The Price Difference Between Men's and Women's Shoes

Alicia Chong Tsui Ying
20074290
School Of Engineering & Technology
Sunway University

Abstract

Shoes are important for all genders; however, shoe brands and shoe prices vary according to gender. The purpose of this report is to explore and discover the popularity of shoe brands according to gender, and to uncover the price differences of shoe brands against gender. The men's and women's shoe prices dataset from Datafiniti's Product Database was used to undergo data exploration. With the problem statements in hand, only a few meaningful variables are kept to undergo further analysis. Furthermore, inconsistent, irrelevant, and duplicated data are treated to increase the accuracy of the results. Next, R Studio is used to generate the results to the problem statements in tables and figures for a better understanding and visualisation. The validated dataset is imported to R Studio, and further adjustments had been made such as merging *WomenShoes* and *MenShoes* datasets, creating new variables, and changing the data type. The results generated shows that among the popular brands for both genders, men's shoes take up a larger portion among the Top 20 Popular Shoe Brands, and the median price for men's shoes are higher than women's shoes.

Introduction

In this assignment, two large sample datasets namely women's shoe prices and men's shoe prices from Datafiniti's Product Database were explored for critical analysis. As a basic overview, each dataset contains information regarding shoe name, brand, price, product description, manufacturer, and other attributes of product. SAS and R programming language is used for the analysis of these datasets.

The Data Exploration of Footwear Against Gender

Problem Statement

In both datasets, the popularity of the shoe brands and the price of the shoes against gender are crucial. Hence, the first problem statement is –Which gender has more products listed among popular brands? With this problem statement, the product listings for men's and women's shoes among popular brands will be displayed, giving the insight on whether popular brands are leaning towards a gender. The second problem statement is – What is the difference between men's and women's shoe prices. With this, the difference in men's median shoe price and women's median shoe price will give an insight on which gender has a higher shoe price on average.

Data Handling

A. Data Import

SAS Enterprise Guide 7.1

SAS Enterprise Guide 7.1 was chosen over SAS Studio as our SAS programming interface because the given datasets were too large for SAS studio (only 6MB of memory space available) to handle.

We started off by creating a permanent library named AECW which stands for Analytics Engineering Course Work with the LIBNAME statement to specify the location of our files. Two SAS macros variables were created for more efficient coding by storing the input and output path of the input and output dataset. OPTIONS VALIDVARNAME=V7 system option is used to change column names in the .csv file to adhere to recommended SAS naming conventions (see Figure 1).

```
%LET inpath=C:\Users\User\Documents\My SAS Files\AE Assignment\Input;
%LET outpath=C:\Users\User\Documents\My SAS Files\AE Assignment\Output;
LIBNAME AECW "C:\Users\User\Documents\My SAS Files\AE Assignment\Output";
OPTIONS validvarname=v7;
```

Figure 1: creation of permanent library, creation of macro variables and usage of OPTIONS VALIDVARNAME=V7

Given that both dataset files are in the .csv format, each dataset was imported separately into SAS Enterprise Guide with the PROC IMPORT procedure and DATA STEP Procedure for men's shoe prices and women's shoe prices dataset respectively (see Figure 2 & 3). DATA STEP was chosen over PROC IMPORT to read women's shoe dataset because SAS could not determine the correct variable types for this dataset with PROC IMPORT (see Figure 3).

```
PROC IMPORT datafile="&inpath\Men shoe prices.csv"
    DBMS=CSV
    OUT= AECW.MenShoe_Import
    REPLACE;
    guessingrows=max;
RUN;
```

Figure 2: Importing Men's shoes dataset

```
data AECW.WomenShoe_import;
    %let _EFIERR_ = 0;

    infile "&inpath\Women shoe prices.csv" delimiter=',' MISSOVER DSD firstobs=2;
    informat id $20.;
    informat asins $100.;
    informat brand $40.;
    informat categories $500.;
    informat colors $450.;
    informat count $1.;
    informat dateAdded B8601DZ35.;
    informat dateUpdated B8601DZ35.;
    informat descriptions $25522.;
    informat dimension $37.;
    informat ean best32.;
    informat features $2056.;
    informat flavors $1.;
    informat imageURLs $3160.;
    informat isbn $1.;
    informat keys $558.;
    informat manufacturer $35.;
    informat manufacturerNumber $94.;
    informat merchants $891.;
    informat name $279.;
    informat prices_amountMin $45.;
    informat prices_amountMax $26.;
```

Figure 3: Snippet of code for Importing Women's shoes dataset

By reading the log, we can observe that Men's shoe dataset contains 19387 observations and 52 variables whereas the women's shoe dataset contains 19045 observations and 47 variables.

B. Data Exploration

After importing the dataset into SAS Enterprise Guide, it was necessary to deploy data exploration techniques to better understand the dataset. As shown in figure 4, to get an overview of the dataset, PROC CONTENTS step was written to view the table attributes as it creates a report of the descriptor portion of the table. Then, PROC PRINT was written to take a glimpse on the first 5 observations of the datasets.

```
/****************************Exploring data*******************************/
proc contents data=AECW.MenShoe_Import varnum;
Run;

proc print data=AECW.MenShoe_Import (firstobs=2 obs=5);
Run;
```

Figure 4: Data exploration to better understand dataset

The CONTENTS Procedure

Data Set Name	AECW.MENSHOE_IMPORT	Observations	19387
Member Type	DATA	Variables	52
Engine	V9	Indexes	0
Created	11/21/2021 14:09:43	Observation Length	114592
Last Modified	11/21/2021 14:09:43	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	229376
Number of Data Set Pages	9694
First Data Page	1
Max Obs per Page	2
Obs in First Data Page	1
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\User\Documents\My SAS Files\AE Assignment\Output\menshoe_import.sas7bdat
Release Created	9.0401M3
Host Created	X64_8HOME

