Alternative Assessment 2

Janice Chong See Wai (S2132420)

2022-06-02

Question(a)

Find and get a dataset from the datasets available within R. Perform exploratory data analysis (EDA) and prepare a codebook on that dataset using a newer method in R. Label your work clearly on EDA and codebook.

In this question, I will be using the dataset called, "USArrests", which shows the Violent Crime Rates by US State.

EDA

```
# Get data, show the head of the data, get the summary of the data.
data("USArrests")
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

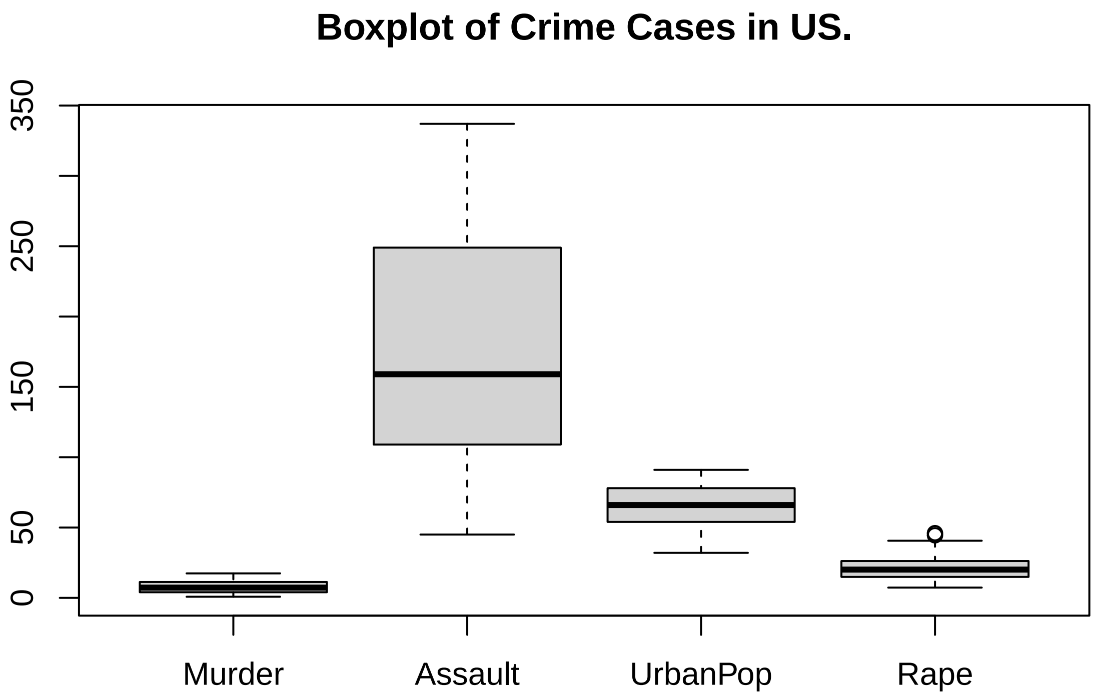
```
summary(USArrests)
```

