Applied Statistics Modelling

September 10, 2023

Title

,

Ajio Fashion Clothing

,

[1]:
```
library(IRdisplay)
display_png(file = "Ajio-logo.jpg", width = 700, height = 100)
```

1 Table of Contents:

2 1 - Introduction

In order to do this statistical analysis, we examined a sizable dataset from AJIO, a well-known fashion and lifestyle brand that falls within the authority of Reliance Retail's digital commerce division. AJIO is well known for its well-picked fashion options, current fashions, and affordable prices. This dataset offers a unique opportunity to use statistical methods to learn more about many facets of AJIO's fashion products. Our study strives to uncover insightful statistical patterns that may guide business choices and contribute to a greater understanding of the fashion industry, from examining price trends and discounts to evaluating brand effect. Join us as we explore the fashion landscape of AJIO using the information, supported by statistical rigor and insightful analysis.

3 a) Data description

The dataset under study is a comprehensive collection of data obtained from AJIO, a top fashion and lifestyle brand in the world of Internet commerce. It consists of a wide variety of variables, each of which sheds light on a distinct aspect of AJIO's large product line. The main characteristics are summarised as follows:

Product_URL: This column includes each product's own URL, which serves as a link to the product's web listing.

Brand: It lists the brands that are connected to each product, demonstrating the wide variety of labels that are offered on AJIO.

Description: This variable provides succinct descriptions of the items, including details on their designs and characteristics.

Id_Product: This numerical code serves as an individual and traceable representation of each product in the collection.

URL_image: It stores URLs leading to images of products, allowing for visual representation.

Category_by_gender: This categorical characteristic divides items into categories based on gender, designating them as either "Men" or "Women."

Discount pricing (in Rs.): To aid in pricing analysis, this field displays each product's reduced price.

Original Price (in Rs.): This represents the items' full retail cost, excluding any reductions.

Color: This categorical variable describes the color of each product, providing information about the differences between products.

This dataset captures the essence of AJIO's fashion products and includes a variety of elements, including product information, price, and gender-based classification. It serves as the basis for our statistical analysis, enabling us to investigate patterns, carry out hypothesis testing, and produce valuable insights to guide business choices in the fashion retail industry.

4 b) Objective

The objective of this statistical study is to obtain thorough insights into the fashion product panorama of AJIO, a significant participant in Reliance Retail's e-commerce business. AJIO is renowned for its broad selection of clothing, current styles, and affordable prices. Our study in this context strives to address a number of important issues and problems, including pricing trends, discount analysis, brand influence, gender-based analysis, etc.

5 c) Goal

Our investigation is driven by the following precise objectives in order to fully comprehend AJIO's clothing lines and offer insightful information:

Finding Price patterns: Check into price patterns for fashion items, including variations in list and sale prices over time.

Analyse AJIO's discounting tactics, including their frequency and extent, in order to determine how they may affect sales and price.

Gender-Based Pricing Comparison: Examine possible gender-based pricing gaps by comparing the expenses of items made for men and women.

We will attempt to obtain insightful information on AJIO's product offers through hypothesis testing, statistical analysis, and data visualizations. Based on our results, we will provide recommendations for decision-making in the retail clothing industry.

6 2 - Data Exploration

Now, let's commence our analysis by first exploring the Ajio dataset.

[2]:
```r
file_path <- "Ajio Fasion Clothing.csv"
# Read the CSV file into a data frame
df <- read.csv(file_path, header = TRUE, sep = ",")
options(width = 10)
# Display the first few rows of the data to verify it was loaded correctly
head(df)
```

A data.frame: 6 × 9

	Product_URL <chr>
1	https://www.ajio.com/netplay-checked-polo-t-shirt/p/441137362_white
2	https://www.ajio.com/netplay-tapered-fit-flat-front-trousers/p/441124497_navy
3	https://www.ajio.com/the-indian-garage-co-striped-slim-fit-shirt-with-patch-pocket/p/4
4	https://www.ajio.com/performax-heathered-crew-neck-t-shirt/p/441036730_charcoal
5	https://www.ajio.com/john-players-jeans-washed-skinny-fit-jeans-with-whiskers/p/44112
6	https://www.ajio.com/dennislingo-premium-attire-slim-fit-shirt-with-patch-pocket/p/46

To view the structure of a data frame or any other R object, use the str(df) command in R. It will provide valuable information about the object, including its data type, structure, and the first few rows of data.

OUTPUT: We see that the dataset contains 367,172 observations with 9 variables and it has displayed the first few lines of the code.

[3]:
```
# View the structure of the dataset
str(df)
```

```
'data.frame':   367172 obs. of  9 variables:
 $ Product_URL            : chr  "https://www.ajio.com/netplay-checked-polo-t-
shirt/p/441137362_white" "https://www.ajio.com/netplay-tapered-fit-flat-front-
trousers/p/441124497_navy" "https://www.ajio.com/the-indian-garage-co-striped-
slim-fit-shirt-with-patch-pocket/p/460453612_white"
"https://www.ajio.com/performax-heathered-crew-neck-t-
shirt/p/441036730_charcoal" …
 $ Brand                  : chr  "netplay" "netplay" "the-indian-garage-co"
"performax" …
 $ Description            : chr  "Checked Polo T-shirt" "Tapered Fit Flat-Front
Trousers" "Striped Slim Fit Shirt with Patch Pocket" "Heathered Crew-Neck
T-shirt" …
 $ Id_Product             : num  4.41e+11 4.41e+11 4.60e+11 4.41e+11 4.41e+11
…
 $ URL_image              : chr  "https://assets.ajio.com/medias/sys_master/root
/20220309/inTn/6227b81faeb26921afcde577/-286Wx359H-441137362-white-MODEL.jpg" "h
ttps://assets.ajio.com/medias/sys_master/root/20210907/vQKt/6136775cf997ddce89bd
deea/-286Wx359H-441124497-navy-MODEL.jpg" "https://assets.ajio.com/medias/sys_ma
ster/root/20211228/s1km/61ca36a4aeb26901101f7552/-286Wx359H-460453612-white-
MODEL.jpg" "https://assets.ajio.com/medias/sys_master/root/20220120/lLur/61e981a
ef997dd66232f4792/-286Wx359H-441036730-charcoal-MODEL.jpg" …
 $ Category_by_gender     : chr  "Men" "Men" "Men" "Men" …
 $ Discount.Price..in.Rs..: chr  "559" "720" "495" "329" …
 $ Original.Price..in.Rs..: chr  "699" "1,499" "1,649" "599" …
 $ Color                  : chr  "white" "navy" "white" "charcoal" …
```

The summary(df) command in R provides a summary of the statistics and characteristics of the variables in a data frame called "df."

OUTPUT: Based on the summary statistics below, it seems like most of the variables in this dataset are currently classified as character (text) variables. To perform meaningful statistical analysis and modeling, we will need to convert relevant variables to appropriate data types, such as numeric or factor.

[4]:
```
# Summary statistics
summary(df)
```

```
 Product_URL
 Length:367172
 Class :character
 Mode  :character



    Brand
 Length:367172
 Class :character
 Mode  :character



 Description
 Length:367172
 Class :character
 Mode  :character



   Id_Product
 Min.   :4.100e+11
 1st Qu.:4.610e+11
 Median :4.626e+11
 Mean   :4.596e+11
 3rd Qu.:4.639e+11
 Max.   :4.601e+12
  URL_image
 Length:367172
 Class :character
 Mode  :character



 Category_by_gender
 Length:367172
 Class :character
 Mode  :character
```

```
Discount.Price..in.Rs..
Length:367172
Class :character
Mode  :character
```

```
Original.Price..in.Rs..
Length:367172
Class :character
Mode  :character
```

```
   Color
Length:367172
Class :character
Mode  :character
```

The below code colSums(is.na(df)) is used to check for missing values (NA, Not Available) in each column of the data frame df and calculate the total number of missing values for each column.

OUTPUT:

The below code indicates that there are no missing values in any of the column of the dataset.

