# Learning Activity 3

```
# 1 Libraries that can import CSV and others

# a default R utils package contains read.table which contains read.csv()

print("read.csv() from utils")
```

```
## [1] "read.csv() from utils"
```

```
csv1 <- utils::read.csv("testdata.csv")
csv1
```

```
##      Name Age Gender          City
## 1    John  25   Male      New York
## 2   Alice  30 Female San Francisco
## 3 Michael  22   Male   Los Angeles
## 4   Emily  28 Female       Chicago
## 5   David  35   Male       Houston
## 6  Sophia  27 Female         Miami
```

```
# b package readr is newer, faster, and auto converts file to data.frame
# https://www.pauloldham.net/importing-csv-files-into-r/#reading-in-a-file-using-read.table-utils-packa

install.packages("readr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(readr)
print("read_csv() from readr")
```

```
## [1] "read_csv() from readr"
```

```
csv2 <- readr::read_csv("testdata.csv")
```

```
## Rows: 6 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Name, Gender, City
## dbl (1): Age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
csv2
```

```
## # A tibble: 6 x 4
##   Name      Age Gender City
##   <chr>   <dbl> <chr>  <chr>
## 1 John       25 Male   New York
## 2 Alice      30 Female San Francisco
```

```
## 3 Michael    22 Male   Los Angeles
## 4 Emily      28 Female Chicago
## 5 David      35 Male   Houston
## 6 Sophia     27 Female Miami
```

```
# c package data.table contains fread which is faster
# https://www.r-bloggers.com/2021/12/import-csv-files-into-r-step-by-step-guide/

install.packages("data.table")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(data.table)
print("fread() from data.table")
```

```
## [1] "fread() from data.table"
```

```
csv3 <- data.table::fread("testdata.csv")
csv3
```

```
##       Name Age Gender          City
## 1:    John  25   Male      New York
## 2:   Alice  30 Female San Francisco
## 3: Michael  22   Male   Los Angeles
## 4:   Emily  28 Female       Chicago
## 5:   David  35   Male       Houston
## 6:  Sophia  27 Female         Miami
```

```
# d read.delim()
csv4 <- utils::read.delim("testdata.csv", sep=",")
csv4
```

```
##      Name Age Gender          City
## 1    John  25   Male      New York
## 2   Alice  30 Female San Francisco
## 3 Michael  22   Male   Los Angeles
## 4   Emily  28 Female       Chicago
## 5   David  35   Male       Houston
## 6  Sophia  27 Female         Miami
```

```
#Becoming Visual Exercise 6.1
# 1 Getting

# 1 Getting to know the data
# a. Import the data (http://becomingvisual.com/rfundamentals/summer_winter_olympics.csv)
medals <- readr::read_csv("http://becomingvisual.com/rfundamentals/summer_winter_olympics.csv")
```

```
## `curl` package not installed, falling back to using `url()`
## New names:
## Rows: 146 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Team..IOC.code.
## dbl (16): ...1, X..Summer, X, X.1, X.2, Total, X..Winter, X.3, X.4, X.5, Tot...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# b. View the data
medals
```

```
## # A tibble: 146 x 17
##     ...1 Team..IOC.code. X..Summer     X   X.1   X.2 Total X..Winter   X.3   X.4
##    <dbl> <chr>                <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>
##  1     1  Afghanistan (~        13     0     0     2     2         0     0     0
##  2     2  Algeria (ALG)         12     5     2     8    15         3     0     0
##  3     3  Argentina (AR~        23    18    24    28    70        18     0     0
##  4     4  Armenia (ARM)          5     1     2     9    12         6     0     0
##  5     5  Australasia (~         2     3     4     5    12         0     0     0
##  6     6  Australia (AU~        25   138   153   177   468        18     5     3
##  7     7  Austria (AUT)         26    18    33    35    86        22    59    78
##  8     8  Azerbaijan (A~         5     6     5    15    26         5     0     0
##  9     9  Bahamas (BAH)         15     5     2     5    12         0     0     0
## 10    10  Bahrain (BRN)          8     0     0     1     1         0     0     0
## # i 136 more rows
## # i 7 more variables: X.5 <dbl>, Total.1 <dbl>, X..Games <dbl>, X.6 <dbl>,
## #   X.7 <dbl>, X.8 <dbl>, Combined.total <dbl>
```

```
# c. Look at column names
summary(medals)
```

```
##       ...1        Team..IOC.code.      X..Summer           X
##  Min.   :  1.00   Length:146         Min.   : 1.00   Min.   :  0.00
##  1st Qu.: 37.25   Class :character   1st Qu.: 8.00   1st Qu.:  0.00
##  Median : 73.50   Mode  :character   Median :13.00   Median :  3.00
##  Mean   : 73.50                      Mean   :13.38   Mean   : 32.94
##  3rd Qu.:109.75                      3rd Qu.:18.00   3rd Qu.: 23.00
##  Max.   :146.00                      Max.   :27.00   Max.   :976.00
##       X.1              X.2              Total           X..Winter
##  Min.   :  0.00   Min.   :  0.00   Min.   :   0.00   Min.   : 0.000
##  1st Qu.:  1.00   1st Qu.:  1.00   1st Qu.:   2.00   1st Qu.: 0.000
##  Median :  4.00   Median :  6.00   Median :  12.00   Median : 5.000
##  Mean   : 32.71   Mean   : 35.13   Mean   : 100.78   Mean   : 6.596
##  3rd Qu.: 26.75   3rd Qu.: 28.75   3rd Qu.:  85.25   3rd Qu.:10.000
##  Max.   :758.00   Max.   :666.00   Max.   :2400.00   Max.   :22.000
##       X.3               X.4               X.5             Total.1
##  Min.   :  0.000   Min.   :  0.000   Min.   :  0.000   Min.   :  0.00
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.00
##  Median :  0.000   Median :  0.000   Median :  0.000   Median :  0.00
##  Mean   :  6.568   Mean   :  6.555   Mean   :  6.493   Mean   : 19.62
##  3rd Qu.:  0.750   3rd Qu.:  1.750   3rd Qu.:  1.000   3rd Qu.:  4.75
##  Max.   :118.000   Max.   :111.000   Max.   :100.000   Max.   :329.00
##     X..Games          X.6               X.7              X.8
##  Min.   : 1.00   Min.   :   0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:11.00   1st Qu.:   0.00   1st Qu.:  1.00   1st Qu.:  1.00
##  Median :15.00   Median :   3.00   Median :  4.00   Median :  6.50
##  Mean   :19.98   Mean   :  39.51   Mean   : 39.27   Mean   : 41.62
##  3rd Qu.:26.00   3rd Qu.:  24.50   3rd Qu.: 28.00   3rd Qu.: 29.00
##  Max.   :49.00   Max.   :1072.00   Max.   :860.00   Max.   :749.00
##  Combined.total
##  Min.   :   1.00
##  1st Qu.:   2.25
##  Median :  12.00
```

```
##  Mean   : 120.40
##  3rd Qu.:  87.75
##  Max.   :2681.00
```

```
names(medals)
```

```
##  [1] "...1"            "Team..IOC.code." "X..Summer"       "X"
##  [5] "X.1"             "X.2"             "Total"           "X..Winter"
##  [9] "X.3"             "X.4"             "X.5"             "Total.1"
## [13] "X..Games"        "X.6"             "X.7"             "X.8"
## [17] "Combined.total"
```

```
# d. Look at dimension of data (rows and columns)
dim(medals)
```

```
## [1] 146  17
```

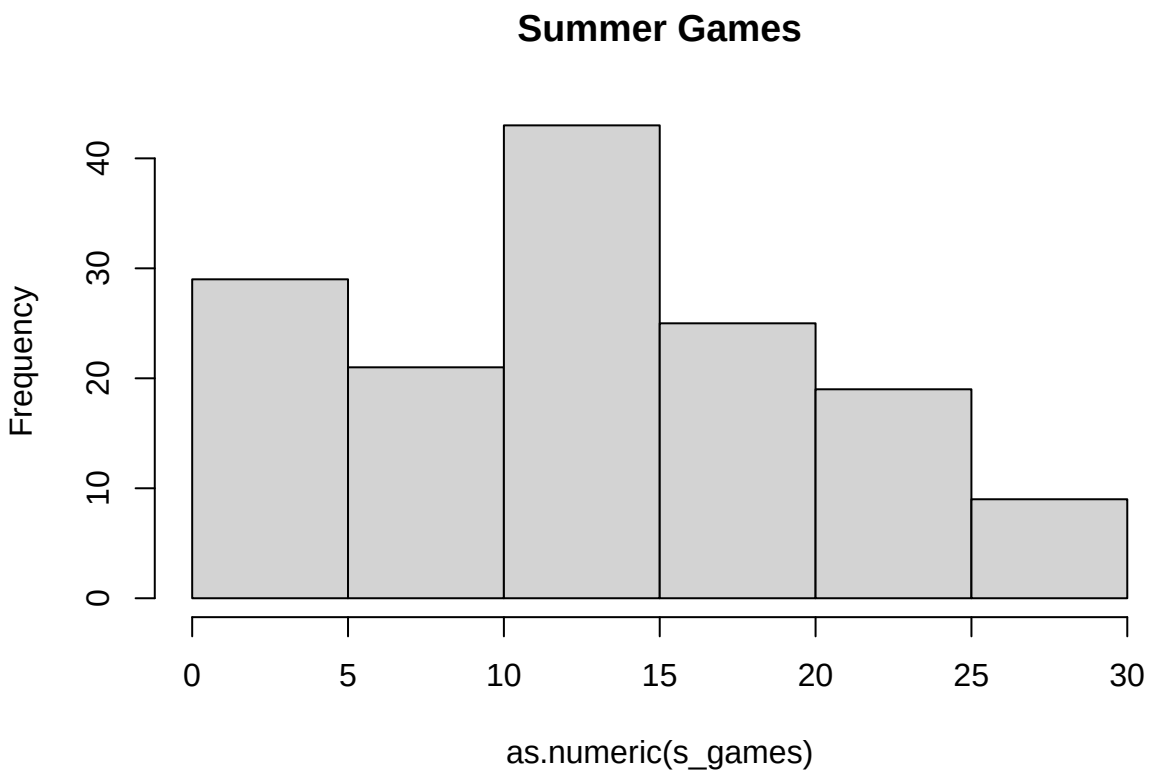
```
# 2. Dealing with Data
# a. Look at the column names and change names to more meaningful names.
names(medals) = c("NA", "country", "s_games", "s_gold", "s_silver", "s_bronze", "s_total", "w_games", "
attach(medals)
# b. The data represent, in order:
#   b.1. country
#   b.2. number of summer games played, gold, silver, bronze, total,
#   b.3 number of winter games played, gold, silver, bronze and total, total
#   b.4 total (Winter + Summer) games, gold, silver, bronze, total
# 3. Summary
# a. use table() to find frequency of total summer games played
table(s_games)
```

```
## s_games
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  3  2  6  1 17  3  1  7  8  2  7 10 13  5  8 11  4  2  3  5  4  5  3  2  5  5
## 27
##  4
```