```
##      Murder          Assault         UrbanPop          Rape      
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30  
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07  
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10  
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23  
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18  
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00  
```

Boxplot

I used boxplot because boxplot can show the distribution of the data of each variable in the five number summary that are minimum, first quartile, median, third quartile and maximum.

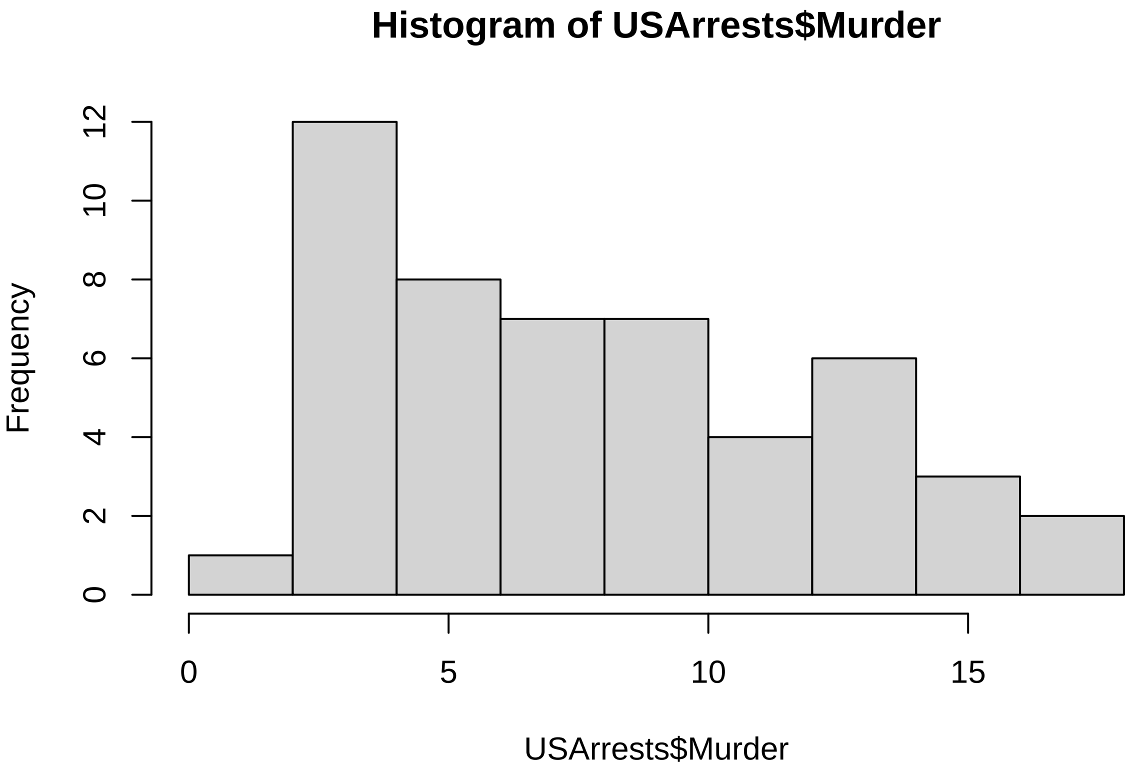
```r
# Create a boxplot to show the summary of the dataset.
boxplot(USArrests,