Variables in Creation Order

#	Variable	Type	Len	Format	Informat
1	id	Char	22	$22.	$22.
2	asins	Char	100	$100.	$100.
3	brand	Char	39	$39.	$39.
4	categories	Char	585	$585.	$585.
5	colors	Char	392	$392.	$392.
6	count	Char	1	$1.	$1.
7	dateAdded	Char	22	$22.	$22.
8	dateUpdated	Char	22	$22.	$22.
9	descriptions	Char	27220	$27220.	$27220.
10	dimension	Char	31	$31.	$31.
11	ean	Char	15	$15.	$15.
12	features	Char	2544	$2544.	$2544.
13	flavors	Char	1	$1.	$1.
14	imageURLs	Char	7983	$7983.	$7983.
15	isbn	Char	1	$1.	$1.
16	keys	Char	529	$529.	$529.
17	manufacturer	Char	37	$37.	$37.
18	manufacturerNumber	Char	61	$61.	$61.
19	merchants	Char	799	$799.	$799.
20	name	Char	156	$156.	$156.

Figure 5: Snippet of output from code in Figure 3

C. Data Preparation

After basic exploration and understanding of the dataset, data cleaning is required for identification, correction, or removal of inaccurate raw data for downstream purposes.

Only variables that are meaningful for analysis are kept whereas variables that are not useful will not be stored in the newly created dataset. Variables *id*, *brand*, *prices_currency*, *prices_amountMin*, *prices_amountMax*, and *price* from both *MenShoe_Import* and *WomenShoe_Import* datasets are copied to new datasets namely *MenShoe_UsefulVar* and *WomenShoe_UsefulVar*. The Mean function is used to find the mean value of *prices_amountMin* and *prices_amountMax*, the output of the function is then rounded with the ROUND function before being assigned to a new variable named *price*. Upon inspection, the values of variable *brand* have inconsistent character case; therefore, the UPCASE function is used to standardize all brand names to capital letter (see Figure 6).

```
data AECW.MenShoe_UsefulVar;
    set AECW.MenShoeImport;
    price = round(mean(prices_amountMin, prices_amountMax),1);
    brand = UPCASE(brand);
    keep id brand prices_currency prices_amountMin prices_amountMax price;
run;
```

Figure 6: Extracting meaningful variables for analysis

After that, PROC SORT with the NODUPKEY option together with the BY _ALL_ statement is used to remove adjacent rows that are entirely duplicated. The OUT= option specifies the output tables *MenShoes_NoDups* and *WomenShoes_NoDups*.

```
proc sort data=AECW.MenShoe_UsefulVar out=AECW.MenShoe_NoDups
    nodupkey;
    by _all_;
run;
```

Figure 7: Removing adjacent rows that are entirely duplicated

```
NOTE: There were 19387 observations read from the data set AECW.MENSHOECLEAN.
NOTE: 1270 observations with duplicate key values were deleted.
NOTE: The data set AECW.MENSHOE_NODUPS has 18117 observations and 5 variables.
```

Figure 8: Snippet of log output from code in Figure 7

As seen in Figure 8, a total of 1270 observations with duplicate key values were deleted from *MenShoe_NoDups* data set. The same process is applied to the *WomenShoe_UsefulVar* data set and 895 observations with duplicate key values were deleted from *WomenShoe_NoDups*.

After the removal of duplicated observations, PROC FREQ is used to create a frequency table for the variable prices_currency and to explore its attributes and to make note of the adjustments needed (see figure 9).

```
title "Frequency count for variable price_currency (before cleaning)";
proc freq data=AECW.MenShoe_NoDups nlevels;
    tables prices_currency / missing nocum nopercent;
run;
```

Figure 9: Check the Frequency count for variable price_currency

As seen in Figure 10, there consists of several incorrect data that must be cleaned. Furthermore, it is noticed that there consists of 5 different price currencies in this dataset, which are USD, AUD, CAD, EUR, and GBP.

Frequency count for variable price_currency before cleaning

The FREQ Procedure

Number of Variable Levels			
Variable	Levels	Missing Levels	Nonmissing Levels
prices_currency	14	1	13

prices_currency	Frequency
	58
107.00	1
118.36	1
119.79	1
147.95	1
35.95	1
AUD	338
CAD	298
EUR	104
GBP	21
New with box	2
New without tags	1
USD	17290
new	1

Figure 10: Result output from code in Figure 9

Another PROC FREQ is also used to explore the variable *brand*. For this procedure, the OUT= option is also used to output a temporary table named where the count of each brand is larger than 10, that way, only brands that are more significant to our interest for this research can be selectively cleaned.

```
title "Frequency count for variable brand";
proc freq data =AECW.MenShoe_NoDups nlevels;
  tables brand/ missing nocum nopercent
                out=MenBrand_freq10(where=(count>10));
run;
```

Figure 11: Check the Frequency count for variable brand

Figure 12 and 13 shows the output results and output table of the frequency count for each brand respectively.

Frequency count for variable brand

The FREQ Procedure

Number of Variable Levels			
Variable	Levels	Missing Levels	Nonmissing Levels
brand	1823	1	1822

brand	Frequency
	251
"""GANESHA HANDICRAFT """	7
"""HANDMADE"""	19
1031	1
12 STEP GOLD	2
14K CO.	36
180S	4
1883 BY WOLVERINE	1
1901	1
20-001707000	3
29 PORTER RD	10

Figure 12: Snippet of result output from code in Figure 11

MENBRAND_FREQ10

Filter and Sort | Query Builder | Where | D

	brand	COUNT	PERCENT
1		251	1.3853626228
2	""HANDMADE""	19	0.104868087
3	14K CO.	36	0.198697428
4	2BHIP	16	0.088309968
5	3N2	22	0.121426206
6	3N2 SPORTS	11	0.060713103
7	5.11 TACTICAL	23	0.126945579
8	A4	13	0.071751849
9	ACACIA	13	0.071751849
10	ACADEMIE	67	0.3697979909

Figure 13: Snippet of output table from code in Figure 11

After exploring the values of variable *price_currency* and *brand*, both variables need to be cleaned before they can be analysed.

