Real estate data analysis

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

# You have just begun an illustrious career in the real estate sector. You’re given a large dataset containing #every non-commercial property for sale in the city of Windsor for the past 6 months. It is your job to #create functions to help interpret and make use of this data.

# Import the dataset

mls\_data <- read.csv("mls\_13.csv")

heck the data

summary(mls\_data)

## price size lsize age   
## Min. : 25000 Min. : 700 Min. : 1102 Min. : 0.00   
## 1st Qu.: 63000 1st Qu.:1050 1st Qu.: 4200 1st Qu.: 7.00   
## Median : 87500 Median :1200 Median : 5617 Median : 20.00   
## Mean : 96377 Mean :1304 Mean : 6063 Mean : 31.95   
## 3rd Qu.:124000 3rd Qu.:1483 3rd Qu.: 6955 3rd Qu.: 60.00   
## Max. :345000 Max. :3573 Max. :95928 Max. :140.00   
## bathp month poolag poolig   
## Min. : 3.000 Min. : 1.00 Min. :0.00000 Min. :0.00000   
## 1st Qu.: 4.000 1st Qu.:18.00 1st Qu.:0.00000 1st Qu.:0.00000   
## Median : 6.000 Median :33.00 Median :0.00000 Median :0.00000   
## Mean : 5.866 Mean :37.76 Mean :0.01966 Mean :0.02648   
## 3rd Qu.: 7.000 3rd Qu.:59.00 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :16.000 Max. :72.00 Max. :1.00000 Max. :1.00000   
## firep sager dagar car\_p   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.0000 Median :0.0000 Median :0.0000 Median :0.00000   
## Mean :0.3804 Mean :0.2925 Mean :0.1132 Mean :0.03291   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000   
## busy\_rd ele   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000   
## Mean :0.1112 Mean :0.1962   
## 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000

dim(mls\_data)

## [1] 2492 14

str(mls\_data)

## 'data.frame': 2492 obs. of 14 variables:  
## $ price : int 95000 122500 75000 117000 143000 140000 113000 125000 85000 119255 ...  
## $ size : int 1360 1290 1056 1510 1544 1250 877 1500 1450 1775 ...  
## $ lsize : int 1855 5790 1962 4500 4750 3700 4400 5016 4506 4330 ...  
## $ age : int 80 18 80 80 60 80 80 80 80 50 ...  
## $ bathp : int 8 7 4 10 4 4 6 6 4 7 ...  
## $ month : int 57 55 67 49 70 71 58 67 49 53 ...  
## $ poolag : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poolig : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ firep : int 0 1 0 0 0 1 0 1 0 0 ...  
## $ sager : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ dagar : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ car\_p : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ busy\_rd: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ ele : int 0 0 0 0 0 0 0 0 0 1 ...

Create a function that calculates the average of a set (without using R’s mean function). It should take an array of numbers as an input, and return the average.

average <- function(x)  
{  
 ave <- sum(x) / length(x)  
 return(ave)  
 }

Create a function that calculates Z scores. It should take in a single value, and an array of numbers as arguments, and return the Z-score.

z\_score <- function(x, y)  
{  
 z <- (x - average(y)) / sd(y)  
 return(z)  
}

Clients frequently approach you with the maximum price they’re willing to spend on a house. With this maximum price, create a function that return an array with 2 elements; the number of houses available at/below this price, and the portion of houses on the market that are at/below this price.

no\_houses <- function(cost)  
{  
 element <- 0  
 element[1] <- sum(mls\_data$price <= cost)  
 element[2] <- element[1] / length(mls\_data$price)  
 return(element)  
}  
  
no\_houses(100000)

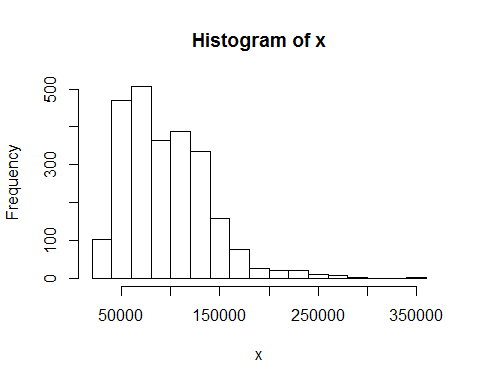
## [1] 1444.0000000 0.5794543

summary(mls\_data)

