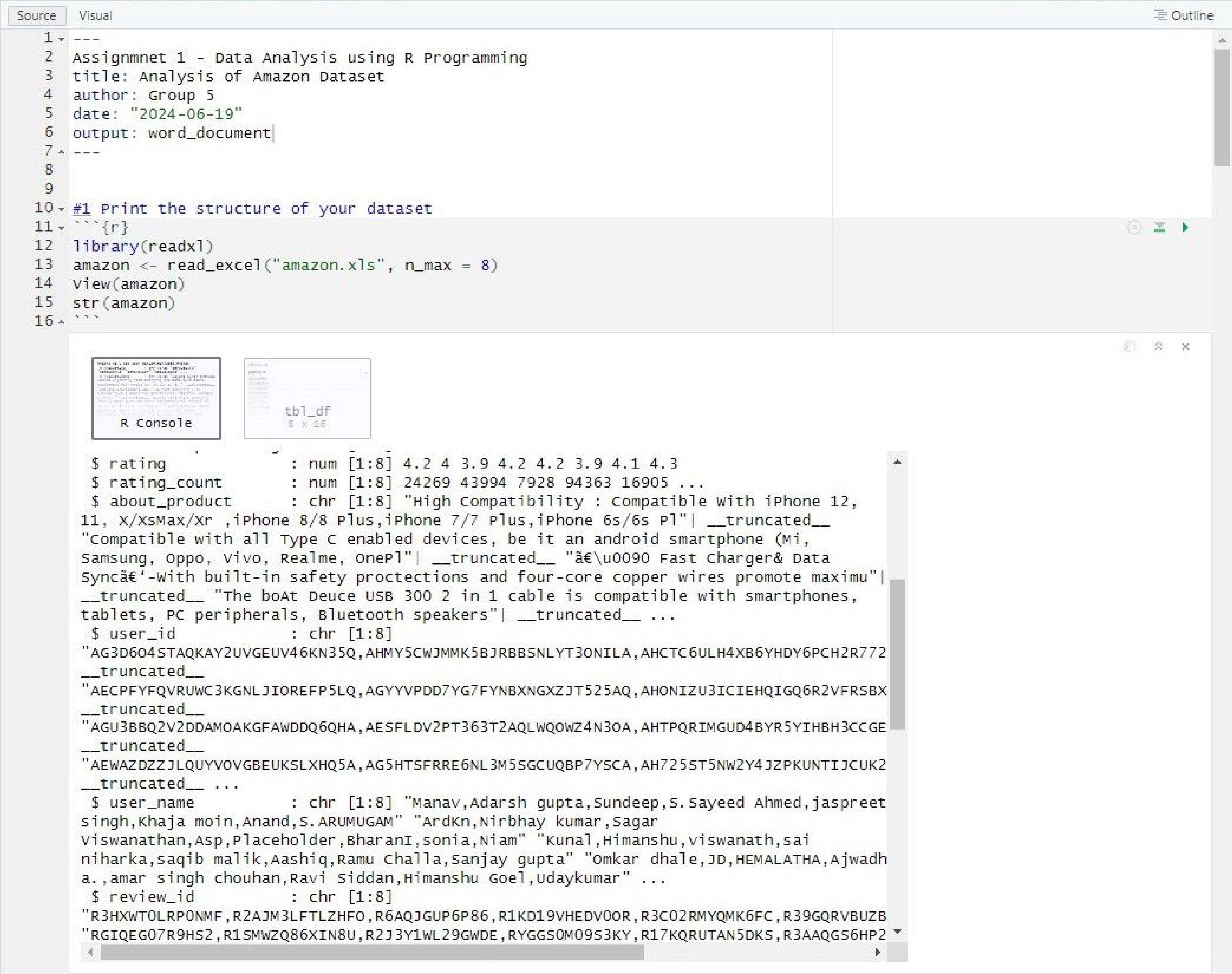
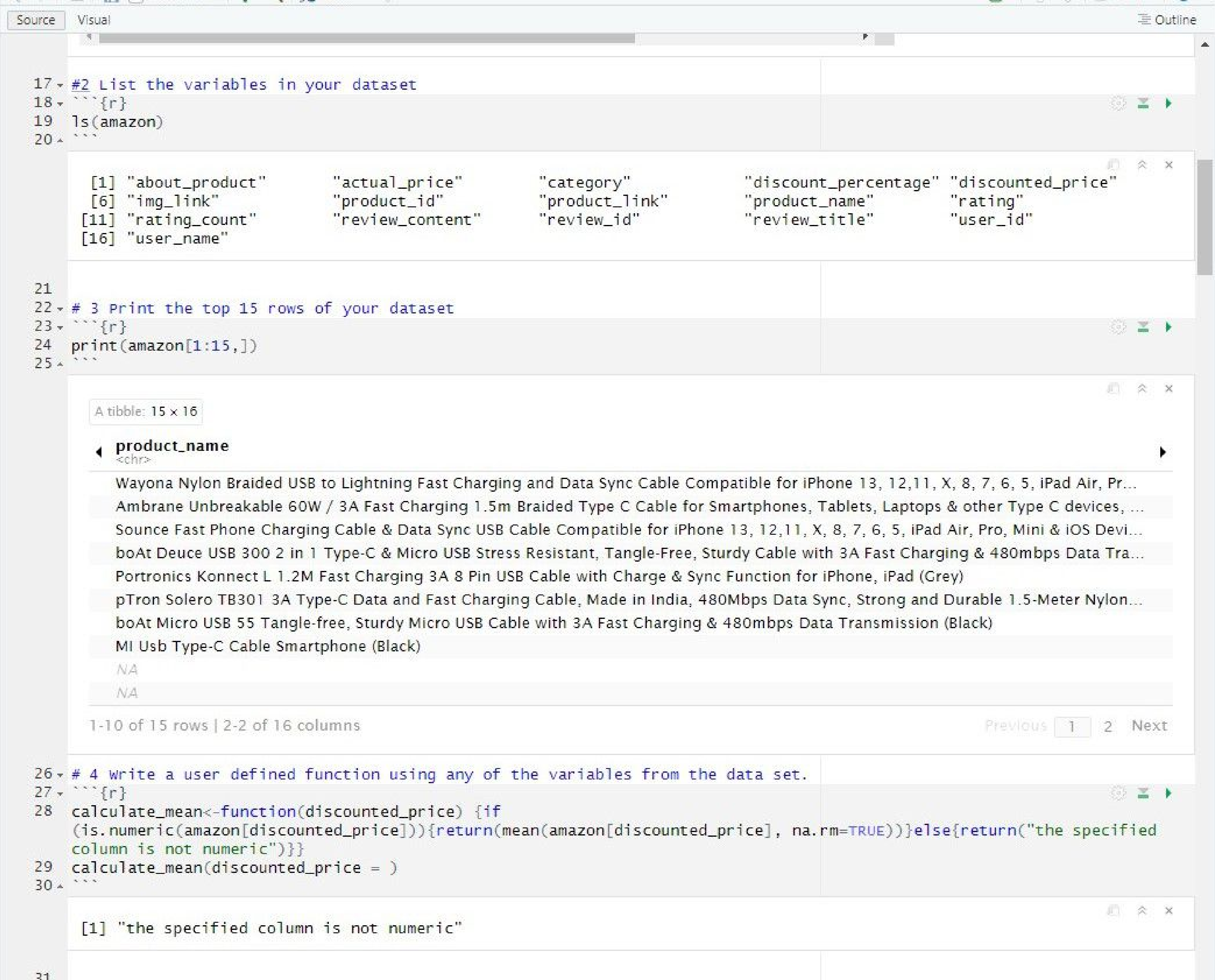


~~~
---
Assignmnet 1 - Data Analysis using R Programming
title: Analysis of Amazon Dataset
author: Group 5
date: "2024-06-19"
output: word_document
---


#1 Print the structure of your dataset
```{r}
library(readxl)
amazon <- read_excel("amazon.xls", n_max = 8)
View(amazon)
str(amazon)
```
~~~

```
 $ rating              : num [1:8] 4.2 4 3.9 4.2 4.2 3.9 4.1 4.3
 $ rating_count        : num [1:8] 24269 43994 7928 94363 16905 ...
 $ about_product       : chr [1:8] "High Compatibility : Compatible With iPhone 12,
11, X/XsMax/Xr ,iPhone 8/8 Plus,iPhone 7/7 Plus,iPhone 6s/6s Pl"| __truncated__
"Compatible with all Type C enabled devices, be it an android smartphone (Mi,
Samsung, Oppo, Vivo, Realme, OnePl"| __truncated__ "ã€\u0090 Fast Charger& Data
Syncã€'-With built-in safety proctections and four-core copper wires promote maximu"|
__truncated__ "The boAt Deuce USB 300 2 in 1 cable is compatible with smartphones,
tablets, PC peripherals, Bluetooth speakers"| __truncated__ ...
