_5000 <dbl>	diffmeanssold <dbl>
3	24.57099	24.65192	-0.08093542
4	23.04784	22.36731	0.68053181
5	24.40123	21.25577	3.14546534
6	23.49228	20.46346	3.02882241

6 rows

```
ggplot(data=boot_diffmeanssold, aes(x = diffmeanssold)) + geom_histogram(fill='blue', co
l='red', binwidth=1) + xlab("Difference Between Mean of units sold") + ggtitle("Distribu
tion of Mean(100 units) - Mean(5000 units)")
```



Building 95% confidence interval.

```
qdata(~ diffmeanssold, c(0.025, 0.975), data=boot_diffmeanssold)
```

```
##      2.5%     97.5%
## -2.459427  4.530100
```

Since 0 falls under the 95% confidence interval, this points to our null hypothesis being true.

Since we have significantly more number of sample count, we can validate our confidence interval using a t-test.

```
t.test(~retail_price|units_sold, conf.level=0.95, alternative = "two.sided", var.equal=F
ALSE, q1_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  retail_price by units_sold
## t = 0.68137, df = 1069, p-value = 0.4958
## alternative hypothesis: true difference in means between group 100 and group 5000 is
not equal to 0
## 95 percent confidence interval:
##  -2.277774  4.701230
## sample estimates:
##  mean in group 100 mean in group 5000
##           23.31173           22.10000
```

This validates our proportion test, and we get a p-value of 0.4958 with a t-value of 0.6813 and degree of freedom of 1069.

Having a p-value greater than the significance level of 0.05, we can accept our null hypothesis.

Hence, we can conclude here that mean prices of products that were sold less than 100 is equal to mean prices of products that were sold more than 5000.

Therefore, we infer that price is not the main factor customers focus on while making a purchase. Sure, it is an important aspect people pay attention to but there might be other factors that are more prevailing than price.

Question 2: Chi-Squared Test

Does the color of the clothes affect the sales?

Choosing one color for clothing over the other is purely subjective and based on the customer's choice. But some colors might be picked more often than others. To find this we perform simple visual analysis as well as chi-squared test.

Getting all the unique color of products:

```
product_color<-sales_data_df[,c('units_sold',"product_color")]
```

```
head(product_color)
```

	units_sold	product_color
	<int>	<chr>
1	100	white
2	20000	green
3	100	leopardprint

	units_sold <int>	product_color <chr>
4	5000	black
5	100	yellow
6	10	navyblue

6 rows

```
unqiue_product_color<-distinct(product_color)
```

```
dim(unqiue_product_color)[1]
```

```
## [1] 270
```

Observing the different colors, we see a total of 270 different colors which can be difficult to analyze which product color affects sales. Therefore, we decided to group the colors in separate categories (warm vs cool) colors to help us make a better prediction.

We will create a category for warm colors which will include the colors: red, orange and yellow, and the second category will be cool colors that will include all other colors such as pink, purple, blue etc.

```
product_color$color_category <- ifelse(product_color$product_color
%in% c('red', 'orange','yellow'), "warm colors", "cool colors")
```

Now, that we have grouped the product colors into two categories of warm and cool colors, we can take a look at the distribution.

```
head(product_color)
```

	units_sold <int>	product_color <chr>	color_category <chr>
1	100	white	cool colors
2	20000	green	cool colors
3	100	leopardprint	cool colors
4	5000	black	cool colors
5	100	yellow	warm colors
6	10	navyblue	cool colors

6 rows

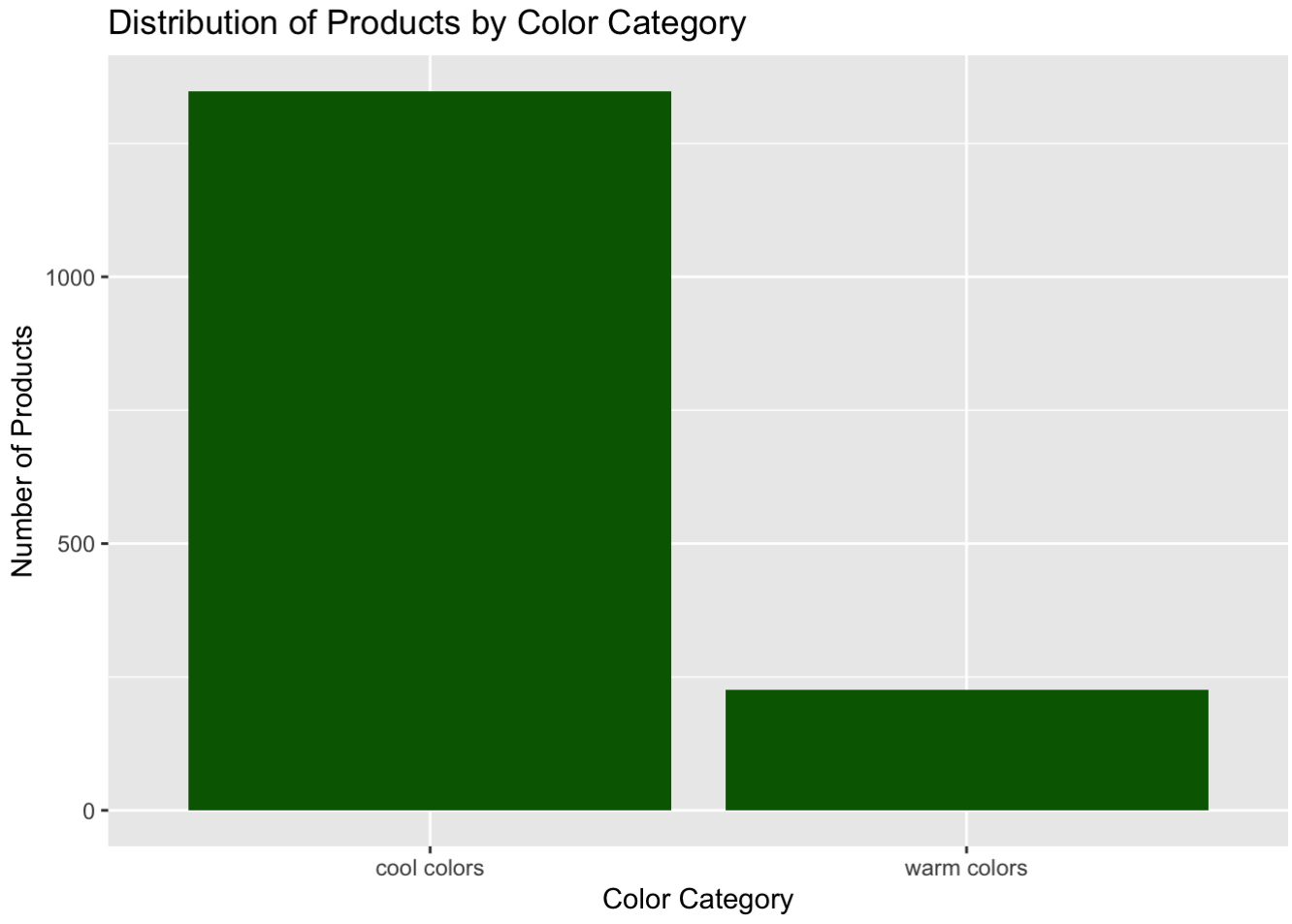
```
color_dist <- product_color %>%
  group_by(color_category) %>%
  summarize(total_units_sold = sum(units_sold), num_products=n())
color_dist
```

color_category <chr>	total_units_sold <int>	num_products <int>
cool colors	6115935	1348
warm colors	709320	225