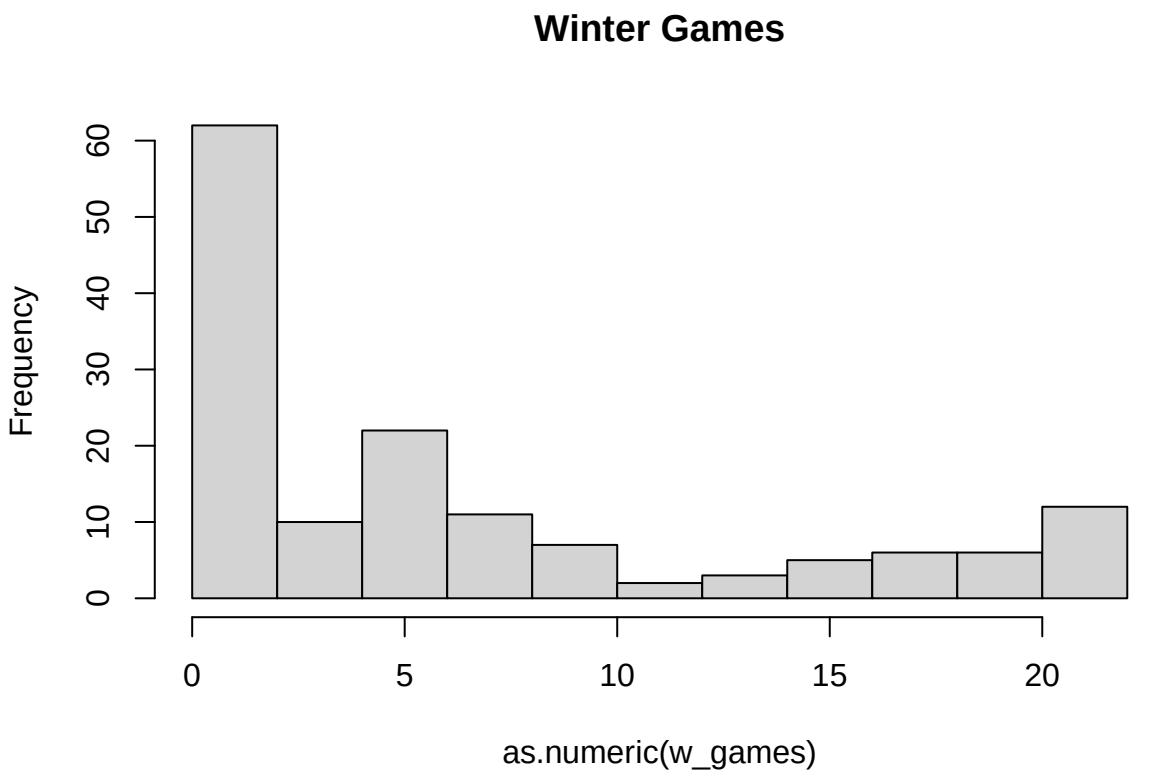
```
# b. explore the data with other variables
table(sw_total)
```

```
## sw_total
##    1    2    3    4    5    6    7    8    9   10   12   13   15   17   18   19
##   26   11    7   10    1    2    4    3    3    3    4    1    1    1    1    1
##   21   22   23   24   25   26   27   28   29   34   40   45   49   59   60   62
##    3    1    2    2    1    4    1    1    1    2    1    1    1    1    1    1
##   67   68   70   76   86   87   88   91  100  108  110  122  133  135  137  147
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  168  180  208  220  243  291  296  302  304  323  376  443  448  463  477  480
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  482  519  521  526  627  663  780  782  806 1204 2681
##    1    1    1    1    1    1    1    1    1    1    1
```

```
# 4. Graphs
# a. do histogram of summer games (total)
hist(as.numeric(s_games),main="Summer Games")
```

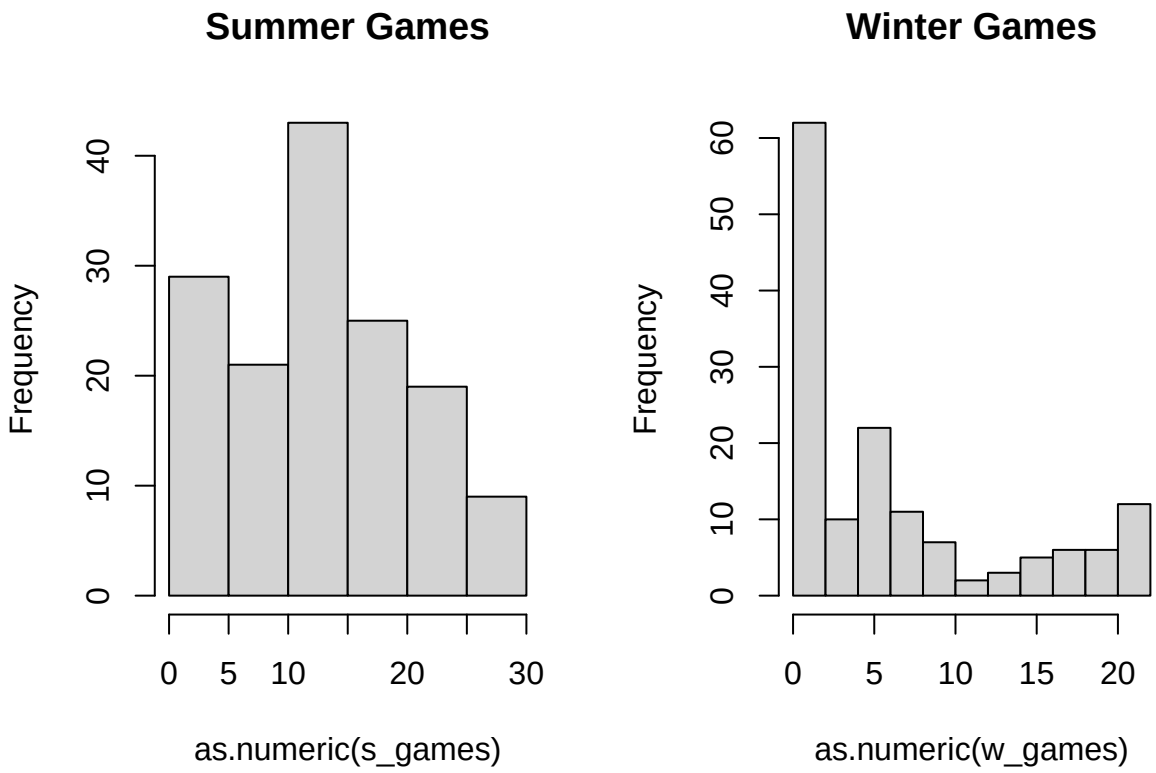



```
# b. do histogram of winter games (total)
hist(as.numeric(w_games), main="Winter Games")
```

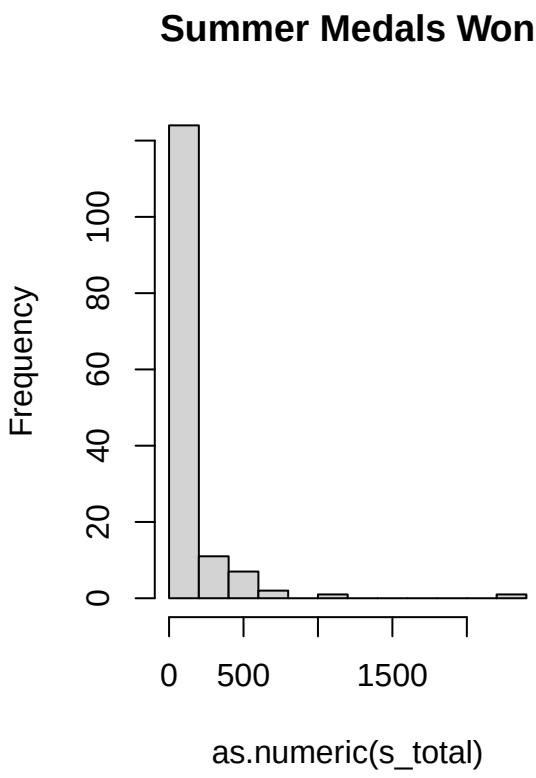


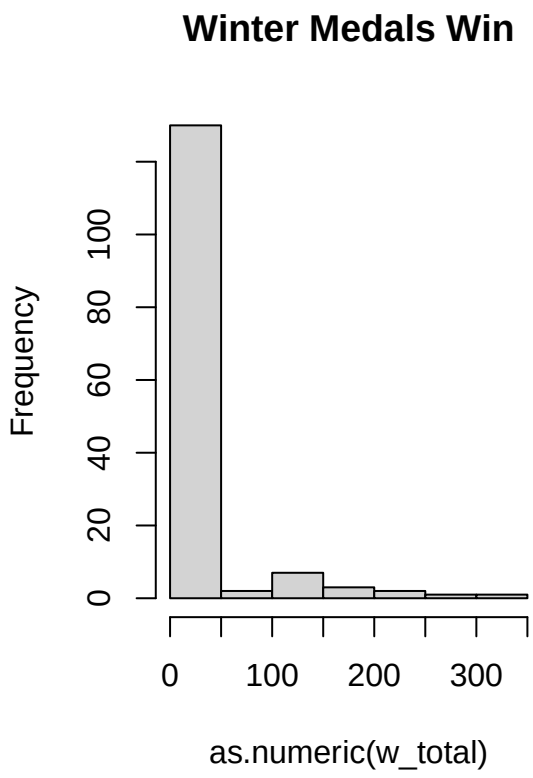

```
# c. put above two histograms on one page
par(mfrow=c(1,2))
```

```
hist(as.numeric(s_games),main="Summer Games")
hist(as.numeric(w_games), main="Winter Games")
```

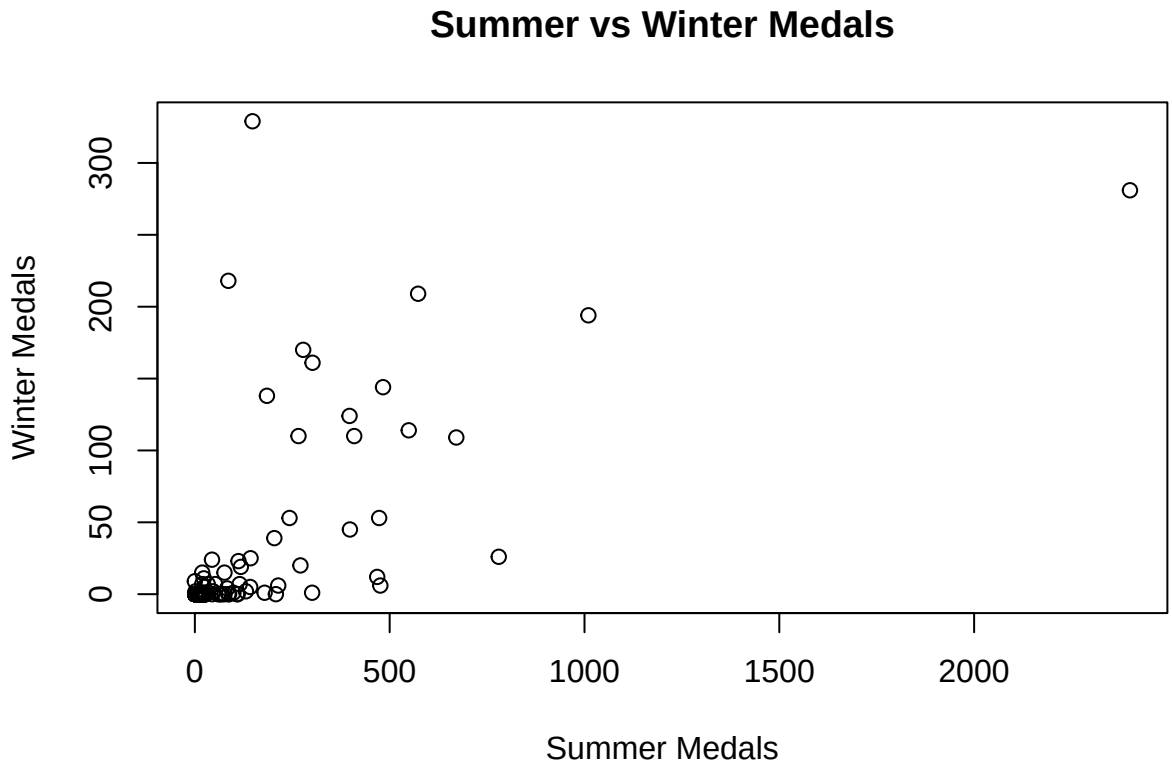



```
# d. do two histograms on one page: total summer, total winter medals won
par(mfrow=c(1,2))
hist(as.numeric(s_total),main="Summer Medals Won")
hist(as.numeric(w_total), main="Winter Medals Win")
```