 $ user_id             : chr [1:8]
"AG3D6O4STAQKAY2UVGEUV46KN35Q,AHMY5CWJMMK5BJRBBSNLYT3ONILA,AHCTC6ULH4XB6YHDY6PCH2R772
__truncated__
"AECPFYFQVRUWC3KGNLJIOREFP5LQ,AGYYVPDD7YG7FYNBXNGXZJT525AQ,AHONIZU3ICIEHQIGQ6R2VFRSBX
__truncated__
"AGU3BBQ2V2DDAMOAKGFAWDDQ6QHA,AESFLDV2PT363T2AQLWQOWZ4N3OA,AHTPQRIMGUD4BYR5YIHBH3CCGE
__truncated__
"AEWAZDZZJLQUYVOVGBEUKSLXHQ5A,AG5HTSFRRE6NL3M5SGCUQBP7YSCA,AH725ST5NW2Y4JZPKUNTIJCUK2
__truncated__ ...
 $ user_name           : chr [1:8] "Manav,Adarsh gupta,Sundeep,S.Sayeed Ahmed,jaspreet
singh,Khaja moin,Anand,S.ARUMUGAM" "ArdKn,Nirbhay kumar,Sagar
Viswanathan,Asp,Placeholder,BharanI,sonia,Niam" "Kunal,Himanshu,viswanath,sai
niharka,saqib malik,Aashiq,Ramu Challa,Sanjay gupta" "Omkar dhale,JD,HEMALATHA,Ajwadh
a.,amar singh chouhan,Ravi Siddan,Himanshu Goel,Udaykumar" ...
 $ review_id           : chr [1:8]
"R3HXWTOLRPONMF,R2AJM3LFTLZHFO,R6AQJGUP6P86,R1KD19VHEDV0OR,R3C02RMYQMK6FC,R39GQRVBUZB
"RGIQEG07R9HS2,R1SMWZQ86XIN8U,R2J3Y1WL29GWDE,RYGGS0M09S3KY,R17KQRUTAN5DKS,R3AAQGS6HP2
```



