Executive Summary

This paper offers recommendations for optimizing website traffic and income production through an analysis of clickstream data from an online retail store. To complete the assigned tasks, the dataset "e-shop clothing 20081.xlsx","shop clothing infor 2008.txt","input.xlsx" was examined using the proper data analytics tools. The report includes descriptive analysis of another dataset using R, sales and revenue trend visualization, and data cleaning. Moreover, insights obtained with R's dplyr on the "starwars" dataset are offered. The findings aim to guide decision-making for enhancing the online retail shop's performance.

Introduction

This report's goal is to use data analytics methods to extract useful information from an online retailer's clickstream dataset. In order to support strategic decisions targeted at boosting website traffic and income generation, this report will look at sales patterns, revenue trends, and descriptive analysis.

Data Cleaning and Preparation

Cleaning and prepping the dataset for analysis was the first stage. This includes verifying data integrity, fixing data formats, and addressing missing values. The identification of pertinent columns for the analysis was predicated on their importance in comprehending sales, revenue, and user activity on the website.

Task A: Extracting Relevant Columns and Data Preparation

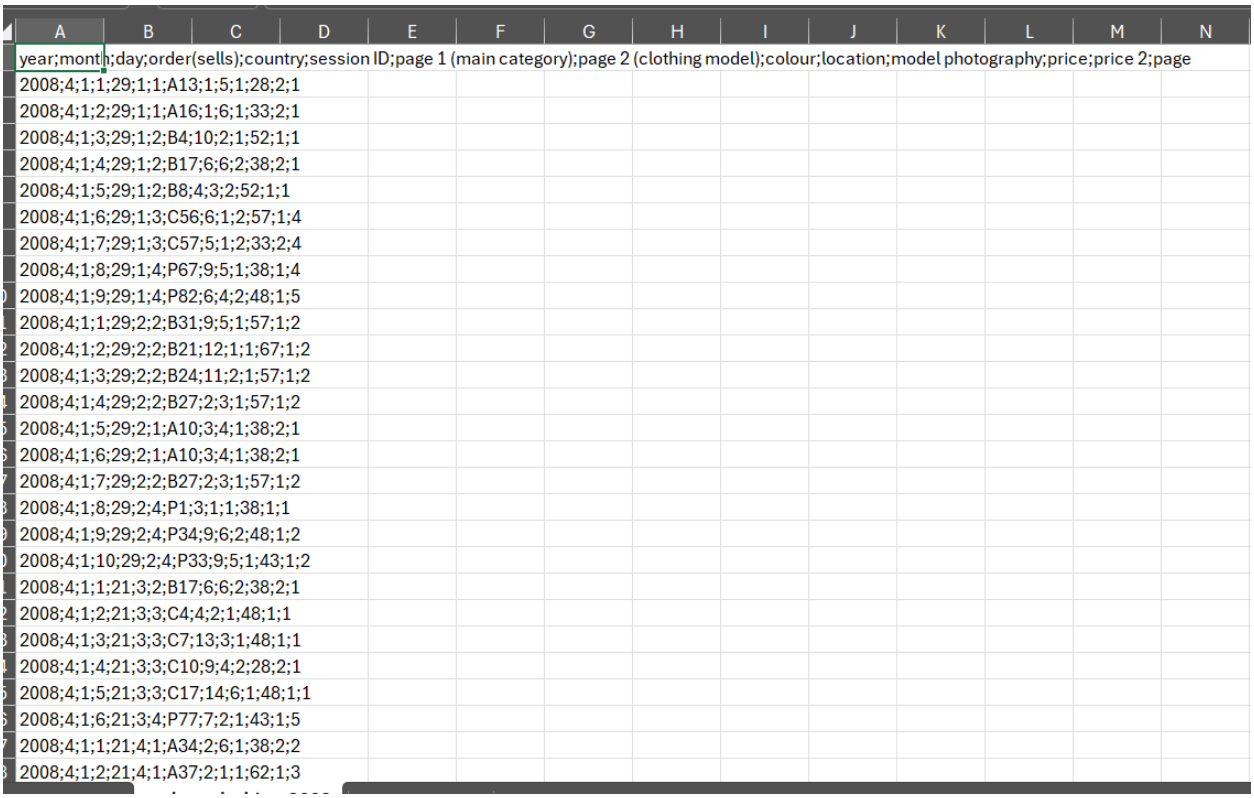
The columns extracted for analysis were chosen based on their relevance to understanding sales and revenue metrics. Key columns included product year, month, day, order, country, page 1 (main category), page 2 (clothing model), colour, location, model photography, price in us dollars, price 2 -> variable informing whether the price of a particular product is higher than

the average price for the entire product category, page -> page number within the e-store website (from 1 to 5) These columns provide insights into customer purchasing behavior and revenue generation trends.

This is the raw of the data, we can see that is the challenging for us to understand clearly which information is provided in the excel file, so initial step is make it look manageable. By using text to column in Data tab, don’t forget select the column contain all the data in this case all data have written on column A.