A DATA STEP was used to clean the data from *MenShoe_NoDups* and *WomenShoe_NoDups* data set. As seen in figure 14, the correct price is reassigned to the *price* variable.

```
data AECW.MenShoe_Clean;
    set AECW.menshoe_nodups;

    * Clean oberservations which data has been read incorrectly in the price column;
        if id = "AVpe7ZLiLJeJML43yglY" then price = 119.79;
        else if id = "AVpe_F4BilAPnD_xSaI2" then price = 107.00;
        else if id = "AVpfEARPilAPnD_xUHSz" then price = 115.00;
        else if id = "AVpfKcs1LJeJML433u4w" then price = 114.87;
        else if id = "AVpfOiTgLJeJML435E6A" then price = 23.59;
        else if id = "AVpfPK_lLJeJML435Sc7" then price = 35.95;
        else if id = "AVpfQ9yKilAPnD_xYbdx" then price = 225.10;
        else if id = "AVpfUpYgLJeJML437D0P" then price = 35.99;
        else if id = "AVpfVOiALJeJML437Pu_" then price = 147.95;
        else if id = "AVpfYJenilAPnD_xaslU" then price = 94.99;
        else if id = "AVpfgGRcilAPnD_xc4DZ" then price = 225.10;
        else if id = "AVpfvrnKLJeJML43C9KK" then price = 225.10;
        else if id = "AVpfyWi1LJeJML43DuW5" then price = 118.36;
        else if id = "AVpfzZbh1cnluZ0-rkRN" then price = 163.99;
        else if id = "AVpgCgp6ilAPnD_xmcuZ" then price = 125.99;
```

Figure 14: Selectively clean price data that has been read correctly

As inspected in Figure 10, there are 5 different price currencies that must be standardized. All prices are converted to USD as it takes up majority of the data. Price currency conversion rate were based on data provided by Morningstar on the 13th of November 2021 (see Figure 15).

```
* Standardise the prices to USD as there contains several currencies
 (based on 13th November 2021 conversion rate);
if prices_currency in ("USD","AUD","CAD","EUR","GBP") then
    do;
        if prices_currency = "AUD" then price = round((price * 0.73),1);
        else if prices_currency = "CAD" then price = round((price * 0.8),1);
        else if prices_currency = "EUR" then price = round((price * 1.15),1);
        else if prices_currency = "GBP" then price = round((price * 1.34),1);
        prices_currency = "USD"; *now all price currency is in USD;
    end;
else prices_currency = .;
```

Figure 15: Standardise price to USD

Variable *price_currency* is then checked again with PROC FREQ for data validation purposes (see Figure 15 & 16).

```
title "Frequency count for variable price_currency (after cleaning)";
proc freq data=AECW.MenShoe_Clean;
    tables prices_currency / missing nocum nopercent;
run;
title;
```

Figure 16: Check the Frequency count for variable brand again (after cleaning)

Frequency count for variable price_currency (after cleaning)

The FREQ Procedure

prices_currency	Frequency
.	67
USD	18051

Figure 17: Snippet of output table from code in Figure 16

Variable brand is selectively cleaned based on the output of the code in figure 11.

```
* Selectively clean Popular Brand Names;
if brand = " " or brand in ("UNBRAND", "UNBRANDED/GENERIC") then brand = "UNBRANDED";
else if brand in ("N I K E", "NIKE - KOBE", "NIKE AIR JORDAN", "NIKE AIR JORDAN I",
                  "NIKE GOLF", "NIKE JORDAN FUTURE LOW", "NIKE LUNARGLIDE 7", "NIKE SB")
                  then brand = "NIKE";
else if brand in ("LAUREN RALPH LAUREN","RALPH LAUREN PURPLE LABEL", "RALPH LAUREN RLX",
                  "RALPH LAUREN RRL" ,"RALPH LAUREN YACHT", "RLX RALPH LAUREN")
                  then brand = "RALPH LAUREN";
else if brand = "PUMA SAFETY SHOES" then brand = "PUMA";
else if brand = "WOLVERINE" then brand = "WOLVERINE WORLDWIDE";
else if brand = "HUGO BY HUGO BOSS"  then brand = "BOSS HUGO BOSS";
else if brand = "ACADEMIE GEAR"  then brand = "ACADEMIE";
else if brand in ( "ALEXANDERS", "ALEXANDER" ) then brand = "ALEXANDER MCQUEEN";
```

Figure 18: Cleaning variable brand

After cleaning the men's shoes data set, similar process is done to the women's shoes data set. Once both men's and women's data set have been cleaned, both data sets are exported to their own respective CSV files using the SAS Output Delivery System (ODS) with PROC PRINT for further analysis in R studio. Only variable id, brand and price is selected to be exported. Both files are exported to a folder located in a local device with the path stored in the macro variable named outpath (See figure 18 & 19).

```
ODS CSVALL FILE="&outpath/MenShoe_Clean.csv";
proc print data=AECW.MenShoe_Clean noobs;
    var id brand price;
run;
ODS CSVALL CLOSE;
```

Figure 19: Cleaning variable brand Output cleaned dataset as a csv file

The purpose of using R studio for further analysis of the two data sets is R studio has more capability and flexibility in terms of producing plots and graphs.