#2 List the variables in your dataset

```{r}
ls(amazon)
```

```
 [1] "about_product"       "actual_price"        "category"            "discount_percentage" "discounted_price"
 [6] "img_link"            "product_id"          "product_link"        "product_name"        "rating"
[11] "rating_count"        "review_content"      "review_id"           "review_title"        "user_id"
[16] "user_name"
```

3 Print the top 15 rows of your dataset

```{r}
print(amazon[1:15,])
```

A tibble: 15 x 16

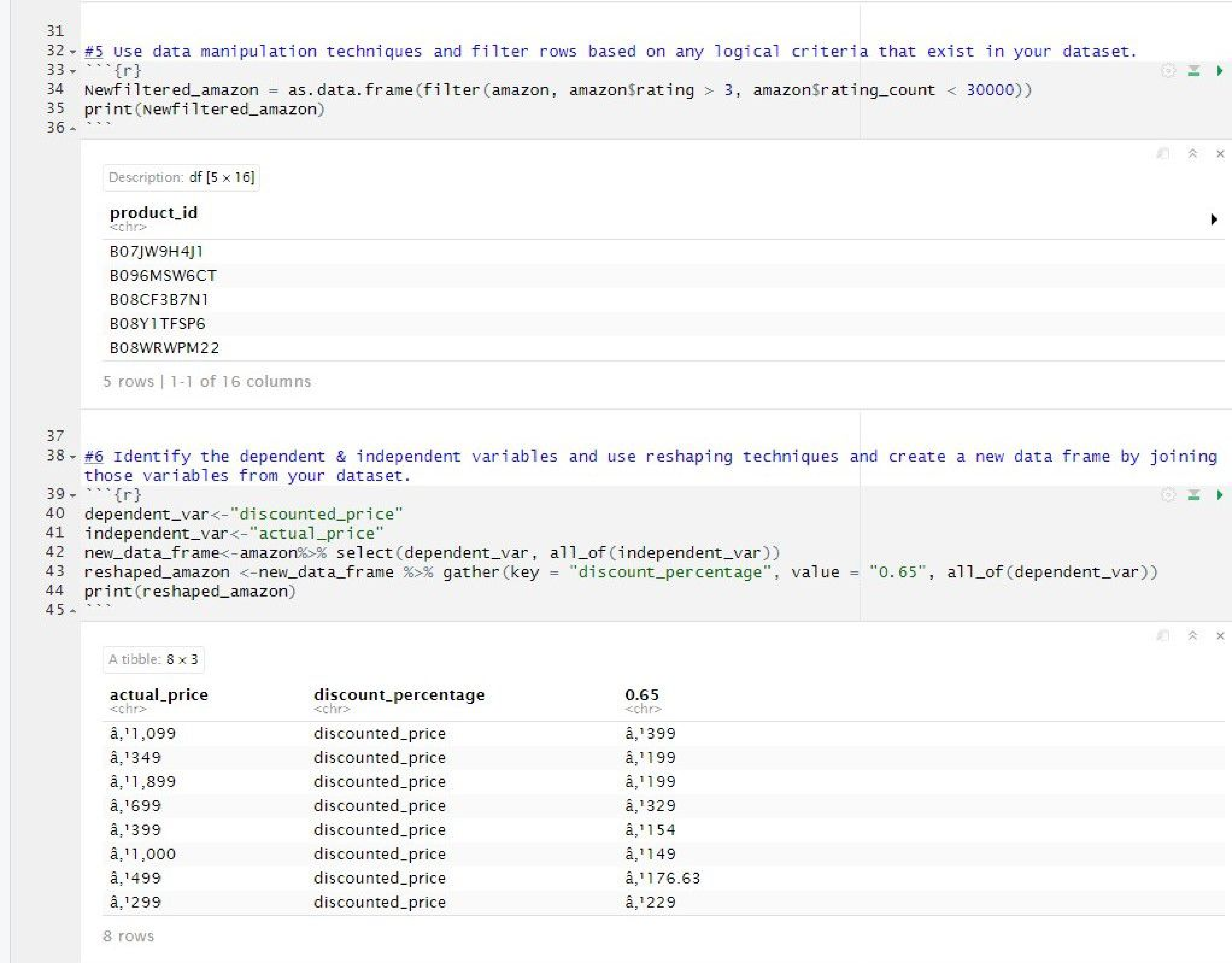
product_name <chr>
Wayona Nylon Braided USB to Lightning Fast Charging and Data Sync Cable Compatible for iPhone 13, 12,11, X, 8, 7, 6, 5, iPad Air, Pr...
Ambrane Unbreakable 60W / 3A Fast Charging 1.5m Braided Type C Cable for Smartphones, Tablets, Laptops & other Type C devices, ...
Sounce Fast Phone Charging Cable & Data Sync USB Cable Compatible for iPhone 13, 12,11, X, 8, 7, 6, 5, iPad Air, Pro, Mini & iOS Devi...
boAt Deuce USB 300 2 in 1 Type-C & Micro USB Stress Resistant, Tangle-Free, Sturdy Cable with 3A Fast Charging & 480mbps Data Tra...
Portronics Konnect L 1.2M Fast Charging 3A 8 Pin USB Cable with Charge & Sync Function for iPhone, iPad (Grey)
pTron Solero TB301 3A Type-C Data and Fast Charging Cable, Made in India, 480Mbps Data Sync, Strong and Durable 1.5-Meter Nylon...
boAt Micro USB 55 Tangle-free, Sturdy Micro USB Cable with 3A Fast Charging & 480mbps Data Transmission (Black)
MI Usb Type-C Cable Smartphone (Black)
NA
NA