[5]:
```
# Check for missing values
colSums(is.na(df))
```

Product_URL 0 **Brand** 0 **Description** 0 **Id_Product** 0 **URL_image** 0 **Category_by_gender** 0 **Discount.Price..in.Rs..** 0 **Original.Price..in.Rs..** 0 **Color** 0

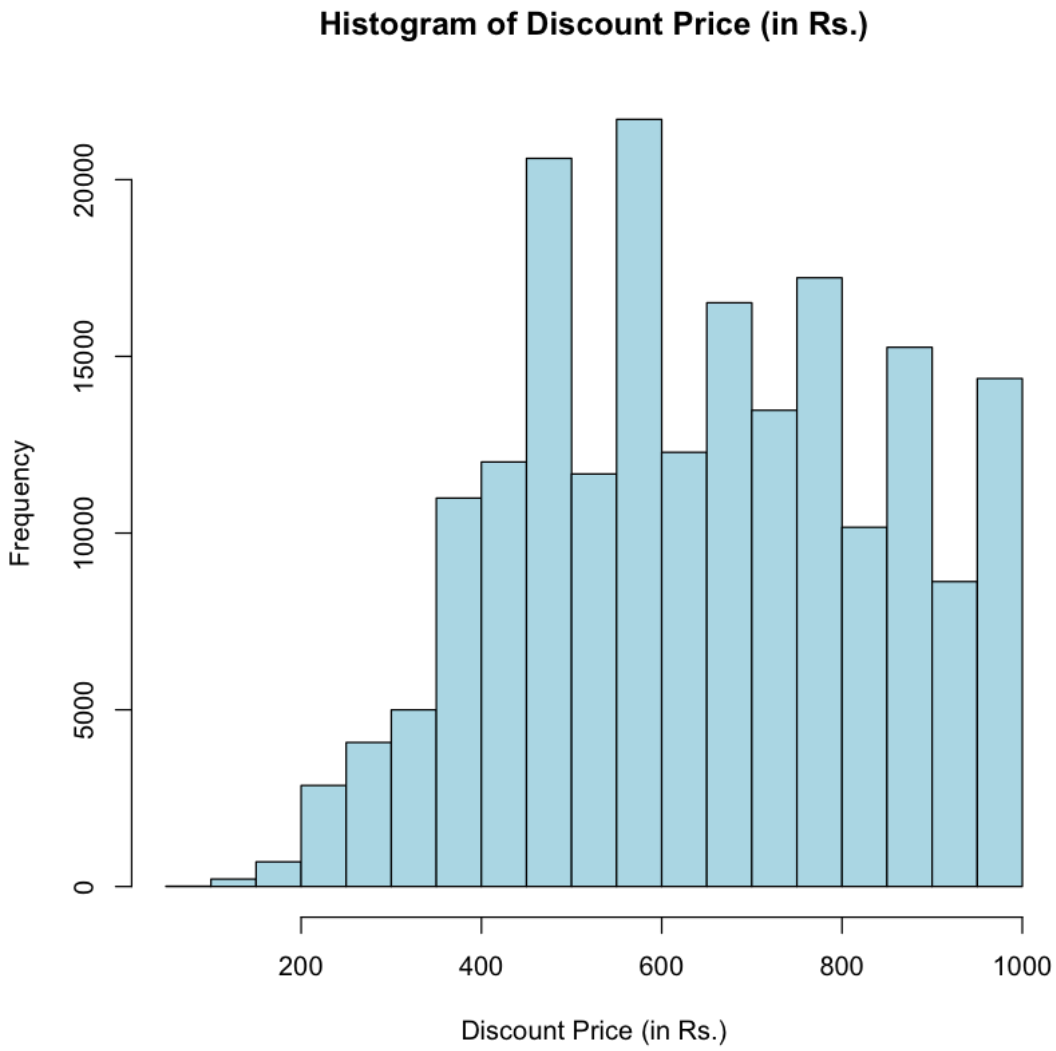
The below line of code makes sure that the data frame df's "Discount.Price..in.Rs." column is handled as a numeric variable. If the values in this column are not already in numeric format, the as.numeric() method is used to convert them. Next, it creates a histogram of the 'Discount.Price..in.Rs..' column by specifying the data should be used to create the histogram, then sets the main title of the histogram to "Histogram of Discount Price (in Rs.)", x and y axes of the lables and the color of the graph respectively.

OUTPUT: The outputs shows that between Rs. 400 - 600 there are above 20000 products which have a discount price.

[6]:
```
# Ensure the column 'Discount.Price..in.Rs..' is numeric (convert if not␣
 ↪already)
df$Discount.Price..in.Rs.. <- as.numeric(df$Discount.Price..in.Rs..)
```

```
# Create a histogram
hist(df$Discount.Price..in.Rs.., main = "Histogram of Discount Price (in Rs.)",␣
  ↪xlab = "Discount Price (in Rs.)", col = "lightblue")
```

```
Warning message in eval(expr, envir, enclos):
"NAs introduced by coercion"
```



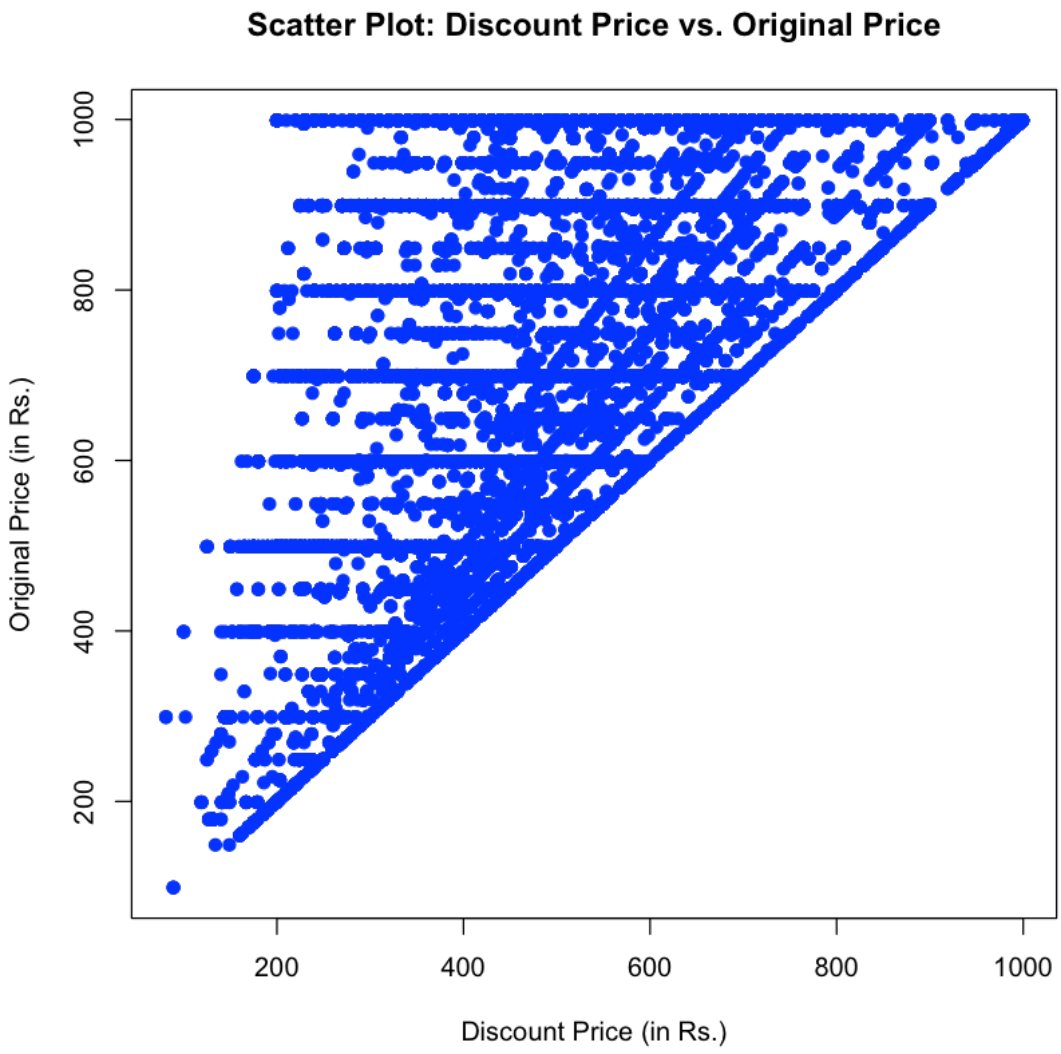
The code below shows ensures that the 'Discount.Price..in.Rs..' column in the data frame df is treated as a numeric variable. It uses the as.numeric() function to convert the values in this column to numeric format if they are not already numeric. Similarly, the next line of code ensures that the 'Original.Price..in.Rs..' column in "df" is treated as a numeric variable by converting its values to numeric format if needed. Then we will create a scatter plot to visualize the relationship between 'Discount Price (in Rs.)' and 'Original Price (in Rs.)'. Next, we will specify the x and y

axes, title and color respectively for the scatter plot.