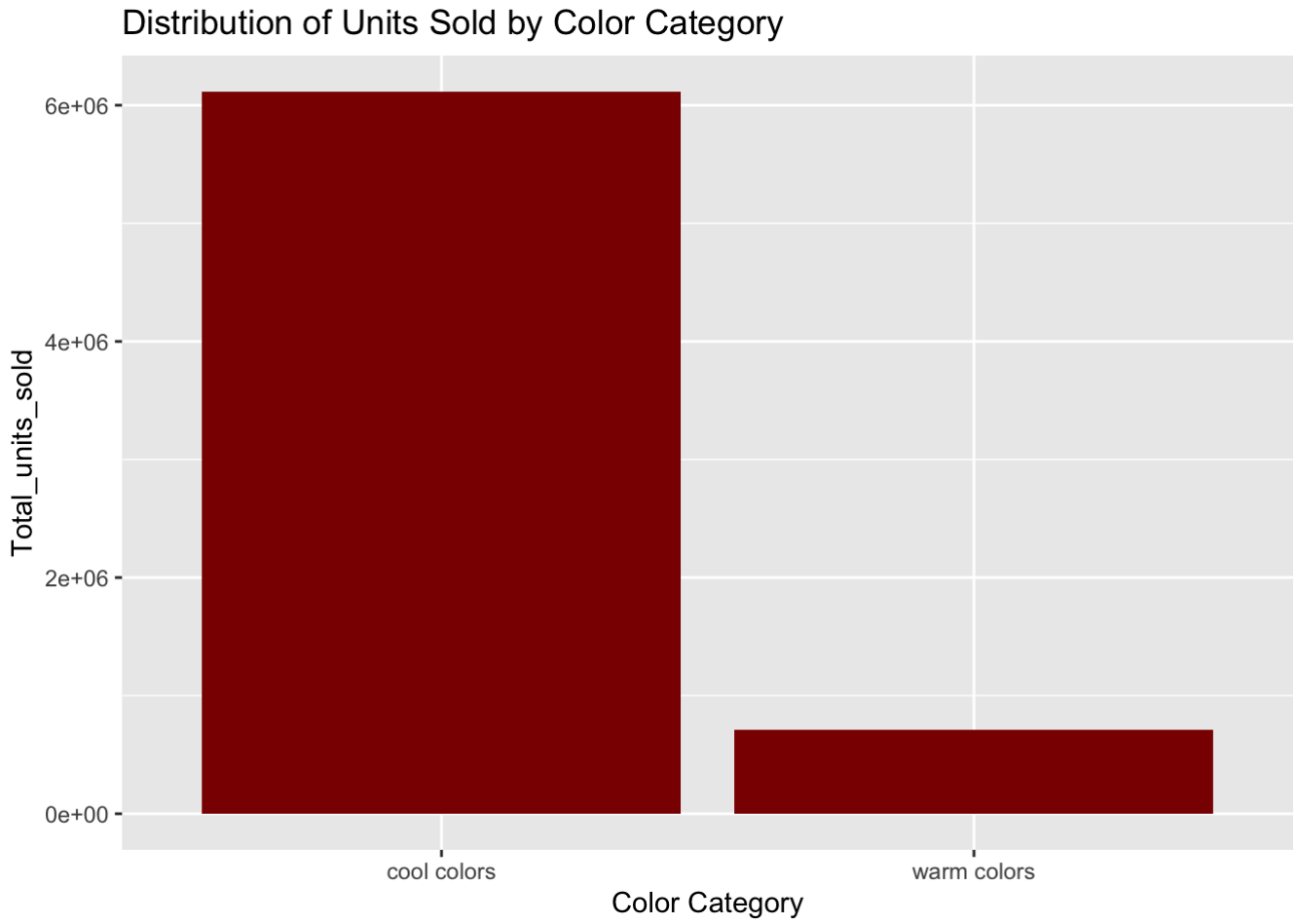
2 rows

We can now have a better idea on the difference in the color distribution and can see that there are more cool colors sold compared to the warm colors. There are also a lot more products with cool colors compared to the warm colors.

```
ggplot(color_dist, aes(x = color_category, y = num_products)) + geom_bar(stat = "identit
y", fill = 'darkgreen') + labs(x = "Color Category", y = "Number of Products", title =
"Distribution of Products by Color Category")
```



```
ggplot(color_dist, aes(x = color_category, y =total_units_sold)) + geom_bar(stat = "iden
tity", fill = 'darkred') + labs(x = "Color Category", y = "Total_units_sold", title = "D
istribution of Units Sold by Color Category")
```



Based on the two bar plots above, we can make our statistical hypothesis:

H0: There is no difference in the total units sold between the two different color categories.

HA: There is a difference in the total units sold between the different color categories.

We will now attempt to see the relationship between units sold and color category using Chi-squared test:

```
sales_color=tally(~units_sold + color_category, data=product_color )
sales_color
```

```
##           color_category
## units_sold cool colors warm colors
##     1                2           1
##     2                1           1
##     3                2           0
##     6                1           0
##     7                1           1
##     8                4           0
##     10              48           1
##     50              64          12
##     100            422          87
##     1000           335          70
##     5000           195          22
##     10000          158          19
##     20000           94           9
##     50000           16           1
##     100000           5           1
```