1-10 of 15 rows | 2-2 of 16 columns

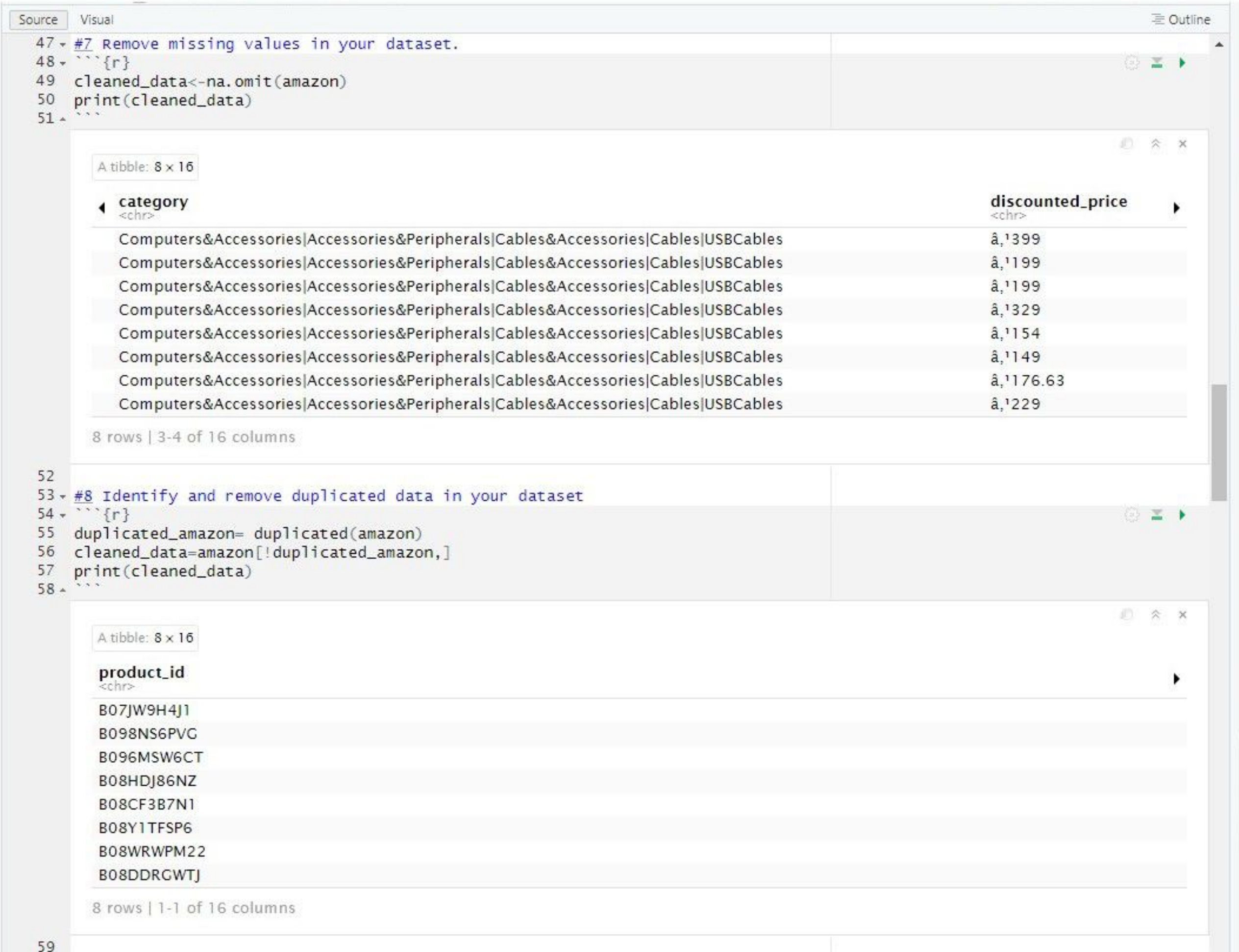
4 Write a user defined function using any of the variables from the data set.

```{r}
calculate_mean<-function(discounted_price) {if
(is.numeric(amazon[discounted_price])){return(mean(amazon[discounted_price], na.rm=TRUE))}else{return("the specified
column is not numeric")}}
calculate_mean(discounted_price = )
```

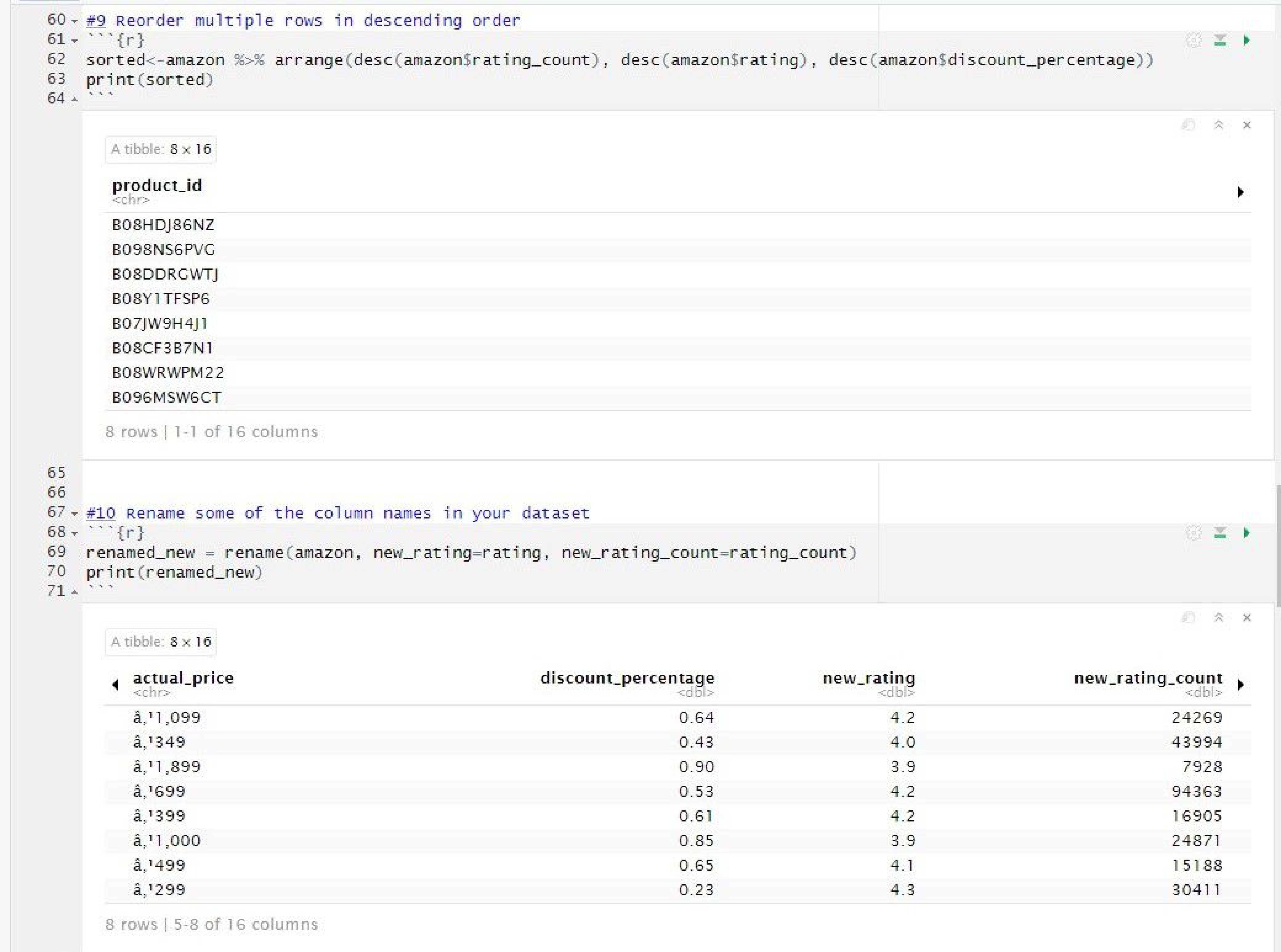
```
[1] "the specified column is not numeric"
```



