Project 2 - Blush Stats from Sephora and ULTA

Hao Cui hc3498

1. Intro

A topic brought up between my friends was how there’s such a plethora of beauty products available it seems hard to decide on which to buy. Hence, this project aims to perhaps give clarity on purchasing choices using statstics. I made my own 2 datasets by recording data observed from the online beauty stores Sephora.com and ULTA.com. I chose blush as my focused product (found in both stores). On the websites, each blush product has its own price, five-star review, and number of colors available which serve as my numeric variables and what type of blush (there are three types: cream, liquid, and powder) which serves as my categorical variable. Each observation is a single blush product and 25 were collected from each store that will merge for a total of 50 (there are more than 50 for both stores, so this is just a sampled list). After scanning through data, I do not expect any particular trends, nothing sticks out to me in particular. But we shall see! My sources are: https://www.sephora.com/shop/blush (https://www.sephora.com/shop/blush) and https://www.ulta.com/makeup-face-blush?N=277v (https://www.ulta.com/makeup-face-blush?N=277v)

Importing Data:

The tidyverse and readxl packages were installed and the two excel sheet datasets were imported with the following code. Then the two datasets were tidied by joining them together into one large dataset called “Blush_data”, using full_join.

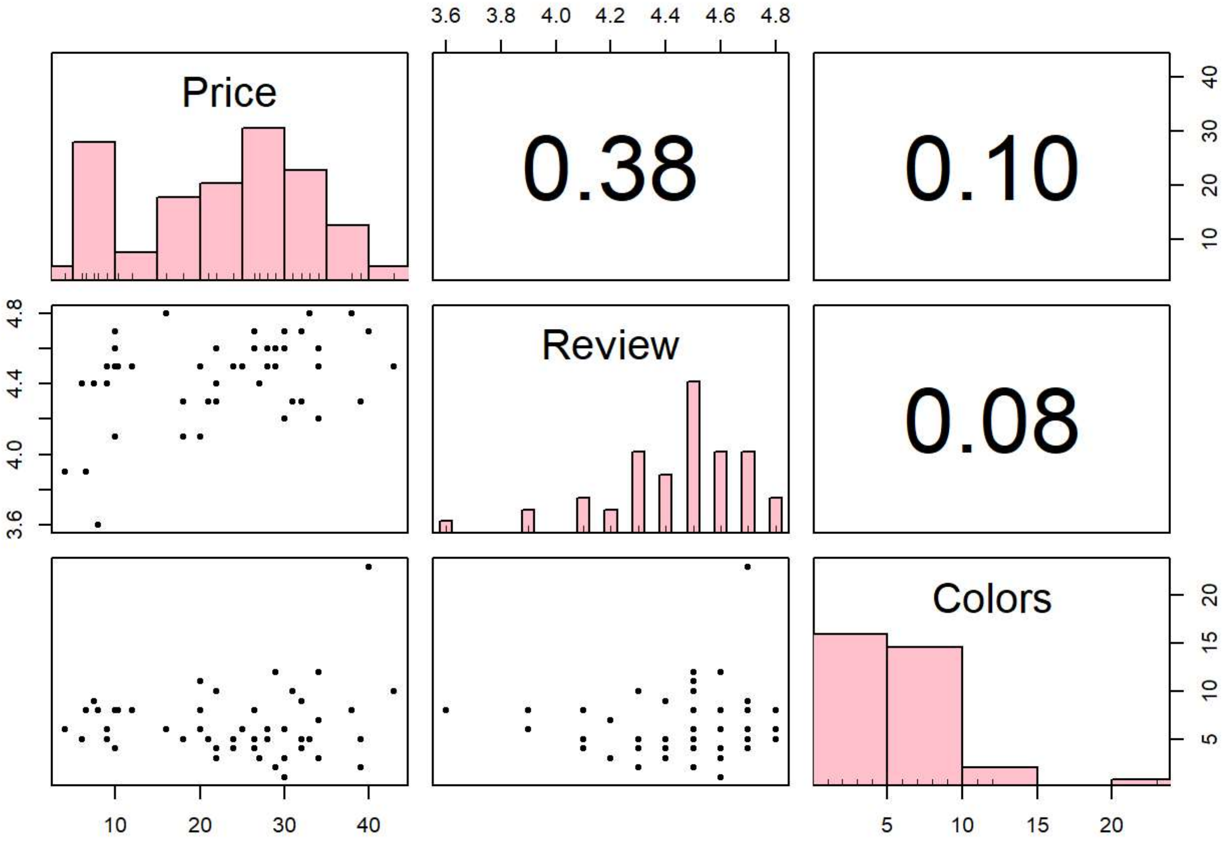
```
## Call needed packages
library(tidyverse)
library(factoextra)
library(cluster)

##install.packages("readxl") in console
library(readxl)



## Import Code had this below:
Sephora_data <- read_excel("C:\\Users\\Hao\\Documents\\Elements of Data Science\\Sephora data.xl
sx")
View(Sephora_data)

ULTA_data <- read_excel("C:\\Users\\Hao\\Documents\\Elements of Data Science\\ULTA data.xlsx")
View(ULTA_data)

## install.packages("dplyr") in console
library(dplyr)
Blush_data <- full_join(Sephora_data, ULTA_data)
```

