

**NAME : NAMAN JAIN**

**COLLEGE ROLL NO. : 19/1425**

**COURSE : B.SC.(H) CS**

**SEMISTER : 4th**

**R Programming**

**Practical File**

Question 1 : Write an R script to find subset of dataset by using subset (), aggregate () functions on iris dataset.

Solution :

Code :

data("iris") # getting iris dataset

str(iris) # iris dataset info

subset(iris,Sepal.Length > 5.0 & Sepal.Width > 4) # subset with Sepal.Length > 5 and Sepal.Width > 4

subset(iris,Petal.Length > 1.0& Petal.Width==0.4) # subset with Petal.Length>5 and Petal.Width>3.5

subset(iris,Species=="setosa" & Sepal.Length > 5.4) # subset with Species of “setosa” and Sepal’s Length > 5.4

iris\_num <- iris[-5] # getting only numeric type column

str(iris\_num) # updated dataset info

aggregate(iris\_num,by=list(iris$Species),FUN = sum) # Sum by species type

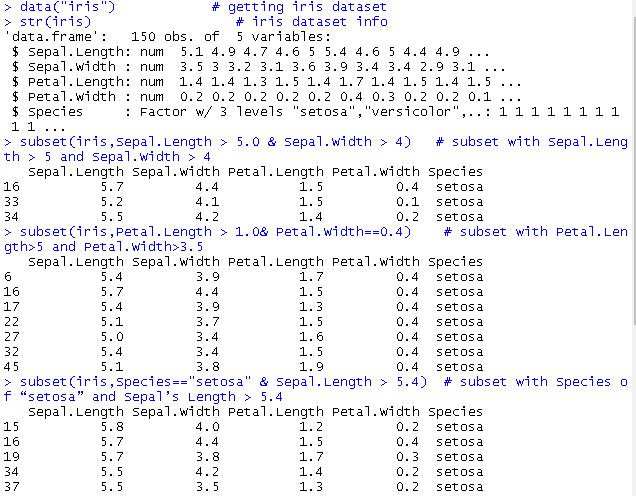
aggregate(iris\_num,by=list(iris$Species),FUN = mean) # Mean by species type

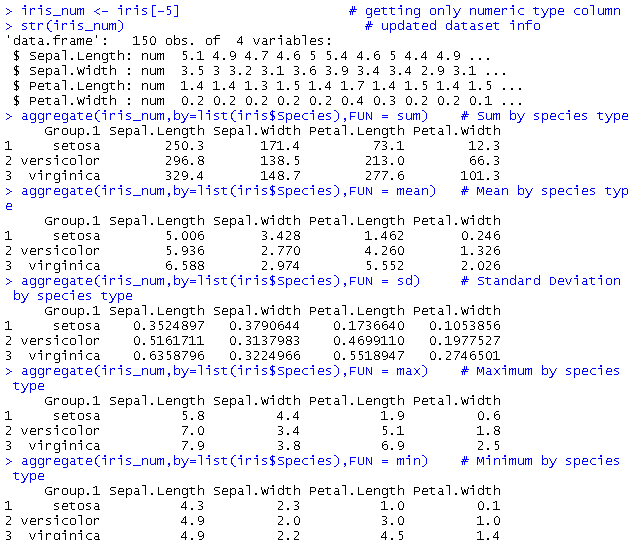
aggregate(iris\_num,by=list(iris$Species),FUN = sd) # Standard Deviation by species type

aggregate(iris\_num,by=list(iris$Species),FUN = max) # Maximum by species type

aggregate(iris\_num,by=list(iris$Species),FUN = min) # Minimum by species type

Output :





Question 2 : Write an R script to find basic descriptive statistics using summary, str, quartile function on mtcars & cars datasets.