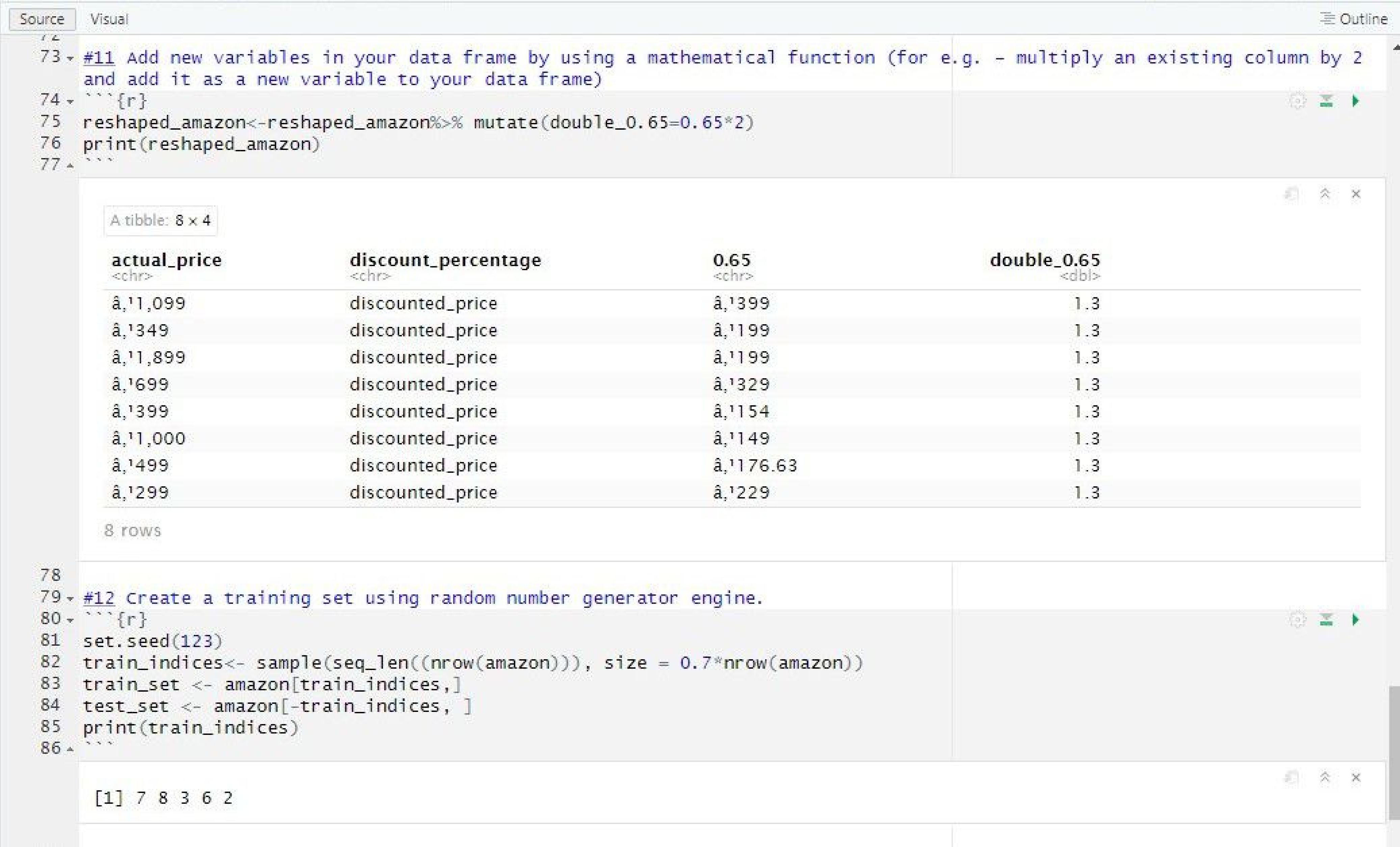
~~~
#5 Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset.
```{r}
Newfiltered_amazon = as.data.frame(filter(amazon, amazon$rating > 3, amazon$rating_count < 30000))
print(Newfiltered_amazon)
~~~
```

Description: df [5 x 16]

product_id <chr>
B07JW9H4J1
B096MSW6CT
B08CF3B7N1
B08Y1TFSP6
B08WRWPM22

5 rows | 1-1 of 16 columns

```
#6 Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset.
```{r}
dependent_var<-"discounted_price"
independent_var<-"actual_price"
new_data_frame<-amazon%>% select(dependent_var, all_of(independent_var))
reshaped_amazon <-new_data_frame %>% gather(key = "discount_percentage", value = "0.65", all_of(dependent_var))
print(reshaped_amazon)
```
```

A tibble: 8 x 3

actual_price <chr>	discount_percentage <chr>	0.65 <chr>
â‚¹1,099	discounted_price	â‚¹399
â‚¹349	discounted_price	â‚¹199
â‚¹1,899	discounted_price	â‚¹199
â‚¹699	discounted_price	â‚¹329
â‚¹399	discounted_price	â‚¹154
â‚¹1,000	discounted_price	â‚¹149
â‚¹499	discounted_price	â‚¹176.63
â‚¹299	discounted_price	â‚¹229