2. Exploratory Data Analysis

```
## Finding correlation coefficients for numeric variables
Blush_num <- Blush_data %>%
  select_if(is.numeric)
cor(Blush_num, use = "pairwise.complete.obs")
```

```
##            Price     Review     Colors
## Price  1.0000000 0.38409690 0.10003001
## Review 0.3840969 1.00000000 0.07730434
## Colors 0.1000300 0.07730434 1.00000000
```

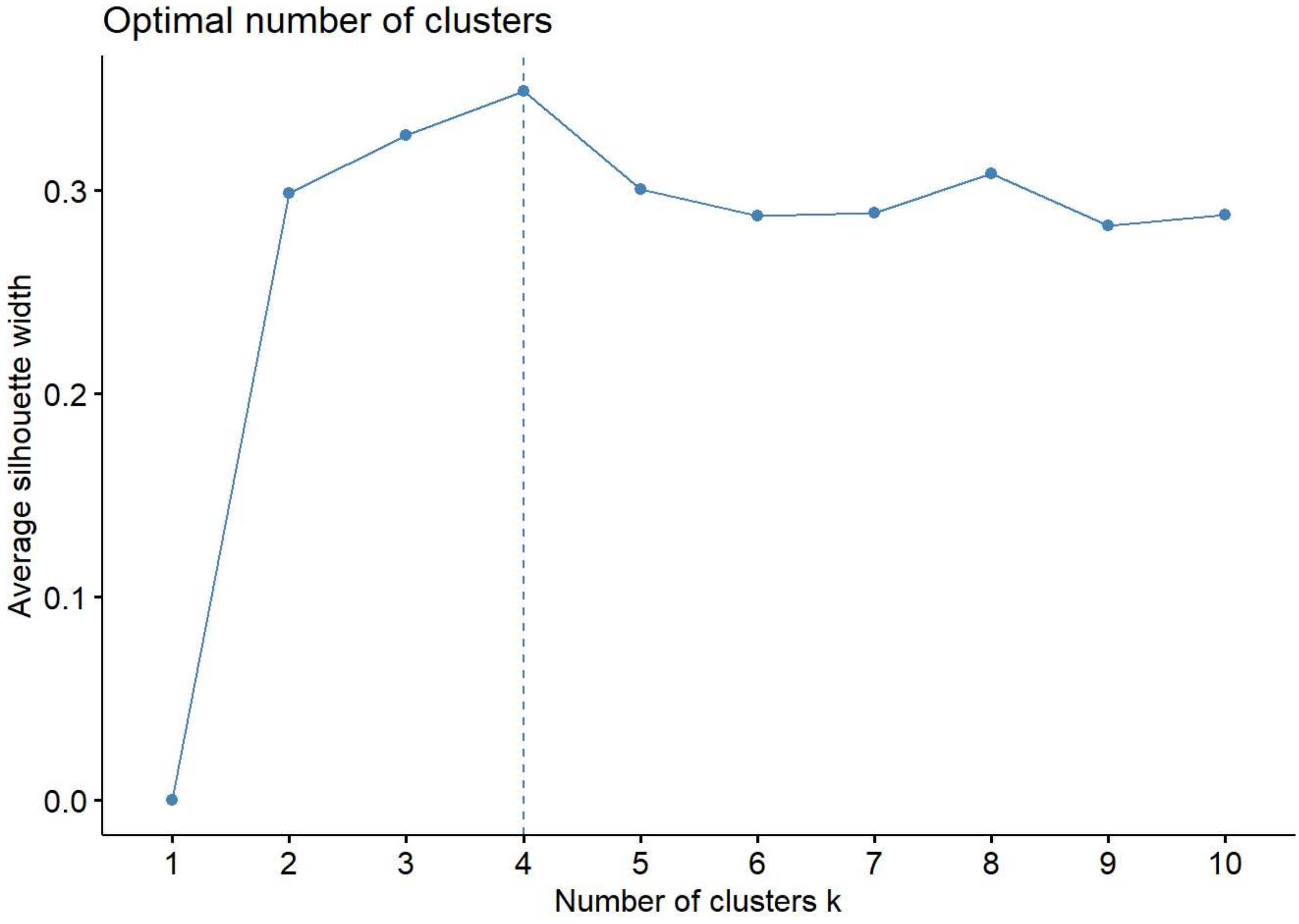
```
## Building univariate and bivariate graphs
## install.packages(psych) in console
library(psych)
pairs.panels(Blush_num,
             method = "pearson", # correlation coefficient method
             hist.col = "pink", # color of histogram
             smooth = FALSE, density = FALSE, ellipses = FALSE)
```