Solution :

Code :

data("mtcars") # getting mtcars dataset

summary(mtcars) # summary of mtcars dataset

str(mtcars) # dataset info

quantile(mtcars$disp) # getting quantiles for displacement

quantile(mtcars$wt) # getting quantiles for weight

data("cars") # getting cars dataset

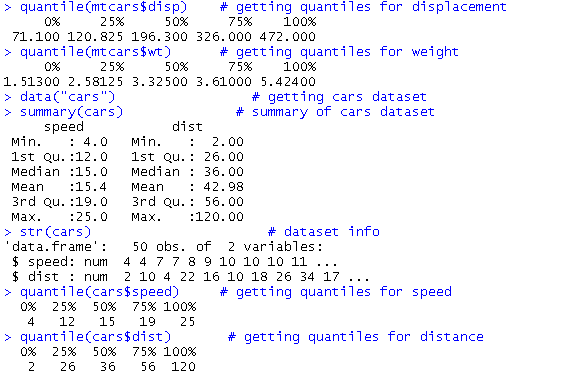
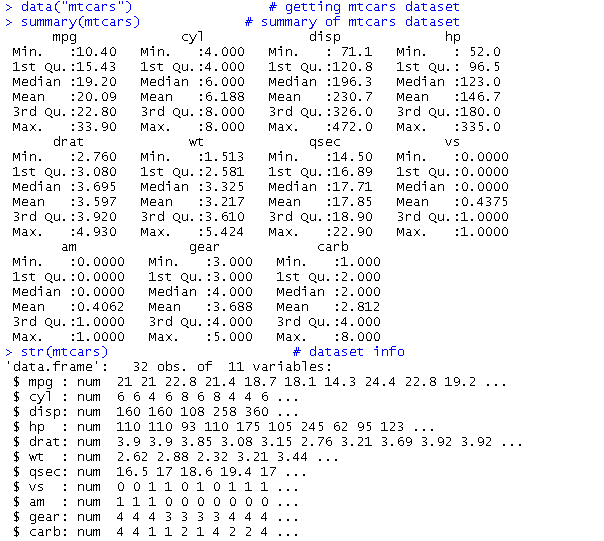
summary(cars) # summary of cars dataset

str(cars) # dataset info

quantile(cars$speed) # getting quantiles for speed

quantile(cars$dist) # getting quantiles for distance

Output :



Question 3 : Reading different types of data sets (.txt, .csv) from web or from existing data set, do the following:

a. Reading Excel data sheet in R.

b. Reading CSV dataset in R.

Solution :

Code :

# Part (a)

install.packages("readxl") # installing required package “readxl”

library(readxl) # getting library of package “readxl”

url <- "<http://www.eia.gov/petroleum/drilling/xls/dpr-data.xlsx>" # url for xlsx file

dm<-basename(url)

download.file(url = url, destfile = dm,mode = "wb") # downloading xlsx file

data\_excel<-read\_excel(dm) # reading xlsx file in read\_excel variable

dim(data\_excel) # dimensions of xlsx file

head(data\_excel) # showing first 5 entries

# part (b)

# url for csv file

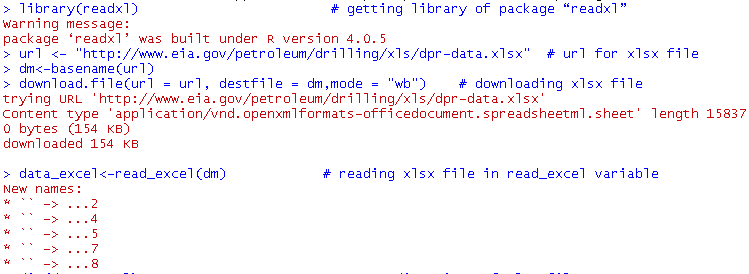
url <- "http://gattonweb.uky.edu/sheather/book/docs/datasets/magazines.csv"

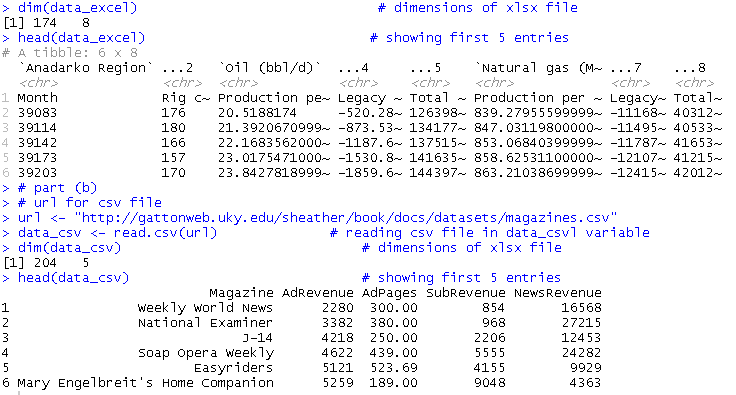
data\_csv <- read.csv(url) # reading csv file in data\_csvl variable

dim(data\_csv) # dimensions of xlsx file

head(data\_csv) # showing first 5 entries

Output :



****

Question 4 : Write an R program to create a vector of a specified type and length. Create a vector of numeric, complex, logical and character types of length 6.