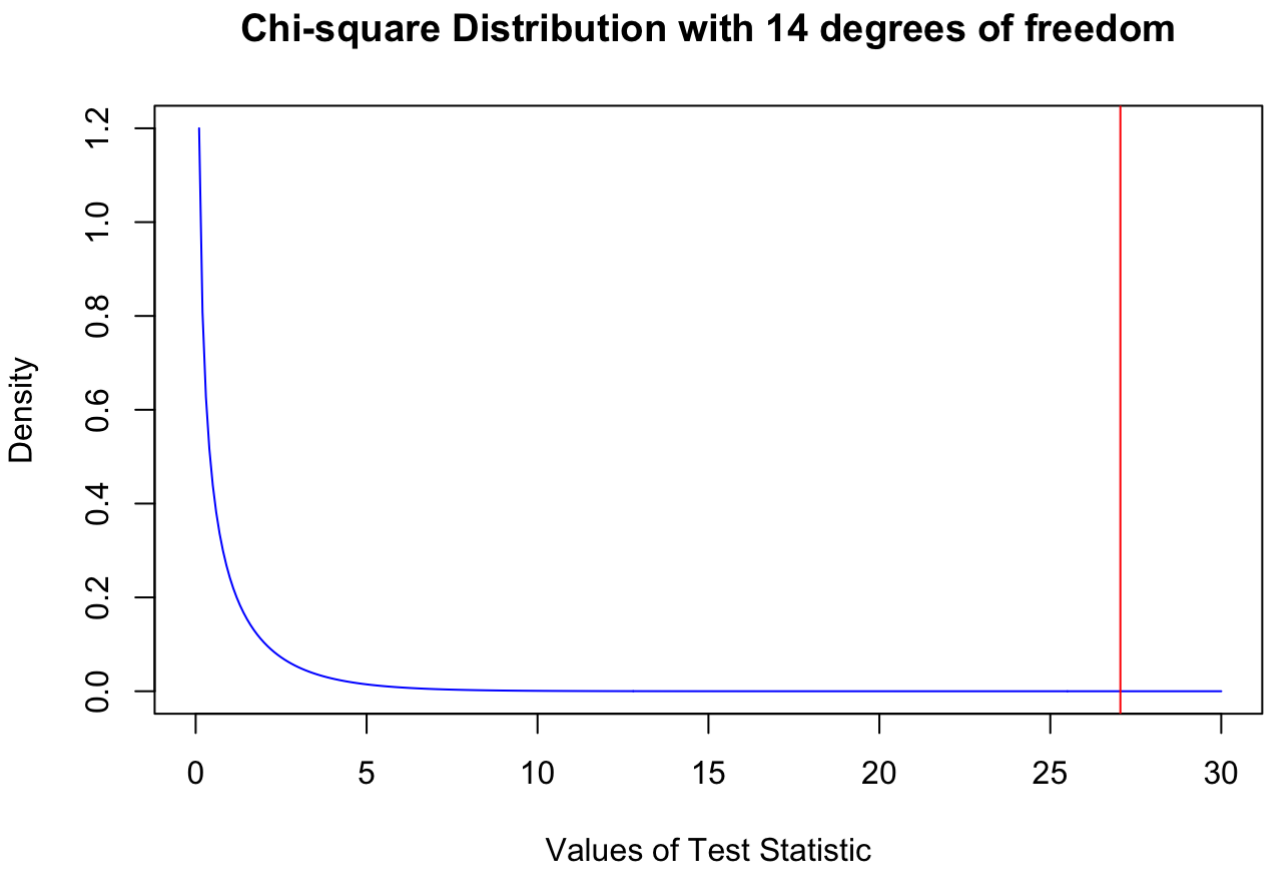
```
chisq.test(sales_color,correct=FALSE)
```

```
## Warning in chisq.test(sales_color, correct = FALSE): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  sales_color
## X-squared = 27.05, df = 14, p-value = 0.01897
```

Computing the Chi-squared test of the two categorical variables, we see that our result for the p-value is 0.01897 which is less than 0.05, therefore, we can reject our Null hypothesis, and accept our alternative Hypothesis. This means that units sold and color category are not independent of each other.

```
chivalues = seq(0, 30, 0.1)
plot(chivalues, dchisq(chivalues,1), xlab="Values of Test Statistic", ylab="Density",typ
e="l", col="blue", main="Chi-square Distribution with 14 degrees of freedom")
abline(v=27.05, col='red')
```



We can now compare the mean units sold between the two color categories using a t-test

```
t.test(units_sold ~ color_category, data = product_color)
```

```
##
##  Welch Two Sample t-test
##
## data:  units_sold by color_category
## t = 2.2123, df = 322.87, p-value = 0.02765
## alternative hypothesis: true difference in means between group cool colors and group
warm colors is not equal to 0
## 95 percent confidence interval:
##   153.297 2615.724
## sample estimates:
## mean in group cool colors mean in group warm colors
##                  4537.044                  3152.533
```

Based on our t-test results, we see that p-value is 0.02765 which is lower than our signi6icance threshold of 0.05, which means that we can reject our Null Hypothesis, and accept our Alternative Hypothesis. Therefore, there is a difference in total units sold between the two color category'). We can also be 95% con6ident that the difference in the mean units sold between the two color category is between 153.297 and 2615.724.

Question 3: AOVA, Linear Regression, Permutation Test

What affects product rating more- ad boosting or product badges (such as good quality, fast shipping, local made)?

Ratings are a crucial part of the research a customer does before buying a product. Now, after a product has been purchased, customers can rate the product based on quality, shipping experience, their satisfaction after the use of the product, etc.

We consider all the mentioned factors one by one and will check if the formulated null hypothesis holds true or not.

1. Shipping cost

We start by looking at if shipping affects the ratings of a product.

H0: Average product ratings are the same between all the shipping price ranges.

HA: Average product ratings are less for higher price ranges compared to low shipping price.

```
ratings_to_shipping<-sales_data_df[,c('rating','shipping_option_price')]
```

```
head(ratings_to_shipping)
```

	rating <dbl>	shipping_option_price <int>
1	3.76	4
2	3.45	2
3	3.57	3
4	4.03	2
5	3.10	1
6	5.00	1