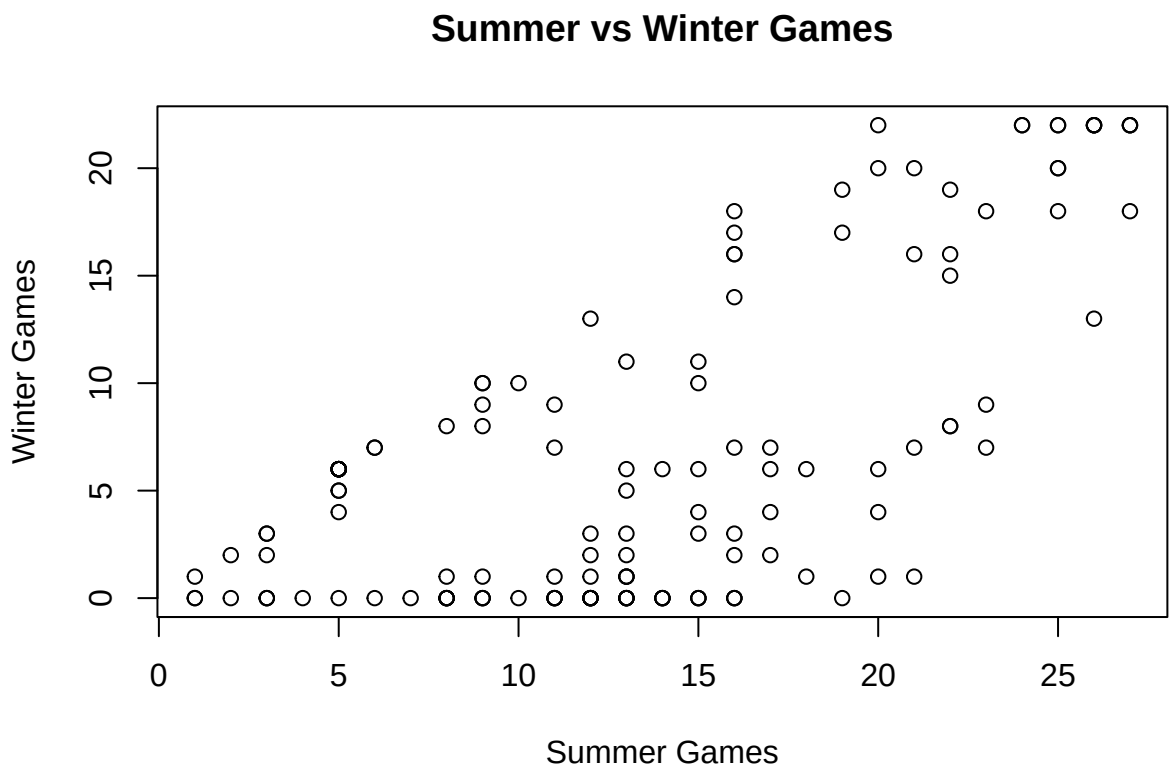


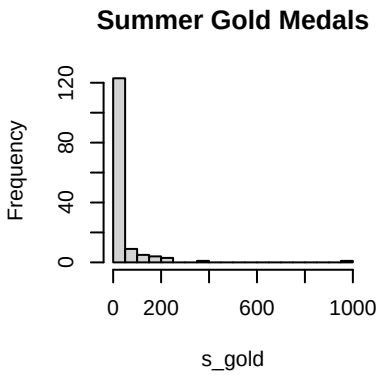

```
# e. is there a correlation between number of medals given out in winter and summer? (do plot)
par(mfrow=c(1,1))
plot(s_total, w_total, xlab="Summer Medals", ylab="Winter Medals", main="Summer vs Winter Medals")
```

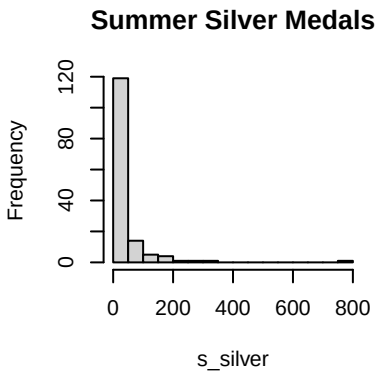


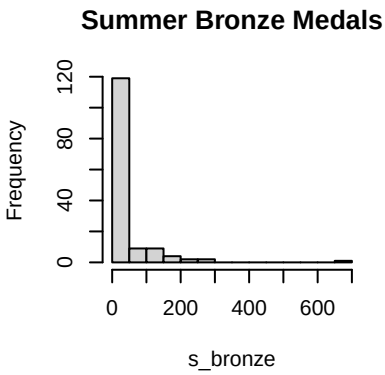
```r
# f. how about number of games each country competes in. Is there correlation between winter and summer
plot(s_games, w_games, xlab="Summer Games", ylab="Winter Games", main="Summer vs Winter Games")
```

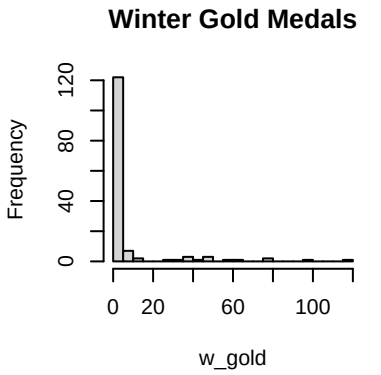


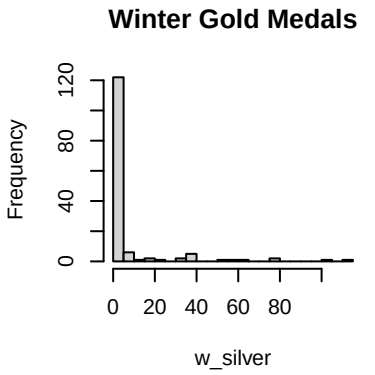

```r
# g. look at distribution of each of the types of medals, by season (6 histograms on one page)
par(mfrow=c(2,3))
bins <- 20
hist(s_gold, breaks=bins, main="Summer Gold Medals")
hist(s_silver, breaks=bins, main="Summer Silver Medals")
hist(s_bronze, breaks=bins, main="Summer Bronze Medals")
hist(w_gold, breaks=bins, main="Winter Gold Medals")
hist(w_silver, breaks=bins, main="Winter Gold Medals")
hist(w_bronze, breaks=bins, main="Winter Gold Medals")
```

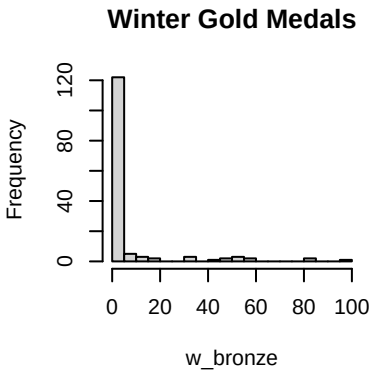


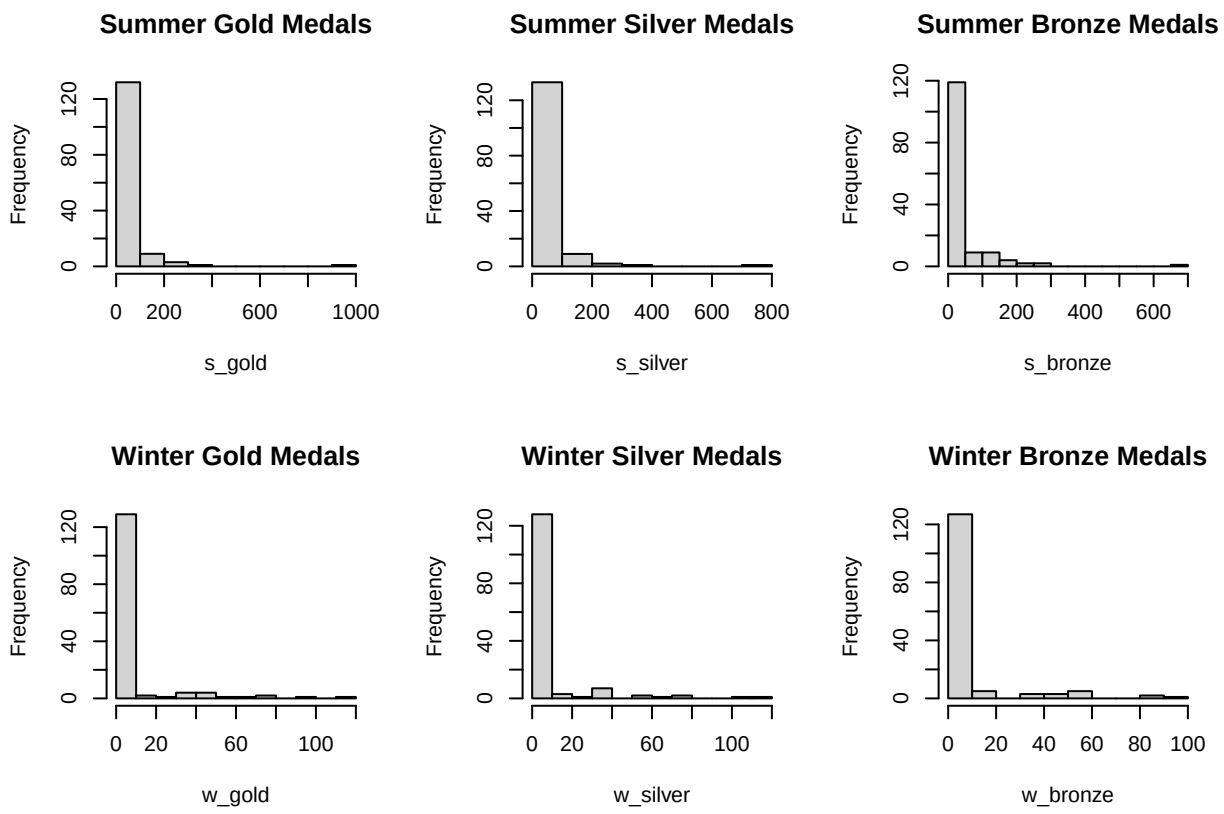



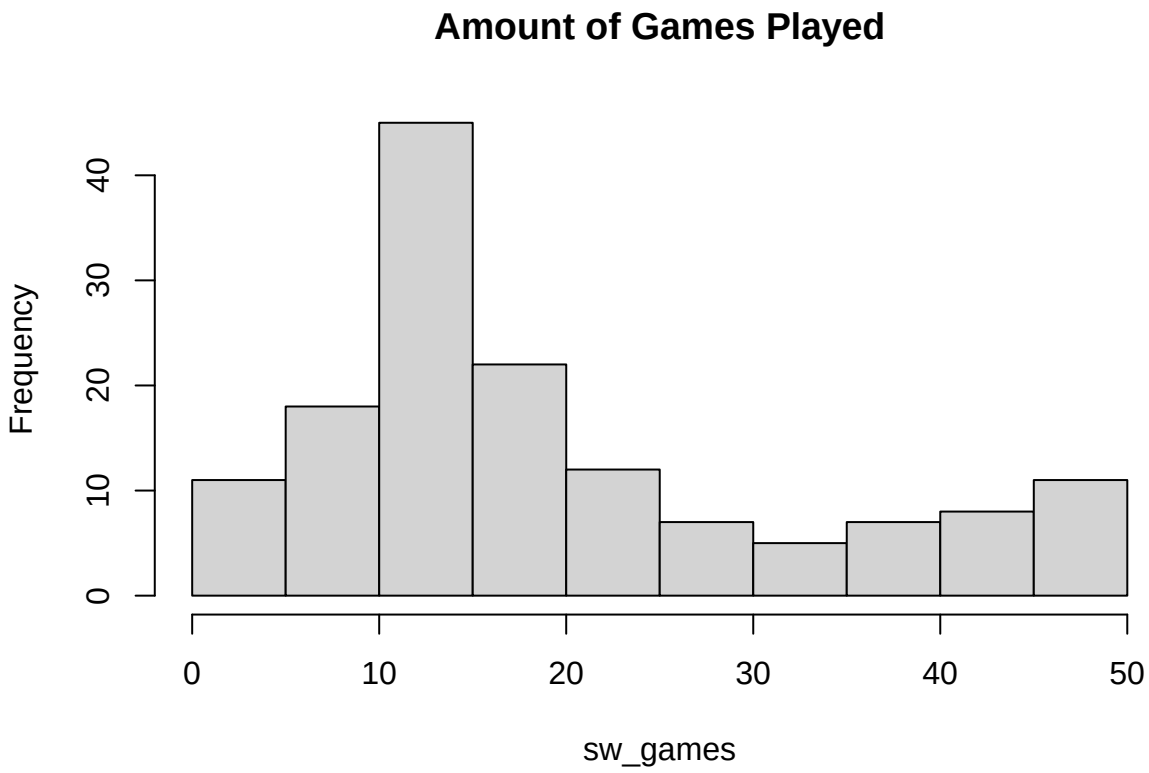




```
# h. redo g with different number of bins (10 instead of 20)
par(mfrow=c(2,3))
bins <- 10
hist(s_gold, breaks=bins, main="Summer Gold Medals")
hist(s_silver, breaks=bins, main="Summer Silver Medals")
hist(s_bronze, breaks=bins, main="Summer Bronze Medals")
hist(w_gold, breaks=bins, main="Winter Gold Medals")
hist(w_silver, breaks=bins, main="Winter Silver Medals")
hist(w_bronze, breaks=bins, main="Winter Bronze Medals")
```




```
# i. explore data on your own
par(mfrow=c(1,1))
hist(sw_games, main="Amount of Games Played")
```



```r
#Becoming Visual Exercise 6.2

# Merge the columns for the year 2016 for GDP, Life Expectancy, and Employment into a new data frame an
gdp <- readr::read_csv("http://becomingvisual.com/rfundamentals/gdp.csv")
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 264 Columns: 60
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Country Name, Country Code
## dbl (58): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
life_expectancy <- readr::read_csv("http://becomingvisual.com/rfundamentals/life_expectancy.csv")
```