        main = "Boxplot of Crime Cases in US.")
```



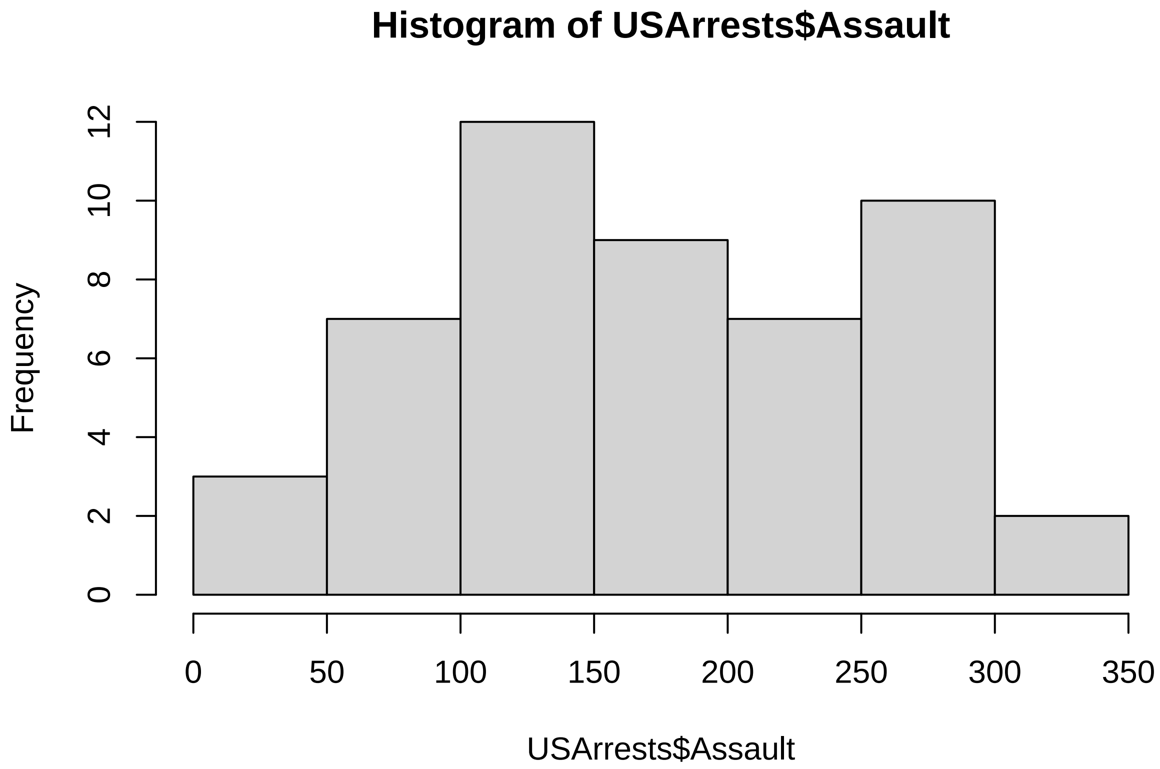
Histogram

I used histogram because histogram can show the frequency distribution of the data of each variable.

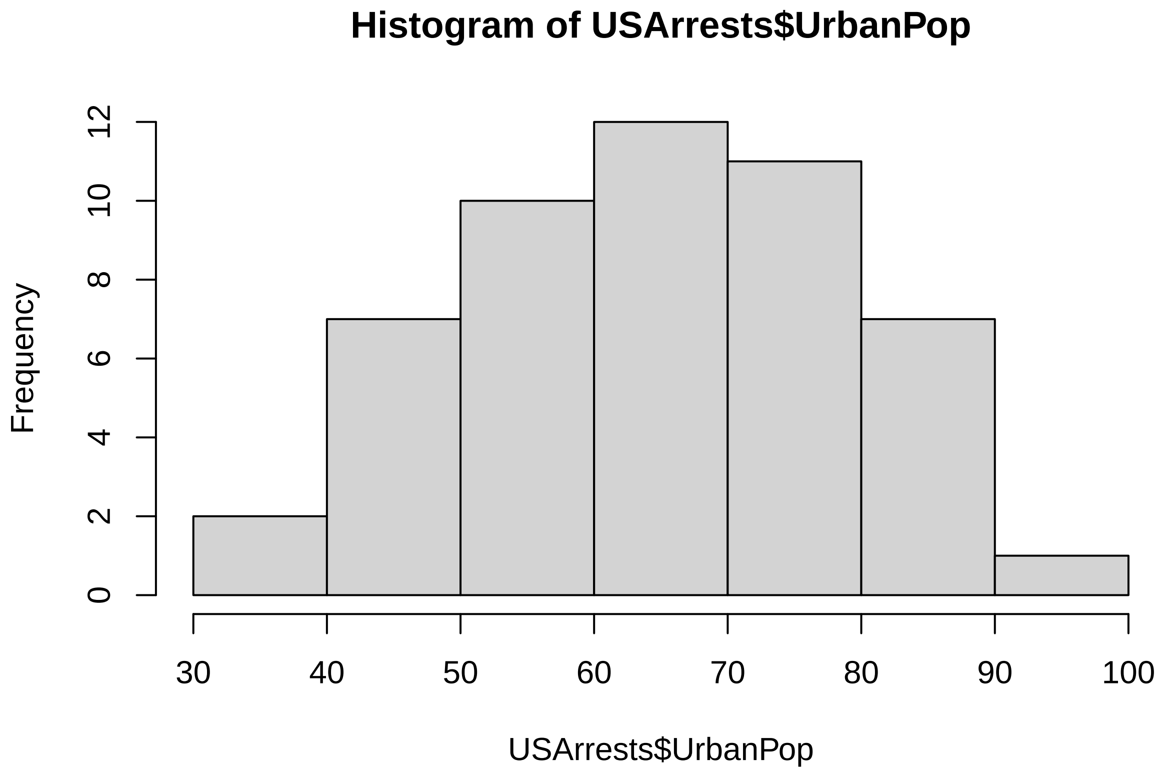
```r
# Plot histogram of each case.
hist(USArrests$Murder)
```



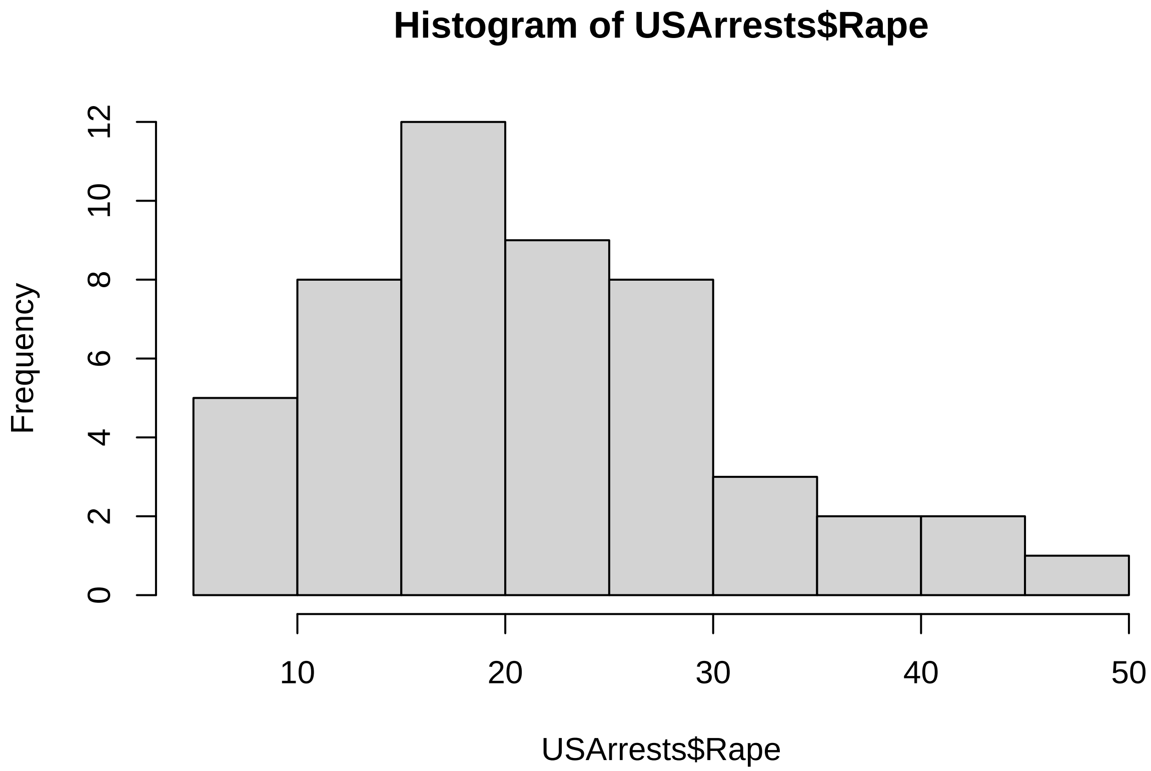
```
hist(USArrests$Assault)
```