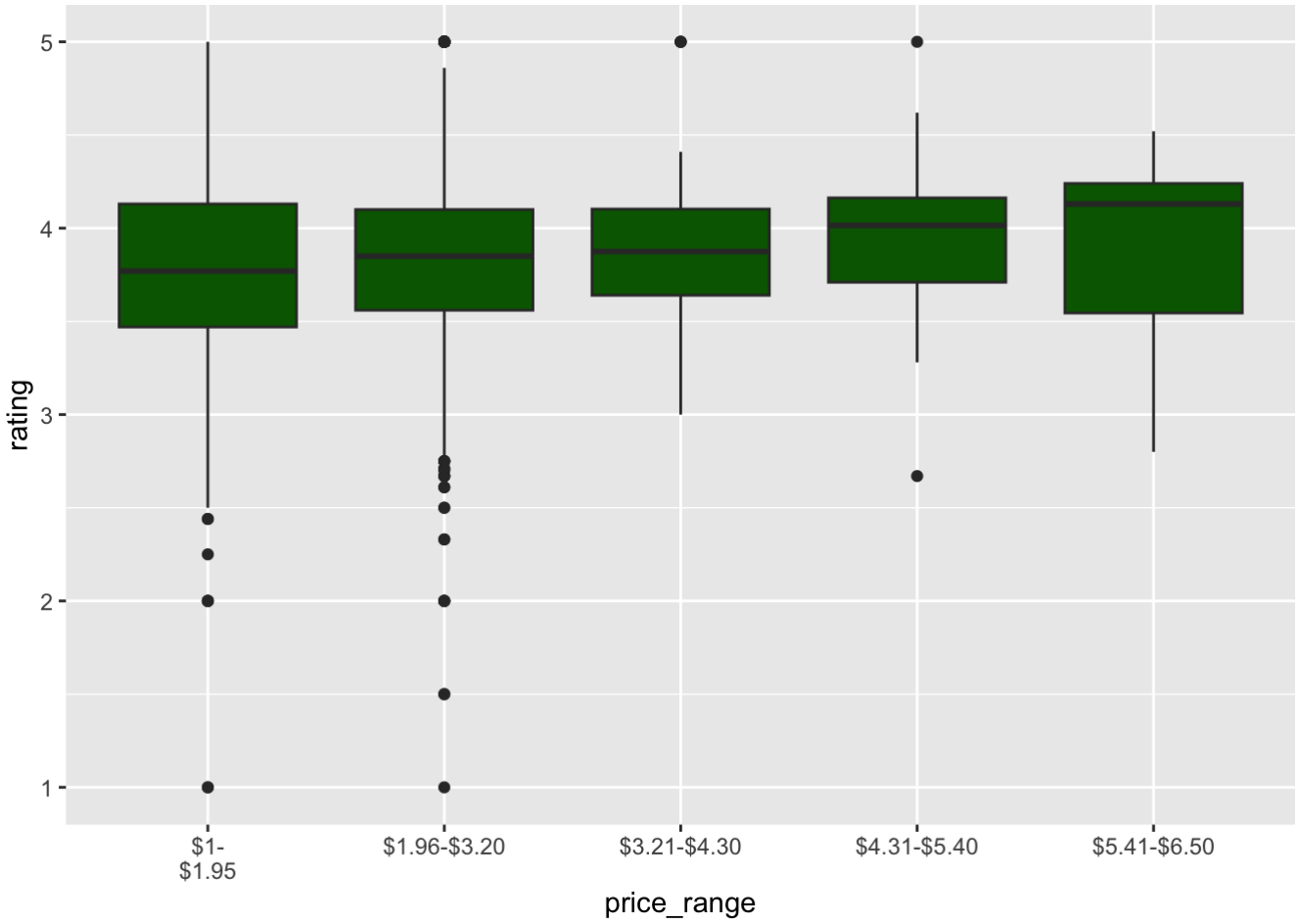
6 rows

```
ratings_to_shipping <- ratings_to_shipping %>%
  mutate(price_range = case_when(
    shipping_option_price >= 1 & shipping_option_price <= 1.95 ~ '$1-
$1.95',
    shipping_option_price > 1.95 & shipping_option_price <= 3.20 ~
'$1.96-$3.20',
    shipping_option_price > 3.20 & shipping_option_price <= 4.30 ~
'$3.21-$4.30',
    shipping_option_price > 4.30 & shipping_option_price <= 5.40 ~
'$4.31-$5.40',
    shipping_option_price > 5.40 & shipping_option_price <= 6.50 ~
'$5.41-$6.50'
))
```

```
ratings_to_shipping_filtered <- ratings_to_shipping %>%
  filter(!is.na(price_range))
```

Visually analyzing shipping price range with the ratings:

```
ggplot(ratings_to_shipping_filtered, aes(x = price_range, y = rating)) + geom_boxplot(fi
ll='darkgreen')
```



Although there are differences in the ratings between the shipping price ranges, we cannot say they are significantly different.

ANOVA test:

```
anova <- aov(rating ~ price_range, data =
ratings_to_shipping_filtered)
summary(anova)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## price_range    4    1.2  0.3039   1.148  0.332
## Residuals   1562  413.6  0.2648
```

Performing ANOVA test, we get p-value = 0.332. Hence, we can accept our null hypothesis that average ratings are the same for all shipping price range.

2. Number of countries shipped

Since we were not able to find any significant relationship between ratings to shipping price range, we wanted to dig further, and see if the ratings were affected by the countries the product was shipping to.

H0: Ratings is not affected by the number of countries a product is shipped to.

HA: Ratings differ based on the number of countries a product is shipped to.

```
ratings_to_shipped_countries <-
sales_data_df[,c('rating','countries_shipped_to')]
head(ratings_to_shipped_countries)
```

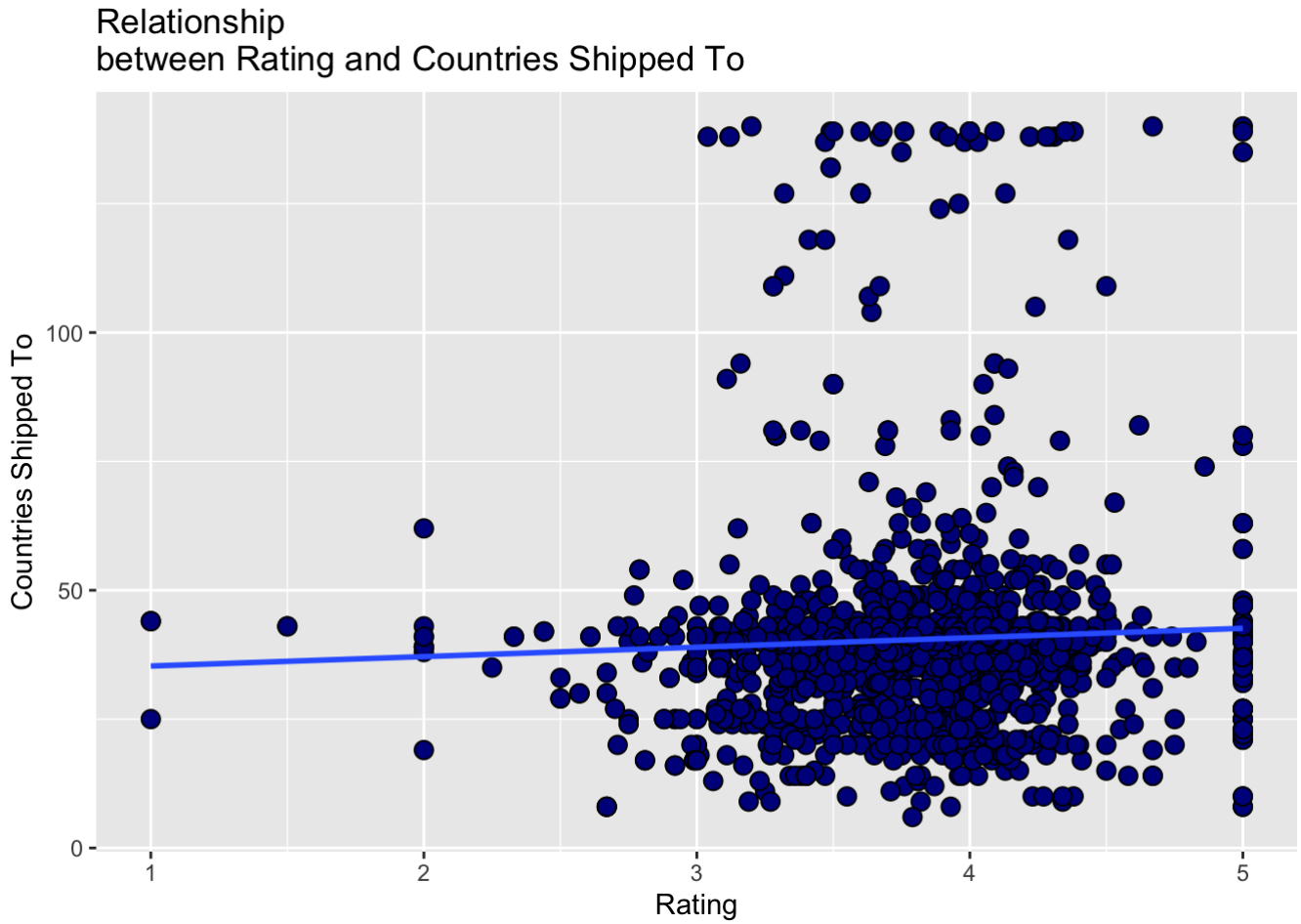
	rating <dbl>	countries_shipped_to <int>
1	3.76	34
2	3.45	41
3	3.57	36
4	4.03	41
5	3.10	35
6	5.00	40