A correlation matrix with correlation coefficients is made to see which of my numeric variables (Price, Review and Colors) are correlated to one another. Seems like none of them are particularly strong in correlation, the highest being a 0.38 between price and review, and the weakest being about 0.08 between review and colors. Univariate and bivariate graphs have also been made. For the univariate graphs: The price histogram shows an almost normal distribution with the exception of a high number of cheaper blushes. Review has a normal distribution with a low outlier. And most blushes have less than 10 colors. For the bivariate graphs: none of the three bivariate graphs show any apparent trends/relationships. There are no obviously distinct shapes (positive or negative linear, curvilinear, etc.).

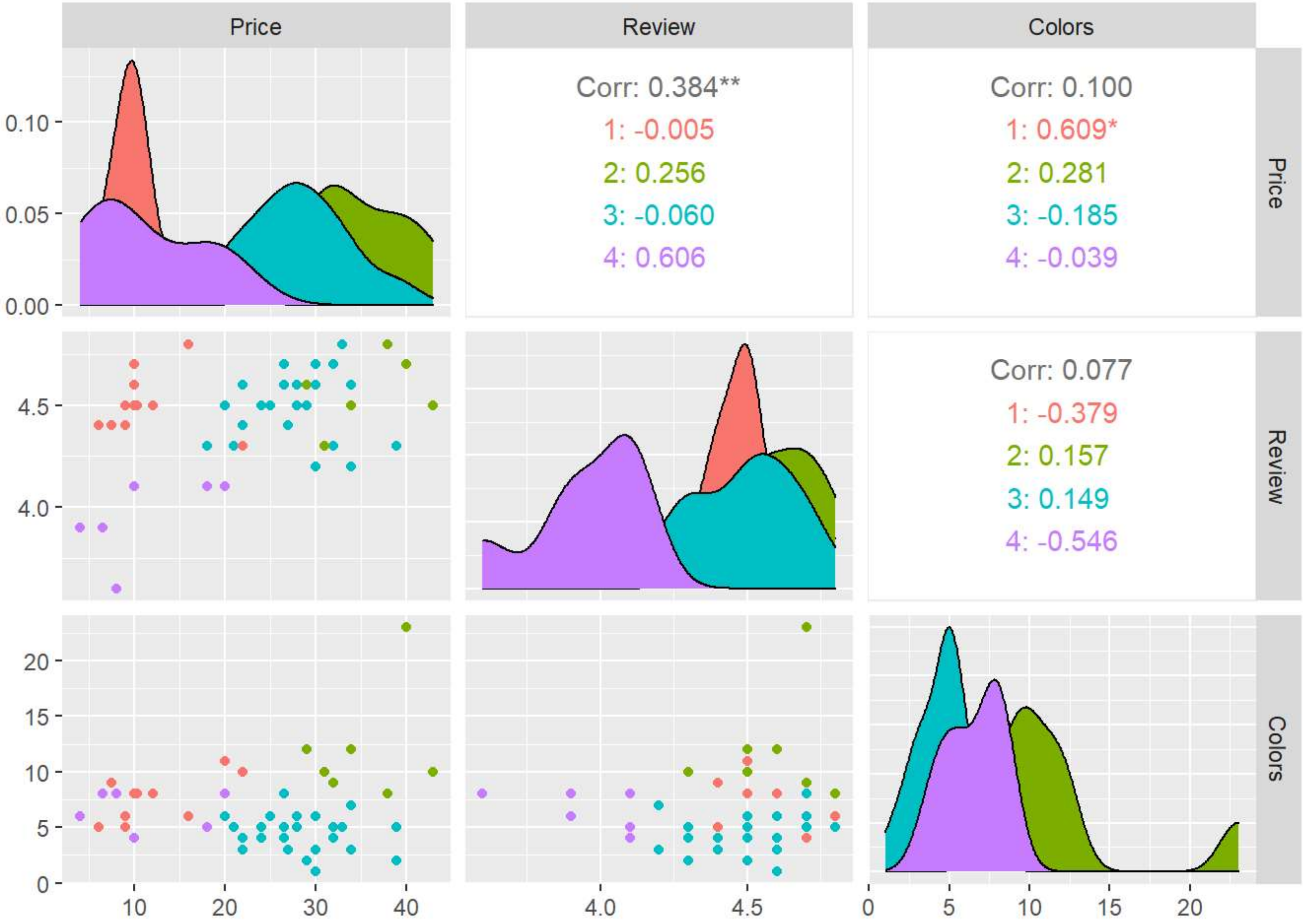
3. Clustering

```
## Prepare data (select variables, scale them)
Blush2 <- Blush_data %>%
  select(Price, Review, Colors) %>%
  scale%>%
  as.data.frame()

## Find optimal number of clusters
fviz_nbclust(Blush2, pam, method = "silhouette")
```