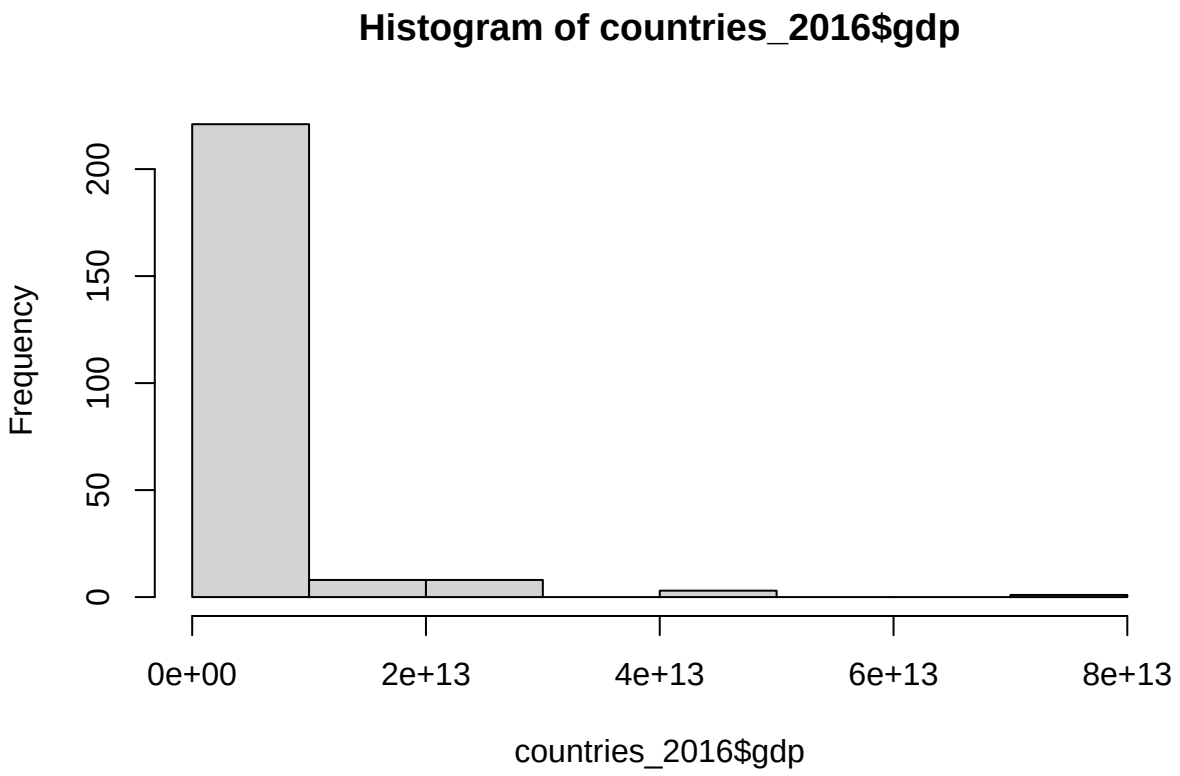
```
## `curl` package not installed, falling back to using `url()`
## Rows: 264 Columns: 60
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Country Name, Country Code
## dbl (57): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
## lgl  (1): 2017
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
employment <- readr::read_csv("http://becomingvisual.com/rfundamentals/employment.csv")
```

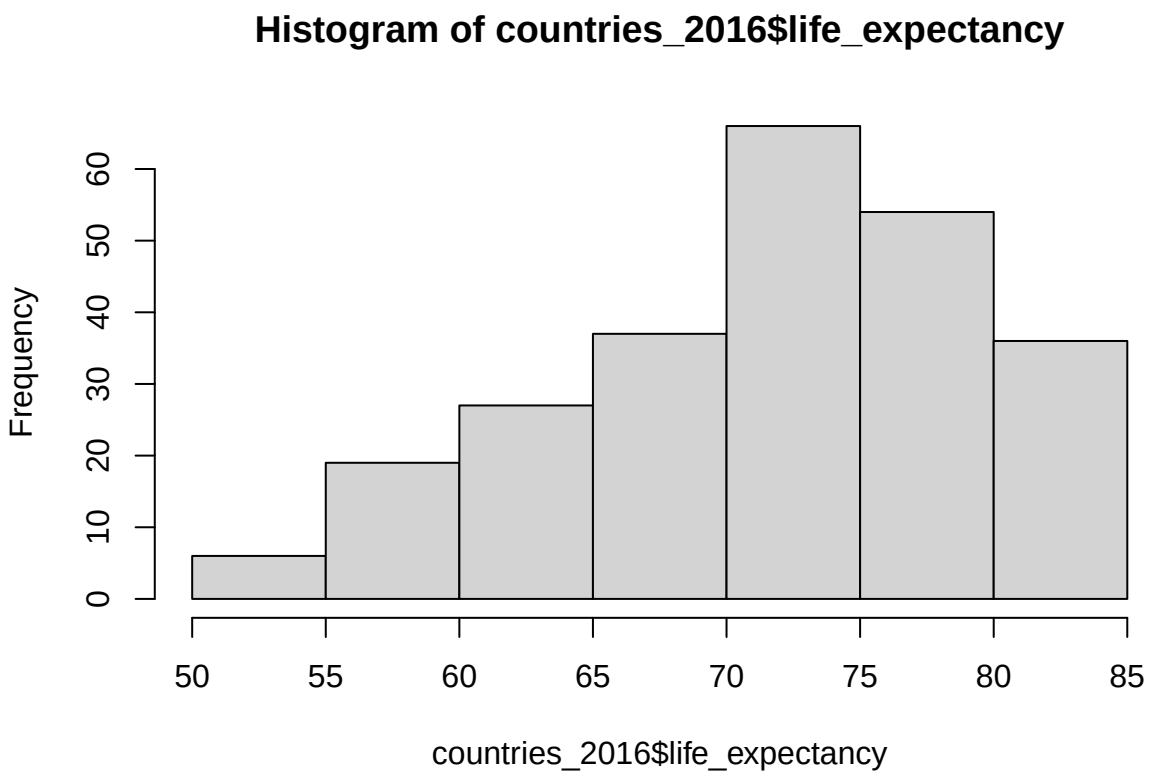
```
## `curl` package not installed, falling back to using `url()`
## Rows: 264 Columns: 60
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Country Name, Country Code
## dbl (27): 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, ...
## lgl (31): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
countries_2016 <- data.frame(gdp$"Country Name", gdp$"2016", life_expectancy$"2016", employment$"2016")

# Rename the appropriate columns to "country", "gdp", "life_expectancy", and "employment".
names(countries_2016) <- c("country", "gdp", "life_expectancy", "employment")

# Convert the employment number to percentages by dividing by 100
countries_2016$employment <- countries_2016$employment / 100

# Then round life expectancy to zero decimals and employment to two decimals
countries_2016$life_expectancy <- round(countries_2016$life_expectancy, digits = 0)
countries_2016$employment <- round(countries_2016$employment, digits = 2)

# Create a frequency table for each variable
gdp_ft <- table(countries_2016$gdp)
life_expectancy_ft <- table(countries_2016$life_expectancy)
```

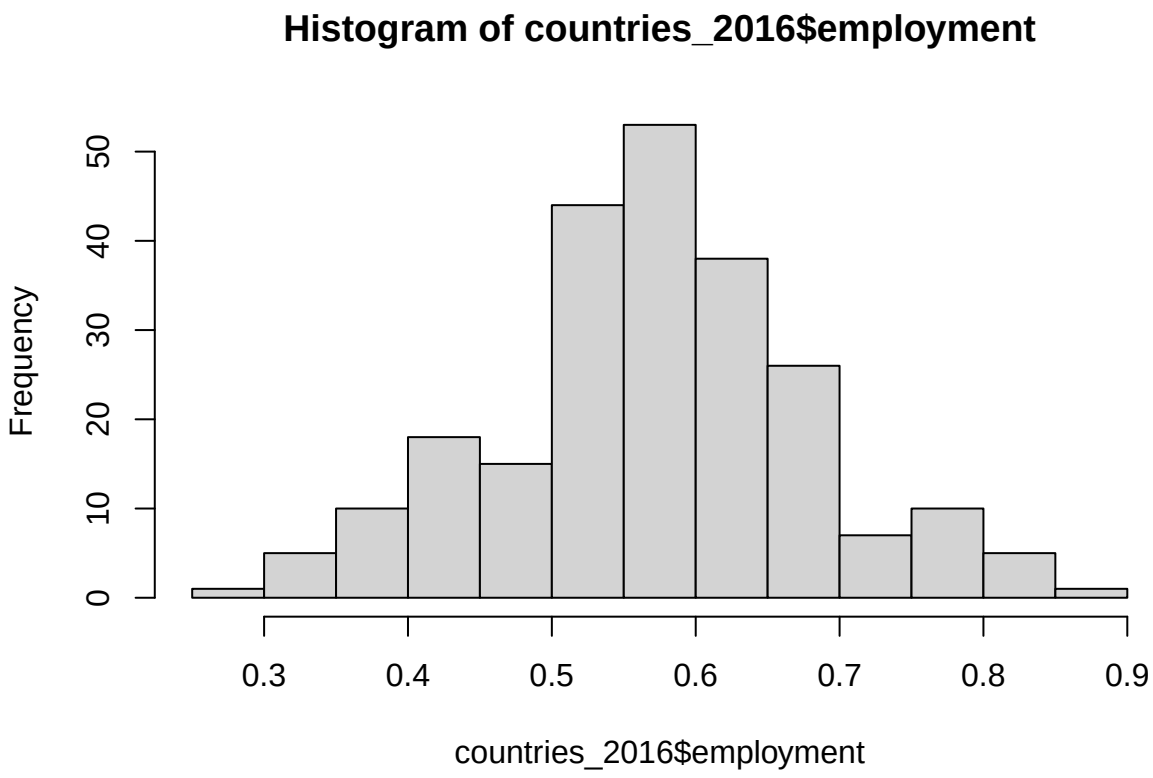
```r
employment_ft <- table(countries_2016$employment)

# Draw histograms for each variable
hist(countries_2016$gdp)
```




```r
hist(countries_2016$life_expectancy)
```