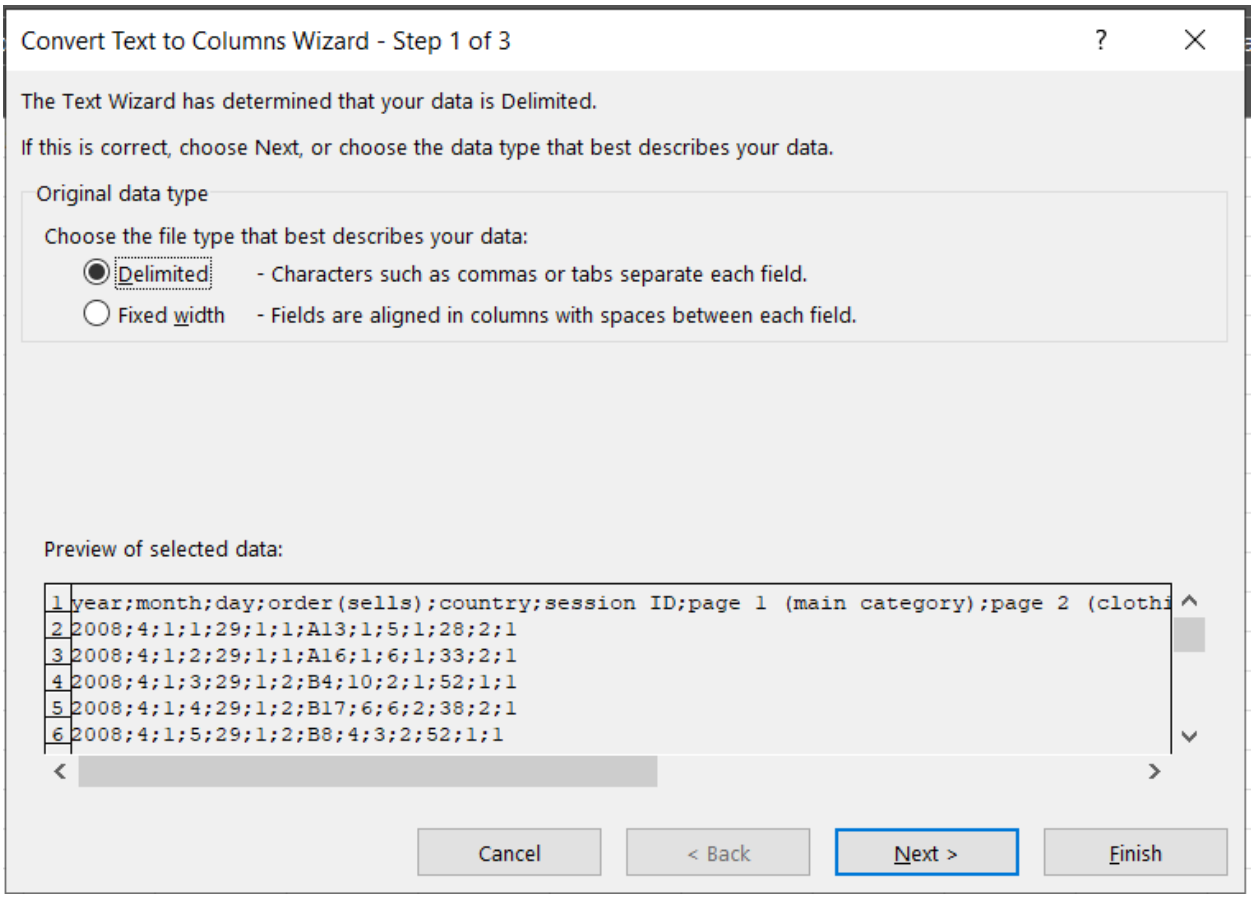
Do the same with the the ‘**input.xlsx’** excel file but the **input.xlsx**

Preparing For Tasks:

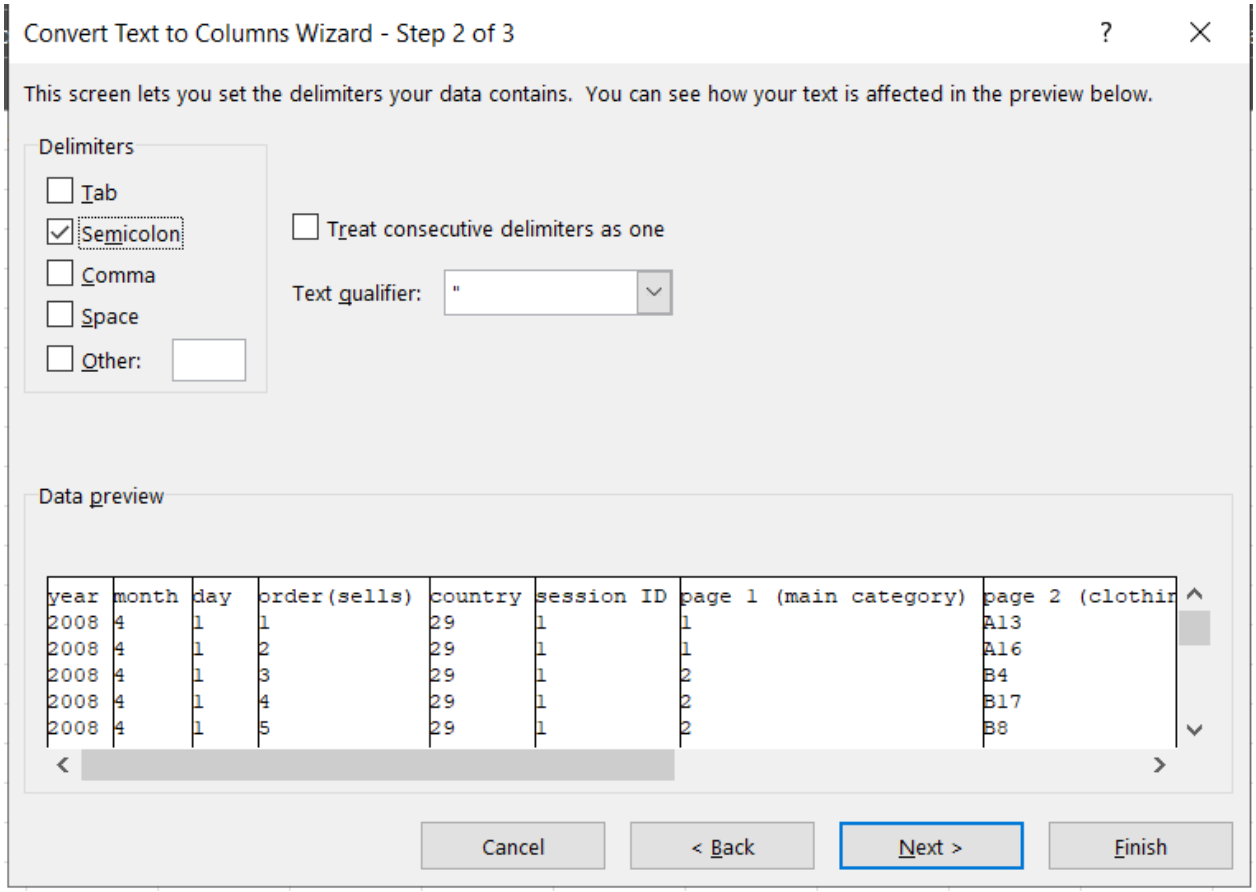


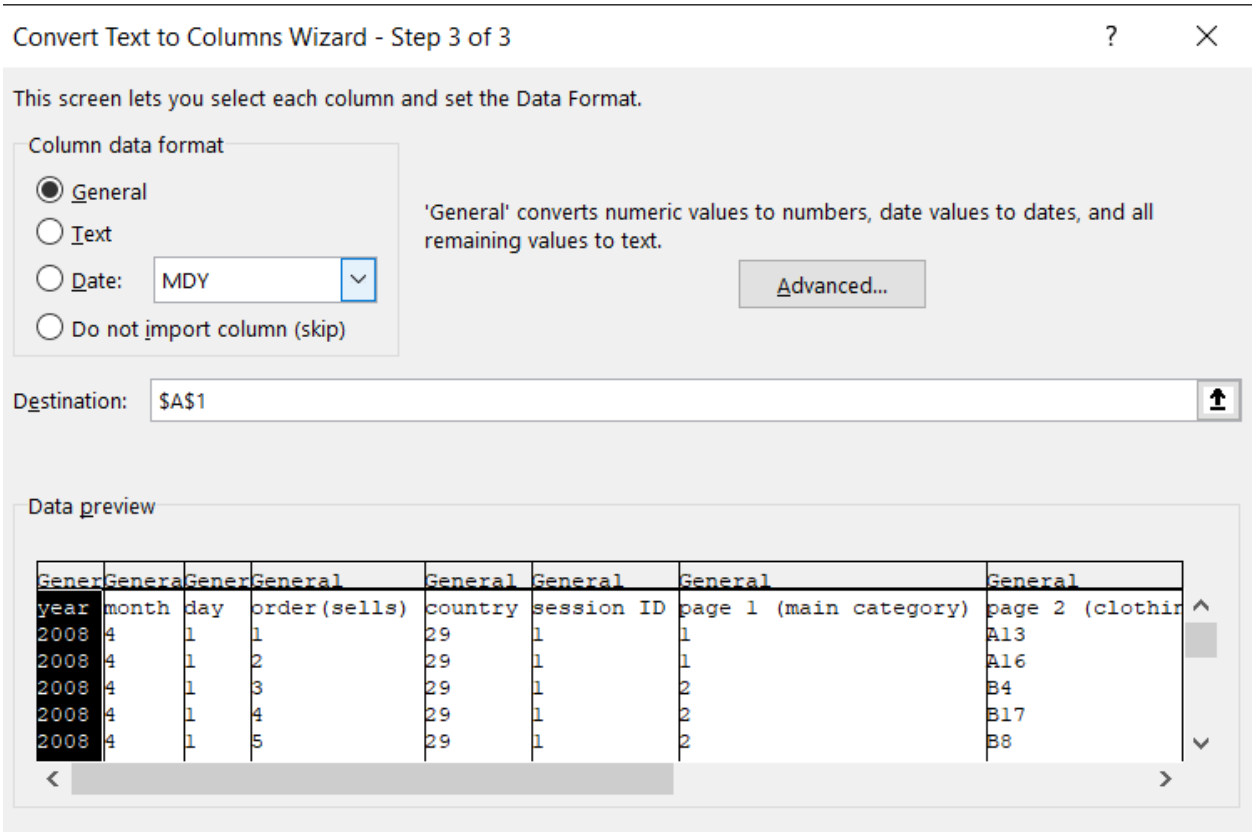
A	B	C	D	E	F	G	H	I	J	K	L	M	N
year;month;day;order(sells);country;session ID;page 1 (main category);page 2 (clothing model);colour;location;model photography;price;price 2;page													
2008;4;1;1;29;1;1;A13;1;5;1;28;2;1													
2008;4;1;2;29;1;1;A16;1;6;1;33;2;1													
2008;4;1;3;29;1;2;B4;10;2;1;52;1;1													
2008;4;1;4;29;1;2;B17;6;6;2;38;2;1													
2008;4;1;5;29;1;2;B8;4;3;2;52;1;1													
2008;4;1;6;29;1;3;C56;6;1;2;57;1;4													
2008;4;1;7;29;1;3;C57;5;1;2;33;2;4													
2008;4;1;8;29;1;4;P67;9;5;1;38;1;4													
2008;4;1;9;29;1;4;P82;6;4;2;48;1;5													
2008;4;1;1;29;2;2;B31;9;5;1;57;1;2													
2008;4;1;2;29;2;2;B21;12;1;1;67;1;2													
2008;4;1;3;29;2;2;B24;11;2;1;57;1;2													
2008;4;1;4;29;2;2;B27;2;3;1;57;1;2													
2008;4;1;5;29;2;1;A10;3;4;1;38;2;1													
2008;4;1;6;29;2;1;A10;3;4;1;38;2;1													
2008;4;1;7;29;2;2;B27;2;3;1;57;1;2													
2008;4;1;8;29;2;4;P1;3;1;1;38;1;1													
2008;4;1;9;29;2;4;P34;9;6;2;48;1;2													
2008;4;1;10;29;2;4;P33;9;5;1;43;1;2													
2008;4;1;1;21;3;2;B17;6;6;2;38;2;1													
2008;4;1;2;21;3;3;C4;4;2;1;48;1;1													
2008;4;1;3;21;3;3;C7;13;3;1;48;1;1													
2008;4;1;4;21;3;3;C10;9;4;2;28;2;1													
2008;4;1;5;21;3;3;C17;14;6;1;48;1;1													
2008;4;1;6;21;3;4;P77;7;2;1;43;1;5													
2008;4;1;1;21;4;1;A34;2;6;1;38;2;2													
2008;4;1;2;21;4;1;A37;2;1;1;62;1;3													