Solution :

Code :

x = vector("numeric", 5)

print("Numeric Vector:")

print(x)

c = vector("complex", 5)

print("Complex Vector:")

print(c)

l = vector("logical", 5)

print("Logical Vector:")

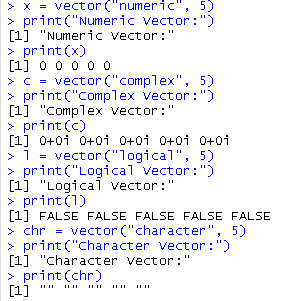
print(l)

chr = vector("character", 5)

print("Character Vector:")

print(chr)

Output :



Question 5 : Write an R program to append value to a given empty vector.

Solution :

Code :

vector = c()

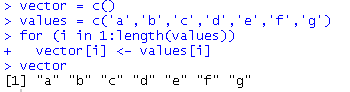
values = c('a','b','c','d','e','f','g')

for (i in 1:length(values))

vector[i] <- values[i]

vector

Output :



Question 6 : Write an R program to reverse the order of given vector.

Solution :

Code :

v = c(0, 10, 10, 10, 20, 30, 40, 40, 40, 50, 60)

print("Original vector-:")

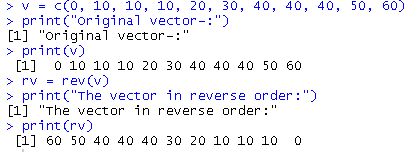
print(v)

rv = rev(v)

print("The vector in reverse order:")

print(rv)

Output :



Question 7 : Write an R program to convert given data frame column(s) to a vector.

Solution :

Code :

dfc1 = c(1, 2, 3, 4, 5)

dfc2 = c(6, 7, 8, 9, 10)

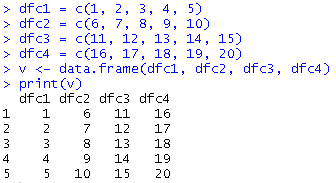
dfc3 = c(11, 12, 13, 14, 15)

dfc4 = c(16, 17, 18, 19, 20)

v <- data.frame(dfc1, dfc2, dfc3, dfc4)

print(v)

Output :



Question 8 : (a) Write an R program to create a list containing strings, numbers, vectors and a logical value.

(b) Write an R program to convert a given list to a vector.

Solution :

Code :

# part (a)

list\_data<-list(3,7,12.5,34.7,"hello","world",TRUE,FALSE)

print("List Data is :")

print(list\_data)

# part (b)

l1<-list(1,3,"Hii",TRUE)

print("Original list :")

print(l1)

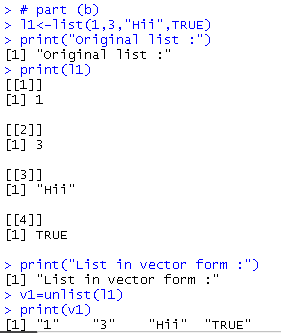
print("List in vector form :")

v1=unlist(l1)

print(v1)

Output :





Question 9 : Write an R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.

Solution :

Code :

list\_data <- list(c("Red","Green","Black"), matrix(c(1,3,5,7,9,11), nrow = 2),

list("Python", "PHP", "Java"))

print("List:")

print(list\_data)

names(list\_data) = c("Color", "Odd numbers", "Language(s)")

print("List with column names:")

print(list\_data)

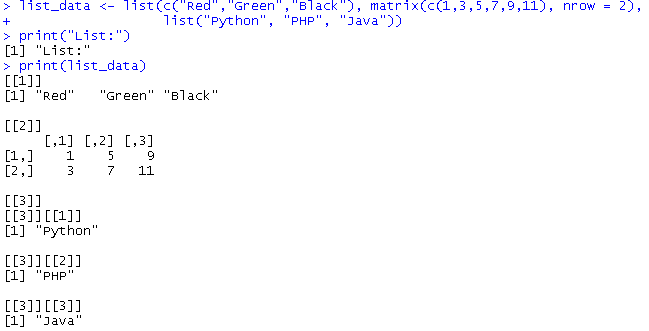
print('1st element:')