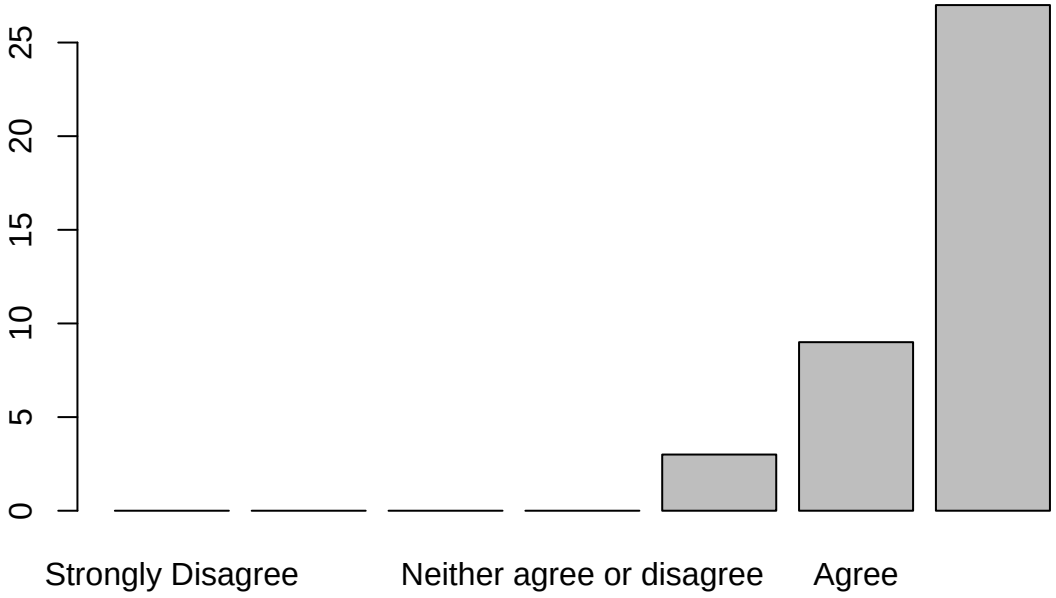

```
hist(countries_2016$employment)
```



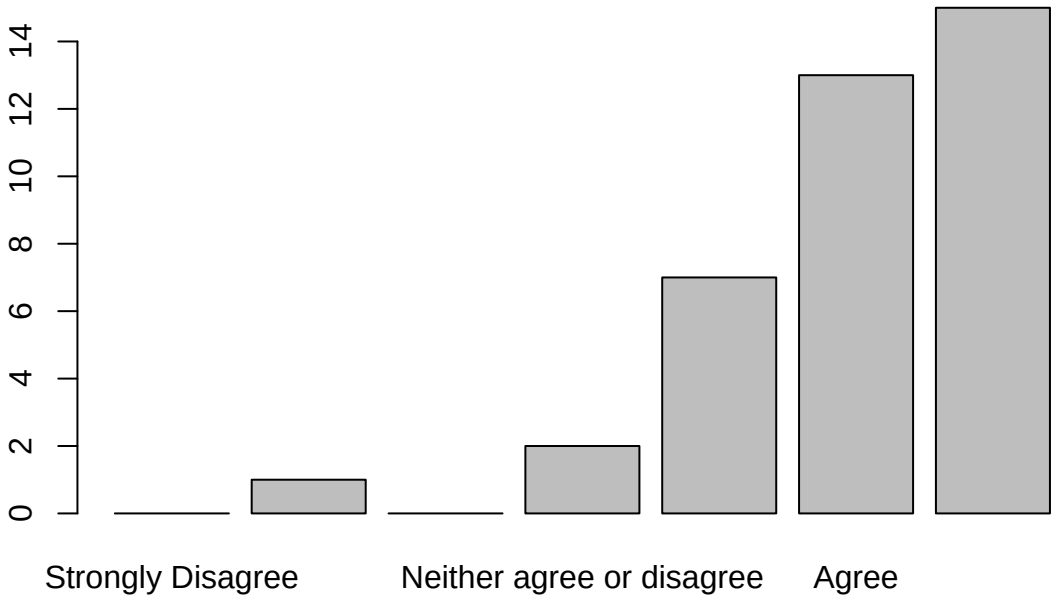

```
#becoming Visual Assignment 6
data <- readr::read_csv("http://becomingvisual.com/rfundamentals/undergrad.csv")
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 39 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): Timestamp, The following tools are important to my future career. [...
## dbl (3): How likely are you to take another information systems course at St...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

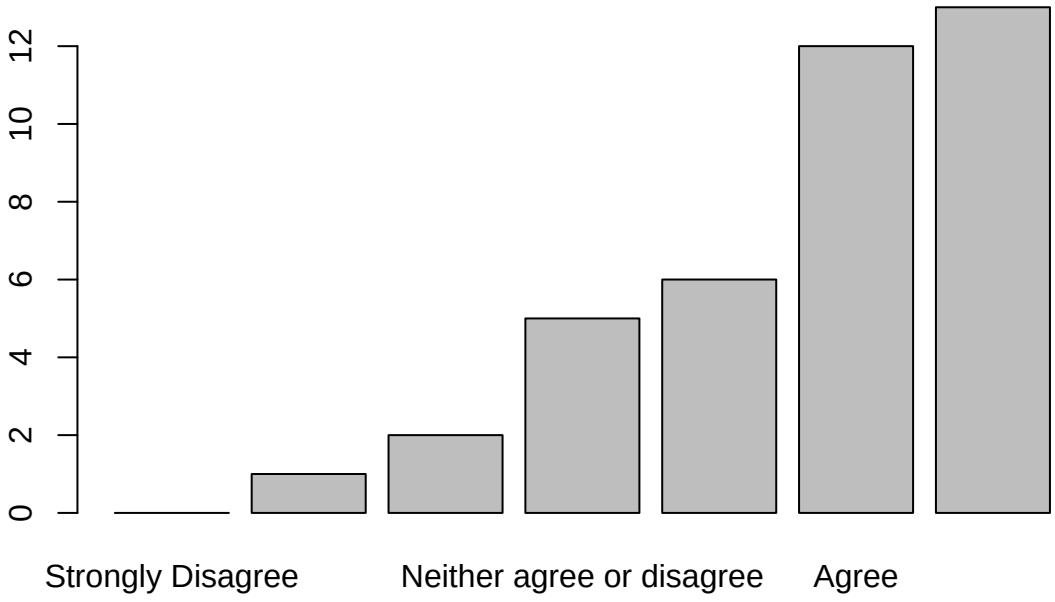
```
# Create ordered factor variables
data$`The following tools are important to my future career. [Excel]` <-  factor(data$`The following too
data$`The following tools are important to my future career. [Statistics]` <- factor(data$`The followin
data$`The following tools are important to my future career. [A programming language]` <- factor(data$`
# Draw histograms for the ordered factor variables
barplot(table(data$`The following tools are important to my future career. [Excel]`))
```




```
barplot(table(data$`The following tools are important to my future career. [Statistics]`))
```




```
barplot(table(data$`The following tools are important to my future career. [A programming language]`))
```




```
# For loop vs While loop vs Apply functions

# For loop
numbers <- c(1, 2, 3, 4, 5)
squared_numbers <- numeric(length(numbers))

for (i in seq_along(numbers)) {
  squared_numbers[i] <- numbers[i] ^ 2
}

print(squared_numbers)
```

```
## [1]  1  4  9 16 25
```

```
print("For loops are preferred if you want to iterate over a sequence or set of items. It is often also
```

```
## [1] "For loops are preferred if you want to iterate over a sequence or set of items. It is often also
```

```
# While loop
numbers <- c(1, 2, 3, 4, 5)
squared_numbers <- numeric(length(numbers))
i <- 1

while (i <= length(numbers)) {
  squared_numbers[i] <- numbers[i] ^ 2
  i <- i + 1
}

print(squared_numbers)
```

```
## [1]  1  4  9 16 25
```

```
print("While loops can be complicated if trying to replicate For loops since you will have to declare va
```

```
## [1] "While loops can be complicated if trying to replicate For loops since you will have to declare v
```

```
# Apply function
numbers <- c(1, 2, 3, 4, 5)
squared_numbers <- lapply(numbers, function(x) x^2)

print(unlist(squared_numbers))
```

```
## [1]  1  4  9 16 25
```

```
print("Apply functions are the most elegant way of applying a standard function to each member of a list
```

```
## [1] "Apply functions are the most elegant way of applying a standard function to each member of a lis
```

```
#Becoming Visual Exercise 7.1
utils::head(attitude)
```

```
##   rating complaints privileges learning raises critical advance
## 1     43         51         30       39     61       92      45
## 2     63         64         51       54     63       73      47
## 3     71         70         68       69     76       86      48
## 4     61         63         45       47     54       84      35
## 5     81         78         56       66     71       83      47
## 6     43         55         49       44     54       49      34
```

```
summary(attitude)
```

```
##      rating        complaints     privileges       learning         raises
##  Min.   :40.00   Min.   :37.0   Min.   :30.00   Min.   :34.00   Min.   :43.00
##  1st Qu.:58.75   1st Qu.:58.5   1st Qu.:45.00   1st Qu.:47.00   1st Qu.:58.25
##  Median :65.50   Median :65.0   Median :51.50   Median :56.50   Median :63.50
##  Mean   :64.63   Mean   :66.6   Mean   :53.13   Mean   :56.37   Mean   :64.63
##  3rd Qu.:71.75   3rd Qu.:77.0   3rd Qu.:62.50   3rd Qu.:66.75   3rd Qu.:71.00
##  Max.   :85.00   Max.   :90.0   Max.   :83.00   Max.   :75.00   Max.   :88.00
##     critical        advance
##  Min.   :49.00   Min.   :25.00
##  1st Qu.:69.25   1st Qu.:35.00
##  Median :77.50   Median :41.00
```

```
##  Mean   :74.77   Mean   :42.93
##  3rd Qu.:80.00   3rd Qu.:47.75
##  Max.   :92.00   Max.   :72.00
```

```
# make function
getSummary <- function(data){
  for(i in 1:ncol(data)){
    print(names(data[i]))
    print(paste("mean:", mean(data[,i])))
    print(paste("median:", median(data[,i])))
    print(paste("min:", min(data[,i])))
    print(paste("max:", max(data[,i])))
    print("")
  }
}
# run function
getSummary(attitude)
```

```
## [1] "rating"
## [1] "mean: 64.6333333333333"
## [1] "median: 65.5"
## [1] "min: 40"
## [1] "max: 85"
## [1] ""
## [1] "complaints"
## [1] "mean: 66.6"
## [1] "median: 65"
## [1] "min: 37"
## [1] "max: 90"
## [1] ""
## [1] "privileges"
## [1] "mean: 53.1333333333333"
## [1] "median: 51.5"
## [1] "min: 30"
## [1] "max: 83"
## [1] ""
## [1] "learning"
## [1] "mean: 56.3666666666667"
## [1] "median: 56.5"
## [1] "min: 34"
## [1] "max: 75"
## [1] ""
## [1] "raises"
## [1] "mean: 64.6333333333333"
## [1] "median: 63.5"
## [1] "min: 43"
## [1] "max: 88"
## [1] ""
## [1] "critical"
## [1] "mean: 74.7666666666667"
## [1] "median: 77.5"
## [1] "min: 49"
## [1] "max: 92"
## [1] ""
## [1] "advance"
```

```
## [1] "mean: 42.9333333333333"
## [1] "median: 41"
## [1] "min: 25"
## [1] "max: 72"
## [1] ""
```

```
#Becoming Visual Exercise 7.2
# import data
airbnb <- readr::read_csv("http://becomingvisual.com/rfundamentals/airbnb.csv")
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 48895 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): name, host_name, neighbourhood_group, neighbourhood, room_type
## dbl  (10): id, host_id, latitude, longitude, price, minimum_nights, number_o...
## date  (1): last_review
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# 1. make function: Create a new function called checkforna that checks to see if a give value is NA an
checkforna <- function(data){
  columns <- names(attitude)
  for(i in 1:ncol(data)){
    for(j in 1:nrow(data)){
      if(is.na(data[j,i])){
        print(paste("Row",j,", Column", columns[i]))
      }
    }
  }
}
#checkforna(airbnb)

# 2. Using a for loop and if / else statement, count the number of AirBnbs that are in a particular nei

count <- 0
for(i in 1:nrow(airbnb)){
  if(airbnb$neighbourhood[i]=="Greenwich Village" | airbnb$neighbourhood[i]=="West Village"){
    count <- count + 1
  }
}
print(paste("Number of AirBnbs in Greenwich Village and West Village neighborhoods:",count))
```

```
## [1] "Number of AirBnbs in Greenwich Village and West Village neighborhoods: 1160"
```

```
#Becoming Visual Assignment 7
# import data
airbnb <- readr::read_csv("http://becomingvisual.com/rfundamentals/airbnb.csv")
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 48895 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): name, host_name, neighbourhood_group, neighbourhood, room_type
## dbl  (10): id, host_id, latitude, longitude, price, minimum_nights, number_o...
```