R studio

In R studio, necessary packages are first invoked with the **library(package)** command to be loaded into the current session. For this project, package *tidyverse* and *ggplot2* is used. The method **setwd()** is then used to set a new directory and establish a save folder in the system. As shown in the figure below, after new directory is set, data is imported as *WomenSheos* and *MenShoes* using the **read.csv()** method (see figure 20).

```
# load libraries
library(tidyverse)
library(ggplot2)

# set working directory where the csv file located
setwd("C:/Users/User/Documents/My SAS Files/AE Assignment/Output")

# Read cleaned Women and Men shoes data set in csv format
WomenShoes = read.csv(file="WomenShoe_clean.csv", header=TRUE, sep=",")
MenShoes = read.csv(file="MenShoe_clean.csv", header=TRUE, sep=",")
```

Figure 20: Loading packages, setting a new directory, importing dataset

In figure 21, price column in *WomenShoes* and *MenShoes* is converted from character to numeric data type with **as.numeric()** method.

```
WomenShoes$price = as.numeric(WomenShoes$price)
MenShoes$price = as.numeric(MenShoes$price)
```

Figure 21: Converting price column to numeric data type

Before merging *WomenShoes* and *MenShoes* data set, a new variable *Gender* is created to differentiate these two data sets in the later merged data set (see figure 22).

```
# Create variable Gender
WomenShoes$gender = "Women"
MenShoes$gender = "Men"
```

Figure 22: Create variable gender

The **rbind**() method is used to combine *WomenShoes* and *MenShoes* data frames by rows to a new data frame named *all_shoes* (see figure 23). After that, the **glimpse()** method is used to reveal the structure of the *all_shoes* data frame (see figure 24).

```
# Merge Women Shoes and Men Shoes data frame
all_shoes = rbind(WomenShoes,MenShoes)

# Structure of the data
glimpse(all_shoes)
```

Figure 23: Merge WomenShoes and MenShoes data frame, check the structure of the merged data frame

```
> glimpse(all_shoes)
Rows: 36,268
Columns: 4
$ id     <chr> "AVpe--5gLJeJML43zzQk", "AVpe--8X1cnluZ0-bU1d", "AVpe--Qb1cnluZ0-bUkZ",~
$ brand  <chr> "ELITES BY WALKING CRADLES", "KLOGS", "NIKE", "PEOPLE FOOTWEAR", "PEOPL~
$ price  <dbl> 45, 120, 106, 22, 60, 120, 35, 20, 29, 10, 21, 18, 38, 69, 49, 200, 205~
$ gender <chr> "Women", "Women", "Women", "Women", "Women", "Women", "Women", "Women",~
```

Figure 24: Output of glimpse(all_shoes)

The median price of each brand is computed and grouped by gender with the **summarise()** method and **median()** method (see figure 25 & 26).

```
all_median_price = all_shoes %>%
                   group_by(brand, gender) %>%
                   summarise(price = median(price, rm.na=true))
```

Figure 25: Create a variable all_median_price to store the median price of each brand grouped by gender

	brand	gender	price
1	""GANESHA HANDICRAFT ""	Men	47.0
2	"HANDMADE""	Men	45.0
3	: MEDLINE	Women	20.0
4	1 WORLD SARONGS	Women	16.0
5	1031	Men	45.0
6	12 STEP GOLD	Men	50.0
7	14K CO.	Men	177.0
8	180S	Men	26.0
9	180S	Women	28.0
10	1883 BY WOLVERINE	Men	150.0

Figure 26: Create a variable all_median_price to store the median price of each brand grouped by gender

Analysis

D. Figures and Tables

To get a better understanding of the price distribution for both men's and women's shoe, a box plot of gender vs price was generated with **qplot()**. Data from *all_median_price* data frame is used to generate this box plot (see figure 27).

```
qplot(gender, price, data = all_median_price,
      geom = "boxplot", fill = gender ) +
      labs(title="Box Plot of Gender vs Price")+
      labs(y="Price", x="Gender")
```

Figure 27: Code used to generate Box Plot of Gender Against Price

Figure 28 shows the generated output; however, no box plot was generated because a boxplot must include the central 50% of the values in the interquartile range. In this case, data from all_median_price is highly skewed to left, hence, there is no finite width of the interquatile range. However, we were able to observe the outliers that exists in the dataset, this implies that further exploration and data cleaning is necessary.

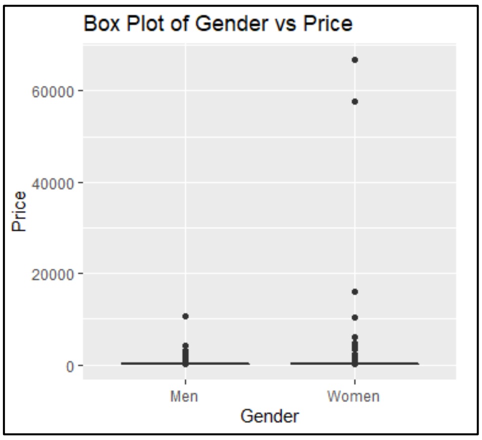

Figure 28: Box Plot of Gender Against Price

Before treating the outliers, further inspection of the outliers is needed. Therefore, a frequency plot was created for visualization of the Top 20 Most Expensive Brands for both *WomenShoes* and *MenShoes* data set. Each brand is sorted by their median price of shoes in descending order (see figure 29 & 31).

```
women_top20 = WomenShoes %>%
          group_by(brand) %>%
          summarise(price = median(price, rm.na=true)) %>%
          arrange(desc(price)) %>%
          top_n(20) %>%
          ggplot(mapping = aes(x=reorder(brand, price), y=price)) +
          geom_bar(stat = "identity", aes(fill=price)) +
          theme_light() +
          scale_colour_gradient() +
          coord_flip() +
          labs(title="Top 20 Expensive brands (Women)",
               x="Brand", y="Median Price (USD)")
```

Figure 29: Code used to generate frequency table of ***Top 20 Most Expensive Brands for Women***

As shown in figure 30, Peacock Diamonds and Peacock Jewels which are ranked 1st and 2nd in the plot, and their median price are US$66966 and US$57576 respectively. These two values were the outliers to this data set. Upon research, the reason why their price value is relatively high in comparison with the other brands is because these two brands are jewellery brands, and not shoes brands. Furthermore, brands such as Diamond Wish and Amoro are also not shoe brands, instead, they are also jewellery brands. Therefore, further data cleaning must be undergone.