The optimal number of clusters is 4.

```
## Apply a clustering algorithm
pam_results <- Blush2 %>%
  pam(k = 4)

## Save cluster assignment as a column in your dataset
Blush_pam <- Blush_data %>%
  mutate(cluster = as.factor(pam_results$clustering))

## Visualize clusters
## install.packages("GGally") in console
library(GGally)
ggpairs(Blush_pam, columns = 3:5, aes(color = cluster))
```



By using ggpairs, the clusters have been visualized, showing all pairwise combinations of variables colored by cluster assignment.

```
## Summary statistics
Blush_pam %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 4 x 4
##   cluster Price Review Colors
##   <fct>   <dbl>  <dbl>  <dbl>
## 1 1        11.8   4.51   7.33
## 2 2        35.3   4.59  12
## 3 3        28.0   4.49   4.48
## 4 4        11.1   3.95   6.5
```

Additional summary statistics have been done for some more info on our clusters.

Above is the coding and output for section 3: Clustering. Here the number of optimal clusters was calculated to be 4 via silhouette. From using ggpairs, the clusters have been visulized showing all pairwise combinations of variables. Combinations of Colors with Review and Colors with Price appear to have a distribution of quite a bit of overlapping clusters, not as visually separated groupings. However, the combination of Review with Price does seem perhaps promising, still showing some overlap, but not as much by comparison to the other two. Especially as the Review with Price combination has the highest correlation 0.38 (mentioned above) and within that a correlation of 0.606 for the fourth cluster, which contains blushes that are the cheapest and also lowest reviewed. While other combinations may have cluster correlations that are just as high, the overall correlation is significantly less, so there is not as much potential significance in these combinations by comparison. The summary statistics show the means of the clusters which can be used as the center of the clusters, though my data is not significant, and it is hard to find an observation that represents the center of each clusters. Once more, the Review with Price combination appears the most promising, with cluster 4 having the highest mean price and review, notaby with contrast with an opposite cluster 2 with the highest mean price and review. While cluster 2 does have the highest number of colors, cluster 3, however, has the lowest number of colors, not cluster 4 (lowest price and review). If cluster 4 had the lowest number of colors, there may be more potential.