8 rows
```



~~~
#7 Remove missing values in your dataset.
```{r}
cleaned_data<-na.omit(amazon)
print(cleaned_data)
~~~
```

A tibble: 8 × 16

category <chr>	discounted_price <chr>
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹399
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹199
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹199
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹329
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹154
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹149
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹176.63
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables	â‚¹229

8 rows | 3-4 of 16 columns

```
#8 Identify and remove duplicated data in your dataset
```{r}
duplicated_amazon= duplicated(amazon)
cleaned_data=amazon[!duplicated_amazon,]
print(cleaned_data)
```
```

A tibble: 8 × 16

| product_id<br><chr> |
|---|
| B07JW9H4J1 |
| B098NS6PVG |
| B096MSW6CT |
| B08HDJ86NZ |
| B08CF3B7N1 |
| B08Y1TFSP6 |
| B08WRWPM22 |
| B08DDRGWTJ |

8 rows | 1-1 of 16 columns
```



~~~
#9 Reorder multiple rows in descending order
```{r}
sorted<-amazon %>% arrange(desc(amazon$rating_count), desc(amazon$rating), desc(amazon$discount_percentage))
print(sorted)
~~~
```

A tibble: 8 x 16

product_id <chr>
B08HDJ86NZ
B098NS6PVG
B08DDRGWTJ
B08Y1TFSP6
B07JW9H4J1
B08CF3B7N1
B08WRWPM22
B096MSW6CT

8 rows | 1-1 of 16 columns

```
#10 Rename some of the column names in your dataset
```{r}
renamed_new = rename(amazon, new_rating=rating, new_rating_count=rating_count)
print(renamed_new)
```
```

A tibble: 8 x 16

actual_price <chr>	discount_percentage <dbl>	new_rating <dbl>	new_rating_count <dbl>
â‚¹1,099	0.64	4.2	24269
â‚¹349	0.43	4.0	43994
â‚¹1,899	0.90	3.9	7928
â‚¹699	0.53	4.2	94363
â‚¹399	0.61	4.2	16905
â‚¹1,000	0.85	3.9	24871
â‚¹499	0.65	4.1	15188
â‚¹299	0.23	4.3	30411