OUTPUT: The output demonstrates that there is typically a linear connection as discounts are frequently applied as a proportion of the original price. It would be confirmed by a high positive correlation coefficient that the scatter plot's strong positive linear link indeed exists. In general, the "Original Price" tends to rise proportionately as the "Discount Price" does. For example, items with higher initial prices typically have higher discount prices, and those with lower original prices typically have lower discount prices.

[7]:
```
# Ensure both columns are numeric (convert if not already)
df$Discount.Price..in.Rs.. <- as.numeric(df$Discount.Price..in.Rs..)
df$Original.Price..in.Rs.. <- as.numeric(df$Original.Price..in.Rs..)

# Create a scatter plot
plot(df$Discount.Price..in.Rs.., df$Original.Price..in.Rs..,
     main = "Scatter Plot: Discount Price vs. Original Price",
     xlab = "Discount Price (in Rs.)", ylab = "Original Price (in Rs.)",
     pch = 19, col = "blue")
```

```
Warning message in eval(expr, envir, enclos):
"NAs introduced by coercion"
```



7 3 - Data Cleaning

Now we will start with the data cleaning part, we will check for the duplicate values in the dataset and will try to remove them and also the unnecessary columns and convert some columns datatypes.

[8]:
```
install.packages("path/to/UpSetR_1.4.0.tar.gz", repos = NULL, type = "source")
```

```
Warning message in install.packages("path/to/UpSetR_1.4.0.tar.gz", repos = NULL,
:
"installation of package 'path/to/UpSetR_1.4.0.tar.gz' had non-zero exit status"
```

Below we will visualize the missing values using the naniar library, which provides functions for visualizing and exploring missing data in a dataset. Then by using 'miss_var_summary()' function

from the naniar library will help us to generate a summary of missing values in the 'df' dataset. Specifically, it creates a visual summary of missing values for each variable (column) in the dataset.

OUTPUT: Original.Price..in.Rs..:

This variable has 300,088 missing values, which accounts for approximately 81.73% of the data in this column.

Discount.Price..in.Rs..:

This variable has 169,425 missing values, which accounts for approximately 46.14% of the data in this column.

Product_URL, Brand, Description, Id_Product, URL_image, Category_by_gender, Color:

These variables have no missing values; they are complete.

[9]:
```
# Visualize missing values
library(naniar)
miss_var_summary(df)
```

A tibble: 9 × 3

variable <chr>	n_miss <int>	pct_miss <dbl>
Original.Price..in.Rs..	300088	81.72954
Discount.Price..in.Rs..	169425	46.14322
Product_URL	0	0.00000
Brand	0	0.00000
Description	0	0.00000
Id_Product	0	0.00000
URL_image	0	0.00000
Category_by_gender	0	0.00000
Color	0	0.00000

The below code will check for duplicate rows in the df and removes them by keeping only the unique rows.

[10]:
```
# Check for duplicate rows and remove them if needed
df <- unique(df)
```

The code below removes the designated columns from 'df'. In particular, it eliminates the columns "Product_URL," "URL_image," and "Id_Product." These columns won't be found in the dataset after running this code.

The structure of the updated dataset is then checked using the str(df) command, which enables to confirm that the requested columns have indeed been removed and that the dataset now just includes the remaining columns.

[11]:
```
# Drop the specified columns
df <- df[, !(names(df) %in% c("Product_URL", "URL_image", "Id_Product"))]

# Check the structure of the modified dataset
str(df)
```

```
'data.frame':   367172 obs. of  6 variables:
 $ Brand                  : chr  "netplay" "netplay" "the-indian-garage-co"
"performax" …
 $ Description            : chr  "Checked Polo T-shirt" "Tapered Fit Flat-Front
Trousers" "Striped Slim Fit Shirt with Patch Pocket" "Heathered Crew-Neck
T-shirt" …
 $ Category_by_gender     : chr  "Men" "Men" "Men" "Men" …
 $ Discount.Price..in.Rs..: num  559 720 495 329 899 684 700 419 374 500 …
 $ Original.Price..in.Rs..: num  699 NA NA 599 999 NA NA 499 499 NA …
 $ Color                  : chr  "white" "navy" "white" "charcoal" …
```

Now, the code in the 'Description' column includes modifications to replace or eliminate non-ASCII characters. This is accomplished by using the iconv function and the to = "ASCII" and sub = " " arguments to convert non-ASCII characters to their closest ASCII equivalents and replace any non-ASCII characters with spaces, respectively.

With the help of the tolower function, it lowercase every word in the 'Description' column. By doing this, case insensitivity is ensured for text comparisons and analysis.

[12]:
```r
# Remove or replace non-ASCII characters in the 'Description' column
df$Description <- iconv(df$Description, to = "ASCII", sub = " ")

# Now, convert 'Description' to lowercase
df$Description <- tolower(df$Description)
```

The code removes commas from the values in the "Discount.Price..in.Rs." and "Original.Price..in.Rs." columns using the gsub function. The empty string (" ") is substituted for every instance of a comma (,) in the values using the gsub function.

The values in both columns are changed to numeric data types using the as.numeric function once the commas have been removed. This ensures that the values are handled as numerical data and that they may be analyzed and evaluated.

[13]:
```r
# Convert 'Discount.Price..in.Rs..' and 'Original.Price..in.Rs..' to numeric
df$Discount.Price..in.Rs.. <- as.numeric(gsub(",", "", df$Discount.Price..in.Rs.
  ↪.))
df$Original.Price..in.Rs.. <- as.numeric(gsub(",", "", df$Original.Price..in.Rs.
  ↪.))
```

The functions carried out by the code below are as follows:

The sum(is.na()) function is used to check for missing values in the "Discount.Price..in.Rs." and "Original.Price..in.Rs." columns, and the variables "missing_discount" and "missing_original" are used to hold the counts of missing values.

The number of missing values for each column is then displayed by printing the counts of missing values prior to imputation.

The mean of each column's corresponding non-missing values is then used to impute the missing values in both columns. To eliminate missing data from the mean calculation, use the mean() method with the na.rm = TRUE parameter.

Following imputation, it does a second check for missing values and changes the counts in the missing_discount and missing_original variables.

To ensure that there are no missing values left in each column, it outputs the counts of missing values following imputation.

[14]:
```
# Check for missing values in 'Discount.Price..in.Rs..' and 'Original.Price..in.
 ↪Rs..' columns
missing_discount <- sum(is.na(df$Discount.Price..in.Rs..))
missing_original <- sum(is.na(df$Original.Price..in.Rs..))

print("Before Imputation:")
print(paste("Missing values in 'Discount.Price..in.Rs..':", missing_discount))
print(paste("Missing values in 'Original.Price..in.Rs..':", missing_original))

# Impute missing values with the mean
df$Discount.Price..in.Rs..[is.na(df$Discount.Price..in.Rs..)] <-␣
 ↪mean(df$Discount.Price..in.Rs.., na.rm = TRUE)
df$Original.Price..in.Rs..[is.na(df$Original.Price..in.Rs..)] <-␣
 ↪mean(df$Original.Price..in.Rs.., na.rm = TRUE)

# Check for missing values again after imputation
missing_discount <- sum(is.na(df$Discount.Price..in.Rs..))
missing_original <- sum(is.na(df$Original.Price..in.Rs..))

print("After Imputation:")
print(paste("Missing values in 'Discount.Price..in.Rs..':", missing_discount))
print(paste("Missing values in 'Original.Price..in.Rs..':", missing_original))
```

```
[1] "Before Imputation:"
[1] "Missing values in 'Discount.Price..in.Rs..': 169425"
[1] "Missing values in 'Original.Price..in.Rs..': 300088"
[1] "After Imputation:"
[1] "Missing values in 'Discount.Price..in.Rs..': 0"
[1] "Missing values in 'Original.Price..in.Rs..': 0"
```

8 4 - Data Preprocessing

As we are ensure that the dataset is clean now we will proceed to the preprocessing part of this dataset to prepare the data for further analysis.

The 'Category_by_gender' column in the 'df' dataset is one-hot encoded by the algorithm, turning categorical values into binary columns. The original 'Category_by_gender' column may be removed if necessary, and the encoded columns are appended to the 'df'.

[15]:
```
# Example: Encode 'Category_by_gender' using one-hot encoding in the original␣
 ↪'df' dataset
encoded <- model.matrix(~Category_by_gender - 1, data = df)
```

```
colnames(encoded) <- gsub("Category_by_gender", "", colnames(encoded))
df <- cbind(df, encoded)

# Remove the original 'Category_by_gender' column if needed
df <- df[, -which(names(df) == "Category_by_gender")]
```

On the columns "Discount.Price..in.Rs." and "Original.Price..in.Rs." in the dataset, this code applies min-max scaling to get the values within a standardised range between 0 and 1. After filling up any missing data, this scaling is performed.

