R_studio

Christoffer Mondrup Kramer

2023-04-22

Ex. 2a: 09-02-2023

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library("scatterplot3d")
```

Custom functions

Here is the collections for custom functions I have created myself

```
#Plotting marginal dot diagram
margin_dot_plot <- function(x, y, xlabel = "x", ylabel = "y") {
layout(mat = matrix(c(2, 0, # First column
                      1, 3), # Second column
                        nrow = 2,
                        ncol = 2),
       heights = c(6, 2),    # Heights of the two rows
       widths = c(1, 6))     # Widths of the two columns
par(mar = c(4, # Bottom
            4, # Left
            0.1, # Top
            0.1))# Right

plot(x, y, xlab = "", ylab = "")

stripchart(y, method = "stack", at = 0,
           pch = 16, col = "darkgreen", frame = FALSE, vertical = TRUE, ylab = ylabel)
stripchart(x, method = "stack", at = 0,
           pch = 16, col = "darkgreen", frame = FALSE, xlab = xlabel)
}


# Arithmetic mean (pp. 6-7)
my_mean <- function(my_list) {
  n = length(my_list)
  return( (1/n) * sum(my_list))
}


# Variance for single variable (p .7)
my_single_sample_variance <- function(my_list) {
  n = length((my_list))
  x_mean = mean(my_list)

  total_res = 0
  for (x in my_list) {
    inner_res = (x - x_mean)^2
    total_res = inner_res + total_res
  }
  return ((1/(n - 1)) * total_res)
}

# Co variance (p. 7)
my_sample_covar <- function(my_list_1, my_list_2) {
  n <- length(my_list_1)

  list_1_mean <- my_mean(my_list_1)
  list_2_mean <- my_mean(my_list_2)

  res <- 0
  for (i in 1:n) {
    list_1_res <- my_list_1[i] - list_1_mean
    list_2_res <- my_list_2[i] - list_2_mean
    temp_res <- (list_1_res * list_2_res)
    res <- res + temp_res
```

```
  }
  return((1/(n-1)) * res)
}

# Correlation coefficient (p. 8)
my_cor_coef <- function(my_list_1, my_list_2){
  covariance <- my_sample_covar(my_list_1, my_list_2)
  var_list_1 <- my_single_sample_variance(my_list_1)
  var_list_2 <- my_single_sample_variance(my_list_2)