Choosing delimited



Select semicolon





Click Finish and let excel handle all for you:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
year	month	day	order(sells	country	session ID	page 1 (ma	page 2 (clo	colour	location	model pho	price	price 2	page
2008	4	1	1	29	1	1	A13	1	5	1	28	2	1
2008	4	1	2	29	1	1	A16	1	6	1	33	2	1
2008	4	1	3	29	1	2	B4	10	2	1	52	1	1
2008	4	1	4	29	1	2	B17	6	6	2	38	2	1
2008	4	1	5	29	1	2	B8	4	3	2	52	1	1
2008	4	1	6	29	1	3	C56	6	1	2	57	1	4
2008	4	1	7	29	1	3	C57	5	1	2	33	2	4
2008	4	1	8	29	1	4	P67	9	5	1	38	1	4
2008	4	1	9	29	1	4	P82	6	4	2	48	1	5
2008	4	1	1	29	2	2	B31	9	5	1	57	1	2
2008	4	1	2	29	2	2	B21	12	1	1	67	1	2
2008	4	1	3	29	2	2	B24	11	2	1	57	1	2
2008	4	1	4	29	2	2	B27	2	3	1	57	1	2
2008	4	1	5	29	2	1	A10	3	4	1	38	2	1
2008	4	1	6	29	2	1	A10	3	4	1	38	2	1
2008	4	1	7	29	2	2	B27	2	3	1	57	1	2
2008	4	1	8	29	2	4	P1	3	1	1	38	1	1
2008	4	1	9	29	2	4	P34	9	6	2	48	1	2
2008	4	1	10	29	2	4	P33	9	5	1	43	1	2
2008	4	1	1	21	3	2	B17	6	6	2	38	2	1
2008	4	1	2	21	3	3	C4	4	2	1	48	1	1
2008	4	1	3	21	3	3	C7	13	3	1	48	1	1
2008	4	1	4	21	3	3	C10	9	4	2	28	2	1
2008	4	1	5	21	3	3	C17	14	6	1	48	1	1
2008	4	1	6	21	3	4	P77	7	2	1	43	1	5
2008	4	1	1	21	4	1	A34	2	6	1	38	2	2
2008	4	1	2	21	4	1	A37	2	1	1	62	1	3

Input.xlsx:

After doing the same like the steps before you will get result

id	name	salary	start_date	dept
1	Rick	623.3	1/1/2012	IT
2	Dan	515.2	9/23/2013	Operations
3	Michelle	611	11/15/2014	IT
4	Ryan	729	5/11/2014	HR
5	Gary	843.25	3/27/2015	Finance
6	Nina	578	5/21/2013	IT
7	Simon	632.8	7/30/2013	Operations
8	Guru	722.5	6/17/2014	Finance

Data Reading Prepare and Chart:

```
library(readxl)
library(dplyr)
library(ggplot2)
```

Using **readxl** library help us in reading excel file, **ddlyr** help us doing the task E which provide the starwar.csv and **ggplot2** provided us creating chart function

Prepare Data for Tasks B and C:

```
# Load the data
eshop <- read_excel('e-shop clothing 2008.xlsx')
```

In **readxl** library provided us **read_excel** function that help us to read excel file, in that function you have to integrate into the function the location of the excel file, in this case the R file and the excel file is the same location so that we just insert into the name of the file. Furthermore, reusing the excel file for another tasks I stored it to eshop variable.