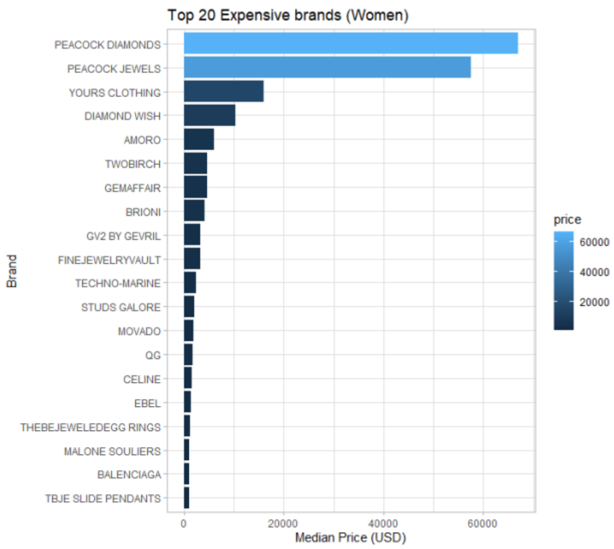

Figure 30: Output from code in figure 27

```
# Top 20 most expensive brands (Men)
men_top20 = MenShoes %>%
          group_by(brand) %>%
          summarise(price = median(price, rm.na=true)) %>%
          arrange(desc(price)) %>%
          top_n(20) %>%
          ggplot(mapping = aes(x=reorder(brand, price), y=price)) +
          geom_bar(stat = "identity", aes(fill=price)) +
          theme_light() +
          scale_colour_gradient() +
          coord_flip() +
          labs(title="Top 20 Expensive brands (Men)",
               x="Brand", y="Median Price (USD)")
men_top20
```

Figure 31: Code used to generate frequency table of ***Top 20 Most Expensive Brands for Men***

Figure 32 displays the frequency table of the Top 20 Most Expensive Brands under the dataset *MenShoes*.

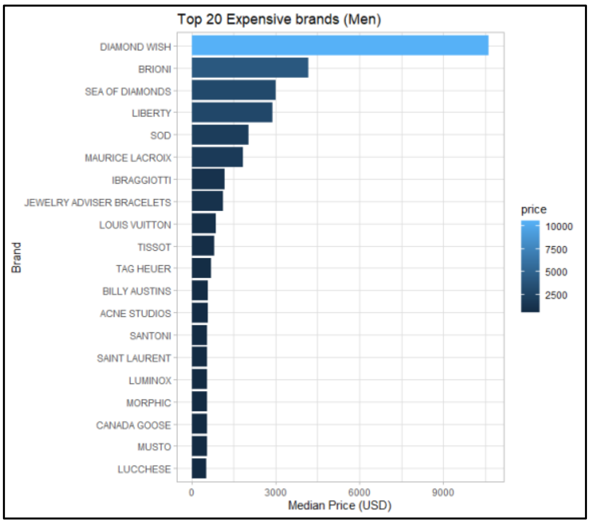

Figure 32: Output from code in figure 31

A Kernel density plot is generated to better visualize the distribution of median price for men's and women's shoes. Given that the outliers values were known, subsets of data with prices below 10000 from *all_median_price* data set were created with **subset()** method. **ggplot()** along with **geom_density()** is used to generate the graph. The **color** and **fill** parameter in **ggplot()** is passed with variable *gender* to generate two density curve based on *gender*. **geom_density()** with the **alpha** parameter is used to produce a smooth distribution curve that has colour fill with transparency (see figure 34).

```
all_median_price %>%
  subset(price < 10000) %>%
  ggplot(aes(x=price, fill=gender, colour=gender)) + geom_density(alpha=.3) +
  labs(title="Density Plot of Median Price (0-10000 USD)")+
  labs(y="Price", x="Density")
```

Figure 34: Code used to generate Density Plot of Median Price

Figure 36 shows the kernel density plot of median price below $ USD 1000. The price distribution curve for men's shoe is in green colour whereas the price distribution curve for women's shoe is in red. It is observed that although the extreme outliers have been filtered, the data is still highly skewed to the left; therefore, no meaningful insights can be observed from the generated plot.

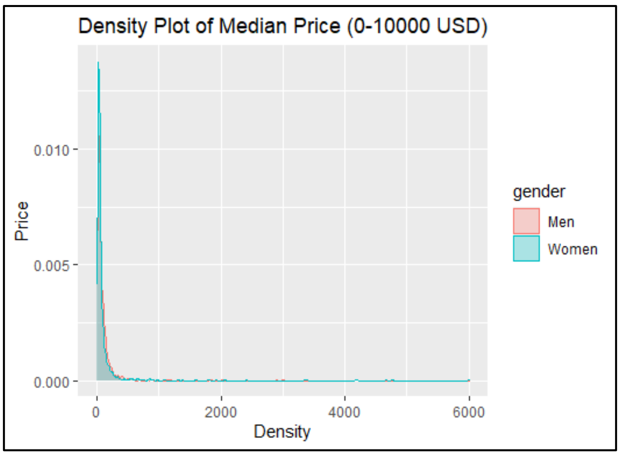

Figure 36: Density Plot of Median Price (0-10000 USD)

To resolve skewness, log transformation is necessary. Log transformation is accomplished by applying the log() function to variable *price* to make highly skewed distribution less skewed. The **+1** from **Log (price + 1)** is added because the data contains zeros and the limitation of using log transformations on data that contains zeros can be solved by doing so. After that, a kernel density plot is created like before (see figure 35).

```
all_median_price %>%
  ggplot(aes(x=log(price+1), fill=gender, colour=gender)) +
  geom_density(alpha=.3) +
  labs(title="Density Plot of Median Price (after log transformation)")+
  labs(y="Price", x="Density")
```

Figure 35: Create Density Plot of Median Price (after log transformation)

In figure 37, the distribution is more normal, which improves the usefulness of our generated plot. Women's shoes median price distribution seems to peak higher than men's shoes median price distribution when density is around 3.5. Men's shoes median price is higher when density is ranged from 4.5 to 6.5. Women's price is relatively higher when the density is in range 6.5 and above. As such, this suggest that there are more women's shoes than men's shoes when the median price of shoes is at the low to low-mid range and high-mid to high range. On the other hand, it is also suggested that there are more men shoes than women shoes when the median price of shoes is at the low-mid range to high-mid range. We can conclude that women have more cheap and expensive shoes as compared to men whereas men has more medium priced shoes.