  res <- covariance / ( sqrt(var_list_1) * sqrt(var_list_2) )
  return(res)
}


# Mean array
my_mean_array <- function(df){
  mean_array <- numeric()

  i <- 1
  for (colname in colnames(df)) {
    mean_array[i] <- my_mean(df[,colname])
    i <- i + 1
  }
  return(mean_array)
}
```

1.4 - p. 38

The world's 10 largest companies yields the following data:

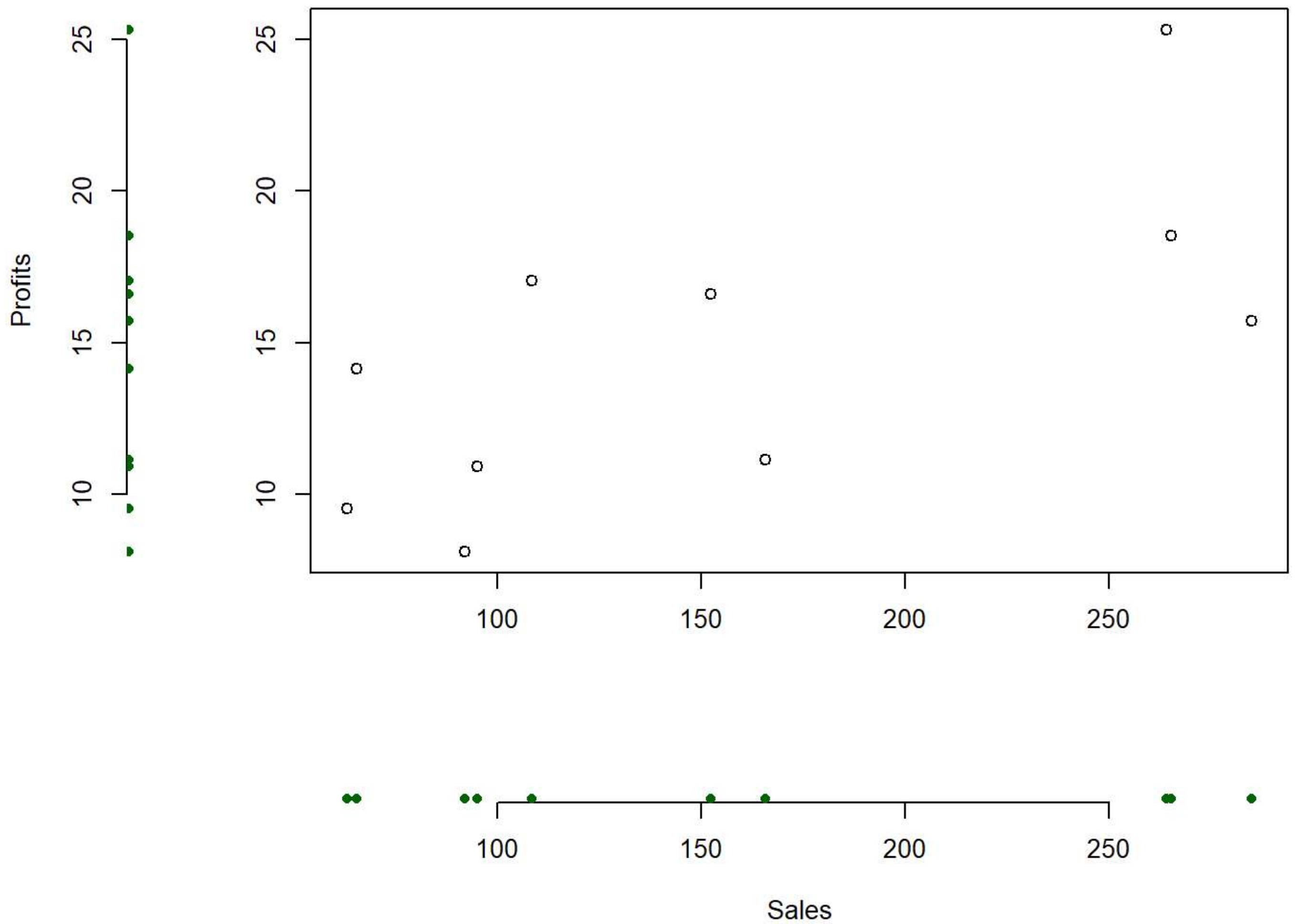
```
company <- c("Citigroup", "General Electric", "American Intl Group", "Bank of America", "HSBC Group", "ExonMobil", "Royal Dutch/shell", "BP", "ING Group", "Toyota Motor")

# x1
sales <- c(108.28, 152.36, 95.04, 65.45, 62.97, 263.99, 265.19, 285.06, 92.01, 165.68)

# x2
profits <- c(17.05, 16.59, 10.91, 14.14, 9.52, 25.33, 18.54, 15.73, 8.10, 11.13)

# x3
assets <- c(1484.10, 750.33, 766.42, 1110.46, 1031.29, 195.26, 193.83, 191.11, 1175.16, 211.15)

company_df <- data.frame(company,
                         sales,
                         profits,
                         assets)
```

```
print(company_df)
```

```
##                 company  sales profits  assets
## 1             Citigroup 108.28   17.05 1484.10
## 2      General Electric 152.36   16.59  750.33
## 3   American Intl Group  95.04   10.91  766.42
## 4       Bank of America  65.45   14.14 1110.46
## 5            HSBC Group  62.97    9.52 1031.29
## 6             ExonMobil 263.99   25.33  195.26
## 7    Royal Dutch/shell 265.19   18.54  193.83
## 8                    BP 285.06   15.73  191.11
## 9             ING Group  92.01    8.10 1175.16
## 10         Toyota Motor 165.68   11.13  211.15
```

a) Plot the scatter diagram and the marignal dot diagrams for variables x_1 and x_2

```
margin_dot_plot(company_df$sales, company_df$profits, xlabel =  "Sales", ylabel = "Profits")
```



b) Compute $\bar{x}_1$, $\bar{x}_2$, s_{11}, s_{22}, s_{12}, r_{12}. Interpret r_{12}