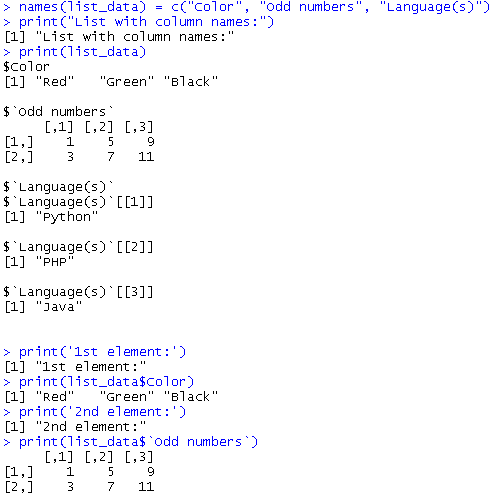
print(list\_data$Color)

print('2nd element:')

print(list\_data$`Odd numbers`)

Output :





Question 10 : Write an R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors.

Solution :

Code :

print("Two vectors of different lengths:")

v1 = c(1,3,4,5)

v2 = c(10,11,12,13,14,15)

print(v1)

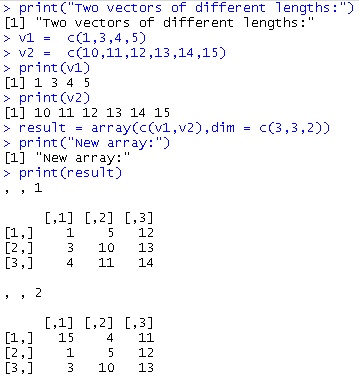
print(v2)

result = array(c(v1,v2),dim = c(3,3,2))

print("New array:")

print(result)

Output :



Question 11 : (a) Write an R program to find the maximum and the minimum value of a given vector.

(b) Write a R program to create a list of heterogeneous data, which include character, numeric and logical vectors. Print the lists.

Solution :

Code :

# part (a)

nums = c(10, 20, 30, 40, 50, 60)

print('Original vector:')

print(nums)

print(paste("Maximum value of the said vector:",max(nums)))

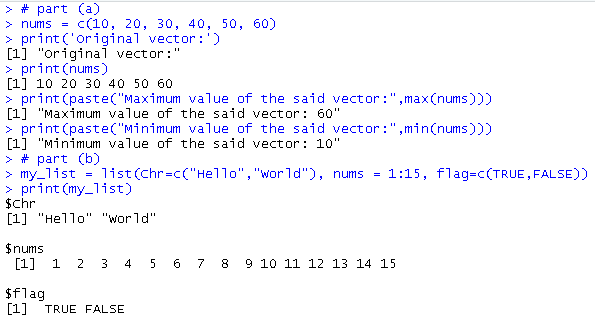
print(paste("Minimum value of the said vector:",min(nums)))

# part (b)

my\_list = list(Chr=c("Hello","World"), nums = 1:15, flag=c(TRUE,FALSE))

print(my\_list)

Output :



Question : 12 Import data set Midwest and do the following :

1. By using ggplot() plot a graph for area v/s poptotal (total population).
2. Plot a bar plot for area v/s poptotal (total population).
3. Plot a pie chart for state, area, poptotal, popdensity, popwhite, popblack.

Solution :

Code :

# part (a)

install.packages("ggplot2")

library(ggplot2)

data("midwest")

ggplot(data = midwest,mapping = aes(x = area,y =poptotal))+geom\_point()

# part (b)

data("midwest")

ggplot(data = midwest, aes(x=area, y=poptotal)) +

geom\_bar(stat="identity", width=.001, fill="tomato3")

# part (c)

midarea <- aggregate(area~state,midwest,FUN = 'sum')

pie(midarea$area, midarea$state, main = "Area Per State")

midpopt <- aggregate(poptotal~state,midwest,FUN = 'sum')

pie(midpopt$poptotal, midpopt$state, main = "Total Population Per State")

middnsty <- aggregate(popdensity~state,midwest,FUN = 'sum')

pie(middnsty$popdensity, middnsty$state, main = "Population density Per State")