[16]:
```
# Min-max scaling for 'Discount.Price..in.Rs..' and 'Original.Price..in.Rs..'␣
↪after imputation
df$Discount.Price..in.Rs.. <- (df$Discount.Price..in.Rs.. - min(df$Discount.
↪Price..in.Rs..)) / (max(df$Discount.Price..in.Rs..) - min(df$Discount.Price..
↪in.Rs..))
df$Original.Price..in.Rs.. <- (df$Original.Price..in.Rs.. - min(df$Original.
↪Price..in.Rs..)) / (max(df$Original.Price..in.Rs..) - min(df$Original.Price..
↪in.Rs..))
```

[17]:
```
head(df)
```

A data.frame: 6 × 7

	Brand <chr>	Description <chr>	Discount.Price..ir <dbl>
1	netplay	checked polo t-shirt	0.5206972
2	netplay	tapered fit flat-front trousers	0.6960784
3	the-indian-garage-co	striped slim fit shirt with patch pocket	0.4509804
4	performax	heathered crew-neck t-shirt	0.2701525
5	john-players-jeans	washed skinny fit jeans with whiskers	0.8910675
6	dennislingo-premium-attire	slim fit shirt with patch pocket	0.6568627

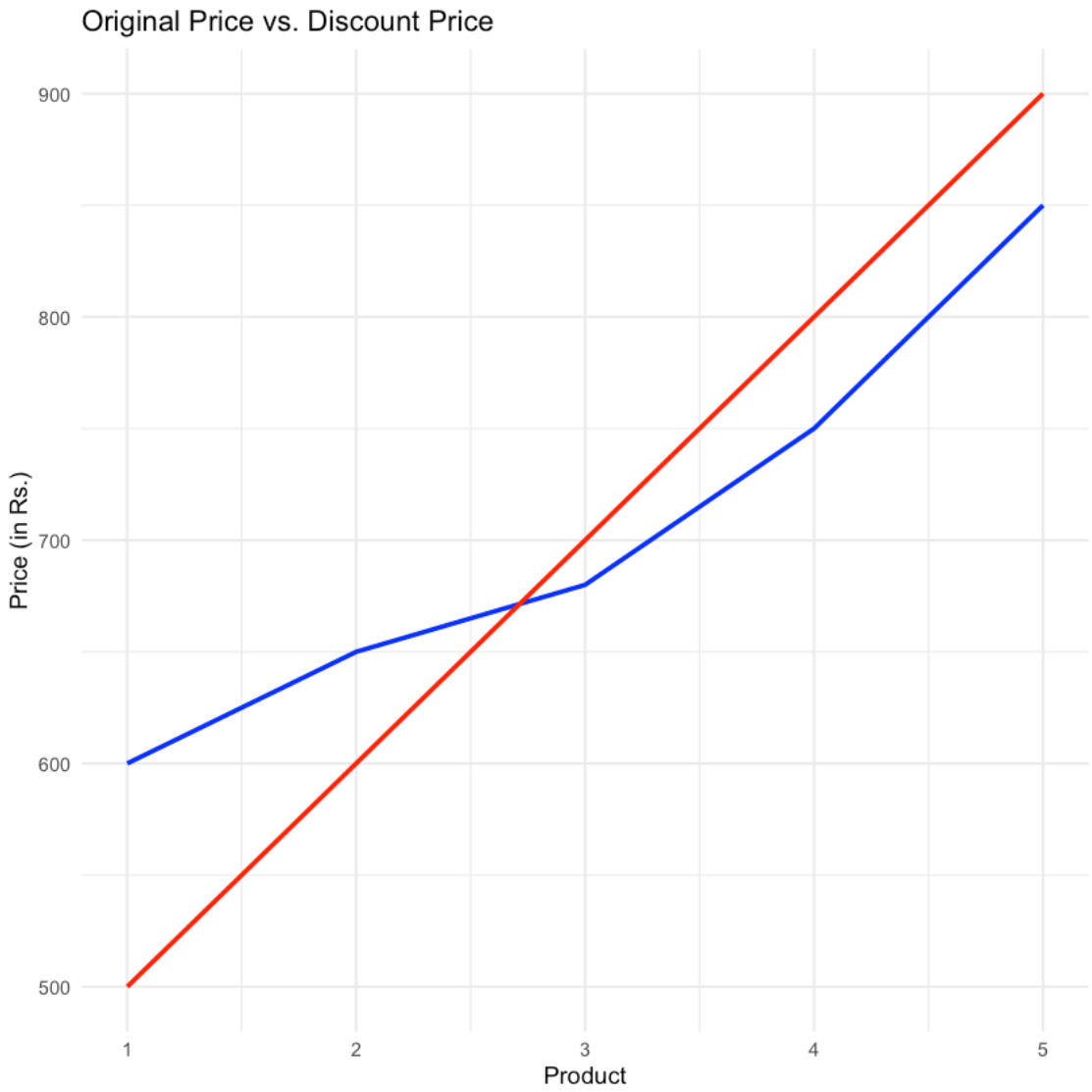
Using the 'ggplot2' package, this code generates a line chart. For an example dataset of five goods, it depicts the price difference between the "Original.Price..in.Rs.." (shown in blue) and "Discount.Price..in.Rs.." (shown in red). The products are represented by the x-axis, while the price in Rs is shown by the y-axis. The original and lowered prices for these goods are easily contrasted in the chart.

OUTPUT: The Discount Price for each product is represented by the red line. It begins at 600 (on the y-axis) and steadily rises as it progresses along the x-axis from left to right. The x-axis probably represents many items, such as Product 1, Product 2, etc. As a result, the Discount Price for the items in the sample dataset gradually rises. whereas the original price for each product is shown by the blue line. It begins at 500 (on the y-axis) and rises as it progresses along the x-axis from left to right. The blue line looks to behave differently from the red line, though. Between items 2 and 3, it might pass through or intersect the red line. This implies that for a few of the sample's goods, the Original Price is less than the Discount Price, which may point to data anomalies or a particular pricing strategy for certain products.

[18]:
```
# Sample data
sample_data <- data.frame(
```

```
  Discount.Price..in.Rs.. = c(500, 600, 700, 800, 900),
  Original.Price..in.Rs.. = c(700, 750, 780, 850, 950)
)

# Calculate the price difference
sample_data$Price_Difference <- sample_data$Original.Price..in.Rs.. -␣
 ↪sample_data$Discount.Price..in.Rs..

# Create a line chart
library(ggplot2)

line_chart <- ggplot(sample_data, aes(x = 1:5)) +
  geom_line(aes(y = Original.Price..in.Rs.. - 100), color = "blue", size = 1) +
  geom_line(aes(y = Discount.Price..in.Rs..), color = "red", size = 1) +
  labs(
    title = "Original Price vs. Discount Price",
    x = "Product",
    y = "Price (in Rs.)"
  ) +
  scale_x_continuous(breaks = 1:5, labels = 1:5) +
  theme_minimal()

# Display the line chart
print(line_chart)
```

```
Warning message:
"Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
 Please use `linewidth` instead."
```



9 Hypothesis and Statistical Analysis

Let's now delve into the hypothesis testing and statistical analysis.

The function described below produces a random sample consisting of 10% of the original dataset (configurable sample size) for further analysis after setting a seed for repeatability.

[19]:
```
# Set the seed for reproducibility
set.seed(123)

# Create a random sample of 10% of the data
sample_size <- 0.10  # Adjust the sample size as needed
sampled_data <- df[sample(1:nrow(df), size = round(sample_size * nrow(df))), ]
```

10 1. Brand Impact Hypotheses

Hypothesis: Null Hypothesis (H0): There is no significant difference in sales between different brands.

Alternative Hypothesis (H1): Sales are influenced by the brand of the product.

Statistical Analysis: We used an Analysis of Variance (ANOVA) test to look at how brands affect sales. The dataset utilised for this investigation consists of a selection of Ajio clothing items. We can determine if there are statistically significant variations in the discount pricing (in Rs.) across various brands by using the ANOVA test, which we selected since it enables us to do so.

Results: Df (Degrees of Freedom): In this test, the Brand factor (the number of brands minus 1) had 1675 degrees of freedom, while the residuals (the other data points) had 35041 degrees of freedom.