6 rows

Visually analyzing relation between ratings and number of countries a product ships:

```
ggplot(data=ratings_to_shipped_countries,aes(x=rating,y=countries_shipped_to)) + geom_po
int(fill='darkblue',size=3,shape=21) + labs(x = "Rating", y = "Countries Shipped To", ti
tle = "Relationship
between Rating and Countries Shipped To") + geom_smooth(method = "lm", se = FALSE)
```

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



Finding correlation between ratings and the number of countries shipped,

```
correlation <- cor(ratings_to_shipped_countries$rating,ratings_to_shipped_countries$coun
tries_shipped_to)
correlation
```

```
## [1] 0.04642301
```

We get a weak positive correlation between ratings and shipped countries. Our Null Hypothesis is that there is no significant correlation between the two variables and our Alternative hypothesis is that there is a significant correlation between the variables.

```
model <- lm(countries_shipped_to ~ rating, data = ratings_to_shipped_countries)
summary(model)
```

```
##
## Call:
## lm(formula = countries_shipped_to ~ rating, data = ratings_to_shipped_countries)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.613  -9.259  -0.985   2.344 100.679
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.4693     3.8276   8.744   <2e-16 ***
## rating        1.8287     0.9928   1.842   0.0657 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.29 on 1571 degrees of freedom
## Multiple R-squared:  0.002155,   Adjusted R-squared:  0.00152
## F-statistic: 3.393 on 1 and 1571 DF,  p-value: 0.06566
```

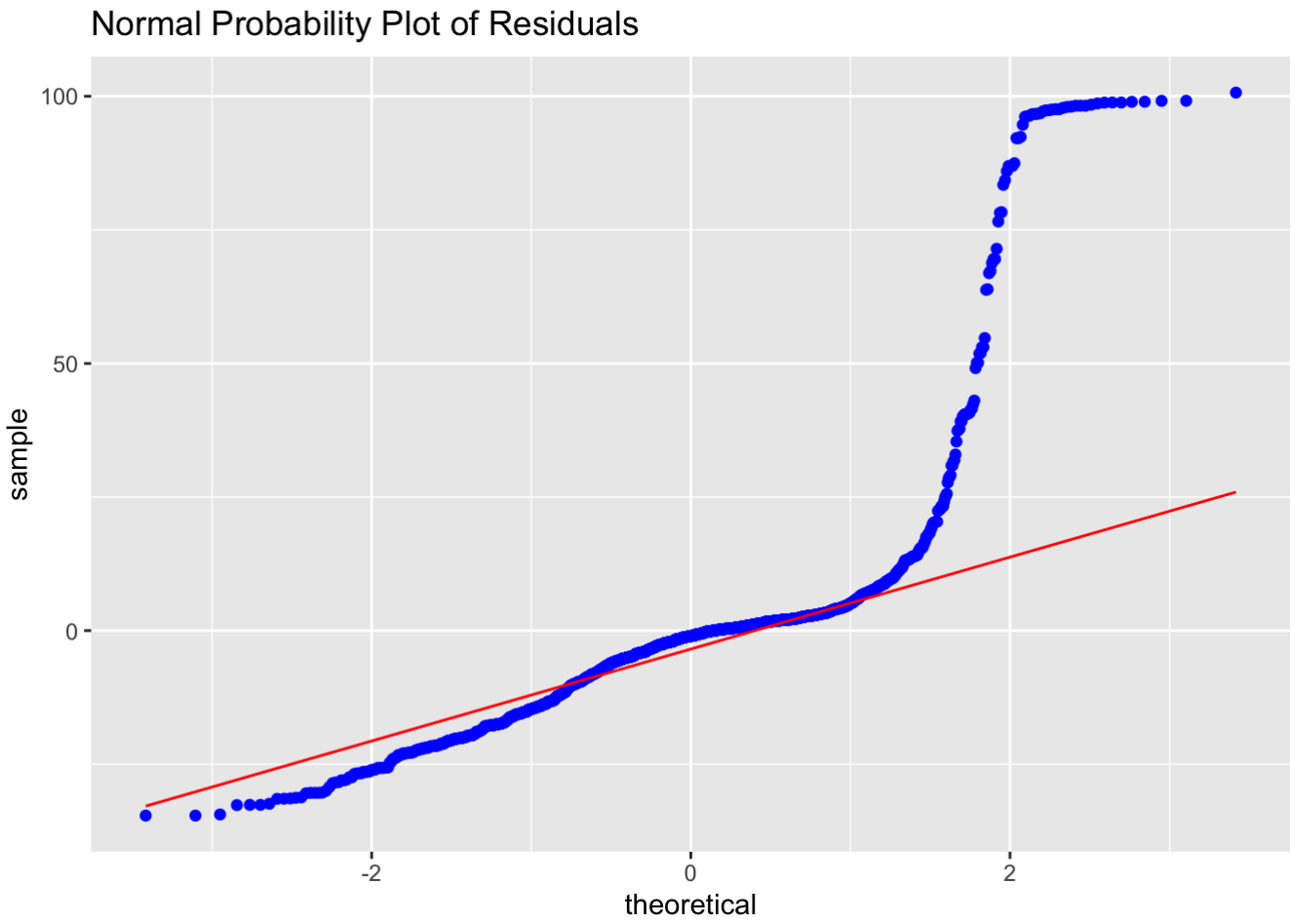
Checking for conditions for linear relation:

```
predicted.values.product_shipped = model$fitted.values
eisproduct_shipped = model$residuals
products_shipped_df = data.frame(predicted.values.product_shipped,
eisproduct_shipped)
```