```
## date  (1): last_review
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# 1.Create a set of functions that compute specific metrics by neighborhood:
#   a. average_number_of_reviews
average_number_of_reviews <- function(data, neighbourhood){
  count <- 0
  mySum <- 0
  for(j in 1:nrow(data)){
    if(data[j,"neighbourhood"][[1]] == neighbourhood){
      count <- count + 1
      mySum <- mySum + data[j, "number_of_reviews"][[1]]
    }

  }
  return(mySum/count)
}
average_number_of_reviews(airbnb, "Greenpoint")
```

```
## [1] 17.42511
```

```
#   b. average_price
average_price <- function(data, neighbourhood){
  count <- 0
  mySum <- 0
  for(j in 1:nrow(data)){
    if(data[j,"neighbourhood"][[1]] == neighbourhood){
      count <- count + 1
      mySum <- mySum + data[j, "price"][[1]]
    }

  }
  return(mySum/count)
}
average_price(airbnb, "Greenpoint")
```

```
## [1] 144.8224
```

```
# per neighbourhood
number_reviews_and_price_per_neighbourhood <- function(data){
  neighbourhoods <- unique(data$neighbourhood)
  for(i in neighbourhoods){
    print(i)
    print(paste("Average Number of Reviews:", average_number_of_reviews(data,i)))
    print(paste("Average Price:", average_price(data,i)))
    print("")
  }
}
number_reviews_and_price_per_neighbourhood(airbnb)
```

```
## [1] "Kensington"
## [1] "Average Number of Reviews: 16.9828571428571"
## [1] "Average Price: 92.8857142857143"
## [1] ""
```

```
## [1] "Midtown"
## [1] "Average Number of Reviews: 12.5851132686084"
## [1] "Average Price: 282.719093851133"
## [1] ""
## [1] "Harlem"
## [1] "Average Number of Reviews: 28.5786305492852"
## [1] "Average Price: 118.974040632054"
## [1] ""
## [1] "Clinton Hill"
## [1] "Average Number of Reviews: 25.5"
## [1] "Average Price: 181.893356643357"
## [1] ""
## [1] "East Harlem"
## [1] "Average Number of Reviews: 32.6284691136974"
## [1] "Average Price: 133.198746642793"
## [1] ""
## [1] "Murray Hill"
## [1] "Average Number of Reviews: 9.35051546391753"
## [1] "Average Price: 220.958762886598"
## [1] ""
## [1] "Bedford-Stuyvesant"
## [1] "Average Number of Reviews: 29.7124394184168"
## [1] "Average Price: 107.678244480345"
## [1] ""
## [1] "Hell's Kitchen"
## [1] "Average Number of Reviews: 25.6521961184883"
## [1] "Average Price: 204.79417773238"
## [1] ""
## [1] "Upper West Side"
## [1] "Average Number of Reviews: 18.2942668696093"
## [1] "Average Price: 210.91831557585"
## [1] ""
## [1] "Chinatown"
## [1] "Average Number of Reviews: 27.0135869565217"
## [1] "Average Price: 161.497282608696"
## [1] ""
## [1] "South Slope"
## [1] "Average Number of Reviews: 36.637323943662"
## [1] "Average Price: 146.725352112676"
## [1] ""
## [1] "West Village"
## [1] "Average Number of Reviews: 19.3815104166667"
## [1] "Average Price: 267.682291666667"
## [1] ""
## [1] "Williamsburg"
## [1] "Average Number of Reviews: 21.7926020408163"
## [1] "Average Price: 143.802806122449"
## [1] ""
## [1] "Fort Greene"
## [1] "Average Number of Reviews: 21.6932515337423"
## [1] "Average Price: 151.374233128834"
## [1] ""
## [1] "Chelsea"
## [1] "Average Number of Reviews: 21.240790655885"
```

```
## [1] "Average Price: 249.738544474394"
## [1] ""
## [1] "Crown Heights"
## [1] "Average Number of Reviews: 23.2787723785166"
## [1] "Average Price: 112.480179028133"
## [1] ""
## [1] "Park Slope"
## [1] "Average Number of Reviews: 28.9288537549407"
## [1] "Average Price: 176.312252964427"
## [1] ""
## [1] "Windsor Terrace"
## [1] "Average Number of Reviews: 27.5414012738853"
## [1] "Average Price: 138.993630573248"
## [1] ""
## [1] "Inwood"
## [1] "Average Number of Reviews: 18.6111111111111"
## [1] "Average Price: 88.8968253968254"
## [1] ""
## [1] "East Village"
## [1] "Average Number of Reviews: 24.1068537506746"
## [1] "Average Price: 186.083108472747"
## [1] ""
## [1] "Greenpoint"
## [1] "Average Number of Reviews: 17.4251121076233"
## [1] "Average Price: 144.822421524664"
## [1] ""
## [1] "Bushwick"
## [1] "Average Number of Reviews: 21.3038539553753"
## [1] "Average Price: 84.8004056795132"
## [1] ""
## [1] "Flatbush"
## [1] "Average Number of Reviews: 20.5909822866345"
## [1] "Average Price: 92.2125603864734"
## [1] ""
## [1] "Lower East Side"
## [1] "Average Number of Reviews: 26.5214050493963"
## [1] "Average Price: 186.312843029638"
## [1] ""
## [1] "Prospect-Lefferts Gardens"
## [1] "Average Number of Reviews: 26.2635514018692"
## [1] "Average Price: 110.401869158879"
## [1] ""
## [1] "Long Island City"
## [1] "Average Number of Reviews: 22.8230912476723"
## [1] "Average Price: 127.465549348231"
## [1] ""
## [1] "Kips Bay"
## [1] "Average Number of Reviews: 12.4191489361702"
## [1] "Average Price: 202.408510638298"
## [1] ""
## [1] "SoHo"
## [1] "Average Number of Reviews: 20.2094972067039"
## [1] "Average Price: 287.103351955307"
## [1] ""
```

```
## [1] "Upper East Side"
## [1] "Average Number of Reviews: 17.622914349277"
## [1] "Average Price: 188.948275862069"
## [1] ""
## [1] "Prospect Heights"
## [1] "Average Number of Reviews: 30.4621848739496"
## [1] "Average Price: 173.372549019608"
## [1] ""
## [1] "Washington Heights"
## [1] "Average Number of Reviews: 19.0889877641824"
## [1] "Average Price: 89.6106785317019"
## [1] ""
## [1] "Woodside"
## [1] "Average Number of Reviews: 21.4255319148936"
## [1] "Average Price: 85.0978723404255"
## [1] ""
## [1] "Brooklyn Heights"
## [1] "Average Number of Reviews: 16.5909090909091"
## [1] "Average Price: 209.064935064935"
## [1] ""
## [1] "Carroll Gardens"
## [1] "Average Number of Reviews: 25.3090128755365"
## [1] "Average Price: 175.914163090129"
## [1] ""
## [1] "Gowanus"
## [1] "Average Number of Reviews: 31.2105263157895"
## [1] "Average Price: 158.801619433198"
## [1] ""
## [1] "Flatlands"
## [1] "Average Number of Reviews: 22.1566265060241"
## [1] "Average Price: 126.433734939759"
## [1] ""
## [1] "Cobble Hill"
## [1] "Average Number of Reviews: 29.1717171717172"
## [1] "Average Price: 211.929292929293"
## [1] ""
## [1] "Flushing"
## [1] "Average Number of Reviews: 34.7840375586854"
## [1] "Average Price: 93.5140845070423"
## [1] ""
## [1] "Boerum Hill"
## [1] "Average Number of Reviews: 24.2655367231638"
## [1] "Average Price: 176.135593220339"
## [1] ""
## [1] "Sunnyside"
## [1] "Average Number of Reviews: 22.2314049586777"
## [1] "Average Price: 84.8650137741047"
## [1] ""
## [1] "DUMBO"
## [1] "Average Number of Reviews: 38.3888888888889"
## [1] "Average Price: 196.305555555556"
## [1] ""
## [1] "St. George"
## [1] "Average Number of Reviews: 41.3958333333333"
```

```
## [1] "Average Price: 118.145833333333"
## [1] ""
## [1] "Highbridge"
## [1] "Average Number of Reviews: 48.8148148148148"
## [1] "Average Price: 71.1111111111111"
## [1] ""
## [1] "Financial District"
## [1] "Average Number of Reviews: 9.31586021505376"
## [1] "Average Price: 225.490591397849"
## [1] ""
## [1] "Ridgewood"
## [1] "Average Number of Reviews: 18.387706855792"
## [1] "Average Price: 77.1843971631206"
## [1] ""
## [1] "Morningside Heights"
## [1] "Average Number of Reviews: 13.2919075144509"
## [1] "Average Price: 114.78323699422"
## [1] ""
## [1] "Jamaica"
## [1] "Average Number of Reviews: 42.9004329004329"
## [1] "Average Price: 95.7705627705628"
## [1] ""
## [1] "Middle Village"
## [1] "Average Number of Reviews: 34.0322580645161"
## [1] "Average Price: 109.58064516129"
## [1] ""
## [1] "NoHo"
## [1] "Average Number of Reviews: 12.7435897435897"
## [1] "Average Price: 295.717948717949"
## [1] ""
## [1] "Ditmars Steinway"
## [1] "Average Number of Reviews: 25.411003236246"
## [1] "Average Price: 95.0291262135922"
## [1] ""
## [1] "Flatiron District"
## [1] "Average Number of Reviews: 17.7375"
## [1] "Average Price: 341.925"
## [1] ""
## [1] "Roosevelt Island"
## [1] "Average Number of Reviews: 15.7662337662338"
## [1] "Average Price: 113.25974025974"
## [1] ""
## [1] "Greenwich Village"
## [1] "Average Number of Reviews: 16.6479591836735"
## [1] "Average Price: 263.405612244898"
## [1] ""
## [1] "Little Italy"
## [1] "Average Number of Reviews: 23.1322314049587"
## [1] "Average Price: 222.066115702479"
## [1] ""
## [1] "East Flatbush"
## [1] "Average Number of Reviews: 24.896"
## [1] "Average Price: 104.222"
## [1] ""
```

```
## [1] "Tompkinsville"
## [1] "Average Number of Reviews: 57.1428571428571"
## [1] "Average Price: 76.1904761904762"
## [1] ""
## [1] "Astoria"
## [1] "Average Number of Reviews: 21.4555555555556"
## [1] "Average Price: 117.187777777778"
## [1] ""
## [1] "Clason Point"
## [1] "Average Number of Reviews: 15.047619047619"
## [1] "Average Price: 112.761904761905"
## [1] ""
## [1] "Eastchester"
## [1] "Average Number of Reviews: 12.0769230769231"
## [1] "Average Price: 141.692307692308"
## [1] ""
## [1] "Kingsbridge"
## [1] "Average Number of Reviews: 20.5"
## [1] "Average Price: 77.9285714285714"
## [1] ""
## [1] "Two Bridges"
## [1] "Average Number of Reviews: 29.5694444444444"
## [1] "Average Price: 127.069444444444"
## [1] ""
## [1] "Queens Village"
## [1] "Average Number of Reviews: 35.7833333333333"
## [1] "Average Price: 83.9333333333333"
## [1] ""
## [1] "Rockaway Beach"
## [1] "Average Number of Reviews: 24.25"
## [1] "Average Price: 132.178571428571"
## [1] ""
## [1] "Forest Hills"
## [1] "Average Number of Reviews: 22.2916666666667"
## [1] "Average Price: 121.625"
## [1] ""
## [1] "Nolita"
## [1] "Average Number of Reviews: 22.2687747035573"
## [1] "Average Price: 230.138339920949"
## [1] ""
## [1] "Woodlawn"
## [1] "Average Number of Reviews: 44"
## [1] "Average Price: 60.0909090909091"
## [1] ""
## [1] "University Heights"
## [1] "Average Number of Reviews: 37.0952380952381"
## [1] "Average Price: 69.5714285714286"
## [1] ""
## [1] "Gravesend"
## [1] "Average Number of Reviews: 19.6470588235294"
## [1] "Average Price: 79.0147058823529"
## [1] ""
## [1] "Gramercy"
## [1] "Average Number of Reviews: 22.7278106508876"
```