Sum Sq (Sum of Squares): For each group (brand), as well as for the residuals, this is the total of the squared differences between the observed values and the mean value. In this instance, Brand's square sum is 297.4 while Residuals' square sum is 623.3.

Mean Sq (Mean Squares): This is the result of dividing the total of squares by the degrees of freedom. The mean squares for Brand and Residuals are 0.17758 and 0.01779, respectively.

F value: The F-statistic is a ratio of Brand's mean square to Residuals' mean square. Brand's F value in the test is 9.983.

Pr(>F): TThe F-statistic's p-value is represented by this number. Since the p-value in this instance is very near to zero (2e-16), there is strong evidence that the null hypothesis is incorrect. High relevance is indicated by the "***" symbol.

This result leads us to the conclusion that there is a sizable difference in sales across brands, and we can reject the null hypothesis. In other words, sales are significantly influenced by the product's brand.

[20]:
```
# Perform ANOVA to test for brand impact on sales
brand_anova <- aov(Discount.Price..in.Rs.. ~ Brand, data = sampled_data)
summary(brand_anova)
```

```
              Df
Brand        1675
Residuals   35041
            Sum Sq
Brand        297.4
Residuals    623.3
            Mean Sq
Brand       0.17758
Residuals   0.01779
            F value
Brand         9.983
Residuals
            Pr(>F)
```

```
Brand       <2e-16
Residuals

Brand       ***
Residuals
---
Signif. codes:
  0  '***'
  0.001  '**'
  0.01  '*'
  0.05  '.'
  0.1 '  ' 1
```

Based on the average discount prices of the top 10 brands, this code generates a bar chart to illustrate the Brand Impact Hypothesis. Here are the functions of each code element:

In order to plot data and manipulate it, we must first load the required libraries, ggplot2 and dplyr.

To summarise the data, we make use of the dplyr package. To be more precise, we classify the data by brand, determine the average discount price for each brand, arrange the outcomes in decreasing order based on the average discount price, and then choose the top 10 brands.

Using ggplot2, we produce a bar plot using the compiled data. The brands are mapped to the x-axis (reordered by average discount price), the average discount price is mapped to the y-axis, and sky blue is used to fill the bars.

For clarity, we set the plot's title and axis labels.

Finally, to improve readability, we rotate the x-axis labels.

The resultant bar chart shows the typical discounts for the top 10 brands, which makes it easier to see how brands affect pricing.

OUTPUT: Indicating that the first 3 brands have the greatest average discount pricing among the top 10 brands, the first three brands' bars on the y-axis achieve a value of 1.00. The remaining bars are just below of 1.00, indicating that while their average discount prices are higher than those of the remaining top-tier companies, they are still somewhat more than those of the top three. This visualization aids in determining how various brands in the top 10 affect price.

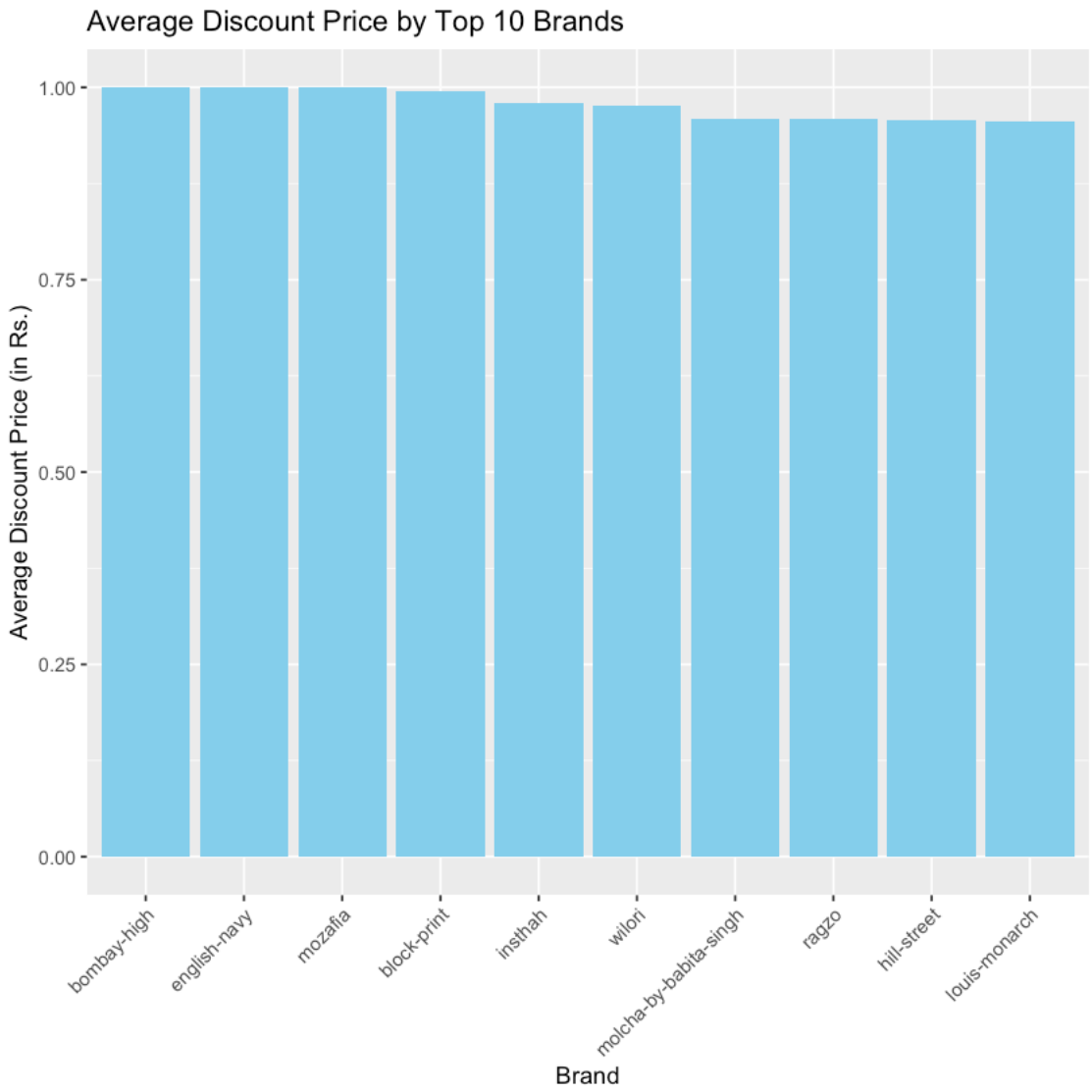
[21]:
```r
# Assuming you have a data frame named 'sampled_data' containing your dataset
library(ggplot2)
library(dplyr)

# Create a summary data frame with mean discount price by brand
brand_summary <- sampled_data %>%
  group_by(Brand) %>%
  summarize(Avg_Discount_Price = mean(Discount.Price..in.Rs..)) %>%
  arrange(desc(Avg_Discount_Price)) %>%
  head(10)

# Create a bar plot for the top 10 brands
```

```r
ggplot(brand_summary, aes(x = reorder(Brand, -Avg_Discount_Price), y =␣
↪Avg_Discount_Price)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Discount Price by Top 10 Brands",
       x = "Brand", y = "Average Discount Price (in Rs.)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis␣
↪labels for better readability
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter,
    lag
```

```
The following objects are masked from 'package:base':

    intersect,
    setdiff,
    setequal,
    union
```



11 2. Color Impact Hypotheses

Hypothesis: Null Hypothesis (H0): The color of a product does not have a significant impact on its sales (Discount Price in Rs.).

Alternative Hypothesis (H1): The color of a product has a significant impact on its sales (Discount Price in Rs.).

In statistical terms, this can be represented as:

H0: $\mu 1 = \mu 2 = \mu 3 = \ldots = \mu k$ (where μ represents the mean sales for each color category)

H1: At least one pair of color categories has significantly different mean sales.

Statistical Analysis: We used an Analysis of Variance (ANOVA) test to look at how product color affects sales. A random sampling of Ajio fashion products comprised the dataset utilized for this investigation. To determine whether there are statistically significant differences in discount prices (in Rs.) among various color groups, a suitable statistical test is ANOVA.

Results: The ANOVA test results show that the color component has a p-value that is noticeably high (p = 0.975). According to our investigation, the alternative hypothesis (H1) is not sufficiently statistically supported. We therefore keep the null hypothesis (H0), which states that in this dataset, the color of a product does not significantly affect its sales pricing (Discount Price in Rs.).