Figure 37: Create Density Plot of Median Price (after log transformation)

After understanding the distribution of price between men's and women's shoes, we were interested to compare the median price and the product listing percentage of shoes for popular brands that is within this dataset by gender. A new data frame must be created for better comparison. To start out, the frequency count for each brand by gender is computed and assigned to *Men_brandfreq* and *Women_brandfreq* with the **count()** method. Next, the median price is computed and grouped by brand for both genders with the **group_by()** method followed by the **summarise()** method along with the **median(),** results are assigned to *men_median_price* and *women_median_price* respectively. New data frames named and *men_by_brand* and *women_by_brand* is created by combining the the median price and frequency count for each brand with **cbind()** method. Then, both *Women_by_brand* and *Men_by_brand* data frame were merged with the **merge()** method by brand and assigned to *brand_group*. Some columns in *brand_group* is renamed with the **rename()** method for better readability. After the combination of *Women_by_brand* and *Men_by_brand* the percent of product listing for men and women can be computed. A variable named *total_count* is created with the assignment of the addition of men_brand_count and women_brand_count. The *total_count* variable is used to find the most popular brands by sorting the values in descending order. A variable named *median_price_diff* is also created by subtracting *men_median_price* and *women_median_price* and passing the results into **abs()** method to ensure that there is no negative values. Lastly, the newly generated columns, *men_percent*, *women_percent*, *total_count* and *median_price_diff* are combined with the *brand_group* data frame with the **cbind()** method. M*en_percent and women_percent* are rounded to the nearest integer with the **round()** method (see appendix figure 38 & 39).

A subset of brand_group named *top20_brand* is created by using **subset()** method to filter out brand that is “UNBRANDED”, sorted by *total_count* in descending order with the **arrange()** method and selected the first 20 observations with **head()** method. A variable named *rank* is created in *top20_brand* by passing the *top20_brand* into **nrow()** and **seq.int().** The columns in *top20_brand* is then reordered for better readability (see appendix figure 40 & 41).

To create a visually appealing table that represents the gender breakdown of popular brands by listing the percentage of product listings for men’s and women’s shoes, flextable package is used. Data from *top20_brand* is first passed into **flextable()** function as an argument with **col_keys()** to select variable *rank*, *brand*, *total_count*, *men_percent* and *women_percent*. Then, the header labels are renamed with **set_header_labels()** function for better readability. Conditional formatting is also applied to column *men_percent* and *women_percent* with the **bg()** function, background colour of column *men_percent* will turn red provided that the values of *men_percent* is greater than *women_percent* whereas the background colour of column *women _percent* will turn green provided that the values of *women _percent* is greater than *men_percent.* As such, the gender that has relatively higher product listing for each brand can easily be differentiated. Titles were also added to the header with the **add_header_lines()** function. Lastly, the table alignment is adjusted with the **align()** function, borders were added to the outer box of the table with **border_outer()** function, text in header and some columns were bolded with **bold()** function, font and font size were changed with the **font()** and **fontsize()** function for better visualization (see appendix figure 42).

From figure 43, we can observe that 14 out of 20 most popular brands has more products listed for men whereas only 6 out of 20 brands has more women products. This suggests that among the 20 most popular brands that sells both men’s and women’s shoes brands, especially sneaker brands are leaning slightly towards men.

Gender Breakdown of Popular Brands				
Percent of Product Listings for Men and Women for Popular Brand				
Rank	Brand Names	Total Count	Men (%)	Women (%)
1	NIKE	2,081	83	17
2	PUMA	766	87	13
3	VANS	616	63	37
4	NEW BALANCE	527	70	30
5	TOMS	434	25	75
6	REEBOK	378	72	28
7	UNIQUE BARGAINS	302	46	54
8	MUK LUKS	292	25	75
9	RALPH LAUREN	285	33	67
10	ADIDAS	279	91	9
11	SKECHERS	250	66	34
12	JORDAN	198	99	1
13	CROCS	195	62	38
14	CONVERSE	186	79	21
15	MICHAEL KORS	183	3	97
16	DICKIES	180	79	21
17	ASICS	160	76	24
18	PLEASERUSA	150	9	91
19	PROPET	150	51	49
20	KINCO	146	75	25

Figure 43: Percent of product listings for men and women for top 20 popular brands table

Another table is also generated to visualize the difference between men's and women's shoes median prices. Similarly, this table is also created with **flextable()** with data from *top20_brand* data frame. **col_keys()** is used to select variable *rank, brand, men_median_price, women_median_price* and *median_price_diff*. Then, the header labels are renamed with **set_header_labels()** function for better readability. A second header is added above the first header with the **add_header_row()** function to display "Median Price (USD)" above columns *men_median_price*, *women_median_price* and *median_price_diff*. Conditional formatting is also applied to column *men_median_price* and *women_median_price* with the **bg()** function, background colour of column *men_median_price* will turn red provided that the values of *men_median_price* is greater than *women_median_price* whereas the background colour of column *women_median_price* will turn green provided that the values of *women_median_price* is greater than *men_median_price*. As such, the gender that has relatively higher median price for each brand can easily be differentiated. Titles were also added to the header with the **add_header_lines()** function. Lastly, the table alignment is adjusted with the **align()** function, borders were added to the outer box of the table with **border_outer()** function, text in header and some columns were bolded with **bold()** function, font and font size were changed with the **font()** and **fontsize()** function for better visualization (see appendix figure 44).

From figure 45, it is observed that the median price for men's shoes is higher than women's shoes in 12 out of 20 brands while the median price for women's shoes is higher than men's shoes in 8 out of 20 brands. The median price for men's and women's shoes for brand PUMA is the same. Although the results from figure 45 suggests that men's shoes have the tendency of having a higher price, the data may be biased since 14 out of 20 most popular brands has more products listed for men according to the output data from figure 43.