```
## [1] "Average Price: 222.754437869822"
## [1] ""
## [1] "Allerton"
## [1] "Average Number of Reviews: 42.9285714285714"
## [1] "Average Price: 87.5952380952381"
## [1] ""
## [1] "East New York"
## [1] "Average Number of Reviews: 31.0045871559633"
## [1] "Average Price: 85.4266055045872"
## [1] ""
## [1] "Theater District"
## [1] "Average Number of Reviews: 15.0763888888889"
## [1] "Average Price: 248.013888888889"
## [1] ""
## [1] "Concourse Village"
## [1] "Average Number of Reviews: 25.21875"
## [1] "Average Price: 73.78125"
## [1] ""
## [1] "Sheepshead Bay"
## [1] "Average Number of Reviews: 19.890243902439"
## [1] "Average Price: 105.774390243902"
## [1] ""
## [1] "Emerson Hill"
## [1] "Average Number of Reviews: 11.2"
## [1] "Average Price: 68.2"
## [1] ""
## [1] "Fort Hamilton"
## [1] "Average Number of Reviews: 13.5454545454545"
## [1] "Average Price: 93.8181818181818"
## [1] ""
## [1] "Bensonhurst"
## [1] "Average Number of Reviews: 24.6"
## [1] "Average Price: 75.7866666666667"
## [1] ""
## [1] "Tribeca"
## [1] "Average Number of Reviews: 11.4915254237288"
## [1] "Average Price: 490.638418079096"
## [1] ""
## [1] "Shore Acres"
## [1] "Average Number of Reviews: 20.8571428571429"
## [1] "Average Price: 152.714285714286"
## [1] ""
## [1] "Sunset Park"
## [1] "Average Number of Reviews: 20.2102564102564"
## [1] "Average Price: 113.041025641026"
## [1] ""
## [1] "Concourse"
## [1] "Average Number of Reviews: 32.86"
## [1] "Average Price: 86.18"
## [1] ""
## [1] "Elmhurst"
## [1] "Average Number of Reviews: 23.4345991561181"
## [1] "Average Price: 80.4599156118144"
## [1] ""
```

```
## [1] "Brighton Beach"
## [1] "Average Number of Reviews: 17.96"
## [1] "Average Price: 131.933333333333"
## [1] ""
## [1] "Jackson Heights"
## [1] "Average Number of Reviews: 27.2096774193548"
## [1] "Average Price: 80.8978494623656"
## [1] ""
## [1] "Cypress Hills"
## [1] "Average Number of Reviews: 33.2"
## [1] "Average Price: 128.903703703704"
## [1] ""
## [1] "St. Albans"
## [1] "Average Number of Reviews: 34"
## [1] "Average Price: 100.828947368421"
## [1] ""
## [1] "Arrochar"
## [1] "Average Number of Reviews: 14.6190476190476"
## [1] "Average Price: 115"
## [1] ""
## [1] "Rego Park"
## [1] "Average Number of Reviews: 25.9811320754717"
## [1] "Average Price: 83.877358490566"
## [1] ""
## [1] "Wakefield"
## [1] "Average Number of Reviews: 25.58"
## [1] "Average Price: 85.58"
## [1] ""
## [1] "Clifton"
## [1] "Average Number of Reviews: 47.2"
## [1] "Average Price: 84.9333333333333"
## [1] ""
## [1] "Bay Ridge"
## [1] "Average Number of Reviews: 18.5673758865248"
## [1] "Average Price: 144.432624113475"
## [1] ""
## [1] "Graniteville"
## [1] "Average Number of Reviews: 41"
## [1] "Average Price: 68.6666666666667"
## [1] ""
## [1] "Spuyten Duyvil"
## [1] "Average Number of Reviews: 17.5"
## [1] "Average Price: 154.75"
## [1] ""
## [1] "Stapleton"
## [1] "Average Number of Reviews: 38.3703703703704"
## [1] "Average Price: 98.962962962963"
## [1] ""
## [1] "Briarwood"
## [1] "Average Number of Reviews: 22.2678571428571"
## [1] "Average Price: 105.875"
## [1] ""
## [1] "Ozone Park"
## [1] "Average Number of Reviews: 24.3870967741935"
```

```
## [1] "Average Price: 85.2741935483871"
## [1] ""
## [1] "Columbia St"
## [1] "Average Number of Reviews: 16.3095238095238"
## [1] "Average Price: 162.952380952381"
## [1] ""
## [1] "Vinegar Hill"
## [1] "Average Number of Reviews: 16.0294117647059"
## [1] "Average Price: 187.176470588235"
## [1] ""
## [1] "Mott Haven"
## [1] "Average Number of Reviews: 42.3666666666667"
## [1] "Average Price: 88.9166666666667"
## [1] ""
## [1] "Longwood"
## [1] "Average Number of Reviews: 26.6612903225806"
## [1] "Average Price: 91.9193548387097"
## [1] ""
## [1] "Canarsie"
## [1] "Average Number of Reviews: 36.312925170068"
## [1] "Average Price: 104.367346938776"
## [1] ""
## [1] "Battery Park City"
## [1] "Average Number of Reviews: 8.28571428571429"
## [1] "Average Price: 367.557142857143"
## [1] ""
## [1] "Civic Center"
## [1] "Average Number of Reviews: 12.25"
## [1] "Average Price: 191.942307692308"
## [1] ""
## [1] "East Elmhurst"
## [1] "Average Number of Reviews: 81.6594594594595"
## [1] "Average Price: 81.1837837837838"
## [1] ""
## [1] "New Springville"
## [1] "Average Number of Reviews: 29.375"
## [1] "Average Price: 76"
## [1] ""
## [1] "Morris Heights"
## [1] "Average Number of Reviews: 27.3529411764706"
## [1] "Average Price: 76.9411764705882"
## [1] ""
## [1] "Arverne"
## [1] "Average Number of Reviews: 29.2597402597403"
## [1] "Average Price: 171.779220779221"
## [1] ""
## [1] "Cambria Heights"
## [1] "Average Number of Reviews: 18.6538461538462"
## [1] "Average Price: 81.7307692307692"
## [1] ""
## [1] "Tottenville"
## [1] "Average Number of Reviews: 19.8571428571429"
## [1] "Average Price: 144.857142857143"
## [1] ""
```

```
## [1] "Mariners Harbor"
## [1] "Average Number of Reviews: 21.625"
## [1] "Average Price: 94.625"
## [1] ""
## [1] "Concord"
## [1] "Average Number of Reviews: 24.9230769230769"
## [1] "Average Price: 58.1923076923077"
## [1] ""
## [1] "Borough Park"
## [1] "Average Number of Reviews: 16.4117647058824"
## [1] "Average Price: 63.0661764705882"
## [1] ""
## [1] "Bayside"
## [1] "Average Number of Reviews: 23.7948717948718"
## [1] "Average Price: 157.948717948718"
## [1] ""
## [1] "Downtown Brooklyn"
## [1] "Average Number of Reviews: 14.3132530120482"
## [1] "Average Price: 150.349397590361"
## [1] ""
## [1] "Port Morris"
## [1] "Average Number of Reviews: 20.6086956521739"
## [1] "Average Price: 79.8913043478261"
## [1] ""
## [1] "Fieldston"
## [1] "Average Number of Reviews: 12.1666666666667"
## [1] "Average Price: 75.0833333333333"
## [1] ""
## [1] "Kew Gardens"
## [1] "Average Number of Reviews: 10.375"
## [1] "Average Price: 88.375"
## [1] ""
## [1] "Midwood"
## [1] "Average Number of Reviews: 15.1834862385321"
## [1] "Average Price: 80.3394495412844"
## [1] ""
## [1] "College Point"
## [1] "Average Number of Reviews: 23.1052631578947"
## [1] "Average Price: 88"
## [1] ""
## [1] "Mount Eden"
## [1] "Average Number of Reviews: 70"
## [1] "Average Price: 58.5"
## [1] ""
## [1] "City Island"
## [1] "Average Number of Reviews: 42.1666666666667"
## [1] "Average Price: 173"
## [1] ""
## [1] "Glendale"
## [1] "Average Number of Reviews: 27.0185185185185"
## [1] "Average Price: 90.7962962962963"
## [1] ""
## [1] "Port Richmond"
## [1] "Average Number of Reviews: 17"
```

```
## [1] "Average Price: 90.1111111111111"
## [1] ""
## [1] "Red Hook"
## [1] "Average Number of Reviews: 28.6455696202532"
## [1] "Average Price: 143.455696202532"
## [1] ""
## [1] "Richmond Hill"
## [1] "Average Number of Reviews: 34.3617021276596"
## [1] "Average Price: 87.1170212765958"
## [1] ""
## [1] "Bellerose"
## [1] "Average Number of Reviews: 8.42857142857143"
## [1] "Average Price: 99.3571428571429"
## [1] ""
## [1] "Maspeth"
## [1] "Average Number of Reviews: 22.6363636363636"
## [1] "Average Price: 83.6454545454545"
## [1] ""
## [1] "Williamsbridge"
## [1] "Average Number of Reviews: 28.45"
## [1] "Average Price: 96.75"
## [1] ""
## [1] "Soundview"
## [1] "Average Number of Reviews: 29.4"
## [1] "Average Price: 53.4666666666667"
## [1] ""
## [1] "Woodhaven"
## [1] "Average Number of Reviews: 31.7272727272727"
## [1] "Average Price: 67.1704545454545"
## [1] ""
## [1] "Woodrow"
## [1] "Average Number of Reviews: 0"
## [1] "Average Price: 700"
## [1] ""
## [1] "Co-op City"
## [1] "Average Number of Reviews: 17"
## [1] "Average Price: 77.5"
## [1] ""
## [1] "Stuyvesant Town"
## [1] "Average Number of Reviews: 8.16216216216216"
## [1] "Average Price: 169.108108108108"
## [1] ""
## [1] "Parkchester"
## [1] "Average Number of Reviews: 23.3589743589744"
## [1] "Average Price: 69.0769230769231"
## [1] ""
## [1] "North Riverdale"
## [1] "Average Number of Reviews: 18.2"
## [1] "Average Price: 79.9"
## [1] ""
## [1] "Dyker Heights"
## [1] "Average Number of Reviews: 34.5833333333333"
## [1] "Average Price: 93.4166666666667"
## [1] ""
```

```
## [1] "Bronxdale"
## [1] "Average Number of Reviews: 24.1052631578947"
## [1] "Average Price: 57.1052631578947"
## [1] ""
## [1] "Sea Gate"
## [1] "Average Number of Reviews: 1.42857142857143"
## [1] "Average Price: 487.857142857143"
## [1] ""
## [1] "Riverdale"
## [1] "Average Number of Reviews: 26.6363636363636"
## [1] "Average Price: 442.090909090909"
## [1] ""
## [1] "Kew Gardens Hills"
## [1] "Average Number of Reviews: 12.7307692307692"
## [1] "Average Price: 112.307692307692"
## [1] ""
## [1] "Bay Terrace"
## [1] "Average Number of Reviews: 41.5"
## [1] "Average Price: 142"
## [1] ""
## [1] "Norwood"
## [1] "Average Number of Reviews: 16.741935483871"
## [1] "Average Price: 75.5483870967742"
## [1] ""
## [1] "Claremont Village"
## [1] "Average Number of Reviews: 12.5"
## [1] "Average Price: 87.4642857142857"
## [1] ""
## [1] "Whitestone"
## [1] "Average Number of Reviews: 22.6363636363636"
## [1] "Average Price: 107.545454545455"
## [1] ""
## [1] "Fordham"
## [1] "Average Number of Reviews: 15.1746031746032"
## [1] "Average Price: 69.4444444444444"
## [1] ""
## [1] "Bayswater"
## [1] "Average Number of Reviews: 11.2352941176471"
## [1] "Average Price: 87.4705882352941"
## [1] ""
## [1] "Navy Yard"
## [1] "Average Number of Reviews: 17.9285714285714"
## [1] "Average Price: 151.642857142857"
## [1] ""
## [1] "Brownsville"
## [1] "Average Number of Reviews: 31.344262295082"
## [1] "Average Price: 76.4590163934426"
## [1] ""
## [1] "Eltingville"
## [1] "Average Number of Reviews: 76"
## [1] "Average Price: 141.666666666667"
## [1] ""
## [1] "Fresh Meadows"
## [1] "Average Number of Reviews: 32.65625"
```