8 rows | 5-8 of 16 columns
```



~~~
#11 Add new variables in your data frame by using a mathematical function (for e.g. - multiply an existing column by 2 and add it as a new variable to your data frame)
```{r}
reshaped_amazon<-reshaped_amazon%>% mutate(double_0.65=0.65*2)
print(reshaped_amazon)
```
~~~

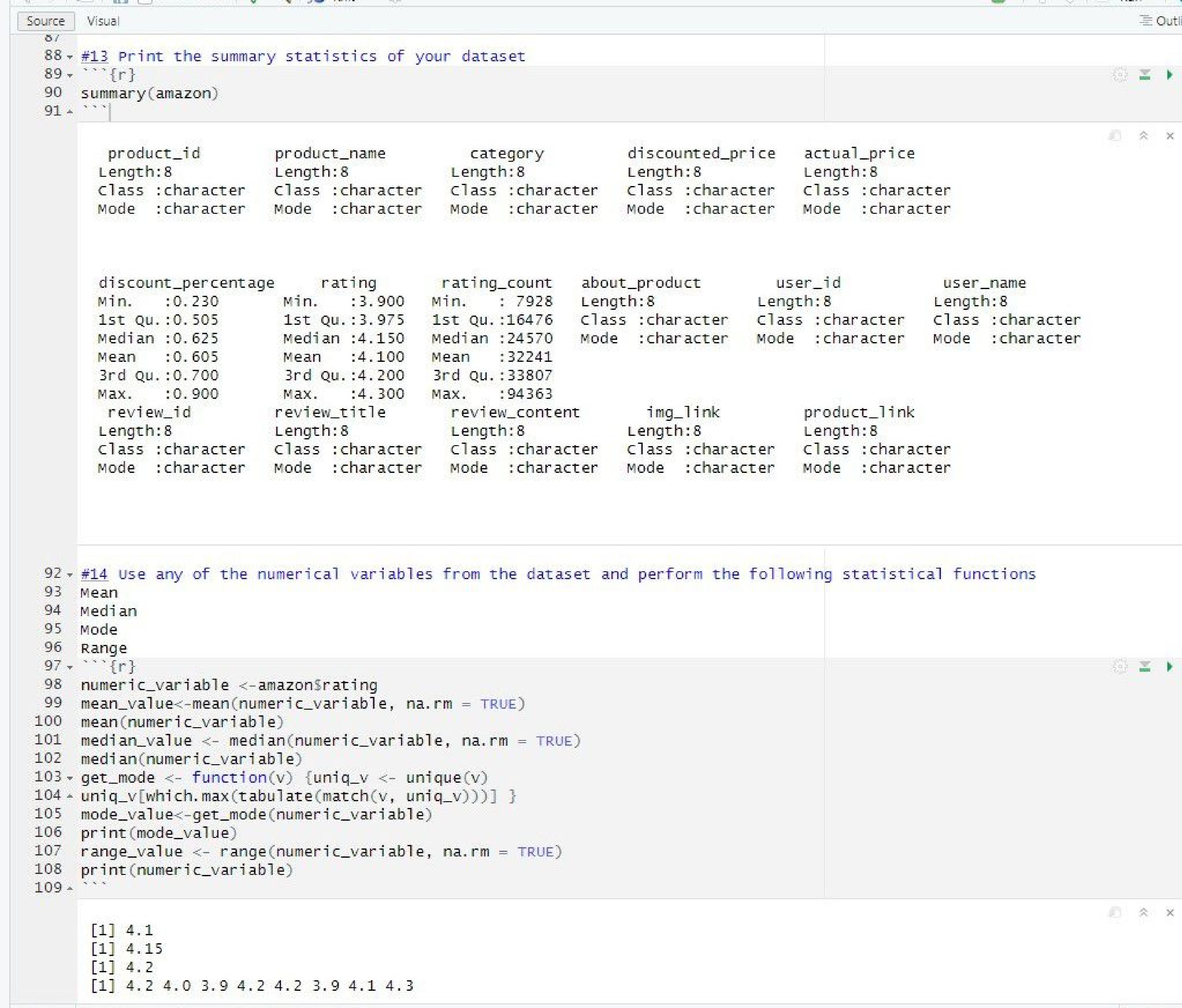
A tibble: 8 x 4

actual_price <chr>	discount_percentage <chr>	0.65 <chr>	double_0.65 <dbl>
â‚¹1,099	discounted_price	â‚¹399	1.3
â‚¹349	discounted_price	â‚¹199	1.3
â‚¹1,899	discounted_price	â‚¹199	1.3
â‚¹699	discounted_price	â‚¹329	1.3
â‚¹399	discounted_price	â‚¹154	1.3
â‚¹1,000	discounted_price	â‚¹149	1.3
â‚¹499	discounted_price	â‚¹176.63	1.3
â‚¹299	discounted_price	â‚¹229	1.3

8 rows

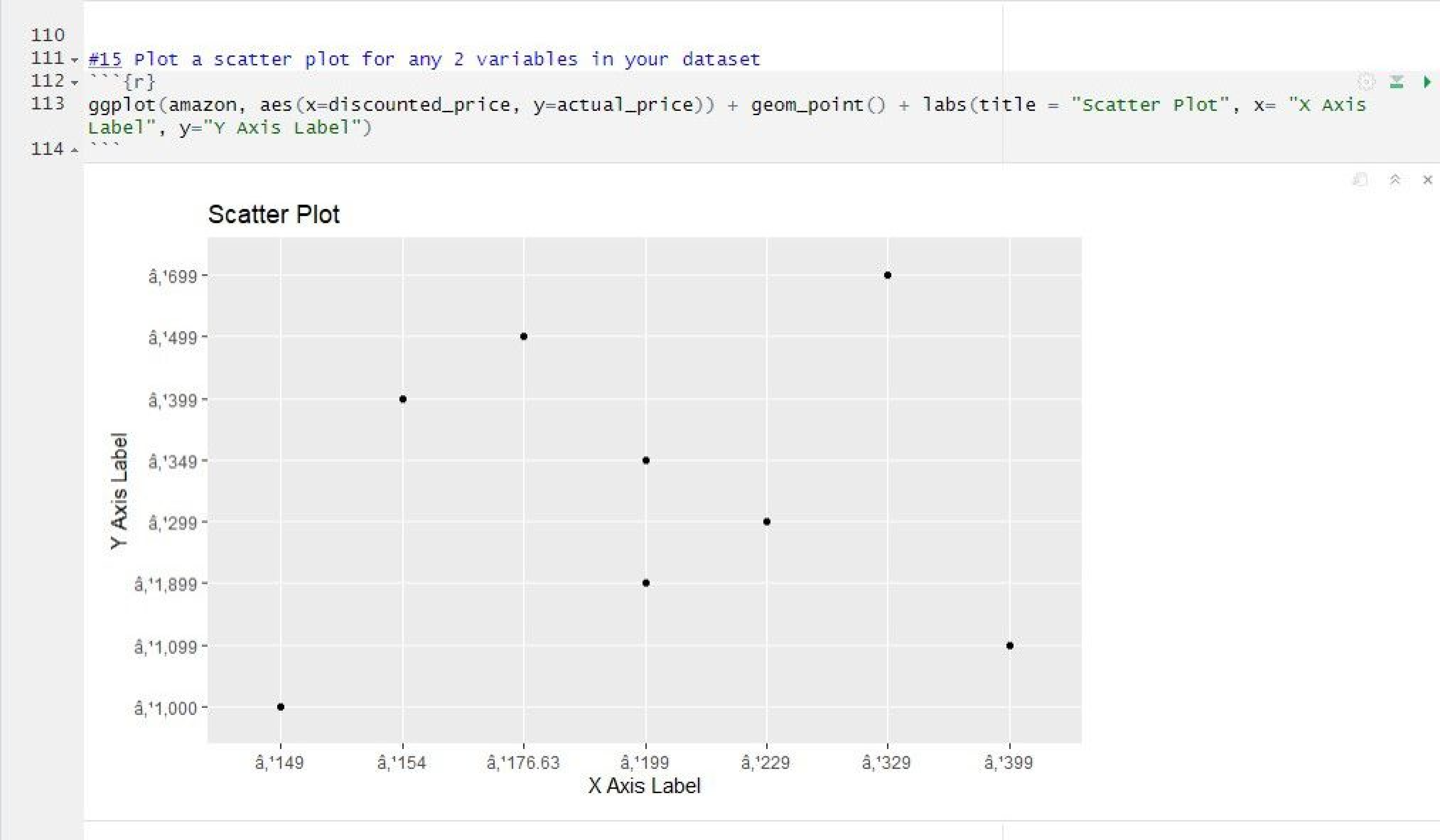
~~~
#12 Create a training set using random number generator engine.
```{r}
set.seed(123)
train_indices<- sample(seq_len((nrow(amazon))), size = 0.7*nrow(amazon))
train_set <- amazon[train_indices,]
test_set <- amazon[-train_indices, ]
print(train_indices)
```
~~~