Difference between Men's and Women's Shoes Prices				
Difference in Median Prices by Gender for the Most Popular Brands				
		Median Price (USD)		
Rank	Brand Names	Men	Women	Difference
1	NIKE	78.5	80.0	1.5
2	PUMA	70.0	70.0	0.0
3	VANS	48.0	44.0	4.0
4	NEW BALANCE	65.0	60.0	5.0
5	TOMS	48.0	49.0	1.0
6	REEBOK	56.0	54.0	2.0
7	UNIQUE BARGAINS	11.5	30.0	18.5
8	MUK LUKS	27.0	38.0	11.0
9	RALPH LAUREN	111.0	106.0	5.0
10	ADIDAS	60.0	44.0	16.0
11	SKECHERS	69.0	58.0	11.0
12	JORDAN	107.0	36.5	70.5
13	CROCS	34.0	33.0	1.0
14	CONVERSE	50.0	52.0	2.0
15	MICHAEL KORS	174.0	76.5	97.5
16	DICKIES	35.0	33.0	2.0
17	ASICS	78.5	84.5	6.0
18	PLEASERUSA	50.0	46.0	4.0
19	PROPET	79.0	60.0	19.0
20	KINCO	128.0	130.5	2.5

Figure 45: Difference in Median Prices by Gender for the top 20 popular brands table

Results

According to our findings, the product listing of brands, especially sneaker brands are leaning towards men among the 20 most popular brands that sells both men's and women's shoes brands. In other words, popular brands are more likely to sell men's shoes. Furthermore, our analysis suggests that men's shoes have the tendency of having a higher price as compared to women's shoes among popular brands that sells both men's and women's shoes.

Conclusion

In conclusion, from the analysis for the given datasets, we were able to know the price distribution of shoe price by gender, identify the percentage of product listing by men and women and find the price difference between men's and women's shoes among the most popular brands. This analysis can provide useful information that helps shoe companies or manufacturers to gain insights of the market trends to help them make better business decisions. However, there exists limitations in our research because the data set contains many observations that are not shoes data, this brought challenges to our analysis as it was hard to filter out brands that do not belong to the shoe category without researching the brand of that particular product.

References

[1] Datafiniti.co, 'How Shoe Brands Change Prices Depending on Gender', 2017. [Online]. Available: https://datafiniti.co/shoe-brands-change-prices-depending-gender/. [Accessed: 10-Oct-2021].

[2] Developer.Datafiniti.co, 'Product Data Schema', n.d. [Online]. Available: https://developer.datafiniti.co/docs/product-data-schema. [Accessed: 3-Oct-2021].

Appendix

```r
# Compute the frequency count for each brand by gender
Men_brandfreq = MenShoes %>% count(brand)
Women_brandfreq = WomenShoes %>% count(brand)

# Compute Median price grouped by Brand for WomenSheos and MenShoes
men_median_price = MenShoes %>%
                    group_by(brand) %>%
                    summarise(price = median(price, na.rm =TRUE))

women_median_price = WomenShoes %>%
                      group_by(brand) %>%
                      summarise(price = median(price, na.rm =TRUE))

# Create new data frame by combining the median price and frequency count
Men_by_brand = cbind(men_median_price, count=Men_brandfreq$n)
Women_by_brand = cbind(women_median_price, count=Women_brandfreq$n)


# Merge both Women_by_brand and Men_by_brand data frame by brand
brand_group = merge(Men_by_brand,Women_by_brand,by="brand",all=TRUE)%>%
              rename(men_median_price = price.x,
                     men_brand_count = count.x,
                     women_median_price = price.y,
                     women_brand_count = count.y)

# Find the percentage of shoes for each brand by gender
men_percent = brand_group$men_brand_count/
              (brand_group$men_brand_count+brand_group$women_brand_count)*100

women_percent = brand_group$women_brand_count/
              (brand_group$men_brand_count+brand_group$women_brand_count)*100

# Find the total count of shoes for each brand regardless of gender
total_count = brand_group$men_brand_count + brand_group$women_brand_count

# Find the price difference between men and women shoe median price
median_price_diff = abs(brand_group$men_median_price -
                          brand_group$women_median_price)

# Combine men_percent, women_percent and total_count with brand_group
brand_group = brand_group %>% cbind(men_percent = round(men_percent),
                                     women_percent = round(women_percent),
                                     total_count,median_price_diff)
```

Figure 38: Create new data frame named brand_group

	brand	men_median_price	men_brand_count	women_median_price	women_brand_count	men_percent	women_percent	total_count	median_price_diff
2075	NIKE	78.5	1726	80.0	355	83	17	2081	1.5
3034	UNBRANDED	44.0	409	70.0	715	36	64	1124	26.0
2347	PUMA	70.0	666	70.0	100	87	13	766	0.0
3074	VANS	48.0	387	44.0	229	63	37	616	4.0
2058	NEW BALANCE	65.0	367	60.0	160	70	30	527	5.0
2952	TOMS	48.0	110	49.0	324	25	75	434	1.0
2425	REEBOK	56.0	272	54.0	106	72	28	378	2.0
3044	UNIQUE BARGAINS	11.5	140	30.0	162	46	54	302	18.5
2005	MUK LUKS	27.0	73	38.0	219	25	75	292	11.0
2375	RALPH LAUREN	111.0	94	106.0	191	33	67	285	5.0
59	ADIDAS	60.0	255	44.0	24	91	9	279	16.0
2825	SUPERIOR GLOVE WORKS	101.0	186	104.0	67	74	26	253	3.0
2661	SKECHERS	69.0	165	58.0	85	66	34	250	11.0
1121	FUSE LENSES	38.0	160	38.0	74	68	32	234	0.0
1512	JORDAN	107.0	196	36.5	2	99	1	198	70.5
695	CROCS	34.0	120	33.0	75	62	38	195	1.0
652	CONVERSE	50.0	147	52.0	39	79	21	186	2.0
1906	MICHAEL KORS	174.0	5	76.5	178	3	97	183	97.5
814	DICKIES	35.0	143	33.0	37	79	21	180	2.0
205	ASICS	78.5	122	84.5	38	76	24	160	6.0
322	BERNE APPAREL	72.0	126	67.0	24	84	16	150	5.0