Prepare Data for Task D:

```
file_input <- read_excel('input.xlsx')
```

Do the same prepare data for tasks B and C steps

Make sure the data is prepared

```
#Make Sure Data Is Lo
head(eshop,4)
tail(file_input,5)
```

Using the head function and tail function in helping us knowing the data have been loaded, if the data is loaded, it led to this result

```
A tibble: 4 × 14
 year month   day `order(sells)` country `session ID` `page 1 (main category)`
<dbl> <dbl> <dbl>          <dbl>   <dbl>        <dbl>                      <dbl>
 2008     4     1              1      29            1                          1
 2008     4     1              2      29            1                          1
 2008     4     1              3      29            1                          2
 2008     4     1              4      29            1                          2
```

Task B: Sales Analysis by Month

Using appropriate charts, the sales trends by month were visualized. This analysis helps in understanding seasonal variations in sales and identifying peak months for revenue generation. The insights gained contribute to planning marketing campaigns and optimizing inventory based on demand fluctuations.

After loading the data, I will use **ggplot2** library to create the chart the task B, before showing the chart we have to determine the right column, in this case calculating the sales by month, I use year, month and total sale of the month.

```
monthly_sales <- eshop %>%
  group_by(year, month) %>%
  summarise(total_sales = sum(`order(sells)`))
```

eshop %>%:
To begin, open the eshop data table.
To transport data through a sequence of operations, use the %>% operator.

group_by(month, year):
The dplyr function group_by is used to group data according to one or more variables.
Year and month are the two columns used to group the data in this instance.
This implies that the year and month values will be used to categorize all of the rows in the eShop table.

summarise(order(sells) + total_sales):
The dplyr function summarise is used to combine previously grouped data groups into a summary table.

Order(sells) + sum(total_sales):
Determine the total value of each group's order(sells) column, or for each pair of years and months.
In the summary table, a new column named total_sales is given this total amount.

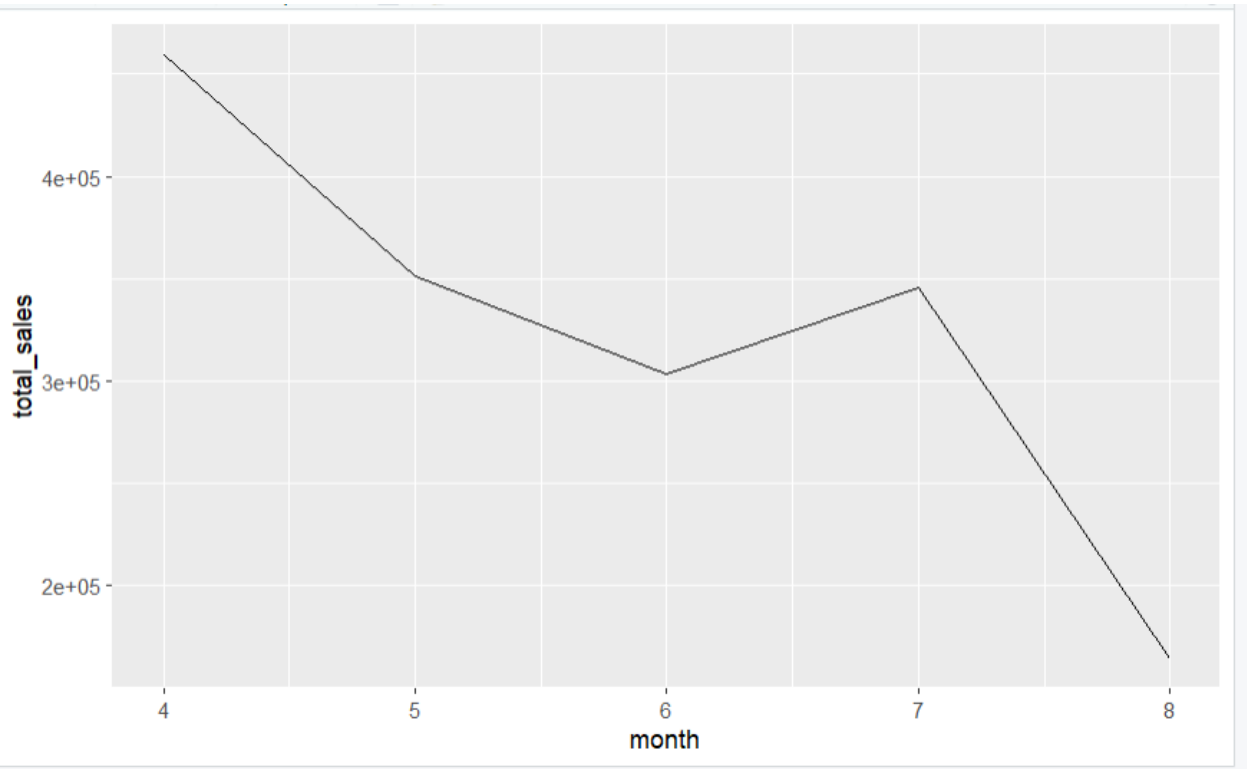
Showing the mothly_sales:

```
  year month total_sales
1 2008     4      459712
2 2008     5      350920
3 2008     6      303562
4 2008     7      345792
5 2008     8      164551
```