```
hist(USArrests$UrbanPop)
```



```
hist(USArrests$Rape)
```



Codebook

Make codebook by using a new method.

```
# I called a package, "dataMaid". Then, I used the function "makeCodebook()" to make a code book.
# The code book will automatically be created as a new PDF document.

library(dataMaid)
makeCodebook(USArrests)
```

```
## Data report generation is finished. Please wait while your output file is being rendered.
```

```
##
##  Is codebook_USArrests.pdf open on your computer? Please close it as fast as possible to avoid probl
```

Question(b)

Demonstrate these FIVE (5) functions of dplyr for data manipulation:

i. filter ( )

ii. arrange ( )

iii. mutate ( )

iv. select ( )

v. summarise ( )

You can create your own sensible dataset in certain context for this question with at least 15 observations (rows) and 4 features (columns) or you can get any suitable dataset online. Show the R code and provide a short explanation on what each function does for each produced output.

```
# Calling dplyr package
library(dplyr)
```

First, we call the dplyr package.

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

## The following object is masked from 'package:dataMaid':
##
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

We will read a new csv dataset.

```
data <- read.csv("supermarket_sales.csv")
head(data)
```

```
##    Invoice.ID Branch      City Customer.type Gender           Product.line
## 1 750-67-8428      A    Yangon        Member Female      Health and beauty
## 2 226-31-3081      C Naypyitaw        Normal Female Electronic accessories
## 3 631-41-3108      A    Yangon        Normal   Male     Home and lifestyle
## 4 123-19-1176      A    Yangon        Member   Male      Health and beauty
## 5 373-73-7910      A    Yangon        Normal   Male      Sports and travel
## 6 699-14-3026      C Naypyitaw        Normal   Male Electronic accessories
##   Unit.price Quantity  Tax.5.    Total      Date  Time     Payment   cogs
## 1      74.69        7 26.1415 548.9715  1/5/2019 13:08     Ewallet 522.83
## 2      15.28        5  3.8200  80.2200  3/8/2019 10:29        Cash  76.40
## 3      46.33        7 16.2155 340.5255  3/3/2019 13:23 Credit card 324.31
## 4      58.22        8 23.2880 489.0480 1/27/2019 20:33     Ewallet 465.76
## 5      86.31        7 30.2085 634.3785  2/8/2019 10:37     Ewallet 604.17
## 6      85.39        7 29.8865 627.6165 3/25/2019 18:30     Ewallet 597.73
##   gross.margin.percentage gross.income Rating
## 1                4.761905      26.1415    9.1
## 2                4.761905       3.8200    9.6
## 3                4.761905      16.2155    7.4
## 4                4.761905      23.2880    8.4
## 5                4.761905      30.2085    5.3
## 6                4.761905      29.8865    4.1
```

```
summary(data)
```

```
##   Invoice.ID           Branch              City           Customer.type     
##  Length:1000        Length:1000        Length:1000        Length:1000       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##     Gender          Product.line         Unit.price       Quantity    
##  Length:1000        Length:1000        Min.   :10.08   Min.   : 1.00  
##  Class :character   Class :character   1st Qu.:32.88   1st Qu.: 3.00  
##  Mode  :character   Mode  :character   Median :55.23   Median : 5.00  
##                                        Mean   :55.67   Mean   : 5.51  
##                                        3rd Qu.:77.94   3rd Qu.: 8.00  
##                                        Max.   :99.96   Max.   :10.00  
##      Tax.5.           Total             Date               Time          
##  Min.   : 0.5085   Min.   :  10.68   Length:1000        Length:1000       
##  1st Qu.: 5.9249   1st Qu.: 124.42   Class :character   Class :character  
##  Median :12.0880   Median : 253.85   Mode  :character   Mode  :character  
##  Mean   :15.3794   Mean   : 322.97                                        
##  3rd Qu.:22.4453   3rd Qu.: 471.35                                        
##  Max.   :49.6500   Max.   :1042.65                                        
##    Payment              cogs        gross.margin.percentage  gross.income    
##  Length:1000        Min.   : 10.17   Min.   :4.762           Min.   : 0.5085  
##  Class :character   1st Qu.:118.50   1st Qu.:4.762           1st Qu.: 5.9249  
##  Mode  :character   Median :241.76   Median :4.762           Median :12.0880  
##                     Mean   :307.59   Mean   :4.762           Mean   :15.3794  
##                     3rd Qu.:448.90   3rd Qu.:4.762           3rd Qu.:22.4453  
##                     Max.   :993.00   Max.   :4.762           Max.   :49.6500  
##      Rating     
##  Min.   : 4.000  
##  1st Qu.: 5.500  
##  Median : 7.000  
##  Mean   : 6.973  
##  3rd Qu.: 8.500  
##  Max.   :10.000
```

i. filter()

The filter() function is to provide a subset of a data frame while maintaing all rows of the specified conditions that we have made.