```
#x1 mean
print("x1 mean")
```

```
## [1] "x1 mean"
```

```
my_mean(company_df$sales)
```

```
## [1] 155.603
```

```
mean(company_df$sales)
```

```
## [1] 155.603
```

```
# x2 mean
print("x2 mean")
```

```
## [1] "x2 mean"
```

```
my_mean(company_df$profits)
```

```
## [1] 14.704
```

```
mean(company_df$profits)
```

```
## [1] 14.704
```

```
#s11 (variance)
print("x1 variance")
```

```
## [1] "x1 variance"
```

```
my_single_sample_variance(company_df$sales)
```

```
## [1] 7476.453
```

```
var(company_df$sales)
```

```
## [1] 7476.453
```

```
# s22 (variance)
print("x2 variance")
```

```
## [1] "x2 variance"
```

```
my_single_sample_variance(company_df$profits)
```

```
## [1] 26.19032
```

```
var(company_df$profits)
```

```
## [1] 26.19032
```

```
# s12 (covariance)
print("x1 and x2 covariance")
```

```
## [1] "x1 and x2 covariance"
```

```
my_sample_covar(company_df$sales, company_df$profits)
```

```
## [1] 303.6186
```

```
cov(company_df$sales, company_df$profits)
```

```
## [1] 303.6186
```

```
# r12 (correlation coefficient)
print("Correlation coeficient")
```

```
## [1] "Correlation coeficient"
```

```
my_cor_coef(company_df$sales, company_df$profits)
```

```
## [1] 0.686136
```

```
cor(company_df$sales, company_df$profits)
```

```
## [1] 0.686136
```

Since the r value or correlation is above 0, there is a positive correlation between sales and profits. Ie. the more sales the higher profits. Which also makes perfect sense. Since it is more than 0.5 there is even a strong positive correlation. see https://www.scribbr.com/statistics/pearson-correlation-coefficient (https://www.scribbr.com/statistics/pearson-correlation-coefficient)

1.5 Use the data from previously

a) Plot the scatter and dot diagrams for (x_2, x_3) and (x_1, x_3).

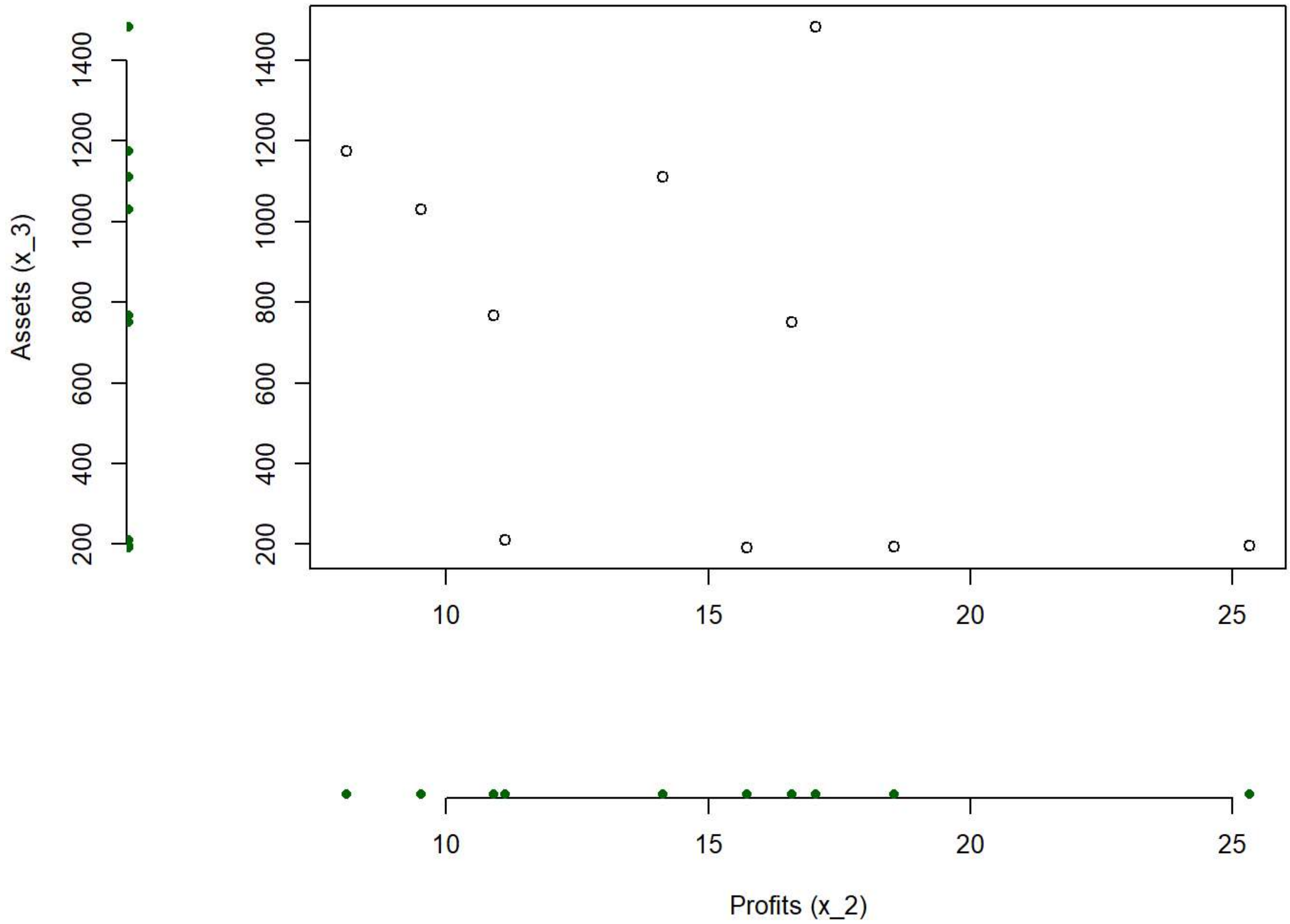
Comment on the patterns