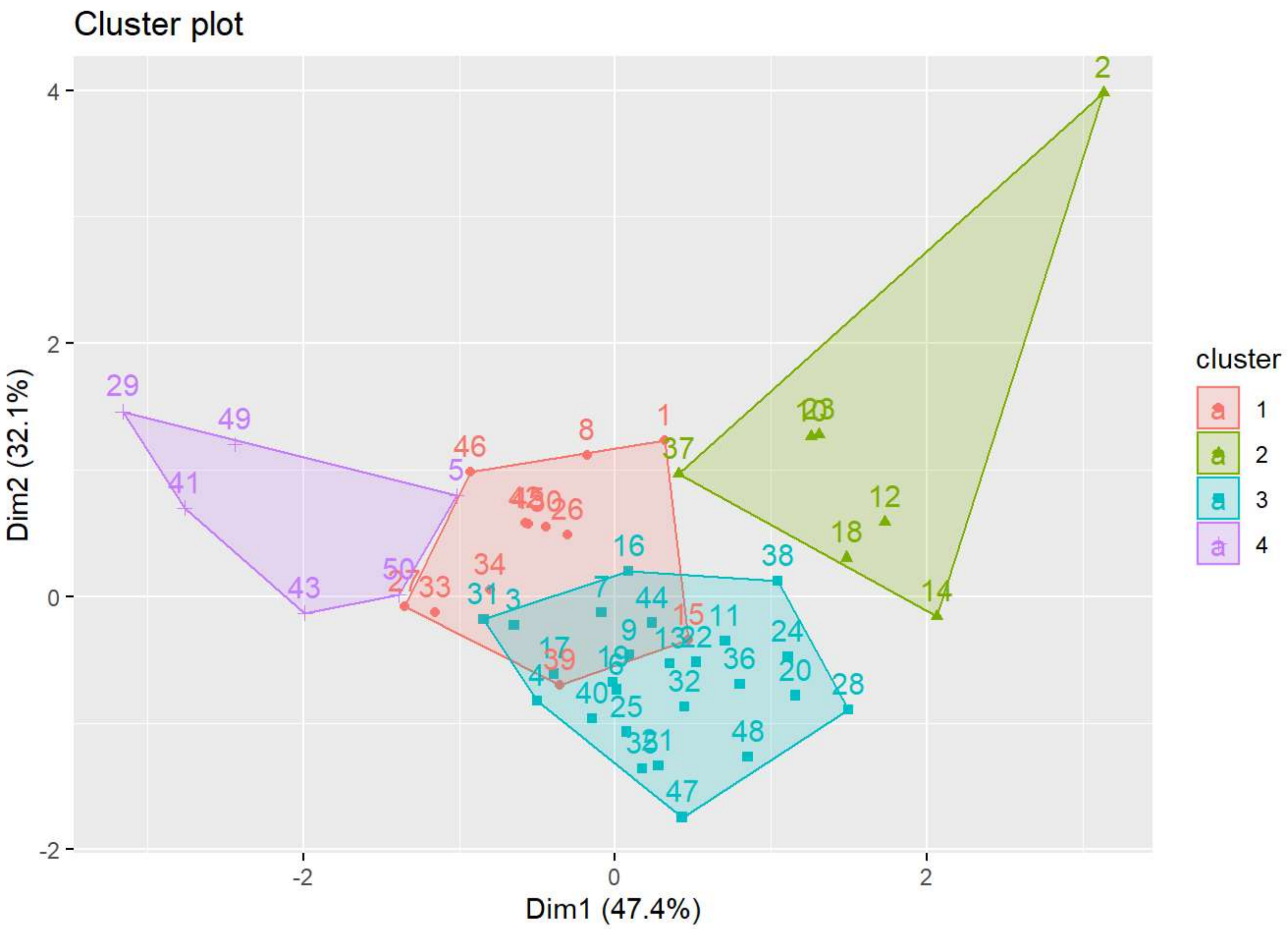
4. Dimentionality Reduction

```
## Use prcomp to find the principal components
pca <- Blush2%>%
  prcomp()

get_pca_var(pca)$coord %>% as.data.frame
```

```
##            Dim.1      Dim.2       Dim.3
## Price  0.8127676 -0.1673102  0.55804664
## Review 0.8031688 -0.2287704 -0.55007642
## Colors 0.3402411  0.9397030 -0.03455801
```

Using PCA, the coefficients for each principal component (PC) are shown.

```
## Visualizing clusters
fviz_cluster(pam_results, data = Blush2)
```



Using fviz_cluster, the optimal 4 clusters have been plotted, showing the first two PCs.