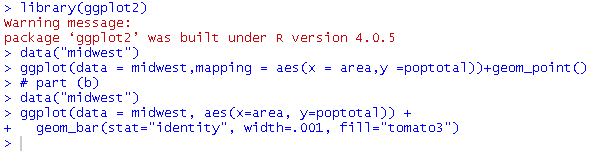
midpopw <- aggregate(popwhite~state,midwest,FUN = 'sum')

pie(midpopw$popwhite, midpopw$state, main = "White Population Per State")

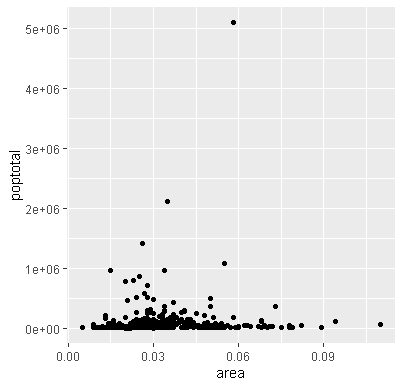
midpopb <- aggregate(popblack~state,midwest,FUN = 'sum')

pie(midpopb$popblack, midpopb$state, main = "Black Population Per State")

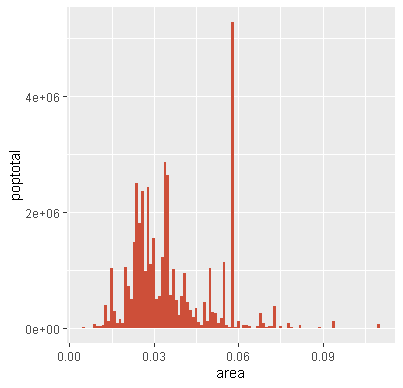
Output :



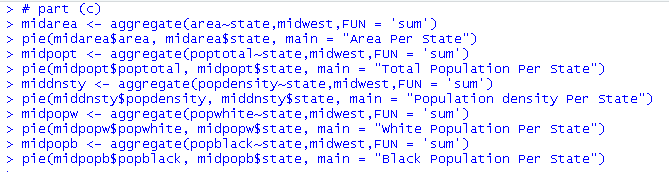
(a)

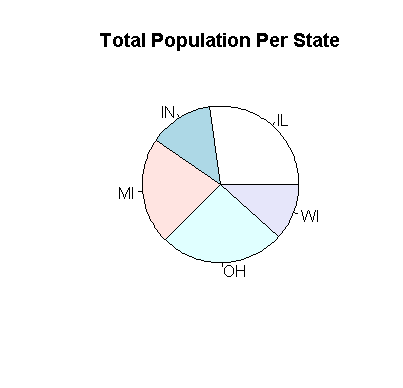
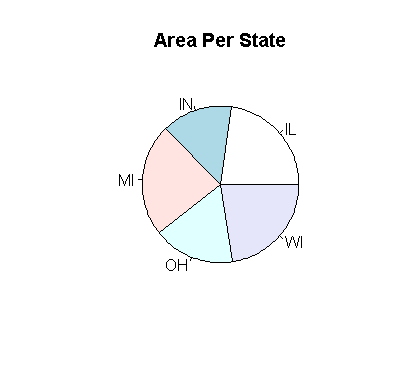


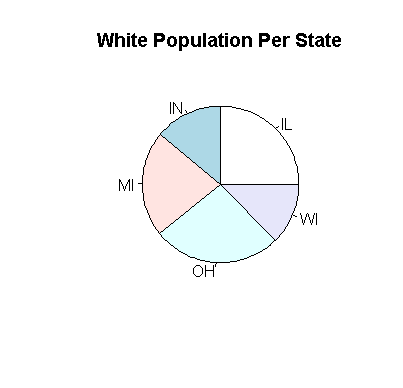
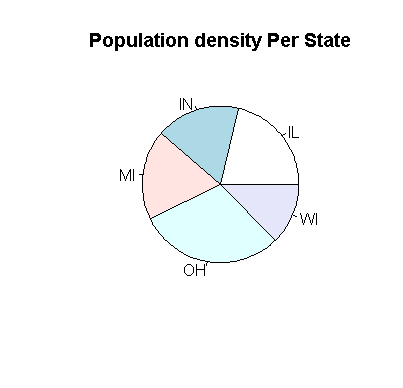
(b)