```
x_1 <- company_df$sales
x_2 <- company_df$profits
x_3 <- company_df$assets

new_df <- data.frame(x_1,
                     x_2,
                     x_3)
print(new_df)
```

```
##       x_1   x_2     x_3
## 1  108.28 17.05 1484.10
## 2  152.36 16.59  750.33
## 3   95.04 10.91  766.42
## 4   65.45 14.14 1110.46
## 5   62.97  9.52 1031.29
## 6  263.99 25.33  195.26
## 7  265.19 18.54  193.83
## 8  285.06 15.73  191.11
## 9   92.01  8.10 1175.16
## 10 165.68 11.13  211.15
```

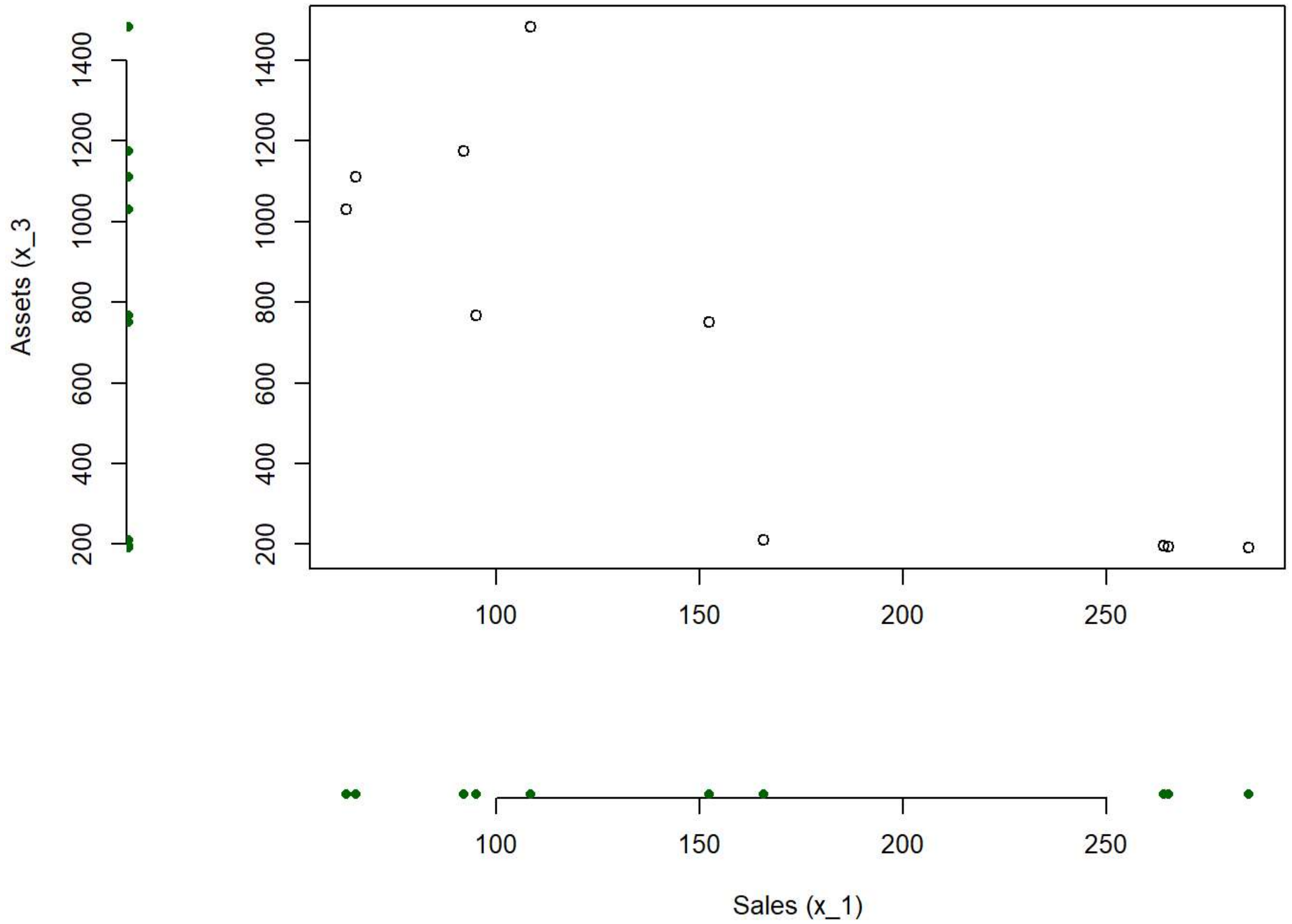
(x_2, x_3)