Above has the 2D plot of PC1 and PC2, which account for the majority of variation in the dataset (total variation 79.5%). Likewise from section 3, there appears to some sort of relationship between Price and Review. Based on the coefficients of the PCs, the largest contributors to PC1 are Price and Review (0.8 for both). However, what was not expected is that Colors had the highest contribution to PC2 (0.9), in comparison to the other two which are small negative values (about -0.2 for both Price and Review). High scores on PC1 means you have high values of price and review but low everything else; low scores mean low price and review, high everything else. Thus, the dimension of greatest variability distinguishes high-price and review from the others. I think these clusters reflect groupings of blushes relatively well, though only to a limited capacity. For one, it’s hard to pick a blush product as the representative center for the clusters. But I say there is possibly some minimal accuracy because it is colloquially known that the review rating on the quality of the product is influenced by the price. More expensive blushes factor in marketing costs into their higher price point, creating a perception of a better quality blush in more expensive products via this marketing, or simply the human bias of buying expensive things makes you believe what you are buying is high quality. Thus, it is natural for the clusters to create distinct groupings. However, what is also realistic is that there are some overlap in clusters 1 and 3, perhaps due to the fact that there are indeed cheaper blushes that are of good quality, contrary to the marketing and human bias. Or another possible reason could be that these clusters contain blushes that have a medium price and review, and therefore it would make sense for customers to have more ambiguity over a product’s quality.

5. Classification and Cross-Reduction

```
library(plotROC)

## Find average five-star review
Blush_data %>%
  summarize(mean_review = mean(Review, na.rm = T))
```

```
## # A tibble: 1 x 1
##   mean_review
##         <dbl>
## 1        4.44
```

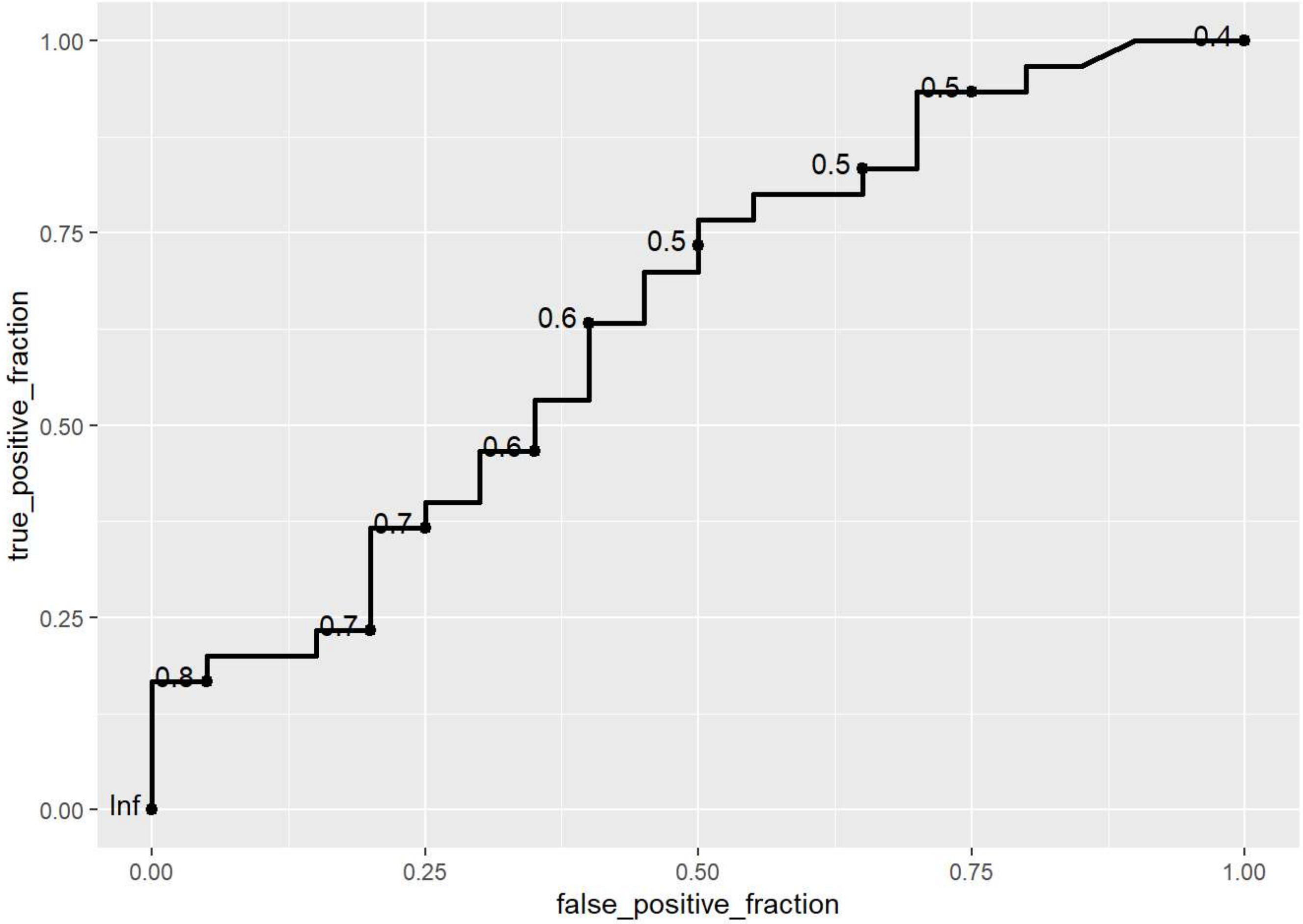
To figure out which blushes are above and below average, we must first find the average, which is 4.444 stars.

```
## Make blushes with above average reviews one, and below average reviews zero.
blush <- Blush_data%>%
 mutate(Above_avg_review = ifelse(Review > 4.444 , 1, 0),
         ID = row_number())

## Logistic regression
fit <- glm(Above_avg_review ~ Price + Colors, data = blush, family = "binomial")

## Calculate a predicted probability for observations
log_blush <- blush %>%
  mutate(probability = predict(fit, type = "response"),
         predicted = ifelse(probability > 0.5, 1, 0)) %>%
  select(Name, Brand, Price, Review, Above_avg_review, probability, predicted)
head(log_blush)
```

```
## # A tibble: 6 x 7
##   Name                 Brand Price Review Above_avg_review probability predicted
##   <chr>                <chr> <dbl>  <dbl>            <dbl>       <dbl>     <dbl>
## 1 Soft Pinch Liquid B~ Rare~    20    4.5                1       0.702         1
## 2 Blush                Nars     40    4.7                1       0.962         1
## 3 Stay Vulnerable Mel~ Rare~    21    4.3                0       0.547         1
## 4 Liquid Blush         Nars     30    4.2                0       0.593         1
## 5 Lip + Cheek Cream B~ Milk~    20    4.1                0       0.622         1
## 6 Dew Blush Liquid Ch~ Saie     24    4.5                1       0.553         1
```