## price size lsize age   
## Min. : 25000 Min. : 700 Min. : 1102 Min. : 0.00   
## 1st Qu.: 63000 1st Qu.:1050 1st Qu.: 4200 1st Qu.: 7.00   
## Median : 87500 Median :1200 Median : 5617 Median : 20.00   
## Mean : 96377 Mean :1304 Mean : 6063 Mean : 31.95   
## 3rd Qu.:124000 3rd Qu.:1483 3rd Qu.: 6955 3rd Qu.: 60.00   
## Max. :345000 Max. :3573 Max. :95928 Max. :140.00   
## bathp month poolag poolig   
## Min. : 3.000 Min. : 1.00 Min. :0.00000 Min. :0.00000   
## 1st Qu.: 4.000 1st Qu.:18.00 1st Qu.:0.00000 1st Qu.:0.00000   
## Median : 6.000 Median :33.00 Median :0.00000 Median :0.00000   
## Mean : 5.866 Mean :37.76 Mean :0.01966 Mean :0.02648   
## 3rd Qu.: 7.000 3rd Qu.:59.00 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :16.000 Max. :72.00 Max. :1.00000 Max. :1.00000   
## firep sager dagar car\_p   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.0000 Median :0.0000 Median :0.0000 Median :0.00000   
## Mean :0.3804 Mean :0.2925 Mean :0.1132 Mean :0.03291   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000   
## busy\_rd ele   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000   
## Mean :0.1112 Mean :0.1962   
## 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000

Create a function that takes in a numeric array and plots it in a histogram, and returns 1 if the set is skewed right, and -1 if it is skewed left. Use this function to plot and determine the skew of the prices.

skewness <- function(x)  
{  
 hist(x)  
 skew <- average(x) - median(x)  
 return(sign(skew))  
}  
  
skewness(mls\_data$price)



## [1] 1

Create a function that has price as an input, and returns the z-score of the average number of bath-pieces (denoted ‘bathp’ in the data set) for houses less than or equal to the inputted price. Also make it print a summary for the number of bath-pieces for houses at or below the inputted price.

bath\_details <- function(input\_price)  
{  
 print(summary(mls\_data$bathp[mls\_data$price <= input\_price]))  
 return(z\_score(average(mls\_data$bathp[mls\_data$price <= input\_price]), mls\_data$price))  
}  
  
bath\_details(85000)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.000 4.000 4.000 5.169 6.000 11.000

## [1] -2.207635

Create a function that takes and upper price and a lower price as an argument, creates a subset of houses that fall into that interval, and returns the head of that subset.

sub\_houses <- function(lower,upper)  
{  
 sub <- 0  
 sub <- mls\_data[which(mls\_data$price >= lower & mls\_data$price <= upper),]  
 return(head(sub))  
}  
  
sub\_houses(75000,100000)

## price size lsize age bathp month poolag poolig firep sager dagar car\_p  
## 1 95000 1360 1855 80 8 57 0 0 0 0 0 0  
## 3 75000 1056 1962 80 4 67 0 0 0 0 0 0  
## 9 85000 1450 4506 80 4 49 0 0 0 0 0 0  
## 12 94000 1100 2640 35 4 59 0 0 0 0 0 0  
## 13 98500 1622 5052 80 4 56 0 0 1 0 0 0  
## 16 94000 1200 5174 80 4 67 0 0 0 0 0 0  
## busy\_rd ele  
## 1 0 0  
## 3 0 0  
## 9 0 0  
## 12 0 0  
## 13 0 0  
## 16 1 0

Finally, clients are very interested in bath-pieces. Create a function that has an input as a number of bath-pieces, and returns a subset of the cheapest house(s) that have that number of bath-pieces.

cheap\_houses <- function(bathpieces)  
{  
 bath <- 0  
 bath<- mls\_data[which(mls\_data$bathp == bathpieces),]  
 bath <- bath[bath$price == min(bath$price),]  
 return(bath)  
}  
  
cheap\_houses(4)

## price size lsize age bathp month poolag poolig firep sager dagar  
## 558 26000 900 3300 80 4 15 0 0 0 0 0  
## 2256 26000 1040 1926 80 4 24 0 0 0 0 0  
## car\_p busy\_rd ele  
## 558 0 0 0  
## 2256 0 0 0