```
margin_dot_plot(x_2, x_3, xlabel = "Profits (x_2)", ylabel = "Assets (x_3)")
```



In General it is hard to see a clear pattern between the profits and the assets visually. IOf anything it might be negative correlated, but I am unsure. The varuance seem to be higher for the profits and assets seem to clump more together. (x_1, x_3)

```
margin_dot_plot(x_1, x_3, xlabel = "Sales (x_1)", ylabel = "Assets (x_3")
```



here there seem to be a negative correlation, with more sales leading to fewer assets, which might indicate that they are emptying there stock, this might explain the negative correlation before. Sales are likewiser more variance, but have a tendency to cluster.

Compute the $\bar{x}, S_n, R$

${x}

```
my_mean_array(new_df)
```

```
## [1] 155.603  14.704 710.911
```

```
colMeans(new_df)
```

```
##     x_1     x_2     x_3
## 155.603  14.704 710.911
```

S_n

```
cov(new_df)
```

```
##             x_1         x_2        x_3
## x_1   7476.4532   303.61862 -35575.960
## x_2    303.6186    26.19032  -1053.827
## x_3 -35575.9596 -1053.82739 237054.270
```

R

```
cor(new_df
    )
```

```
##            x_1        x_2        x_3
## x_1  1.0000000  0.6861360 -0.8450549
## x_2  0.6861360  1.0000000 -0.4229366
## x_3 -0.8450549 -0.4229366  1.0000000
```

1.7 (p. 40)

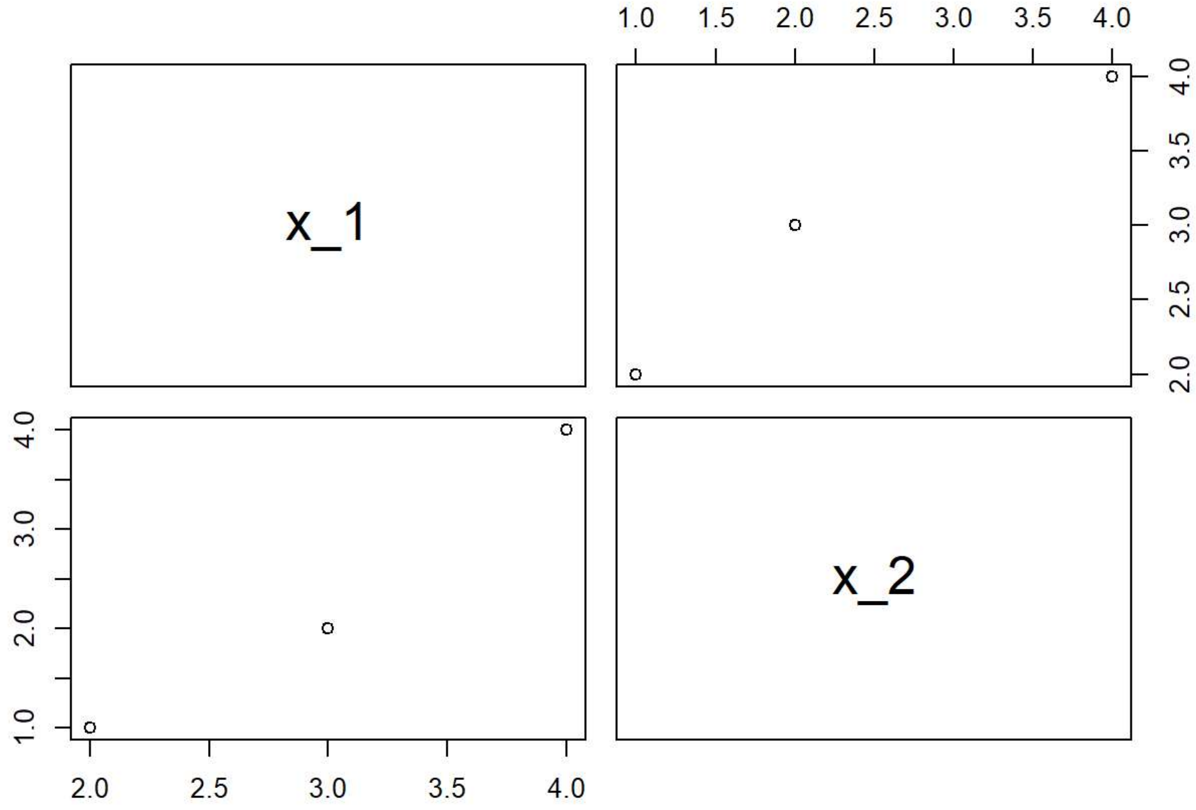
You are given the following n = 3 observations and p = 2 variables:

```
x_1 <- c(2, 3, 4)
x_2 <- c(1, 2, 4)

df <- data.frame(x_1,
                 x_2)
print(df)
```

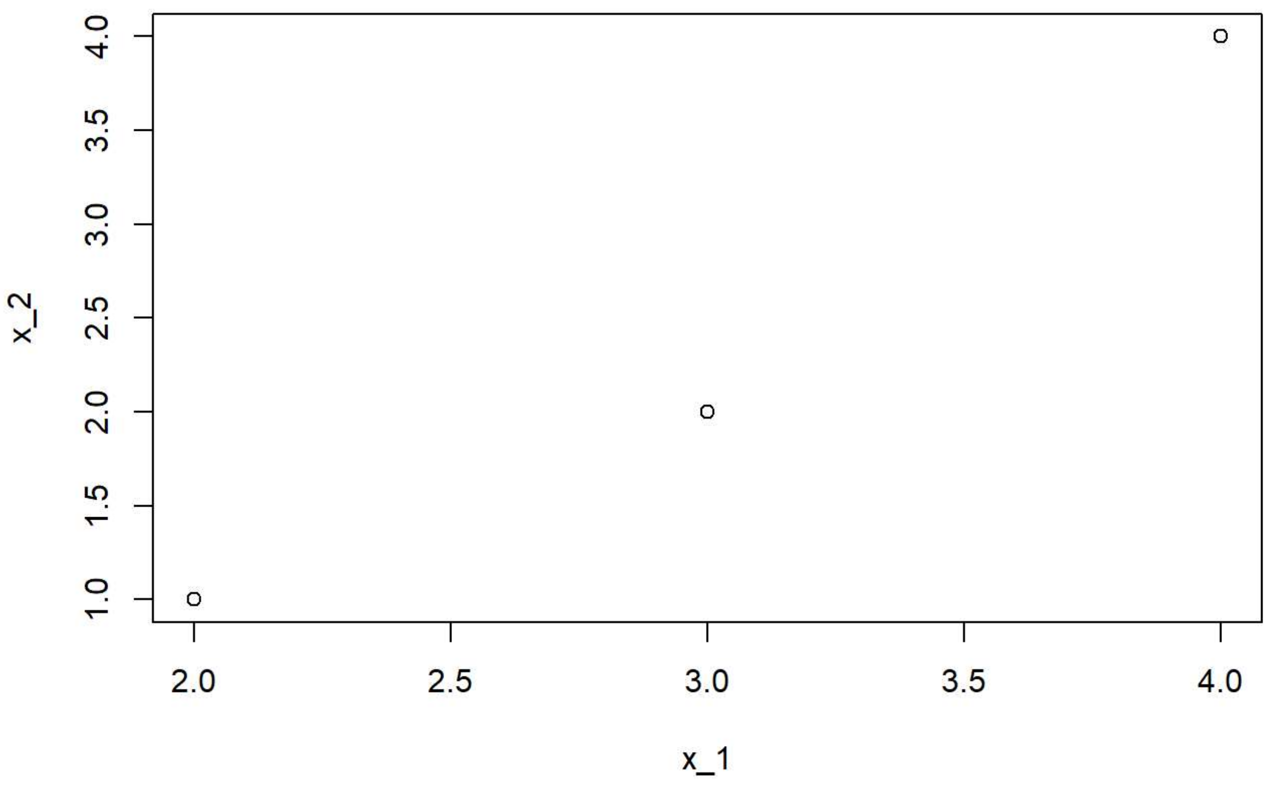
```
##   x_1 x_2
## 1   2   1
## 2   3   2
## 3   4   4
```

a) Plot the pairs of observations in the two dimensional variable space. That is, construct a two-dimensional scatter plot of the data

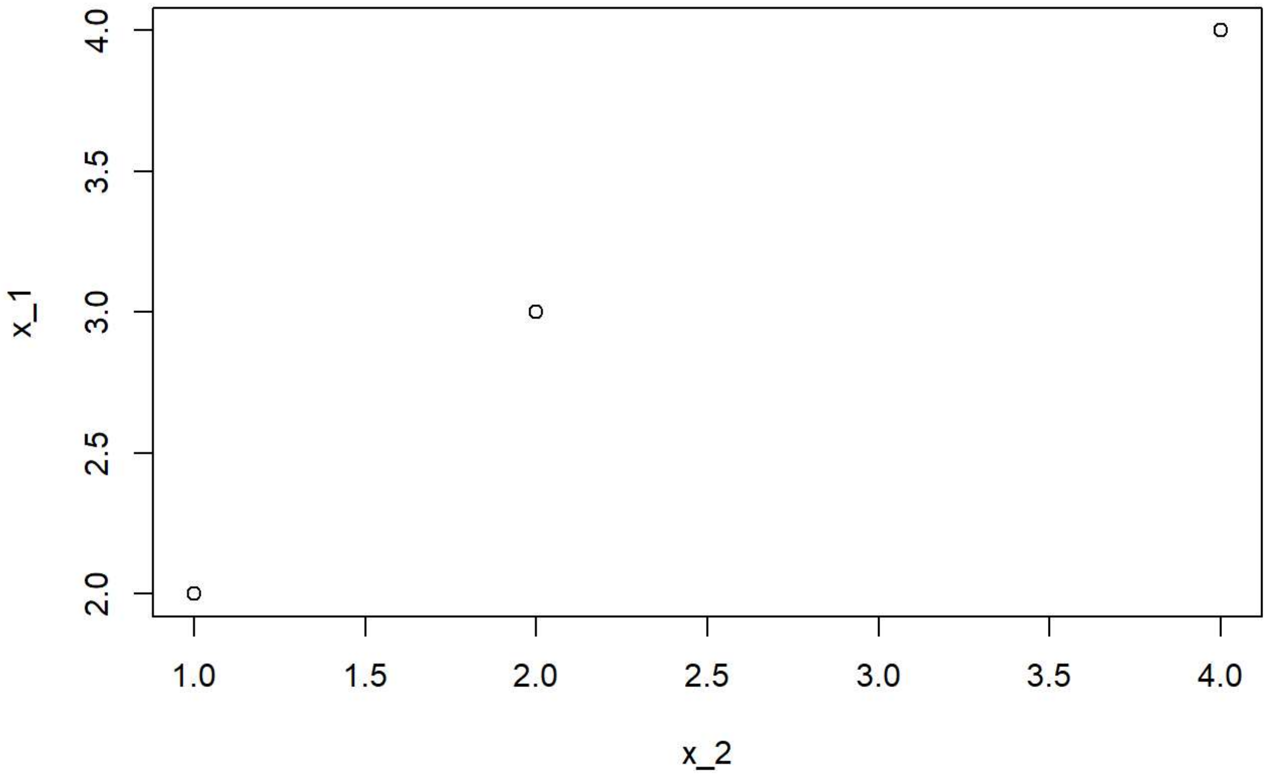
```
pairs(df)
```