The table log_blush contains the predicted probabilities of each product, above shown are the first six rows of the table.

```
## ROC curve
ROC <- ggplot(log_blush) +
  geom_roc(aes(d = Above_avg_review, m = probability))
ROC
```



The ROC curve is shown above, creating a diagonal, almost linear shape with a shallow curve, which is not ideal. A better model would have the curve approach the top left corner.

```
## Calculate the area under the curve
calc_auc(ROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.6458333
```

The AUC value is calculated above, being 0.646.

```
## k-fold cross-validation
set.seed(1)
# Choose number of folds:
k = 10

# Randomly order rows in the dataset
data <- blush[sample(nrow(blush)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Use a for loop to get diagnostics for each test set
diags_k <- NULL

for(i in 1:k){
  # Create training and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

  # Train model on training set (all but fold i)
  fit <- glm(Above_avg_review ~ Price + Colors, data = train, family = "binomial")

  # Test model on test set (fold i)
  df <- data.frame(
    prob = predict(fit, newdata = test, type = "response"),
    y = test$Above_avg_review)

  # Consider the ROC curve for the test dataset
  ROCplot <- ggplot(df) + geom_roc(aes(d = y, m = prob), n.cuts = 0)

  # Get diagnostics for fold i (AUC)
  diags_k[i] <- calc_auc(ROCplot)$AUC
}

# Average performance
mean(diags_k)
```

```
## [1] 0.575
```

Above is the k-fold cross validation calculation, which is 0.575, falling in the bad model range.

The AUC value (0.646) is poor, falling under the 0.6-0.7 range. This is synonymous to the shape of the curve which does not curve that much towards the top left corner of the graph. The AUC tells that a randomly selected individual from the “success” group has a test value larger than for a randomly chosen individual from the “failure” group [AUC] about 65% percent of the time, which is a poor model. This means that Price and Colors are not useful for finding whether or not a blush product’s review falls above or below average. After doing the k-fold cross validation the value (0.575) becomes even worse, as it falls in the ‘bad’ model range of 0.5-0.6. This means that the our model of price and color of a blush product to predict above and below average review quality is bad. Though, this does not necessarily mean bad news, in fact it could be good. This could indicate that the number of colors and how expensive a blush product does not have an influence on the review. Equating higher reviews to higher quality blush, that means that you can find high quality blushes at a cheaper price with also a wide variety of colors rather than only expensive ones.