Homework 1

1. The Iowa data set iowa.csv is a toy example that summarises the yield of wheat (bushels per acre) for the state of Iowa between 1930-1962. In addition to yield, year, rainfall and temperature were recorded as the main predictors of yield.
   1. First, we need to load the data set into R using the command read.csv(). Use the help function to learn what arguments this function takes. Once you have the necessary input, load the data set into R and make it a data frame called iowa.df.

iowa.df<-read.csv("data/iowa.csv",header = TRUE,sep = ';')

b. How many rows and columns does `iowa.df` have?   
 answer:33 rows and 10 columns

dim(iowa.df)

## [1] 33 10

c. What are the names of the columns of `iowa.df`?   
 answer: Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield

str(iowa.df)

## 'data.frame': 33 obs. of 10 variables:  
## $ Year : int 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 ...  
## $ Rain0: num 17.8 14.8 28 16.8 11.4 ...  
## $ Temp1: num 60.2 57.5 62.3 60.5 69.5 55 66.2 61.8 59.5 66.4 ...  
## $ Rain1: num 5.83 3.83 5.17 1.64 3.49 7 2.85 3.8 4.67 5.32 ...  
## $ Temp2: num 69 75 72 77.8 77.2 65.9 70.1 69 69.2 71.4 ...  
## $ Rain2: num 1.49 2.72 3.12 3.45 3.85 3.35 0.51 2.63 4.24 3.15 ...  
## $ Temp3: num 77.9 77.2 75.8 76.4 79.7 79.4 83.4 75.9 76.5 76.2 ...  
## $ Rain3: num 2.42 3.3 7.1 3.01 2.84 2.42 3.48 3.99 3.82 4.72 ...  
## $ Temp4: num 74.4 72.6 72.2 70.5 73.4 73.6 79.2 77.8 75.7 70.7 ...  
## $ Yield: num 34 32.9 43 40 23 38.4 20 44.6 46.3 52.2 ...

d. What is the value of row 5, column 7 of `iowa.df`?  
 answer: 79.7

iowa.df[5,7]

## [1] 79.7

e. Display the second row of `iowa.df` in its entirety.

iowa.df[2,]

## Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield  
## 2 1931 14.76 57.5 3.83 75 2.72 77.2 3.3 72.6 32.9

1. Syntax and class-typing.
   1. For each of the following commands, either explain why they should be errors, or explain the non-erroneous result

* vector1 <- c("5", "12", "7", "32")   
  answer:"5" "12" "7" "32"  
  max(vector1)   
  answer:"7"  
  sort(vector1)   
  answer:"12" "32" "5" "7"   
  sum(vector1)   
  answer:the data structure is chr
  1. For the next series of commands, either explain their results, or why they should produce errors.

vector2 <- c("5",7,12)   
answer:"5" "7" "12"  
vector2[2] + vector2[3]   
answer:Error because the default is to character type  
dataframe3 <- data.frame(z1="5",z2=7,z3=12)   
answer: z1 z2 z3  
 5 7 12  
dataframe3[1,2] + dataframe3[1,3]  
answer: 19  
list4 <- list(z1="6", z2=42, z3="49", z4=126)  
answer:  
$z1  
[1] "6"  
  
$z2  
[1] 42  
  
$z3  
[1] "49"  
  
$z4  
[1] 126  
list4[[2]]+list4[[4]]  
answer:168  
list4[2]+list4[4]  
answer:list4[2]=z2,list4[4]=z4，cannot calculate

1. Working with functions and operators.
   1. The colon operator will create a sequence of integers in order. It is a special case of the function seq() which you saw earlier in this assignment. Using the help command ?seq to learn about the function, design an expression that will give you the sequence of numbers from 1 to 10000 in increments of 372. Design another that will give you a sequence between 1 and 10000 that is exactly 50 numbers in length.

seq(1,1000,372)

## [1] 1 373 745

seq(1,1000,length.out = 50)

## [1] 1.00000 21.38776 41.77551 62.16327 82.55102 102.93878  
## [7] 123.32653 143.71429 164.10204 184.48980 204.87755 225.26531  
## [13] 245.65306 266.04082 286.42857 306.81633 327.20408 347.59184  
## [19] 367.97959 388.36735 408.75510 429.14286 449.53061 469.91837  
## [25] 490.30612 510.69388 531.08163 551.46939 571.85714 592.24490  
## [31] 612.63265 633.02041 653.40816 673.79592 694.18367 714.57143  
## [37] 734.95918 755.34694 775.73469 796.12245 816.51020 836.89796  
## [43] 857.28571 877.67347 898.06122 918.44898 938.83673 959.22449  
## [49] 979.61224 1000.00000

b. The function `rep()` repeats a vector some number of times. Explain the difference between `rep(1:3, times=3) and rep(1:3, each=3).  
answer:One is to circulate 3 times according to 1.2.3; The other is to circulate for 3 times according to 1.2.3 Respectively

rep(1:3, times=3)

## [1] 1 2 3 1 2 3 1 2 3

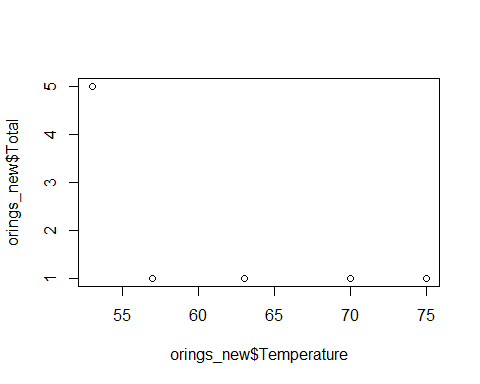
rep(1:3, each=3)

## [1] 1 1 1 2 2 2 3 3 3

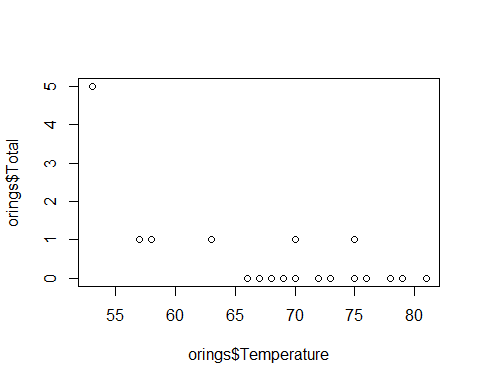
MB.Ch1.2. The orings data frame gives data on the damage that had occurred in US space shuttle launches prior to the disastrous Challenger launch of 28 January 1986. The observations in rows 1, 2, 4, 11, 13,and 18 were included in the pre-launch charts used in deciding whether to proceed with the launch, while remaining rows were omitted.

Create a new data frame by extracting these rows from orings, and plot total incidents against temperature for this new data frame. Obtain a similar plot for the full data set.

orings\_new=orings[c(1,2,4,11,13,18),]  
plot(orings\_new$Temperature,orings\_new$Total)



plot(orings$Temperature,orings$Total)

 MB.Ch1.4. For the data frame ais (DAAG package)

1. Use the function str() to get information on each of the columns. Determine whether any of the columns hold missing values.

str(ais)

## 'data.frame': 202 obs. of 13 variables:  
## $ rcc : num 3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...  
## $ wcc : num 7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...  
## $ hc : num 37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...  
## $ hg : num 12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...  
## $ ferr : num 60 68 21 69 29 42 73 44 41 44 ...  
## $ bmi : num 20.6 20.7 21.9 21.9 19 ...  
## $ ssf : num 109.1 102.8 104.6 126.4 80.3 ...  
## $ pcBfat: num 19.8 21.3 19.9 23.7 17.6 ...  
## $ lbm : num 63.3 58.5 55.4 57.2 53.2 ...  
## $ ht : num 196 190 178 185 185 ...  
## $ wt : num 78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...  
## $ sex : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...  
## $ sport : Factor w/ 10 levels "B\_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...

1. Make a table that shows the numbers of males and females for each different sport. In which sports is there a large imbalance (e.g., by a factor of more than 2:1) in the numbers of the two sexes? answer:Gym,Netball,T\_sprnt,W\_polo have a large imbalance

ais %>% group\_by(sport) %>% summarise(male=sum(sex=='m'),female=sum(sex=='f')) %>% mutate(proportion=male/female)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 10 x 4  
## sport male female proportion  
## <fct> <int> <int> <dbl>  
## 1 B\_Ball 12 13 0.923  
## 2 Field 12 7 1.71   
## 3 Gym 0 4 0   
## 4 Netball 0 23 0   
## 5 Row 15 22 0.682  
## 6 Swim 13 9 1.44   
## 7 T\_400m 18 11 1.64   
## 8 T\_Sprnt 11 4 2.75   
## 9 Tennis 4 7 0.571  
## 10 W\_Polo 17 0 Inf

MB.Ch1.6.Create a data frame called Manitoba.lakes that contains the lake’s elevation (in meters above sea level) and area (in square kilometers) as listed below. Assign the names of the lakes using the row.names() function. elevation area Winnipeg 217 24387 Winnipegosis 254 5374 Manitoba 248 4624 SouthernIndian 254 2247 Cedar 253 1353 Island 227 1223 Gods 178 1151 Cross 207 755 Playgreen 217 657