Showing the Chart:

```
#The Sale Chart in Line Chart
ggplot(monthly_sales, aes(x = month, y = total_sales)) + geom_line()
       + labs(title = "Monthly Sales", x = "Month", y ="Sale")
       +theme(axis.text.x =  element_text(angle = 90, hjust = 1))
```

Using **ggplot** to provide the chart, in this case I want to show the trend of the sale by month so I use the Line Chart



Task C: Revenue Analysis by Month

Revenue trends were analyzed by aggregating sales data for each month and plotting total revenue over time. This visualization highlights revenue growth patterns and identifies opportunities for increasing revenue through targeted strategies such as promotional offers during peak sales months.

Do the same Task B, in this case I determine that I have to use year, month and price column.

```
#Calculate revenue per month
revenue <- eshop %>%
  group_by(year, month) %>%
  summarise(revenue_per_month = sum(price))
```

eshop %>%:

Start with the eshop data table.

Use the %>% operator to move data through a series of operations.

group_by(year, month):

group_by is a function in dplyr used to group data by one or more variables.

In this case, the data is grouped by two columns year and month.

This means that all rows in the eshop table will be grouped by year and month values.

summarise(revenue_per_month = sum(price)):

summarise is a function in dplyr used to create a summary table from previously grouped groups of data.

revenue_per_month = sum(price):

Calculate the total value of the price column for each group (that is, for each pair of year and month).

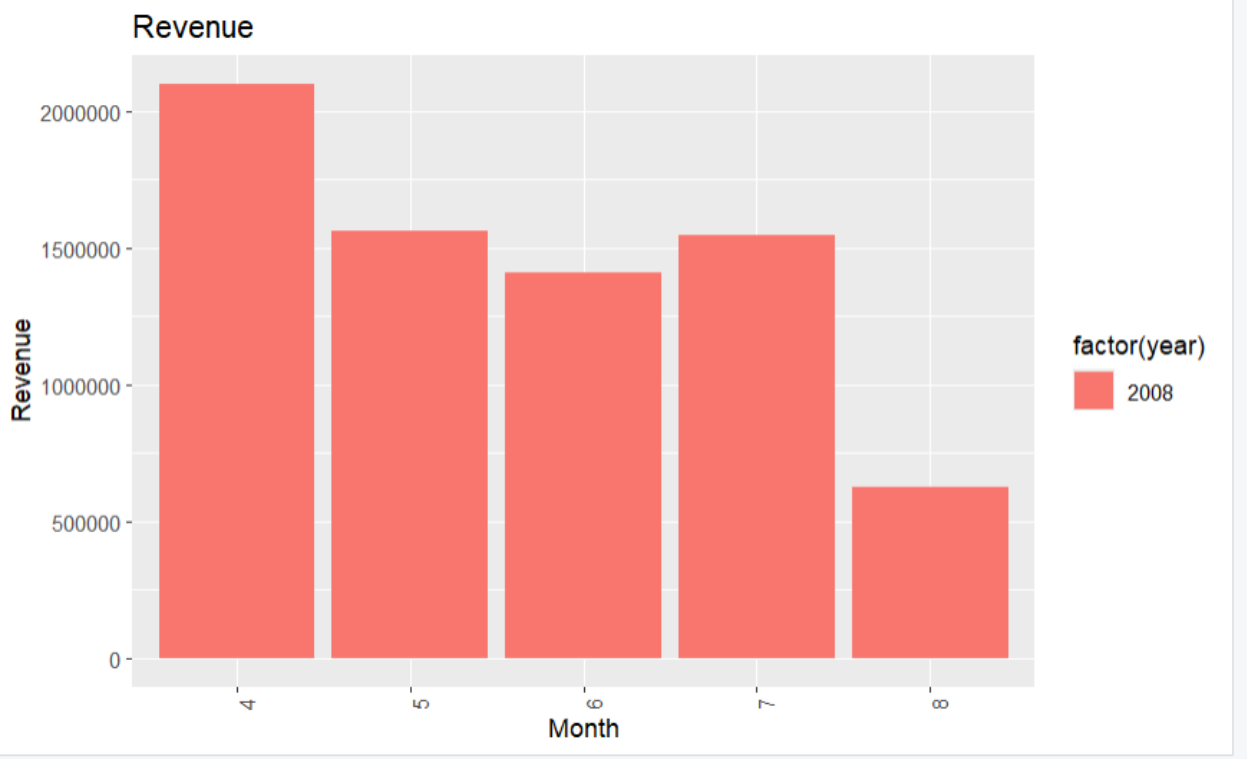
This total value is assigned to a new column called revenue_per_month in the summary table.