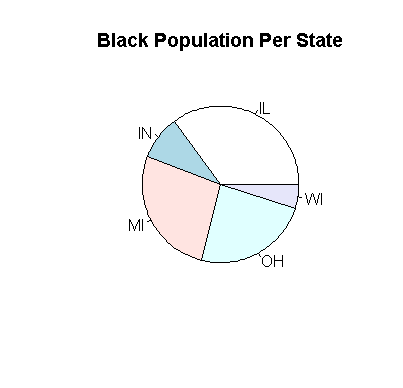


(c)

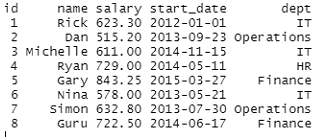








Question 13 : Create an employe table having :



1. Get the max salary from the data frame.
2. Get the person details having max salary.
3. Get all the people working in the IT department.
4. Get the persons in the IT department whose salary is greater than 600.
5. Get the people who joined on or after 2014.

Solution :

Code :

# creating employee table

id<-1:8

name<-c("Rick","Dan","Michelle","Ryan","Gary","Nina","Simon","Guru")

salary<-c(623.30,515.20,611.00,729.00,843.25,578.00,632.80,722.50)

start\_date<-c(as.Date('2012-01-01'),as.Date('2013-09-23'),as.Date('2014-11-15'),as.Date('2014-05-11'),as.Date('2015-03-27'),as.Date('2013-05-21'),as.Date('2013-07-30'),as.Date('2014-06-17'))

dept<-c("IT","Operations","IT","HR","Finance","IT","Operations","Finance")

employee<-data.frame(id,name,salary,start\_date,dept)

names(employee)<-c("ID","Name","Salary","Start\_Date","Dept")

# part (a)

max\_salary<-max(employee["Salary"])

max\_salary

employee

install.packages("tidyverse")

library(tidyverse)

# part (b)

employee %>% filter(employee$Salary==max\_salary)

# part (c)

employee %>% filter(employee$Dept=="IT")

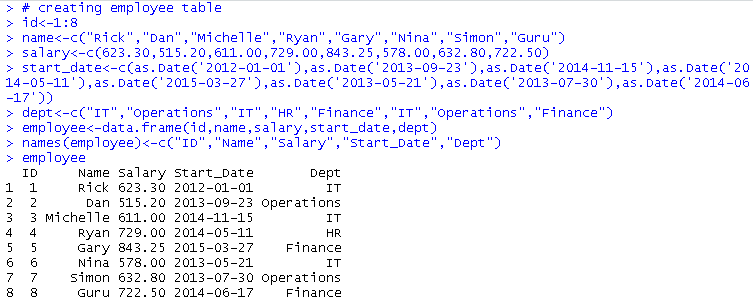
# part (d)

employee %>% filter(employee$Dept=="IT",employee$Salary>600)

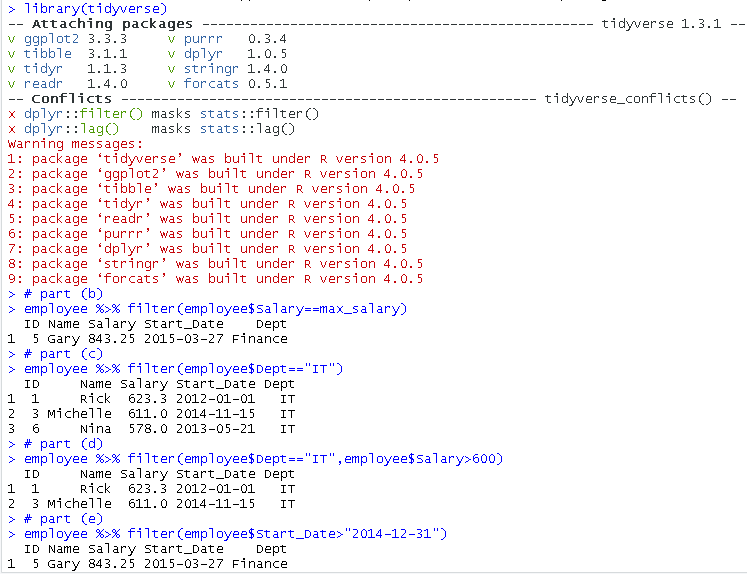
# part (e)

employee %>% filter(employee$Start\_Date>"2014-12-31")