```
# Filtering data with Branch A, Gross income > 15 and has a Rating of > 8
filtered <- filter(data, Branch=="A", gross.income>15, Rating>8)
head(filtered)
```

```
##    Invoice.ID Branch   City Customer.type Gender       Product.line Unit.price
## 1 750-67-8428      A Yangon        Member Female  Health and beauty      74.69
## 2 123-19-1176      A Yangon        Member   Male  Health and beauty      58.22
## 3 252-56-2699      A Yangon        Normal   Male Food and beverages      43.19
## 4 227-03-5010      A Yangon        Member Female Home and lifestyle      52.59
## 5 287-21-9091      A Yangon        Normal   Male Home and lifestyle      74.67
## 6 212-62-1842      A Yangon        Normal   Male Food and beverages      58.26
##   Quantity  Tax.5.    Total      Date  Time     Payment   cogs
## 1        7 26.1415 548.9715  1/5/2019 13:08     Ewallet 522.83
## 2        8 23.2880 489.0480 1/27/2019 20:33     Ewallet 465.76
## 3       10 21.5950 453.4950  2/7/2019 16:48     Ewallet 431.90
## 4        8 21.0360 441.7560 3/22/2019 19:20 Credit card 420.72
## 5        9 33.6015 705.6315 1/22/2019 10:55     Ewallet 672.03
## 6        6 17.4780 367.0380 3/28/2019 16:44        Cash 349.56
##   gross.margin.percentage gross.income Rating
## 1                4.761905      26.1415    9.1
## 2                4.761905      23.2880    8.4
## 3                4.761905      21.5950    8.2
## 4                4.761905      21.0360    8.5
## 5                4.761905      33.6015    9.4
## 6                4.761905      17.4780    9.9
```

```
# Filtering data with Gender as Female, Payment with Ewallet and has a Rating of > 8
filtered <- filter(data, Gender=="Female", Payment=="Ewallet",
                   Rating>8)
head(filtered)
```

```
##    Invoice.ID Branch      City Customer.type Gender        Product.line
## 1 750-67-8428      A    Yangon        Member Female   Health and beauty
## 2 347-34-2234      B  Mandalay        Member Female   Sports and travel
## 3 109-28-2512      B  Mandalay        Member Female Fashion accessories
## 4 225-32-0908      C Naypyitaw        Normal Female   Sports and travel
## 5 663-86-9076      C Naypyitaw        Member Female  Food and beverages
## 6 685-64-1609      A    Yangon        Member Female Fashion accessories
##   Unit.price Quantity  Tax.5.    Total      Date  Time Payment   cogs
## 1      74.69        7 26.1415 548.9715  1/5/2019 13:08 Ewallet 522.83
## 2      55.07        9 24.7815 520.4115  2/3/2019 13:40 Ewallet 495.63
## 3      97.61        6 29.2830 614.9430  1/7/2019 15:01 Ewallet 585.66
## 4      44.86       10 22.4300 471.0300 1/26/2019 19:54 Ewallet 448.60
## 5      68.54        8 27.4160 575.7360  1/8/2019 15:57 Ewallet 548.32
## 6      30.14       10 15.0700 316.4700 2/10/2019 12:28 Ewallet 301.40
##   gross.margin.percentage gross.income Rating
## 1                4.761905      26.1415    9.1
## 2                4.761905      24.7815   10.0
## 3                4.761905      29.2830    9.9
## 4                4.761905      22.4300    8.2
## 5                4.761905      27.4160    8.5
## 6                4.761905      15.0700    9.2
```

ii. arrange()

The arrange() function is to arrange the dataset based on their column names.

```
# We will arrange the dataset according to the payment method.
arranged <- arrange(data, Payment)
head(arranged)
```