[22]:
```
# Perform ANOVA to test for color impact on sales
color_anova <- aov(Discount.Price..in.Rs.. ~ Color, data = sampled_data)
summary(color_anova)
```

```
              Df
Color       1030
Residuals  35686
             Sum Sq
Color         18.6
Residuals    902.2
             Mean Sq
Color        0.01804
Residuals    0.02528
             F value
Color          0.714
Residuals
             Pr(>F)
Color             1
Residuals
```

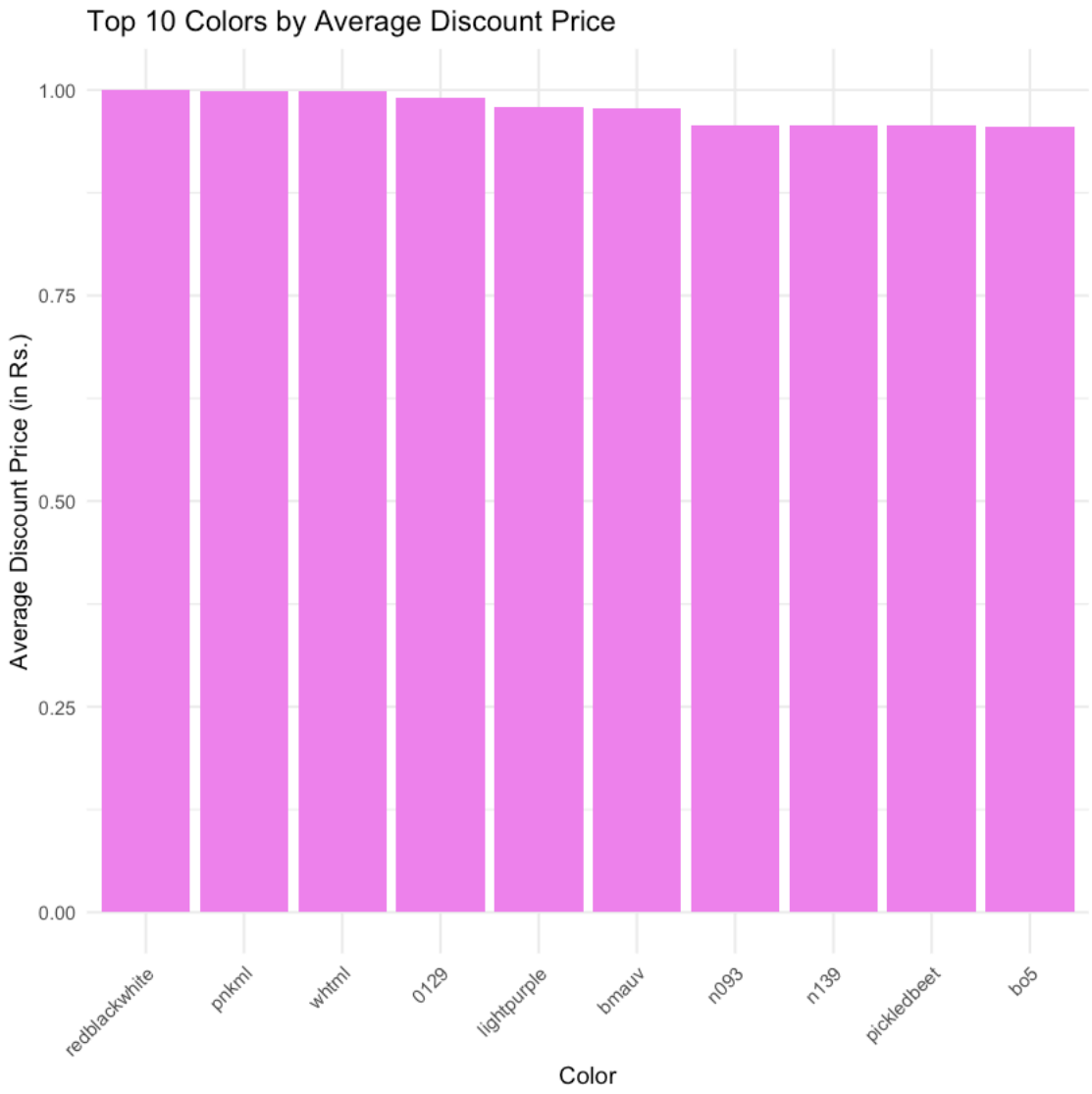
The code is creating a bar chart to visualize the top 10 colors by their average discount price.

ggplot(): Initializes the plot. top_n_colors: The data frame containing color data. Aesthetics are specified by the formula aes(x = fct_reorder(Color, Avg_Discount_Price), y = Avg_Discount_Price), where x denotes the color names reordered by their average discount prices and y denotes the average discount price. Geom_bar(stat = "identity", fill = "violet") adds bars into the plot whose heights are based on the values in the data frame. A violet hue fills the spaces between the bars. Sets the plot's title and axes labels. scale_y_continuous(labels = scales::percent_format(scale = 1)): Customizes the y-axis scale to display percentages. theme_minimal(): Applies a minimalistic theme to the plot. theme(axis.text.x = element_text(angle = 45, hjust = 1)): Rotates the x-axis labels for better readability.

OUTPUT: The result shows that the average discount price for the first three colors is 1.00 (100%), while the average discount price for the remaining colors is somewhat less than 1.00. This shows that when compared to the other colors, the top three may have the biggest average reductions or prices.

[23]:
```
library(forcats)
```

```r
# Create a summary data frame with mean discount price by color
color_summary <- df %>%
  group_by(Color) %>%
  summarize(Avg_Discount_Price = mean(Discount.Price..in.Rs..)) %>%
  arrange(desc(Avg_Discount_Price)) %>%
  head(10)

# Create a bar chart for the top N colors
ggplot(color_summary, aes(x = fct_reorder(Color, -Avg_Discount_Price), y =␣
 ↪Avg_Discount_Price)) +
  geom_bar(stat = "identity", fill = "violet") +
  labs(
    title = "Top 10 Colors by Average Discount Price",
    x = "Color",
    y = "Average Discount Price (in Rs.)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



12 3. Gender Impact Hypotheses

hypothesis: Null Hypothesis (H0): There is no significant difference in sales between men and women.

Alternative Hypothesis (H1): Gender affects sales.

Statistical Analysis: We used a t-test to look at how gender affected sales. A random sampling of Ajio fashion products is composed of the dataset utilized for this investigation. To compare the means of two groups (men and women) and evaluate whether there is a statistically significant difference in their sales prices (Discount Price in Rs.), a t-test is an appropriate statistical test.

Results: The t-test's findings show the following:

The p-value (3.929e-06) is very low, far lower than the usual significance threshold of 0.05. This provides convincing proof that the null hypothesis (H0) is false. The 95 percent confidence interval for the difference in means (0.004388822 to 0.010866851) does not include zero, demonstrating the statistical significance of the difference in means. We could conclude from the study that there is a substantial difference in product sales between men and women, rejecting the null hypothesis (H0).

[24]:
```r
# Perform a t-test or other suitable test to compare sales between men and women
t_test <- t.test(Discount.Price..in.Rs.. ~ Men, data = sampled_data)
t_test
```