Checking normality of residuals:

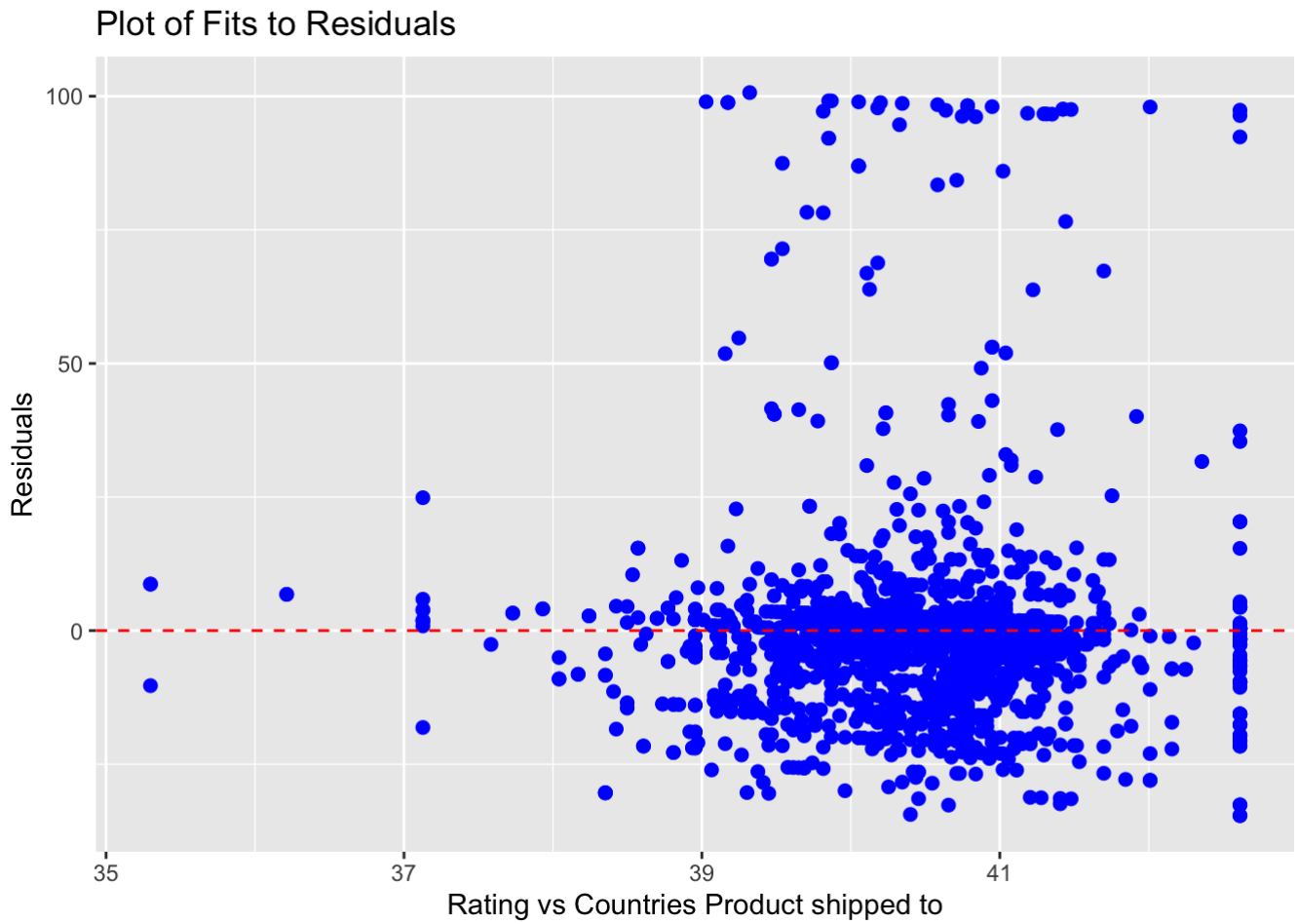
```
ggplot(products_shipped_df, aes(sample = eisproduct_shipped)) + stat_qq(col='blue') + st
at_qqline(col='red') + ggtitle("Normal Probability Plot of Residuals")
```

```
## Warning: The following aesthetics were dropped during statistical transformation: sam
ple
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Checking for homoscedasticity:

```
ggplot(products_shipped_df, aes(x = predicted.values.product_shipped, y = eisproduct_shi
pped)) +  geom_point(size=2, col='blue', position="jitter") + xlab('Rating vs Countries
Product shipped to') + ylab("Residuals") + ggtitle("Plot of Fits to Residuals") + geom_h
line(yintercept=0, color="red", linetype="dashed")
```



Our model doesn't seem to have normally distributed residuals but are homoscedastic. Hence, the model should not be taken as a representative for defining linearity between product ratings and the number of countries shipped.

The results obtained from the linear regression model shows that the estimated intercept is 33.4693 when the rating is 0. Which means if a product is sold to 33.4693, it is likely going to get 0 ratings. The estimated slope is 1.8287 which means that for one unit increase in the ratings (slope), we can expect to see an increase of 1.8287 in the number of countries the product is shipped. The p-value computed for the slope coefficient is 0.0657 which is quite close to our standard threshold of 0.05, but we still cannot reject our Null Hypothesis that the slope is equal to zero. Therefore, we can conclude that there is a very weak positive correlation between ratings and countries shipped to, but to make a more confident conclusion, we would need more data.

3. Ad-boosting and product badges

Since shipping is not a factor at all for the product ratings, let's check if ad-boosting or having product badges make a difference. Product badges are special badges awarded to a product based on quality, fast shipping or if it is locally made.

Here, we will compare if ad-boost affects ratings more or the product badges by performing a permutation test.

H0: Average ratings for a product that is ad-boosted is same as the ratings of a product with badges.

HA: Average ratings of a product with ad-boosting is more than products with badges.

Filtering out required columns.

```
rating_df = data.frame(rating = sales_data_df$rating, rating_count = sales_data_df$rating_count, ad_boost = sales_data_df$uses_ad_boosts, total_badges = sales_data_df$badges_count, quality = sales_data_df$badge_product_quality, shipping = sales_data_df$badge_fast_shipping)
```

```
head(rating_df)
```

	rating <dbl>	rating_count <int>	ad_boost <int>	total_badges <int>	quality <int>	shipping <int>
1	3.76	54	0	0	0	0
2	3.45	6135	1	0	0	0
3	3.57	14	0	0	0	0
4	4.03	579	1	0	0	0
5	3.10	20	1	0	0	0
6	5.00	1	0	0	0	0

6 rows

Filtering product ratings that have got badges and are ad boosted.