```
[1] 7 8 3 6 2
```



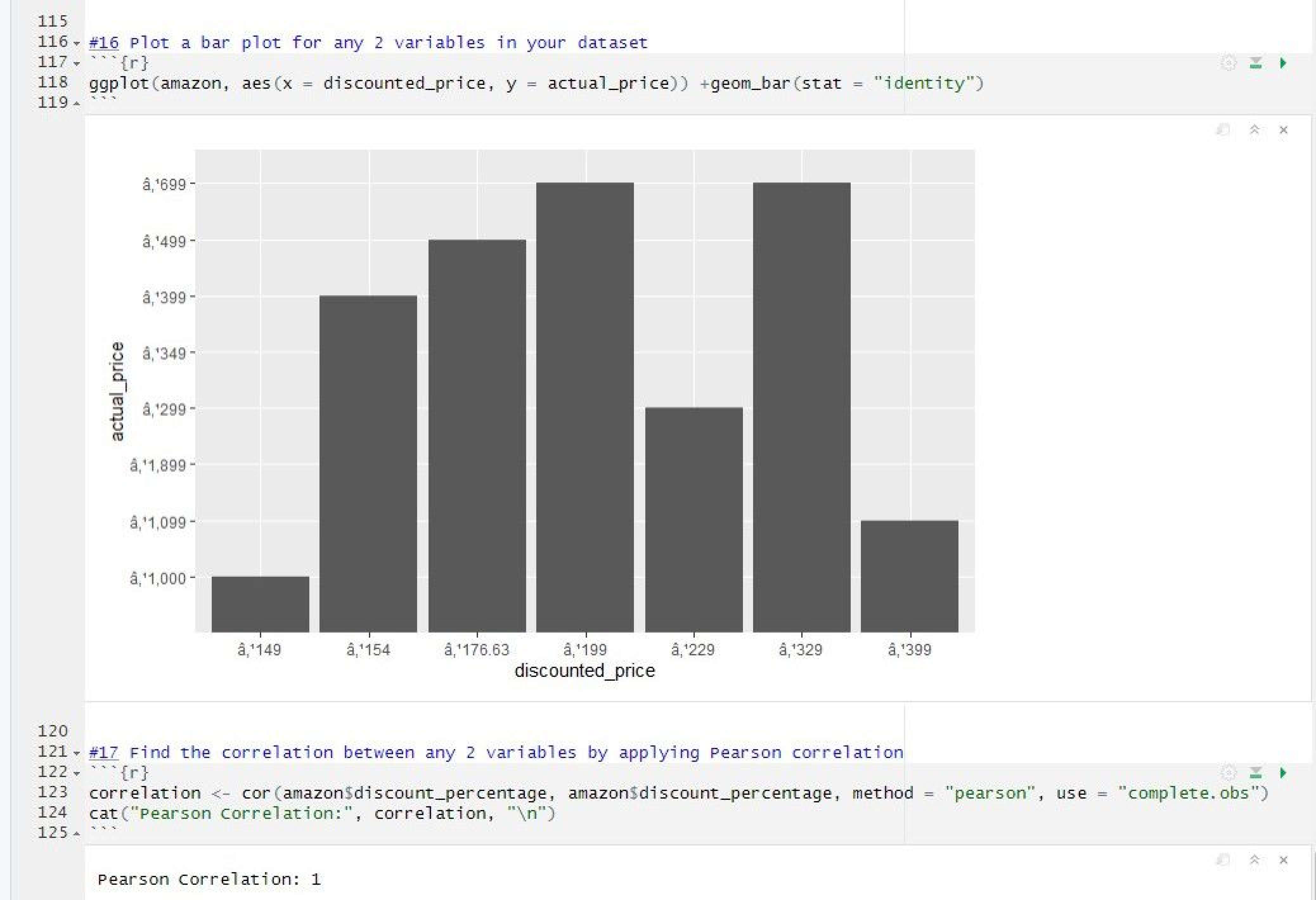
~~~
#13 Print the summary statistics of your dataset
```{r}
summary(amazon)
~~~
```

```
  product_id        product_name         category         discounted_price   actual_price
 Length:8           Length:8           Length:8           Length:8           Length:8
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character

 discount_percentage     rating       rating_count   about_product        user_id            user_name
 Min.   :0.230       Min.   :3.900   Min.   : 7928   Length:8           Length:8           Length:8
 1st Qu.:0.505       1st Qu.:3.975   1st Qu.:16476   Class :character   Class :character   Class :character
 Median :0.625       Median :4.150   Median :24570   Mode  :character   Mode  :character   Mode  :character
 Mean   :0.605       Mean   :4.100   Mean   :32241
 3rd Qu.:0.700       3rd Qu.:4.200   3rd Qu.:33807
 Max.   :0.900       Max.   :4.300   Max.   :94363
  review_id         review_title       review_content       img_link         product_link
 Length:8           Length:8           Length:8           Length:8           Length:8
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
#14 Use any of the numerical variables from the dataset and perform the following statistical functions
Mean
Median
Mode
Range
```{r}
numeric_variable <-amazon$rating
mean_value<-mean(numeric_variable, na.rm = TRUE)
mean(numeric_variable)
median_value <- median(numeric_variable, na.rm = TRUE)
median(numeric_variable)
get_mode <- function(v) {uniq_v <- unique(v)
uniq_v[which.max(tabulate(match(v, uniq_v)))] }
mode_value<-get_mode(numeric_variable)
print(mode_value)
range_value <- range(numeric_variable, na.rm = TRUE)
print(numeric_variable)
```
```

```
[1] 4.1
[1] 4.15
[1] 4.2
[1] 4.2 4.0 3.9 4.2 4.2 3.9 4.1 4.3
```
```



~~~
#15 Plot a scatter plot for any 2 variables in your dataset
```{r}
ggplot(amazon, aes(x=discounted_price, y=actual_price)) + geom_point() + labs(title = "Scatter Plot", x= "X Axis Label", y="Y Axis Label")
```
~~~

Scatter Plot



~~~
#16 Plot a bar plot for any 2 variables in your dataset
```{r}
ggplot(amazon, aes(x = discounted_price, y = actual_price)) +geom_bar(stat = "identity")
~~~
```

â‚¹699
â‚¹499
â‚¹399
â‚¹349
â‚¹299
â‚¹1,899
â‚¹1,099
â‚¹1,000
actual_price
â‚¹149
â‚¹154
â‚¹176.63
â‚¹199
â‚¹229
â‚¹329
â‚¹399
discounted_price

```
#17 Find the correlation between any 2 variables by applying Pearson correlation
```{r}
correlation <- cor(amazon$discount_percentage, amazon$discount_percentage, method = "pearson", use = "complete.obs")
cat("Pearson Correlation:", correlation, "\n")
```
```

```
Pearson Correlation: 1
```
```