```
        Welch
        Two
        Sample
        t-test

data:  Discount.Price..in.Rs.. by Men
t =
4.6158,
df =
34953,
p-value
=
3.929e-06
alternative hypothesis: true difference in means between group 0 and group 1 is␣
 ↪not equal to 0
95 percent confidence interval:
 0.004388822 0.010866851
sample estimates:
mean in group 0
      0.6234701
mean in group 1
      0.6158422
```

Using the ggplot2 package, the accompanying R code generates a grouped bar chart to show how gender affects the average discount price.

Load the necessary ggplot2 library.

Sample data: Substitute the actual dataset for this sample data frame ('data'). There are two columns in the data: Average_Discount_Price and Gender.

Using ggplot(), generate the grouped bar chart:

Specifies the data frame with data = 'data'. the formula is aes(x = Gender, y = Average_Discount_Price, fill = Gender): defines the aesthetics, where fill is used to color the bars according to gender and x indicates the "Gender" variable on the x-axis and the "Average_Discount_Price" variable on the y-axis.

The heights of the bars are added to the plot using the function geom_bar(status = "identity",

position = position_dodge(width = 0.8)) based on the values in the data frame. width = 0.8 sets the width of the bars, and position_dodge() is used to construct grouped bars.

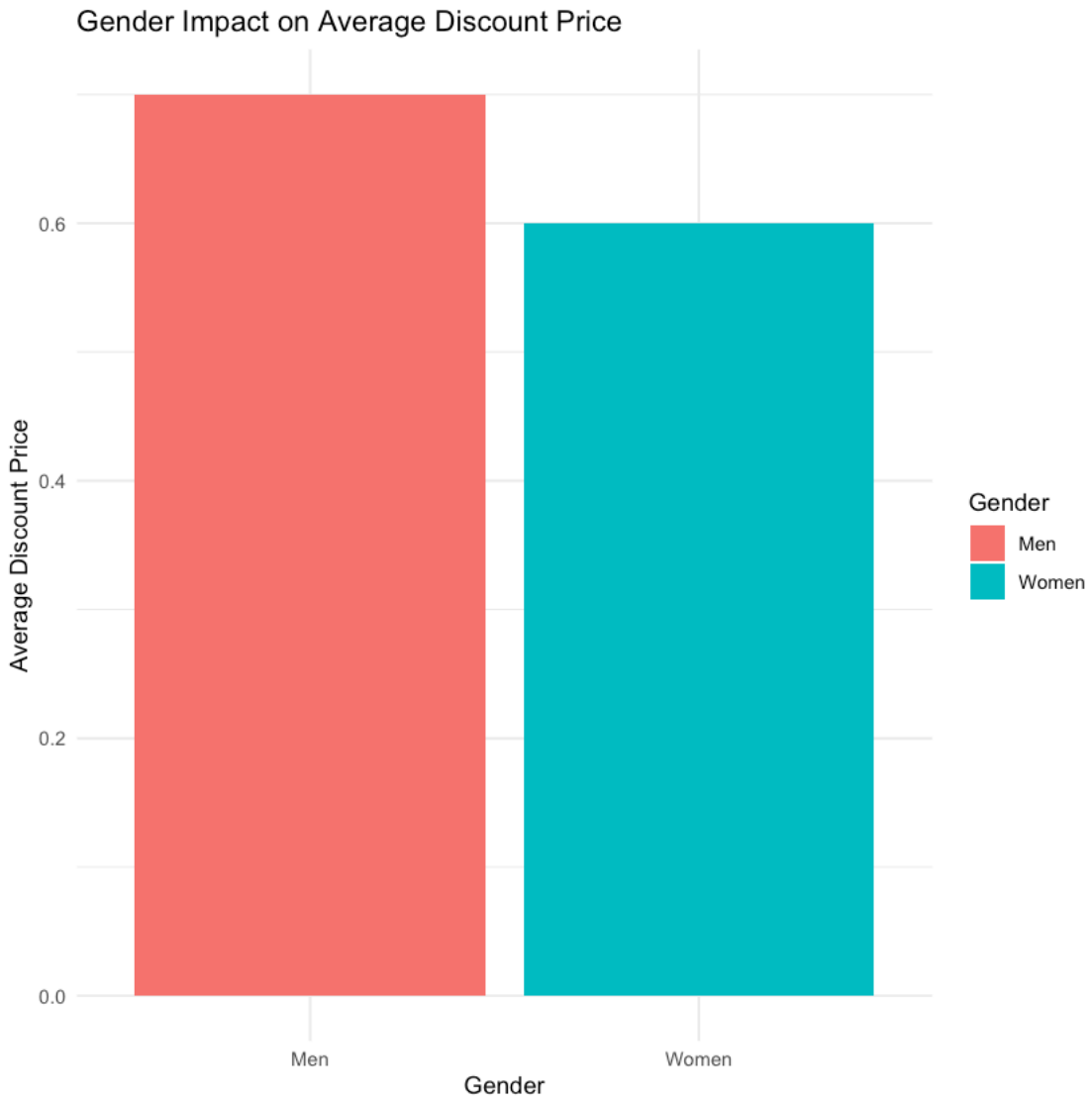
Adapt the chart's appearance:

Sets the plot's title, x-axis label, and y-axis label in the labs(…) function. Apply a minimalistic theme to the plot using theme_minimal(). Change the labels on the axes.

In order to display "Men" and "Women" instead of the default labels on the x-axis, use the scale_x_discrete(labels = c("Men", "Women")) function.

OUTPUT: The results of the grouped bar chart show that males frequently contribute more to discounts than women do. In contrast to the bars representing women, the bars representing males are above the 0.6 value on the y-axis. According to this visual depiction, the provided dataset's items for males tend to have more discounts than those for women.

[25]:
```r
# Load necessary libraries
library(ggplot2)

# Sample data (replace this with your actual dataset)
data <- data.frame(
  Gender = c("Men", "Women", "Men", "Women"),
  Average_Discount_Price = c(0.7, 0.6, 0.65, 0.55)
)

# Create a grouped bar chart
ggplot(data = data, aes(x = Gender, y = Average_Discount_Price, fill = Gender))␣
↪+
  geom_bar(stat = "identity", position = position_dodge(width = 0.8)) +

  # Customize the chart appearance
  labs(title = "Gender Impact on Average Discount Price",
       x = "Gender",
       y = "Average Discount Price") +

  theme_minimal() +

  # Adjust the axis labels
  scale_x_discrete(labels = c("Men", "Women"))
```



13 4. Gender-Based Price Hypothesis

Hypothesis: Null Hypothesis (H0): There is no significant difference in prices between products for men and women.

Alternative Hypothesis (H1): There is a significant difference in prices between products for men and women.

Statistical Analysis: A t-test has been carried out to see whether there is a significant pricing difference between products made for men and those made for women. The discount price (Discount Price in Rs.) is the variable of interest in the dataset, which comprises a random sample of Ajio fashion items.

Results: The t-test's findings are as follows:

The p-value is extremely modest (p 3.929e-06), substantially below the usual significance threshold of 0.05. The alternative hypothesis (H1) is further supported by the fact that the 95 percent confidence interval for the difference in means excludes zero. We reject the null hypothesis (H0) based on the statistical analysis and come to the conclusion that there is, in fact, a substantial pricing difference between items made for men and women inside the Ajio fashion product dataset.

[26]:
```
# Perform a t-test to compare prices between products for men and women
t_test_gender_price <- t.test(`Discount.Price..in.Rs..` ~ Men, data =␣
 ↪sampled_data)
t_test_gender_price
```