```
size_df = nrow(rating_df)
required_products = numeric(sum(rating_df$total_badges) + sum(rating_df$ad_boost))
required_products = as.data.frame(matrix(nrow=sum(rating_df$total_badges) + sum(rating_df$ad_boost),ncol=6))
colnames(required_products) <- c('rating', 'rating_count','ad_boost','total_badges','quality','shipping')
for(i in 1:size_df){
  if(rating_df$total_badges[i] > 0 & rating_df$ad_boost[i] == 1){
    next
  }
  if(rating_df$total_badges[i] > 0 | rating_df$ad_boost[i] == 1){
    required_products[i,] = rating_df[i,]
  }
}
required_products = na.omit(required_products)
head(required_products)
```

	rating <dbl>	rating_count <int>	ad_boost <int>	total_badges <int>	quality <int>	shipping <int>
2	3.45	6135	1	0	0	0
4	4.03	579	1	0	0	0
5	3.10	20	1	0	0	0
9	3.47	15	1	0	0	0

	rating <dbl>	rating_count <int>	ad_boost <int>	total_badges <int>	quality <int>	shipping <int>
12	3.31	13	1	0	0	0
13	3.45	141	1	0	0	0

6 rows

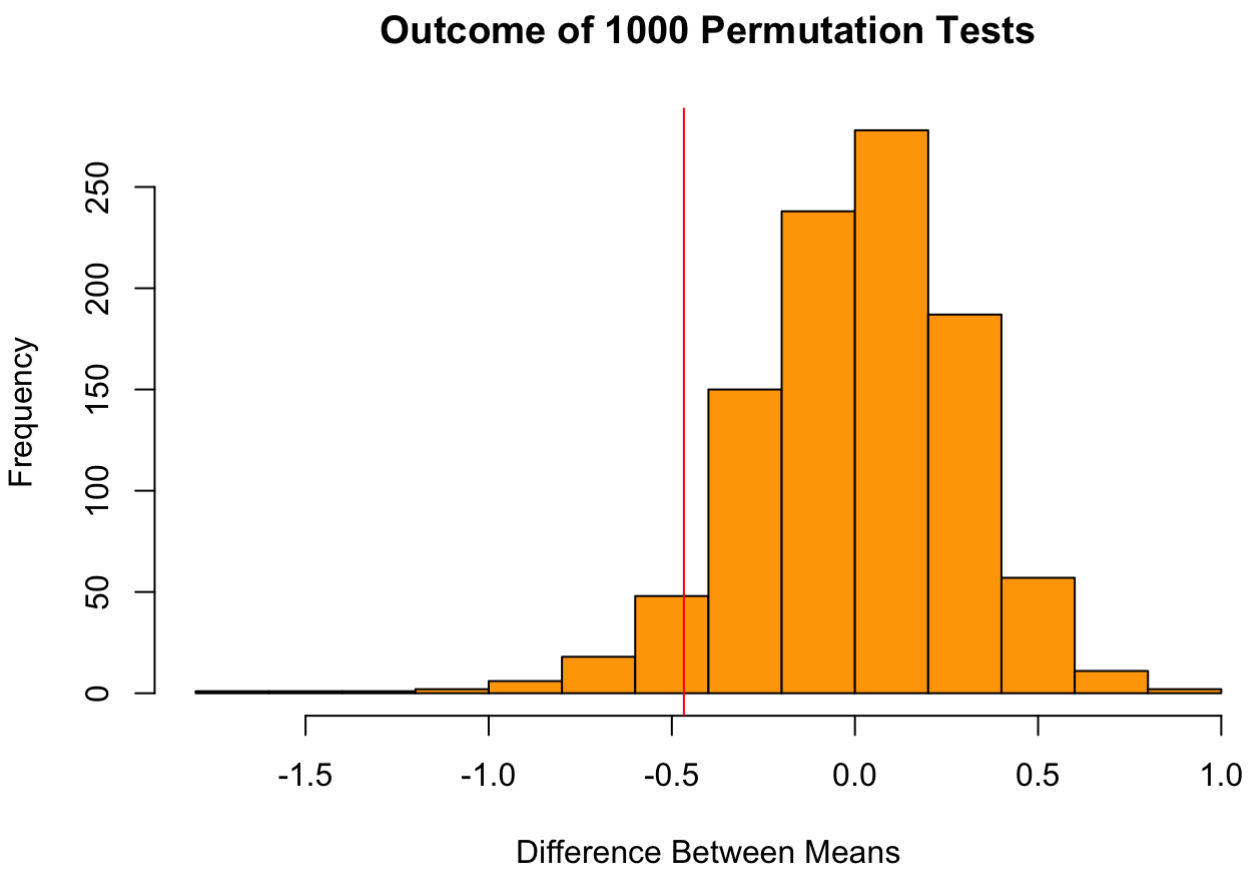
Performing a permutation test,

```
obsdiffmean = mean(required_products$rating[required_products$ad_boost == 1]) - mean(req
uired_products$rating[required_products$total_badges > 0])
total_required_products = nrow(required_products)
total_ad_boost = count(required_products$rating[required_products$ad_boost == 1])
N = 1000
outcome = numeric(N)
for(i in 1:N)
{ index = sample(total_required_products, total_ad_boost, replace=FALSE)#taking adboost
first
  outcome[i] = mean(required_products$rating[index]) - mean(required_products$rating[-in
dex])
}
head(outcome)
```

```
## [1] -0.18924309  0.29285298 -0.00845706 -0.20263464  0.18906841  0.33972344
```

Plotting the distribution:

```
hist(outcome, xlab="Difference Between Means", ylab="Frequency", main="Outcome of 1000 P
ermutation Tests", col='orange')
abline(v = obsdiffmean, col="red")
```



95% confidence interval:

```
outcome_df = data.frame(outcome)
qdata(~ outcome, c(0.025, 0.975), data=outcome_df)
```

```
##       2.5%      97.5%
## -0.6112445  0.5272052
```

```
sum(outcome < obsdiffmean)/1000 #P(mean(ad-boost) < mean(badges)) our p-value
```