Showing the revenue_per_month:

year	month	revenue_per_month
<dbl>	<dbl>	<dbl>
2008	4	2100924
2008	5	1562573
2008	6	1411981
2008	7	1547518
2008	8	625180

Showing The Chart:

In this case I would like to use bar chart to show the comparation of the revenue monthly.



Task D: Descriptive Analysis Using R

A separate dataset ("input.csv") was analyzed using R to compute descriptive statistics such as mean, median, mode, standard deviation, and variance of the salary column. These statistics provide a comprehensive overview of the salary distribution within the dataset, aiding in workforce planning and salary benchmarking.

Get the salary column:

```
#Getting the salary column
salary <- file_input$salary
print(salary)
```

file_input have been loaded before and **$salary** is the column of the file_input make sure it contains the column or you will get error

```
[1] 623.30 515.20 611.00 729.00 843.25 578.00 632.80 722.50
>
```

Mean of the salary:

```
#Mean Of salary
print(mean(salary,na.rm = TRUE))
```

Result of Mean:

```
> print(mean(sa
[1] 656.8813
>
```

Median of salary:

```
#Median of salary
print(median(salary,na.rm = TRUE))
```

Result of Median:

```
] 628.05
```

Mode of Salary:

```
#Mode of Salary
Mode_function <- function(value){
  ux <- unique(value)
  ux[which.max(tabulate(match(value,ux)))]
}

print(Mode_function(salary))
```

Result of Mode:

```
] 623.3
```

Variance of Salary:

```
#Variance of Salary
print(var(salary,na.rm = TRUE))
```

Result of Variance:

```
> print(var(salary,na.rm = TRUE))
[1] 10621.25
```

Standard Deviation:

```
#Standard Deviation of salary
print(sd(salary,na.rm = TRUE))
```

Result of Standard Deviation:

```
> print(sd(salary,na.rm = TRUE))
[1] 103.0595
```

Task E: Analysis Using the Starwars Dataset

The "starwars" dataset from the dplyr package in R was utilized to perform specific analyses:

Part i: Actors whose eye color is not black and height is over 150 cm were extracted. This segmentation helps in identifying distinct groups within the dataset based on physical attributes.

Part ii: A new column for Body Mass Index (BMI) was added to the dataset using the formula BMI=mass(height100)2BMI = \frac{mass}{(\frac{height}{100})^2}BMI=(100height)2mass. A scatter plot of height against BMI was created to visualize the relationship between these

variables. This visualization aids in understanding the distribution of BMI across different heights among the actors.

i:

```
data("starwars")

# Extract actors whose eye color is not black and height is over 150 meters
filtered_data <- starwars %>%
  filter(eye_color != "black", height > 150)

# Display the result
print(filtered_data)
```

In this part, I used the **data("starwars")** to get all data inside starwars which mean starwars is already existed inside Rstudio.

Using **%>%** I can get all data in starwars then I use filter function to get the eye_color which is a column in R and height is a column, then I stored it to **filtered_data.**

Finally, I used the **print()** function to display the result. After that, you will get this result.

	name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender	homeworld	species
	<chr>	<int>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
1	Luke…	172	77	blond	fair	blue	19	male	mascu…	Tatooine	Human
2	C-3PO	167	75	NA	gold	yellow	112	none	mascu…	Tatooine	Droid
3	Dart…	202	136	none	white	yellow	41.9	male	mascu…	Tatooine	Human
4	Owen…	178	120	brown, gr…	light	blue	52	male	mascu…	Tatooine	Human
5	Beru…	165	75	brown	light	blue	47	fema…	femin…	Tatooine	Human
6	Bigg…	183	84	black	light	brown	24	male	mascu…	Tatooine	Human
7	Obi-…	182	77	auburn, w…	fair	blue-gray	57	male	mascu…	Stewjon	Human
8	Anak…	188	84	blond	fair	blue	41.9	male	mascu…	Tatooine	Human
9	Wilh…	180	NA	auburn, g…	fair	blue	64	male	mascu…	Eriadu	Human
10	Chew…	228	112	brown	unknown	blue	200	male	mascu…	Kashyyyk	Wookiee

ii:

```
# Add BMI column to the dataset
starwars_with_bmi <- starwars %>%
  mutate(BMI = mass / ((height / 100) ^ 2))

# Display the updated dataset with BMI
print(starwars_with_bmi)

# Plot height against BMI
ggplot(starwars_with_bmi, aes(x = height, y = BMI)) +
  geom_point() +
  labs(title = "Height vs. BMI in Star Wars Characters",
       x = "Height (cm)",
       y = "BMI") +
  theme_minimal()
```

To get the part **ii** done, we have to add BMI column to the dataset, applying the same methods in the part **i** but we have to calculate the BMI with this formula:

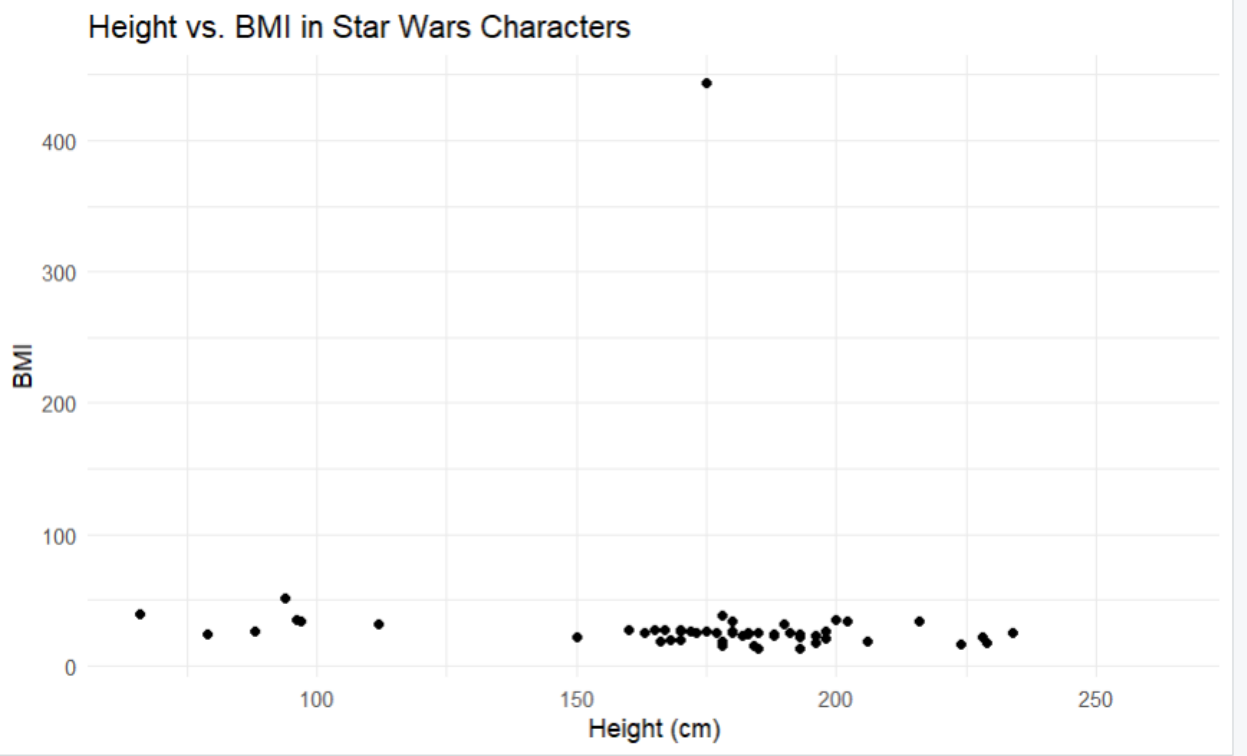
$$BMI = \frac{weight\,(kg)}{height^2\,(m^2)}$$

In this case, mass is the weight then we display the result, you will get this result, if you do it right.

name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender	homeworld	species
<chr>	<int>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Luke…	172	77	blond	fair	blue	19	male	mascu…	Tatooine	Human
C-3PO	167	75	NA	gold	yellow	112	none	mascu…	Tatooine	Droid
R2-D2	96	32	NA	white, bl…	red	33	none	mascu…	Naboo	Droid
Dart…	202	136	none	white	yellow	41.9	male	mascu…	Tatooine	Human
Leia…	150	49	brown	light	brown	19	fema…	femin…	Alderaan	Human
Owen…	178	120	brown, gr…	light	blue	52	male	mascu…	Tatooine	Human
Beru…	165	75	brown	light	blue	47	fema…	femin…	Tatooine	Human
R5-D4	97	32	NA	white, red	red	NA	none	mascu…	Tatooine	Droid
Bigg…	183	84	black	light	brown	24	male	mascu…	Tatooine	Human
Obi-…	182	77	auburn, w…	fair	blue-gray	57	male	mascu…	Stewjon	Human

i 77 more rows

After that, I use **ggplot** to create visualization the BMI that we have gotten before:



In this graph you can see the consistence base on the BMI that we have visualized

Conclusion:

The clickstream information from an online retail company was thoroughly analyzed in this research, which offers insightful advice on how to maximize website traffic and income production. Following thorough data preparation, cleaning, and visualization along with descriptive statistical analysis in R, the following main conclusions and suggestions have been established:

Data Preparation and Cleaning:

A crucial first step was guaranteeing the usability and integrity of the data. This required dealing with data types, dealing with null values, and extracting pertinent columns that were essential to comprehending revenue and sales indicators.

Trends in Sales and Revenue:

Task B: Seasonal fluctuations and peak sales months were revealed by utilizing line charts to visualize the monthly sales trends. Planning successful marketing efforts and controlling inventory to meet demand require these knowledge.
Task C: Bar charts were used to show monthly income trends and emphasize patterns of revenue growth. This study helps find ways to increase revenue by using focused tactics, including special deals during months with high sales.

Analyzing Descriptively:

Task D: Using R, descriptive statistics such as mean, median, mode, variance, and standard deviation were calculated for the wage column from a different dataset. These figures help with personnel planning and salary benchmarking by giving a thorough picture of the salary distribution.
Analyzing the Starwars dataset:

Task E(i): We extracted actors with non-black eye color and heights greater than 150 cm from the "starwars" dataset. This division facilitates the identification of discrete groups according to physical characteristics.
Task E(ii): A scatter plot of height against BMI was made and a new BMI column was added to the "starwars" dataset. This graphic offers important insights into the relationship between height and BMI by illustrating how BMI is distributed throughout a range of heights.