Output :







Question 14 : In the library MASS is a dataset UScereal which contains information about popular breakfast cereals. Attach the data set and use different kinds of plots to investigate the following relationships :

1. Relationship between manufacturer and shelf
2. Relationship between fat and vitamins
3. Relationship between fat and shelf
4. Relationship between carbohydrates and sugars
5. Relationship between fibre and manufacturer
6. Relationship between sodium and sugars

Solution :

Code :

install.packages("MASS")

library(MASS)

data("UScereal")

attach(UScereal)

library(ggplot2)

library(lattice)

# a) relationship between manufacturer and shelf

table(mfr,shelf)

# b) relationship between fat and vitamins

xyplot(vitamins~fat, UScereal)

xyplot(vitamins~jitter(fat,100), UScereal)

stripchart(fat~vitamins, pch = 19, method = "jitter")

stripplot(vitamins~jitter(fat,100))

ggplot(UScereal, aes(fat, vitamins)) + geom\_point(position = "jitter")

# c) relationship between fat and shelf

ggplot(UScereal,aes(fat, shelf)) + geom\_point(position = "jitter")

# d) relationship between carbohydrates and sugars

ggplot(UScereal,aes(carbo,sugars)) + geom\_point()

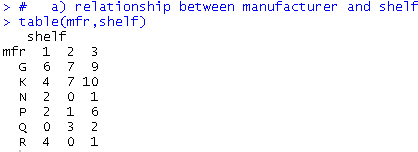
# e) relationship between fibre and manufacturer

ggplot(UScereal,aes(mfr,fibre)) + geom\_point(position = "jitter")

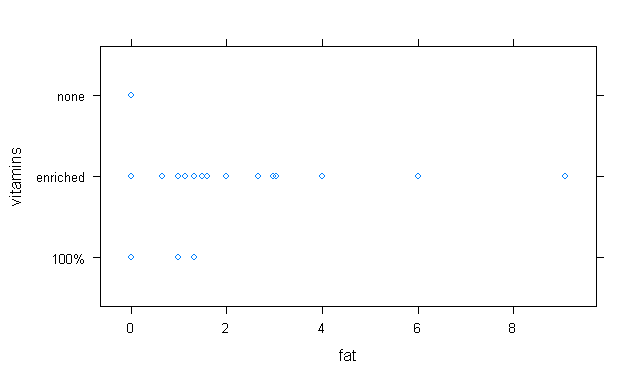
# f) relationship between sodium and sugars

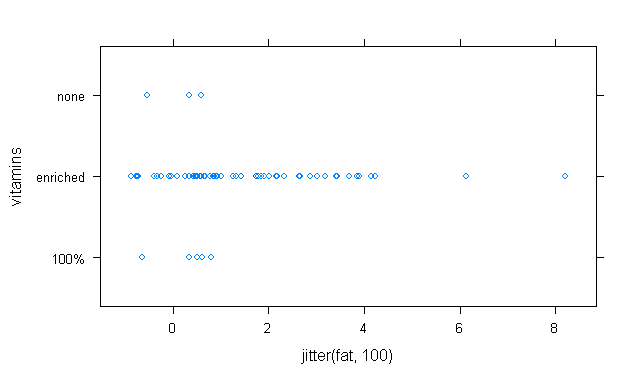
ggplot(UScereal,aes(sodium,sugars)) + geom\_point()

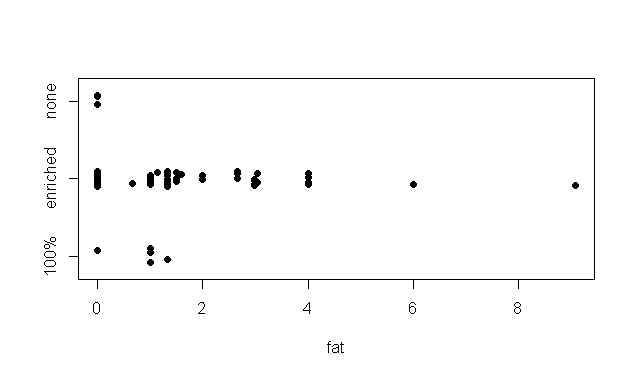
Output : (a)