```
## [1] 0.043
```

```
sum(outcome >= obsdiffmean)/1000 #P(mean(badges) > mean(ad-boost))
```

```
## [1] 0.957
```

Hence, the p-value less than 0.05 signifies that we can reject our null hypothesis in favor of the alternate hypothesis. Therefore, ad-boosting a product gives better average ratings compared to awarding badges to the products.

4. Ad-boosting

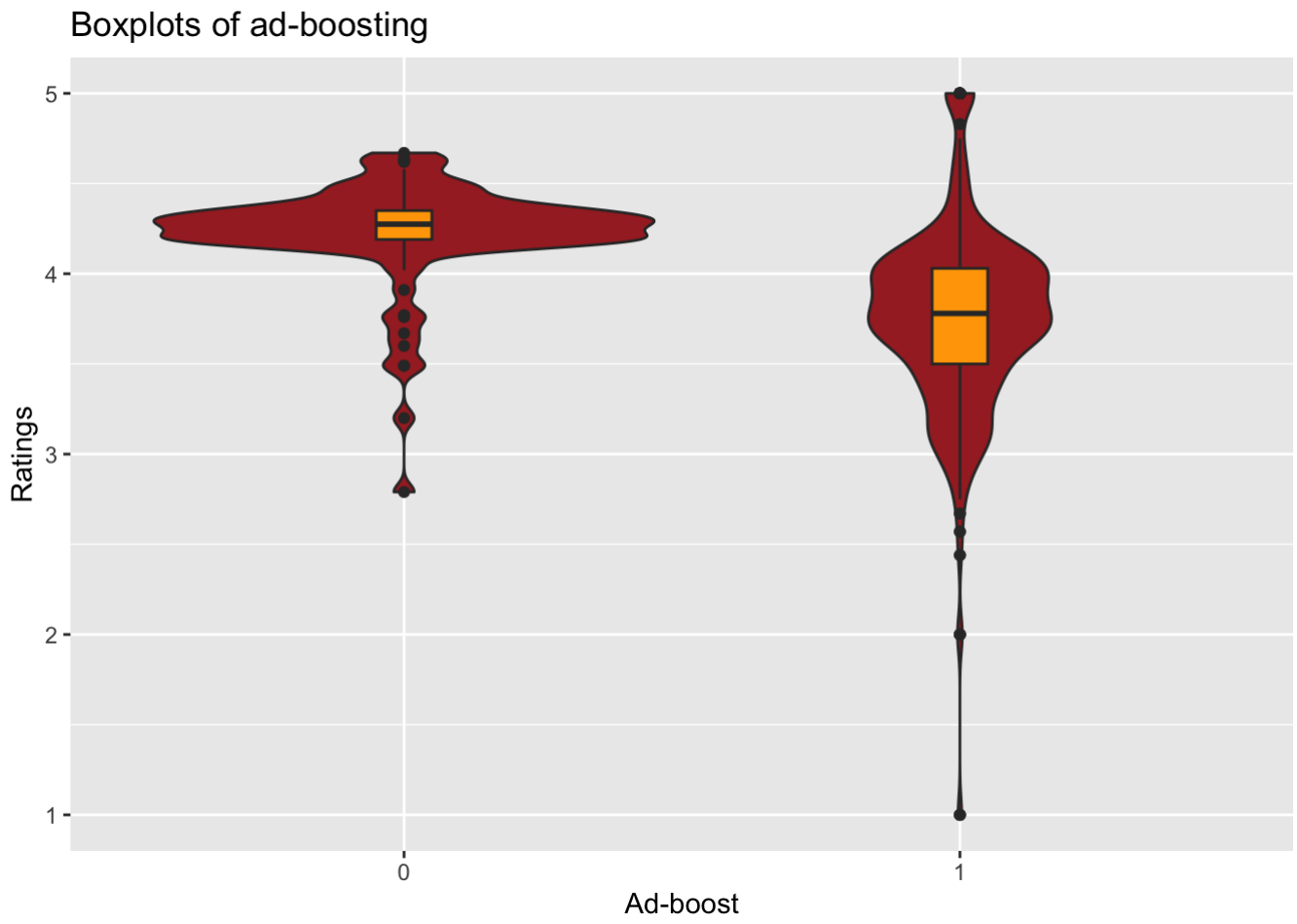
Since ad-boosting affects the product ratings, we are now curious to see what difference it makes to the ratings if a product is ad-boosted versus a product that is not ad-boosted.

H0: ad-boosted products have same average ratings compared to non-ad-boosted products.

HA: ad-boosted products have different ratings compared to non-ad-boosted products.

Plotting ratings with ad-boost:

```
ggplot(data=required_products, aes(x = rating, y = factor(ad_boost))) + geom_violin(fill
="brown") + geom_boxplot(width = 0.1, fill="orange") + xlab("Ratings") + ylab("Ad-boos
t") + ggtitle("Boxplots of ad-boosting") + coord_flip()
```



Right away we can make an interesting observation from the plot. Non-ad-boosted products have higher rating density compared to ad-boosted products.

Performing a t-test with samples more than 25 (therefore no need to check for normality):

```
t.test(~rating|uses_ad_boosts, conf.level=0.95, alternative = "greater", var.equal=FALS
E, sales_data_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  rating by uses_ad_boosts
## t = 1.9582, df = 1456.2, p-value = 0.0252
## alternative hypothesis: true difference in means between group 0 and group 1 is great
er than 0
## 95 percent confidence interval:
##  0.008192764         Inf
## sample estimates:
## mean in group 0 mean in group 1
##        3.843139        3.791762
```

p-value of 0.0252 shows that ad-boosting has an affect on product ratings(rejecting H0). Interestingly, ad-boosted products actually had worse ratings compared to non ad-boosted products.

Conclusion

- It was very interesting data set to work on as our Null Hypothesis was rejected in some scenarios and in some cases, we did not reject our Null Hypothesis.
- To give the marketing or sales team an analysis of what factors affected Sales in the UK on Wish's platform, we can confidently say that:
    - Price does not affect sales volume (inferred from question 1).
    - Product color affects sales volume. Buyers preferred cool colors over warm colors (inferred from question 2).
    - Shipping fee did not affect product ratings while products that were shipped to more countries had a weak positive correlation to product ratings. Therefore, making a product available globally doesn't increase product ratings (inferred from question 3).
    - Ad-boost affected product ratings in a negatively way. This means that customers do not like products that are boosted through advertisement or else there are factors in play here that are having a negative effect on the ratings for products there are ad-boosted (inferred from question 3).

References

Morgan Stanley,2022, Global E-Commerce Growth Forecast 2022,

https://www.morganstanley.com/ideas/global-ecommerce-growth-forecast-2022 (https://www.morganstanley.com/ideas/global-ecommerce-growth-forecast-2022)

Crunching the Data. (n.d.). Data Science Project Proposals. Retrieved from

https://crunchingthedata.com/data-science-project-proposals/ (https://crunchingthedata.com/data-science-project-proposals/)

JEFFREY MVUTU MABILAMA, 2020, Summer Clothes Sales [Dataset], Kaggle

https://www.kaggle.com/datasets/jmmvutu/summer-products-and-sales-in-ecommerce-wish (https://www.kaggle.com/datasets/jmmvutu/summer-products-and-sales-in-ecommerce-wish)

Wish.com, 2023, https://www.wish.com/ (https://www.wish.com/)