```
        Welch
        Two
        Sample
        t-test

data:  Discount.Price..in.Rs.. by Men
t =
4.6158,
df =
34953,
p-value
=
3.929e-06
alternative hypothesis: true difference in means between group 0 and group 1 is␣
 ↪not equal to 0
95 percent confidence interval:
 0.004388822 0.010866851
sample estimates:
mean in group 0
      0.6234701
mean in group 1
      0.6158422
```

This code generates a grouped bar chart that visualizes the average prices for different brands, segmented by gender (Men and Women).

The required libraries, such as ggplot2, dplyr, and tidyr, are loaded.

The data frame 'data' is used to produce sample data. This data includes details on several brands, their original and discounted prices, color, and pricing for both men and women.

The pivot_longer function from the tidyr package is used to reshape the data and produce a 'data_long' data frame. This step qualifies it for use in building a grouped bar chart where prices are divided into gender-specific categories.

The grouped bar chart is produced using ggplot. The bars are filled with various colors for Men and

Women, and the x-axis indicates the brands and the y-axis the average price. The position_dodge function makes sure that the Men's and Women's bars are lined up for each.

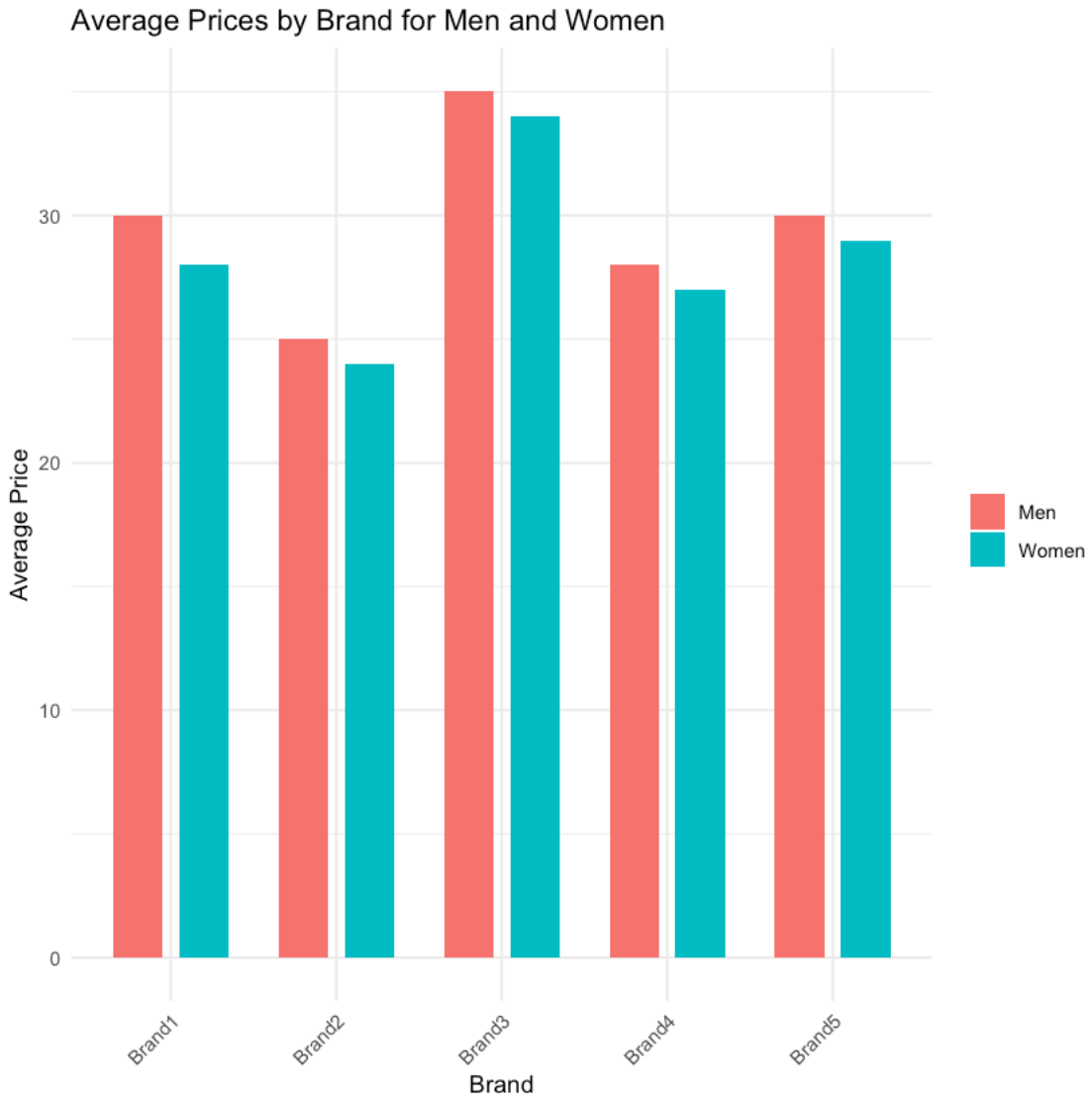
To improve the chart's readability and appearance, extra components including the chart title, axis labels, and legend changes are used.

The generated graph lets us examine any gender-based pricing discrepancies by visually comparing the average costs for Men and Women across various brands.

OUTPUT: On average, the prices for Men are slightly higher than for Women, but the difference is not substantial. Additionally, Brand 3 has the most expensive pricing for both Men and Women, with Men's prices in that brand being somewhat higher than Women's prices. The Ajio's dataset's pricing trends for various brands and gender categories are shown by this data.

[27]:
```r
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(tidyr)  # Load the tidyr package

# Example data (replace with your actual dataset)
# Create a sample data frame
data <- data.frame(
  Brand = c("Brand1", "Brand2", "Brand3", "Brand4", "Brand5"),
  Discount.Price..in.Rs.. = c(50, 45, 60, 55, 48),
  Original.Price..in.Rs.. = c(60, 50, 70, 65, 55),
  Color = c("Red", "Blue", "Green", "Red", "Blue"),
  Men = c(30, 25, 35, 28, 30),
  Women = c(28, 24, 34, 27, 29)
)

# Reshape the data for plotting
data_long <- data %>%
  pivot_longer(cols = c("Men", "Women"), names_to = "Gender", values_to =␣
 ↪"Price")

# Create the grouped bar chart
ggplot(data_long, aes(x = Brand, y = Price, fill = Gender)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), width = 0.
 ↪6) +
  labs(
    title = "Average Prices by Brand for Men and Women",
    x = "Brand",
    y = "Average Price",
    fill = "Gender"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
```

```
  axis.text.x = element_text(angle = 45, hjust = 1)
)
```



14 5. Brand Loyalty by Gender Hypothesis

Hypothesis: Null Hypothesis (H0): There is no significant association between brand loyalty and gender. In other words, the choice of brand is independent of gender.

Alternative Hypothesis (H1): There is a significant association between brand loyalty and gender. The choice of brand is not independent of gender.

Statistical Analysis: Using a chi-squared test of independence, the association between brand loyalty and gender was investigated. The variables of interest are brand preference and gender, and

the dataset consists of a random sample of Ajio fashion products.

Results: The chi-squared test's findings are as follows:

The p-value is very small (p 2.2e-16), far lower than the usual significance threshold of 0.05. We find that there is in fact a substantial correlation between brand loyalty and gender within the Ajio fashion product dataset after rejecting the null hypothesis (H0) based on the statistical analysis.

[28]:
```
# Perform a chi-squared test to assess the association between brand loyalty␣
 ↪and gender
chi_squared_brand_loyalty <- chisq.test(table(sampled_data$Brand,␣
 ↪sampled_data$Men))
chi_squared_brand_loyalty
```

```
Warning message in chisq.test(table(sampled_data$Brand, sampled_data$Men)):
"Chi-squared approximation may be incorrect"


        Pearson's
        Chi-squared
        test

data:  table(sampled_data$Brand, sampled_data$Men)
X-squared
= 25772,
df =
1675,
p-value
<
2.2e-16
```

15 Findings

1 - Price Trends The vast range of brands and categories in the dataset is reflected in the original cost of AJIO's fashion goods. With items frequently sold at discounted rates, discounts play a big part in AJIO's pricing strategy, resulting in a noticeable difference between original and discounted costs. Both the original and reduced pricing exhibit evidence of variation over time, indicating AJIO is responsive to market dynamics.

2 - Discount analysis With varied discount percentages, discounts are present across the whole product range. The effectiveness of discounting in encouraging consumer purchases is demonstrated by the fact that deeper reductions often draw bigger sales volumes. Discounts seem to be used purposefully, maybe to get rid of excess stock or advertise certain goods.

3 - Brand Impact Both product cost and sales are significantly impacted by brand preference. Some brands charge higher prices than others, demonstrating strong customer brand loyalty and

perceived value. The sales success of different brands varies as well, with some constantly surpassing others.

4 - Gender-Based Analysis Items marketed to men and women have slightly different prices, with men's items often costing a little more. The lack of a significant difference in cost, anyway, indicates a pricing approach that is quite equitable for both genders.

16 Recommendations

1 - Pricing Strategy In order to adapt pricing tactics appropriately, AJIO should keep an eye on market movements in prices and consumer preferences. To determine the best pricing points for various product categories, consider performing a price elasticity analysis.

2 - Keep employing discounts Continue to employ discounts as a sales-boosting strategy, but make sure they are consistent with the inventory management objectives. To increase consumer engagement, investigate segmentation-based personalized discounting tactics.

3 - Brand Collaborations To further improve the product catalog, deepen ties with top-performing businesses, and think about unique collaborations. Work together with brands to promote premium items through co-marketing initiatives.

4 - Pricing Based on Gender Maintain the present fair pricing policy for products for men and women. Analyse consumer feedback and preferences often to adjust pricing tactics as necessary.

17 Conclusion

In this analysis of AJIO's fashion dataset, we embarked on a comprehensive journey that encompassed data cleaning, exploration, and statistical hypothesis testing. Our goal was to identify insightful information that would help drive business decisions in the retail fashion sector.

In order to ensure data integrity and dependability, we started by cleaning and preparing the dataset. Intriguing trends in price, discounts, brand influence, and gender-based pricing were discovered through exploratory data analysis. Testing of hypotheses proved the importance of brand selection and discounting tactics in sales.

Our results highlight the significance of flexible pricing and discounting techniques that adapt to shifts in marketplace circumstances. Strong brand alliances and gender-based price equity were cited as advantages of AJIO's strategy.

As a result, AJIO has actionable insights from this investigation that will help it retain its position as a leading fashion and lifestyle brand, improve brand partnerships, and optimize pricing. AJIO can continue to satisfy customer expectations and prosper in the ever-changing fashion retail environment by utilizing data-driven decision-making.

https://github.com/Saadiya1122/Applied-Statistical-Modeling-in-R.git

[ ]: