# ASSIGNMENT TASK:

## Download the txt file (Forest718.txt) from Future Learn and save it to your R working directory.

**Solution**

data = read.csv("C:/Users/Aleson/Desktop/Ahmed Ali Qayyum Tasks/R/R programming tasks 3T7665535K250260B/forestfires.csv")

## Assign the data to a matrix, e.g. using

the.data <- as.matrix(read.table("Forest718.txt"))

**Solution**

data200 <-head(data$Y, 200)

# OR

head(data$Y, 200)

## Your variable of interest is X13=Y: area - the burned area of the forest (in ha): 0.00 to

## 1090.84 (the thirteenth column in the dataset). Generate a subset of 200 data e.g.

## Using:

my.data <- the.data[sample(1:517,200),c(1:13)]

**Solution**

head(data$Y, 200)

[1] 5 4 4 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 4 3 4 4 4 4 4 4 6 6 6 4 4

[51] 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 5 5 5 4 4 4 9 9 3 2 2 2 2 2 2 2 2 2 2 2 5 5 6 6 6 6 4 4 4 44

[101]4 4 4 4 4 5 5 5 5 5 4 4 4 4 4 5 4 4 4 4 4 4 4 4 4 4 5 5 5 5 6 6 6 6 5 5 6 69 4 5 2 6 2 5 5 4 32 6

[151]5 9 4 4 5 4 4 2 4 4 4 4 3 6 5 5 5 5 5 6 4 6 4 4 4 4 5 55 6 3 6 4 4 6 5 65 4 4 5 5 2 5 2 5 5 5 4 4

## Choose any FOUR variables from X5 to X11. Using scatter plots and histograms.

## Report on the general relationship between each of the variables and your variable of interest Y. Include 4 scatter plots, 5 histograms and 1 or 2 sentences for each of the variables

**Solution**

newdata <- data[c(5,6,7,8,9,2)]

# Scatterplot

attach(newdata)

plot(newdata$FFMC, data$Y, main="Scatterplot",

xlab="FFMC ", ylab="Y ", pch =19)

plot(newdata$temp, data$Y, main="Scatterplot",

xlab="FFMC ", ylab="Y ", pch=19)

plot(newdata$RH, data$Y, main="Scatterplot",

xlab="FFMC ", ylab="Y ", pch=19)

plot(newdata$wind, data$Y, main="Scatterplot",

xlab="FFMC ", ylab="Y ", pch=19)

# Histogram

hist(newdata$FFMC,xlab = "Weight",col = "yellow",border = "blue")

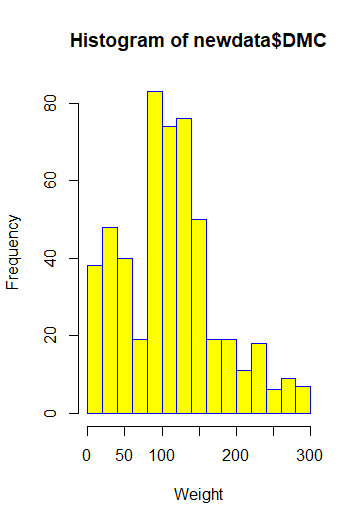
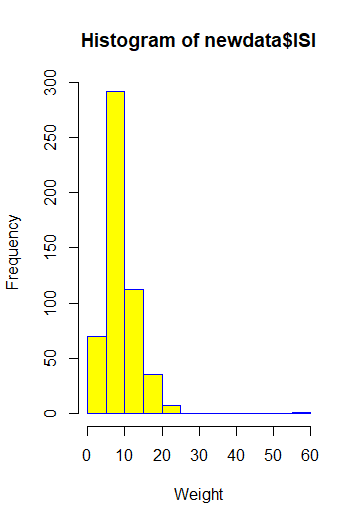
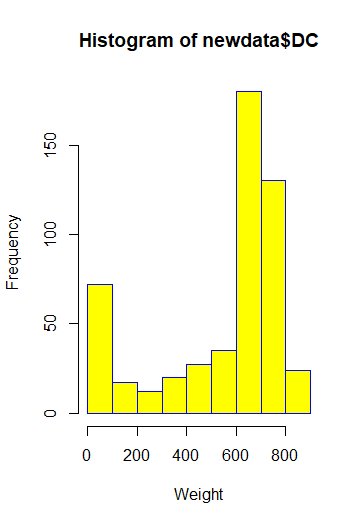
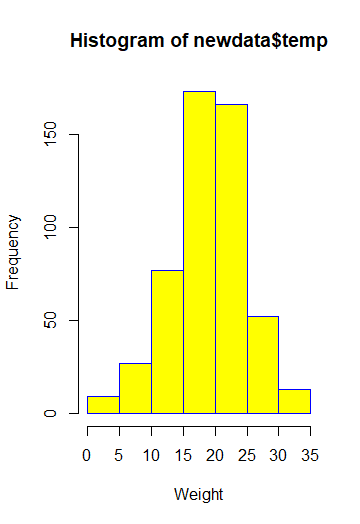
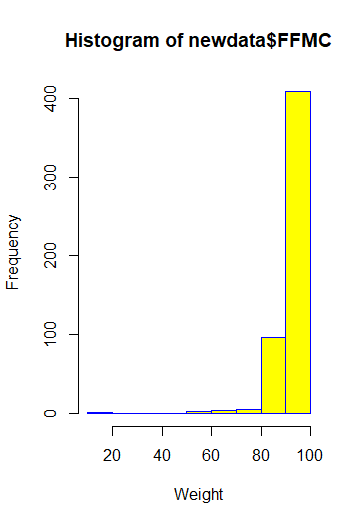
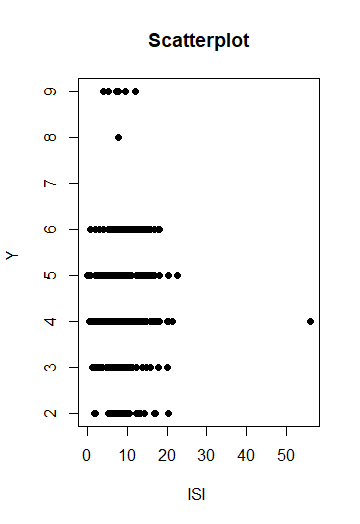
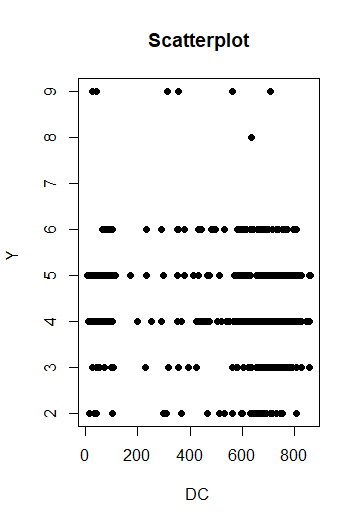
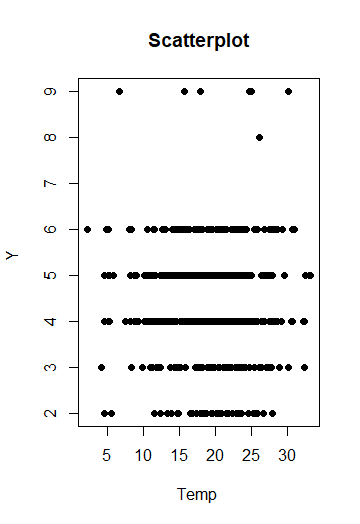
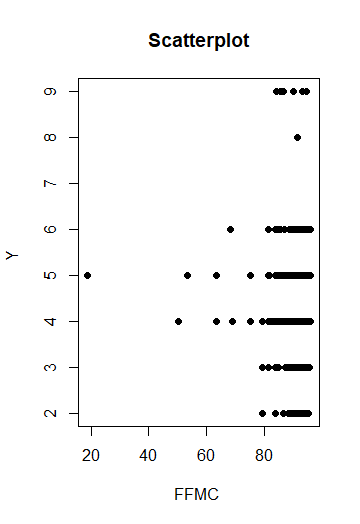
hist(newdata$temp,xlab = "Weight",col = "yellow",border = "blue")

hist(newdata$RH,xlab = "Weight",col = "yellow",border = "blue")

hist(newdata$wind,xlab = "Weight",col = "yellow",border = "blue")

hist(newdata$rain,xlab = "Weight",col = "yellow",border = "blue")

#



# 2. Transform the data

## For the chosen four variables and the variable of interest Y make appropriate

write.table(your.data,"name-transformed.txt",)

**Solution**

newdata1 <- data[c(3,4,5,6,2)]

newdata1 <- head(newdata1, 200)

write.table(newdata1, file = "C:/Users/Aleson/Desktop/Ahmed Ali Qayyum Tasks/R/R programming tasks 3T7665535K250260B/name-transformed.txt", sep = ",",

row.names = TRUE, col.names = NA)

## Briefly explain the general relationship between each of your transformed variables and your variable of interest (the area).

**Solution**

Transforming data is one step in addressing data that does not fit model assumptions, and is also used to coerce different variables to have similar distributions.

A **variable** is any item or quantity that can vary or take on different values. The **variables of interest** in a statistical study are the items or quantities that the study seeks to measure. The explanatory **variable** may explain or cause the effect.