Figure 39: Snippet of output from code in figure 38 (brand_group data frame)

```
# Top 20 brand from brand_group based on brand total count
top20_brand = brand_group %>%
  subset(brand != "UNBRANDED" & brand != "SUPERIOR GLOVE WORKS"
         & brand != "BERNE APPAREL" & brand != "FUSE LENSES") %>%
  arrange(desc(total_count)) %>%
  head(20)

# Crate rank variable
top20_brand$rank = seq.int(nrow(top20_brand))

# reorder column in top20_brand
col_order =  c( "rank", "brand", "total_count", "men_brand_count",
                "women_brand_count","men_percent", "women_percent",
                "men_median_price","women_median_price",
                "median_price_diff")
top20_brand  =  top20_brand[, col_order]
```

Figure 40: Create new dataframe named top20_brand

	rank	brand	total_count	men_brand_count	women_brand_count	men_percent	women_percent	men_median_price	women_median_price	median_price_diff
1	1	NIKE	2081	1726	355	83	17	78.5	80.0	1.5
2	2	PUMA	766	666	100	87	13	70.0	70.0	0.0
3	3	VANS	616	387	229	63	37	48.0	44.0	4.0
4	4	NEW BALANCE	527	367	160	70	30	65.0	60.0	5.0
5	5	TOMS	434	110	324	25	75	48.0	49.0	1.0
6	6	REEBOK	378	272	106	72	28	56.0	54.0	2.0
7	7	UNIQUE BARGAINS	302	140	162	46	54	11.5	30.0	18.5
8	8	MUK LUKS	292	73	219	25	75	27.0	38.0	11.0
9	9	RALPH LAUREN	285	94	191	33	67	111.0	106.0	5.0
10	10	ADIDAS	279	255	24	91	9	60.0	44.0	16.0
11	11	SUPERIOR GLOVE WORKS	253	186	67	74	26	101.0	104.0	3.0
12	12	SKECHERS	250	165	85	66	34	69.0	58.0	11.0
13	13	FUSE LENSES	234	160	74	68	32	38.0	38.0	0.0
14	14	JORDAN	198	196	2	99	1	107.0	36.5	70.5
15	15	CROCS	195	120	75	62	38	34.0	33.0	1.0
16	16	CONVERSE	186	147	39	79	21	50.0	52.0	2.0
17	17	MICHAEL KORS	183	5	178	3	97	174.0	76.5	97.5
18	18	DICKIES	180	143	37	79	21	35.0	33.0	2.0
19	19	ASICS	160	122	38	76	24	78.5	84.5	6.0
20	20	BERNE APPAREL	150	126	24	84	16	72.0	67.0	5.0

Figure 41: Output from code in figure 40 (top20_brand dataframe)

```
# Create table of product listing percentage with flextable
flextable( data = top20_brand, col_keys = c("rank", "brand", "total_count",
                                            "men_percent","women_percent")) %>%

  # rename header labels
  set_header_labels(rank = "Rank",brand = "Brand Names",total_count = "Total Count",
                    men_percent = "Men (%)", women_percent = "Women (%)") %>%

  # apply conditional formatting to column men_percent and women_percent
  bg(~ men_percent > women_percent, bg = "#FC7676", ~ men_percent) %>%
  bg(~ men_percent < women_percent, bg = "#71CA97", ~ women_percent) %>%

  # add titles in header
  add_header_lines(values = "Percent of Product Listings for Men and Women
                   for Popular Brand") %>%
  add_header_lines(values = "Gender Breakdown of Popular Brands") %>%

  # adjust alignment, add borders, bold columns and change fonts and fontsize
  autofit() %>%
  align(align = "center", part = "header") %>%
  align_nottext_col(align = "center") %>%
  border_outer(part="all") %>%
  bold(j = c("brand","men_percent", "women_percent"), bold = TRUE) %>%
  bold(bold = TRUE, part = "header") %>%
  font(fontname = "Courier", part = "all") %>%
  fontsize(i = 1, size = 12, part = "header") %>%
  fontsize(i = 2, size = 8, part = "header")
```

Figure 42: Create table of product listing percentage by gender with flextable

```
# Create table of median price comparison with Flextable
flextable( data = top20_brand, col_keys = c("rank", "brand", "men_median_price",
                                            "women_median_price",
                                            "median_price_diff")) %>%

  # rename header labels
  set_header_labels(rank = "Rank", brand = "Brand Names",
                    men_median_price = "Men",
                    women_median_price = "Women",
                    median_price_diff = "Difference") %>%

  # add 2nd header that displays "Median Price (USD)"
  add_header_row(values = c(" "," ","Median Price (USD)"), colwidths = c(1,1,3)) %>%

  bg(~ men_median_price > women_median_price, bg = "#FC7676", ~ men_median_price) %>%
  bg(~ men_median_price < women_median_price, bg = "#71CA97", ~ women_median_price) %>%

  # add titles in header
  add_header_lines(values = "Difference in Median Prices by Gender for the Most
                             Popular Brands") %>%
  add_header_lines(values = "Difference between Men's and Women's Shoes Prices") %>%

  # adjust alignment, add borders, bold columns and change fonts and fontsize
  autofit() %>%
  align(align = "center", part = "header") %>%
  align_nottext_col(align = "center") %>%
  border_outer() %>%
  bold(bold = TRUE, part = "all") %>%
  font(fontname = "Courier", part = "all") %>%
  fontsize(i = 1, size = 12, part = "header") %>%
  fontsize(i = 2, size = 8, part = "header")
```

Figure 44: Create table of product listing percentage by gender with flextable