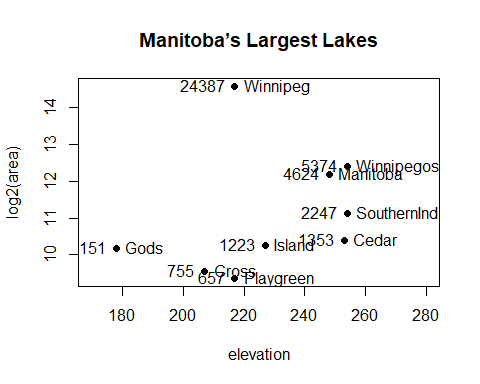
elevation=c(217,254,248,254,253,227,178,207,217)  
area=c(24387,5374,4624,2247,1353,1223,1151,755,657)  
Manitoba.lakes=data.frame(elevation,area)  
row.names(Manitoba.lakes)=c("Winnipeg","Winnipegosis","Manitoba","SouthernIndian","Cedar","Island","Gods","Cross","Playgreen")

1. Use the following code to plot log2(area) versus elevation, adding labeling information (there is an extreme value of area that makes a logarithmic scale pretty much essential):

attach(Manitoba.lakes)

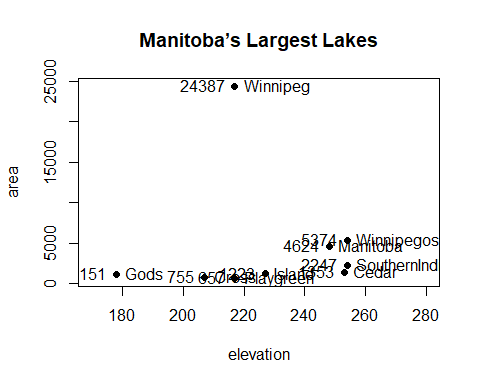
## The following objects are masked \_by\_ .GlobalEnv:  
##   
## area, elevation

plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))  
# NB: Doubling the area increases log2(area) by 1.0  
text(log2(area) ~ elevation, labels=row.names(Manitoba.lakes), pos=4)  
text(log2(area) ~ elevation, labels=area, pos=2)   
title("Manitoba’s Largest Lakes")

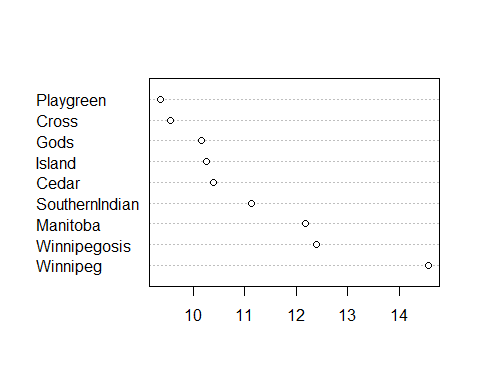
 Devise captions that explain the labeling on the points and on the y-axis. It will be necessary to explain how distances on the scale relate to changes in area.

1. Repeat the plot and associated labeling, now plotting area versus elevation, but specifying log=“y” in order to obtain a logarithmic y-scale.

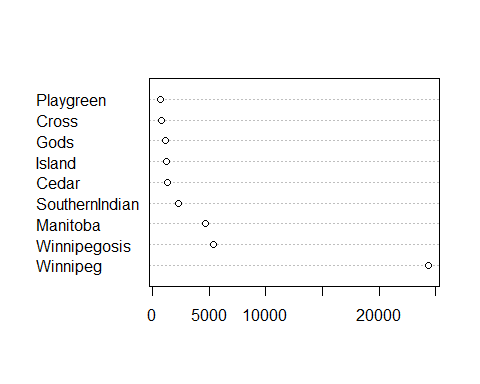
plot(area ~ elevation, pch=16, xlim=c(170,280), ylog=T)  
text(area ~ elevation, labels=row.names(Manitoba.lakes), pos=4, ylog=T)  
text(area ~ elevation, labels=area, pos=2, ylog=T)   
title("Manitoba’s Largest Lakes")

 MB.Ch1.7. Look up the help page for the R function dotchart(). Use this function to display the areas of the Manitoba lakes (a) on a linear scale, and (b) on a logarithmic scale. Add, in each case, suitable labeling information.

dotchart(log2(area),labels = row.names(Manitoba.lakes))



dotchart(area,labels = row.names(Manitoba.lakes))



MB.Ch1.8. Using the sum() function, obtain a lower bound for the area of Manitoba covered by water.

sum(Manitoba.lakes$area)

## [1] 41771