```
plot(x_1, x_2)
```



```
plot(x_2, x_1)
```

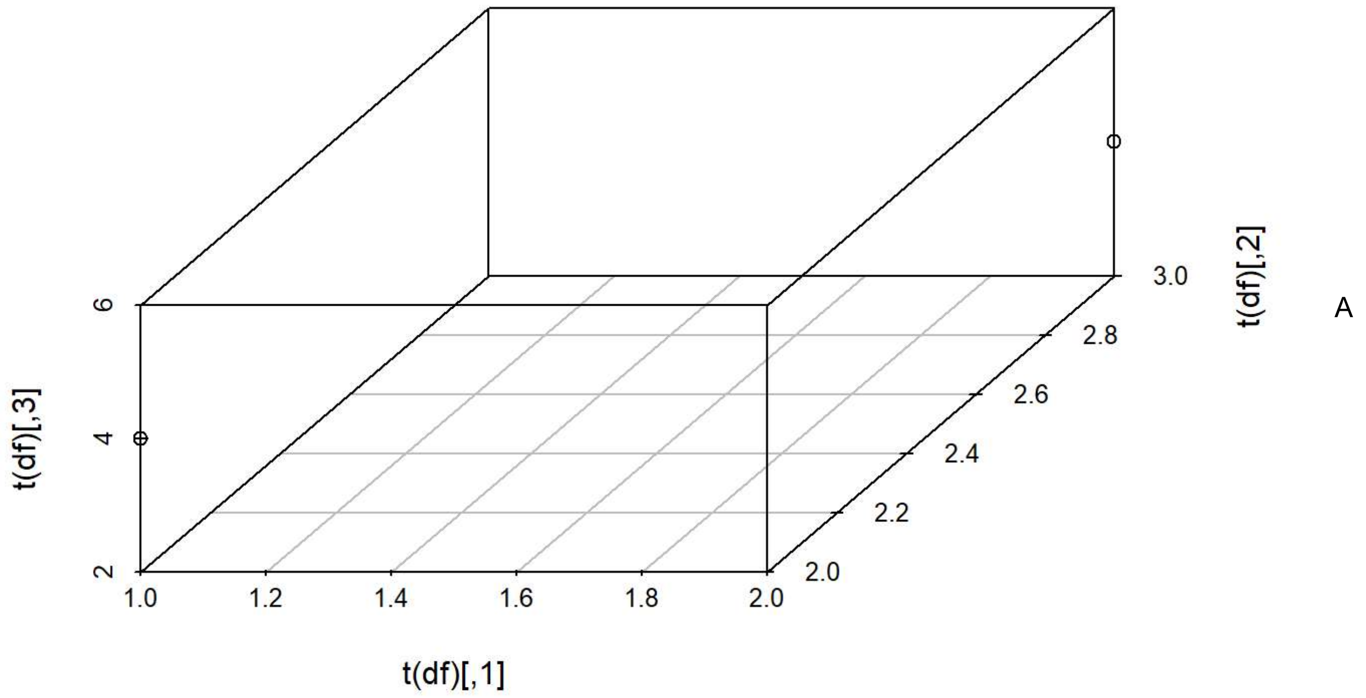


b) Plot the data as two points in the three dimensional item space