# 3. Build models and investigate the importance of each variable

## Download the AggWaFit.R file (from CloudDeakin) to your working directory and load into the R workspace using

**Solution**

source ("AggWaFit718.R")

Using the fitting functions to learn the parameters for:

• A weighted arithmetic mean,

**code:**

x <- c(0:10, 50)

xm <- mean(x)

c(xm, mean(x, trim = 0.10))

• Weighted power means with p = 0:5, and p = 2,

**Code:**

PM <- function(x,w =array(1/length(x),length(x)),p) { # 1. pre-defining the function inputs

if(p == 2 | p == 0:5) { # 2. condition for `if' statement

prod(x^w) # 3. what to do when (p==2 and p == 0:5) is TRUE

} else {(sum(w\*(x^p)))^(1/p)} # 4. what to do when (p==0) is FALSE }

• An ordered weighted averaging function,

**Code:**

OWA <- function(x,w=array(1/length(x),length(x)) ) {

# 1. pre-defining the inputs (with equal weights default)

w <- w/sum(w)

# 2. normalise the vector in case weights don't add to 1

sum(w\*sort(x))

# 3. OWA calculation }

• A Choquet integral.

Code:

choquet <- function(x,v) { # 1. pre-defining the inputs (no default)

n <- length(x) # 2. store the length of x

w <- array(0,n) # 3. create an empty weight vector

for(i in 1:(n-1)) { # 4. define weights based on order

v1 <- v[sum(2^(order(x)[i:n]-1))] # 4i. v1 is f-measure of set of all

# elements greater or equal to

# i-th smallest input.

v2 <- v[sum(2^(order(x)[(i+1):n]-1)) # 4ii. v2 is same as v1 except

without i-th smallest

w[i] <- v1 - v2 # 4iii. subtract to obtain w[i]

} #

w[n] <- 1- sum(w) #4iv. final weight leftover

x <- sort(x) # 5. sort our vector

sum(w\*x) # 6. calculate as we would WAM

}

weight/Parameter

code:

library(dplyr)

set.seed(1234)

df <- iris

df[,"weights"] <- rnorm(nrow(df),1,0.1 ) # generate randomized weights

head(df)

df %>%

group\_by(Species) %>%

summarise\_each(funs(sum(. \* weights , na.rm = TRUE), # Weighted Sum

weighted.mean(.,w = weights, na.rm = TRUE))) -> agg.df # Weighted Mean

agg.df

#Including Table add calculate error

newdata122 <- data[c(5,13)]

stderr (c(12,13))

## Include two tables in your report - one on the error measures, and one summarising the weights/parameters that were learned for your data.

**Solution**

The role of this additional bias term is really important. It significantly increases the capacity of your model.

## Compare and interpret the data in your tables. Be sure to comment on:

## How good the model is.

## b. The importance of each of the variables (the four variables that you have selected),

## c. Any interaction between any of those variables (are they complementary or redundant?) and

## d. Better models favor higher or lower inputs.

**Solution**

**Data** **Collection and** **Interpretation**.

**Data interpretation** is a part of daily life for most people. Interpretation is the process of making sense of numerical data that has been collected, analyzed, and presented. Models are a primary way to estimate multiple effects of alternative water resources system design and operating policies. Models predict the values of various performance indicators of a system. Their outputs are based on model structure, hydrological and other time-series inputs and a host of parameters whose values describe the system being simulated. The predictive importance and causal importance of a given input is that both depends on other inputs used in the network. Marginal importance considers each input in isolation.

# 4. Use your model for prediction

## Using your best fitting model, predict the area for the following input:

## X5=91.6; X6=181.3; X7=613; X8=7.6; X9=24.6; X10=44; X11=4; X12=0.

**Solution**

newdata121 <- data[c(5,6,7,8,9,10,11,12,13)]

> subset(newdata121, FFMC == 91.6 | DMC == 191.3 | DC == 613 | ISI == 7.6 | temp == 24.6 | RH == 44 | wind == 4 | rain == 0)

## Give your result and comment on whether you think it is reasonable.

**Solution**

Its important to be able to make predictions about something to foresee the effects or to consider a course of action. It’s even better if you know that your predictions are sound. In this post, the use of [regression analysis](http://statisticsbyjim.com/glossary/regression-analysis/) has been shown to make predictions and determine whether they are both unbiased and precise.

## Comment generally on the ideal conditions (in terms of your chosen four variables) under which an area will result.

**Solution**

Choosing a correct linear regression model can be difficult. Trying to model it with only a sample doesn’t make it any easier. It starts when a researcher wants to mathematically describe the relationship between some [predictors and the response variable](https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/what-are-response-and-predictor-variables/).