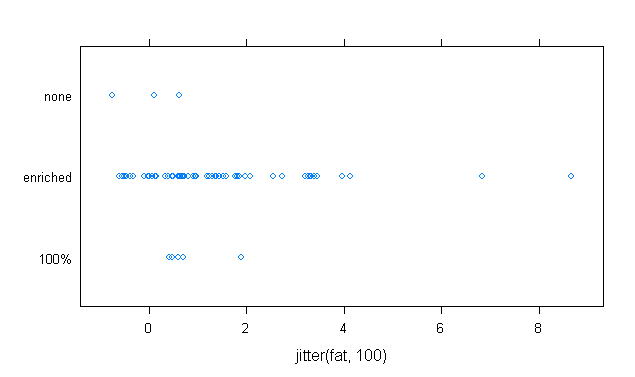


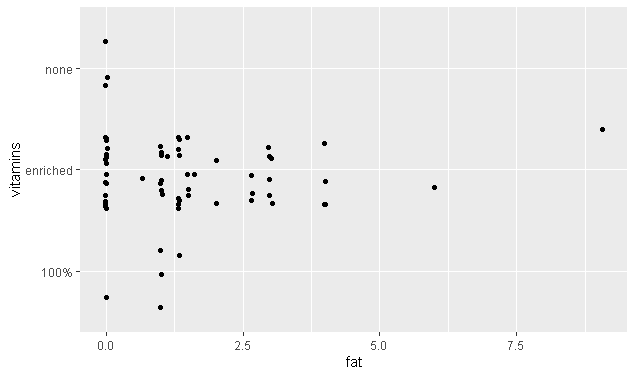
(b)



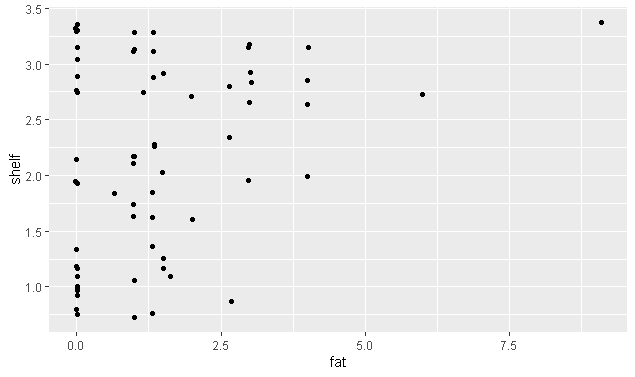




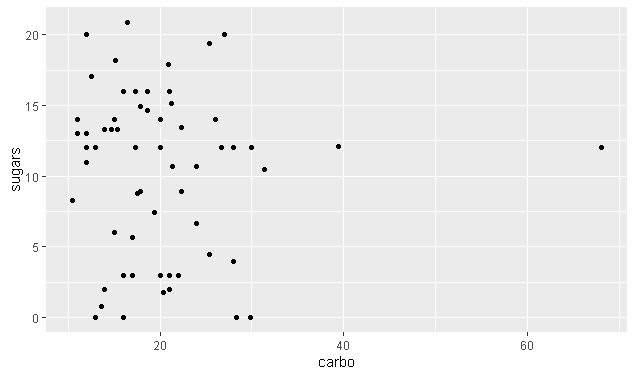




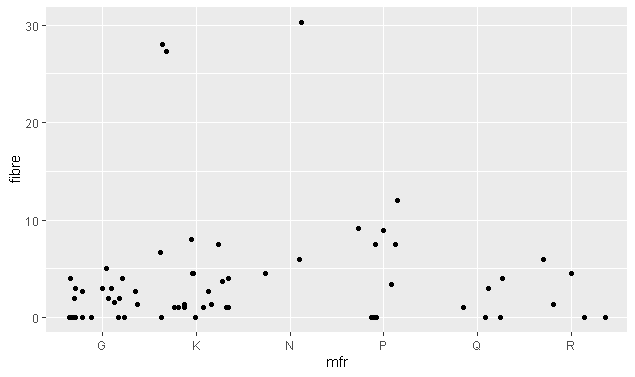
(c)



(d)



(e)



(f)