```
## [1] "Average Price: 99.5"
## [1] ""
## [1] "Mount Hope"
## [1] "Average Number of Reviews: 21.75"
## [1] "Average Price: 77.5"
## [1] ""
## [1] "Lighthouse Hill"
## [1] "Average Number of Reviews: 39"
## [1] "Average Price: 157.5"
## [1] ""
## [1] "Springfield Gardens"
## [1] "Average Number of Reviews: 69.0941176470588"
## [1] "Average Price: 94.2352941176471"
## [1] ""
## [1] "Howard Beach"
## [1] "Average Number of Reviews: 13.6"
## [1] "Average Price: 115.4"
## [1] ""
## [1] "Belle Harbor"
## [1] "Average Number of Reviews: 9.25"
## [1] "Average Price: 171.5"
## [1] ""
## [1] "Jamaica Estates"
## [1] "Average Number of Reviews: 31.3684210526316"
## [1] "Average Price: 182.947368421053"
## [1] ""
## [1] "Van Nest"
## [1] "Average Number of Reviews: 18.7272727272727"
## [1] "Average Price: 113.818181818182"
## [1] ""
## [1] "Morris Park"
## [1] "Average Number of Reviews: 18.0666666666667"
## [1] "Average Price: 69.3333333333333"
## [1] ""
## [1] "West Brighton"
## [1] "Average Number of Reviews: 36.4444444444444"
## [1] "Average Price: 80.5555555555556"
## [1] ""
## [1] "Far Rockaway"
## [1] "Average Number of Reviews: 12.8965517241379"
## [1] "Average Price: 165.862068965517"
## [1] ""
## [1] "South Ozone Park"
## [1] "Average Number of Reviews: 48.675"
## [1] "Average Price: 82.4"
## [1] ""
## [1] "Tremont"
## [1] "Average Number of Reviews: 20.6363636363636"
## [1] "Average Price: 51.5454545454545"
## [1] ""
## [1] "Corona"
## [1] "Average Number of Reviews: 28.21875"
## [1] "Average Price: 59.171875"
## [1] ""
```

```
## [1] "Great Kills"
## [1] "Average Number of Reviews: 26.4"
## [1] "Average Price: 100.6"
## [1] ""
## [1] "Manhattan Beach"
## [1] "Average Number of Reviews: 50.625"
## [1] "Average Price: 103.5"
## [1] ""
## [1] "Marble Hill"
## [1] "Average Number of Reviews: 14"
## [1] "Average Price: 89.1666666666667"
## [1] ""
## [1] "Dongan Hills"
## [1] "Average Number of Reviews: 22.5714285714286"
## [1] "Average Price: 79.4285714285714"
## [1] ""
## [1] "Castleton Corners"
## [1] "Average Number of Reviews: 25.25"
## [1] "Average Price: 139.75"
## [1] ""
## [1] "East Morrisania"
## [1] "Average Number of Reviews: 48"
## [1] "Average Price: 85"
## [1] ""
## [1] "Hunts Point"
## [1] "Average Number of Reviews: 9.77777777777778"
## [1] "Average Price: 50.5"
## [1] ""
## [1] "Neponsit"
## [1] "Average Number of Reviews: 15.3333333333333"
## [1] "Average Price: 274.666666666667"
## [1] ""
## [1] "Pelham Bay"
## [1] "Average Number of Reviews: 26"
## [1] "Average Price: 105"
## [1] ""
## [1] "Randall Manor"
## [1] "Average Number of Reviews: 18"
## [1] "Average Price: 336"
## [1] ""
## [1] "Throgs Neck"
## [1] "Average Number of Reviews: 32.0416666666667"
## [1] "Average Price: 91.0416666666667"
## [1] ""
## [1] "Todt Hill"
## [1] "Average Number of Reviews: 4"
## [1] "Average Price: 169"
## [1] ""
## [1] "West Farms"
## [1] "Average Number of Reviews: 3.5"
## [1] "Average Price: 122"
## [1] ""
## [1] "Silver Lake"
## [1] "Average Number of Reviews: 118.5"
```

```
## [1] "Average Price: 70"
## [1] ""
## [1] "Morrisania"
## [1] "Average Number of Reviews: 10.3333333333333"
## [1] "Average Price: 83.4444444444444"
## [1] ""
## [1] "Laurelton"
## [1] "Average Number of Reviews: 29.3333333333333"
## [1] "Average Price: 95.3333333333333"
## [1] ""
## [1] "Grymes Hill"
## [1] "Average Number of Reviews: 10.5714285714286"
## [1] "Average Price: 159.142857142857"
## [1] ""
## [1] "Holliswood"
## [1] "Average Number of Reviews: 3.75"
## [1] "Average Price: 135.75"
## [1] ""
## [1] "Pelham Gardens"
## [1] "Average Number of Reviews: 40.7142857142857"
## [1] "Average Price: 93.6071428571429"
## [1] ""
## [1] "Belmont"
## [1] "Average Number of Reviews: 12.3333333333333"
## [1] "Average Price: 77.125"
## [1] ""
## [1] "Rosedale"
## [1] "Average Number of Reviews: 28.3559322033898"
## [1] "Average Price: 76.6949152542373"
## [1] ""
## [1] "Edgemere"
## [1] "Average Number of Reviews: 10.2727272727273"
## [1] "Average Price: 94.7272727272727"
## [1] ""
## [1] "New Brighton"
## [1] "Average Number of Reviews: 40.2"
## [1] "Average Price: 101.8"
## [1] ""
## [1] "Midland Beach"
## [1] "Average Number of Reviews: 6.16666666666667"
## [1] "Average Price: 91.8333333333333"
## [1] ""
## [1] "Baychester"
## [1] "Average Number of Reviews: 44.2857142857143"
## [1] "Average Price: 75.4285714285714"
## [1] ""
## [1] "Melrose"
## [1] "Average Number of Reviews: 9"
## [1] "Average Price: 83.3"
## [1] ""
## [1] "Bergen Beach"
## [1] "Average Number of Reviews: 20.5"
## [1] "Average Price: 106.7"
## [1] ""
```

```
## [1] "Richmondtown"
## [1] "Average Number of Reviews: 79"
## [1] "Average Price: 78"
## [1] ""
## [1] "Howland Hook"
## [1] "Average Number of Reviews: 20.5"
## [1] "Average Price: 100"
## [1] ""
## [1] "Schuylerville"
## [1] "Average Number of Reviews: 32.5384615384615"
## [1] "Average Price: 69.2307692307692"
## [1] ""
## [1] "Coney Island"
## [1] "Average Number of Reviews: 21.9411764705882"
## [1] "Average Price: 123.705882352941"
## [1] ""
## [1] "New Dorp Beach"
## [1] "Average Number of Reviews: 14.8"
## [1] "Average Price: 57.4"
## [1] ""
## [1] "Prince's Bay"
## [1] "Average Number of Reviews: 5.75"
## [1] "Average Price: 409.5"
## [1] ""
## [1] "South Beach"
## [1] "Average Number of Reviews: 11.875"
## [1] "Average Price: 89.25"
## [1] ""
## [1] "Bath Beach"
## [1] "Average Number of Reviews: 21.3529411764706"
## [1] "Average Price: 81.7647058823529"
## [1] ""
## [1] "Jamaica Hills"
## [1] "Average Number of Reviews: 8.625"
## [1] "Average Price: 132.125"
## [1] ""
## [1] "Oakwood"
## [1] "Average Number of Reviews: 1.8"
## [1] "Average Price: 81.2"
## [1] ""
## [1] "Castle Hill"
## [1] "Average Number of Reviews: 13.1111111111111"
## [1] "Average Price: 63"
## [1] ""
## [1] "Hollis"
## [1] "Average Number of Reviews: 11.7142857142857"
## [1] "Average Price: 88.6428571428571"
## [1] ""
## [1] "Douglaston"
## [1] "Average Number of Reviews: 20.875"
## [1] "Average Price: 82.75"
## [1] ""
## [1] "Huguenot"
## [1] "Average Number of Reviews: 55.6666666666667"
```

```
## [1] "Average Price: 118.333333333333"
## [1] ""
## [1] "Olinville"
## [1] "Average Number of Reviews: 7.5"
## [1] "Average Price: 64"
## [1] ""
## [1] "Edenwald"
## [1] "Average Number of Reviews: 15.3846153846154"
## [1] "Average Price: 82"
## [1] ""
## [1] "Grant City"
## [1] "Average Number of Reviews: 26.8333333333333"
## [1] "Average Price: 57.6666666666667"
## [1] ""
## [1] "Westerleigh"
## [1] "Average Number of Reviews: 9"
## [1] "Average Price: 71.5"
## [1] ""
## [1] "Bay Terrace, Staten Island"
## [1] "Average Number of Reviews: 1.5"
## [1] "Average Price: 102.5"
## [1] ""
## [1] "Westchester Square"
## [1] "Average Number of Reviews: 15.5"
## [1] "Average Price: 122.2"
## [1] ""
## [1] "Little Neck"
## [1] "Average Number of Reviews: 3.8"
## [1] "Average Price: 75.2"
## [1] ""
## [1] "Fort Wadsworth"
## [1] "Average Number of Reviews: 0"
## [1] "Average Price: 800"
## [1] ""
## [1] "Rosebank"
## [1] "Average Number of Reviews: 30.7142857142857"
## [1] "Average Price: 111.857142857143"
## [1] ""
## [1] "Unionport"
## [1] "Average Number of Reviews: 14.8571428571429"
## [1] "Average Price: 137.142857142857"
## [1] ""
## [1] "Mill Basin"
## [1] "Average Number of Reviews: 17.25"
## [1] "Average Price: 179.75"
## [1] ""
## [1] "Arden Heights"
## [1] "Average Number of Reviews: 7.75"
## [1] "Average Price: 67.25"
## [1] ""
## [1] "Bull's Head"
## [1] "Average Number of Reviews: 15.3333333333333"
## [1] "Average Price: 47.3333333333333"
## [1] ""
```

```
## [1] "New Dorp"
## [1] "Average Number of Reviews: 0"
## [1] "Average Price: 57"
## [1] ""
## [1] "Rossville"
## [1] "Average Number of Reviews: 21"
## [1] "Average Price: 75"
## [1] ""
## [1] "Breezy Point"
## [1] "Average Number of Reviews: 1.66666666666667"
## [1] "Average Price: 213.333333333333"
## [1] ""
## [1] "Willowbrook"
## [1] "Average Number of Reviews: 9"
## [1] "Average Price: 249"
## [1] ""
```

```
# 2.Use the functions created in part 1 to compute these metrics by room type and neighborhood.
average_number_of_reviews_room <- function(data, room_type){
  count <- 0
  mySum <- 0
  for(j in 1:nrow(data)){
    if(data[j,"room_type"][[1]] == room_type){
      count <- count + 1
      mySum <- mySum + data[j, "number_of_reviews"][[1]]
    }

  }
  return(mySum/count)
}


#   b. average_price
average_price_room <- function(data, room_type){
  count <- 0
  mySum <- 0
  for(j in 1:nrow(data)){
    if(data[j,"room_type"][[1]] == room_type){
      count <- count + 1
      mySum <- mySum + data[j, "price"][[1]]
    }

  }
  return(mySum/count)
}

number_reviews_and_price_per_room_type <- function(data){
  room_types <- unique(data$room_type)
  for(i in room_types){
    print(i)
    print(paste("Average Number of Reviews:", average_number_of_reviews_room(data,i)))
    print(paste("Average Price:", average_price_room(data,i)))
    print("")
  }
```

```
}
number_reviews_and_price_per_room_type(airbnb)
```

```
## [1] "Private room"
## [1] "Average Number of Reviews: 24.1129624652871"
## [1] "Average Price: 89.7809728567589"
## [1] ""
## [1] "Entire home/apt"
## [1] "Average Number of Reviews: 22.8424180408517"
## [1] "Average Price: 211.79424613326"
## [1] ""
## [1] "Shared room"
## [1] "Average Number of Reviews: 16.6"
## [1] "Average Price: 70.1275862068966"
## [1] ""
```