```
##    Invoice.ID Branch      City Customer.type Gender           Product.line
## 1 226-31-3081      C Naypyitaw        Normal Female Electronic accessories
## 2 529-56-3974      B  Mandalay        Member   Male Electronic accessories
## 3 829-34-3910      A    Yangon        Normal Female      Health and beauty
## 4 299-46-1805      B  Mandalay        Member Female      Sports and travel
## 5 649-29-6775      B  Mandalay        Normal   Male    Fashion accessories
## 6 145-94-9061      B  Mandalay        Normal Female     Food and beverages
##   Unit.price Quantity Tax.5.   Total      Date  Time Payment   cogs
## 1      15.28        5  3.820  80.220  3/8/2019 10:29    Cash  76.40
## 2      25.51        4  5.102 107.142  3/9/2019 17:03    Cash 102.04
## 3      71.38       10 35.690 749.490 3/29/2019 19:21    Cash 713.80
## 4      93.72        6 28.116 590.436 1/15/2019 16:19    Cash 562.32
## 5      33.52        1  1.676  35.196  2/8/2019 15:31    Cash  33.52
## 6      88.36        5 22.090 463.890 1/25/2019 19:48    Cash 441.80
##   gross.margin.percentage gross.income Rating
## 1                4.761905        3.820    9.6
## 2                4.761905        5.102    6.8
## 3                4.761905       35.690    5.7
## 4                4.761905       28.116    4.5
## 5                4.761905        1.676    6.7
## 6                4.761905       22.090    9.6
```

iii. mutate()

The mutate() function creates a neew variable from existing data.

```
# We will create a new variable called "Mean.Gross.Income" which contains the mean of gross income.
mutated <- mutate(data, mean.gross.income = mean(data$gross.income))
head(mutated)
```

```
##    Invoice.ID Branch      City Customer.type Gender           Product.line
## 1 750-67-8428      A    Yangon        Member Female      Health and beauty
## 2 226-31-3081      C Naypyitaw        Normal Female Electronic accessories
## 3 631-41-3108      A    Yangon        Normal   Male     Home and lifestyle
## 4 123-19-1176      A    Yangon        Member   Male      Health and beauty
## 5 373-73-7910      A    Yangon        Normal   Male      Sports and travel
## 6 699-14-3026      C Naypyitaw        Normal   Male Electronic accessories
##   Unit.price Quantity  Tax.5.    Total      Date  Time     Payment   cogs
## 1      74.69        7 26.1415 548.9715  1/5/2019 13:08     Ewallet 522.83
## 2      15.28        5  3.8200  80.2200  3/8/2019 10:29        Cash  76.40
## 3      46.33        7 16.2155 340.5255  3/3/2019 13:23 Credit card 324.31
## 4      58.22        8 23.2880 489.0480 1/27/2019 20:33     Ewallet 465.76
## 5      86.31        7 30.2085 634.3785  2/8/2019 10:37     Ewallet 604.17
## 6      85.39        7 29.8865 627.6165 3/25/2019 18:30     Ewallet 597.73
##   gross.margin.percentage gross.income Rating mean.gross.income
## 1                4.761905      26.1415    9.1          15.37937
## 2                4.761905       3.8200    9.6          15.37937
## 3                4.761905      16.2155    7.4          15.37937
## 4                4.761905      23.2880    8.4          15.37937
## 5                4.761905      30.2085    5.3          15.37937
## 6                4.761905      29.8865    4.1          15.37937
```

iv. select ( )

The select() function is used to select a particular column in a dataset.

```
# We will select only the Invoice.ID, Branch, Product.line, gross.margin.percentage from the dataset.
selected <- select(data, Invoice.ID, Branch, Product.line,
       gross.margin.percentage)
head(selected)
```

```
##    Invoice.ID Branch           Product.line gross.margin.percentage
## 1 750-67-8428      A      Health and beauty                4.761905
## 2 226-31-3081      C Electronic accessories                4.761905
## 3 631-41-3108      A     Home and lifestyle                4.761905
## 4 123-19-1176      A      Health and beauty                4.761905
## 5 373-73-7910      A      Sports and travel                4.761905
## 6 699-14-3026      C Electronic accessories                4.761905
```

v. summarise()

The summarise ( ) function is to make a summary of a data frame by creating a new data frame but with lesser variables depending on what we want.

```
# We will make a summary of the dataset that is group by Rating>8 and contains mean gross income.
data %>% group_by(Rating>8) %>% summarise(mean.gross.icome = mean(data$gross.income))
```

```
## # A tibble: 2 x 2
##   `Rating > 8` mean.gross.icome
##   <lgl>                   <dbl>
## 1 FALSE                    15.4
## 2 TRUE                     15.4
```