```
transposed <- t(df)
print(transposed)
```

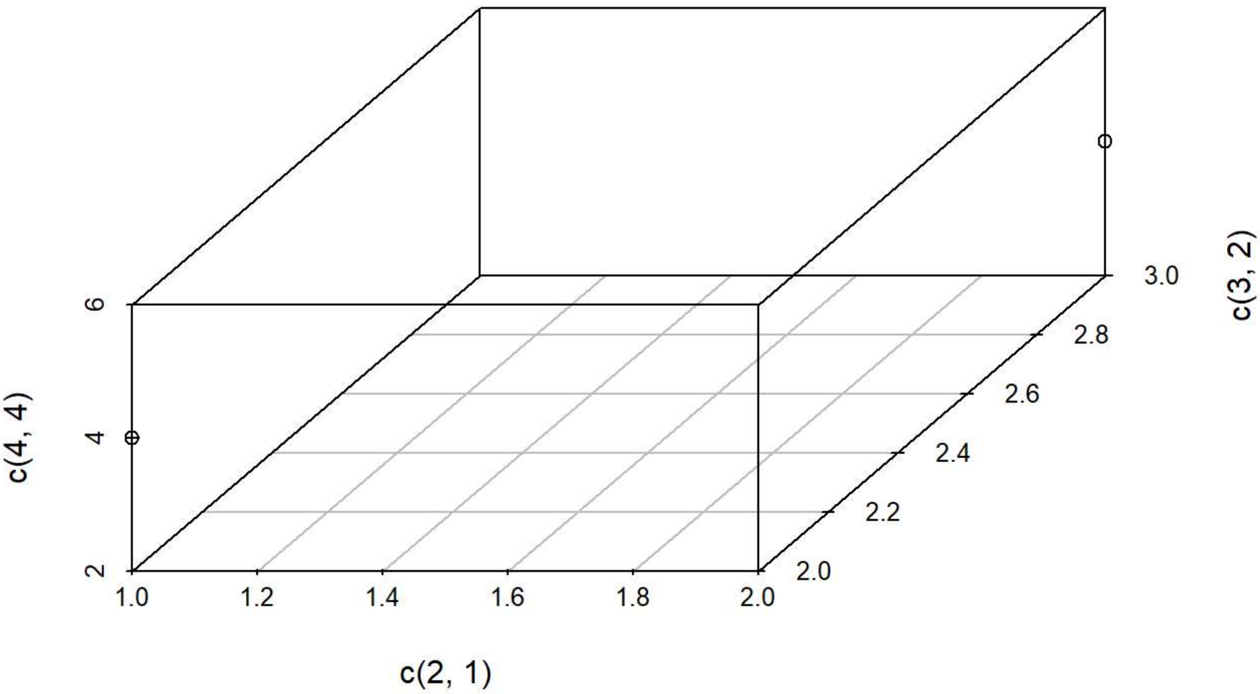
```
##     [,1] [,2] [,3]
## x_1    2    3    4
## x_2    1    2    4
```

```
scatterplot3d(t(df))
```



bit prettier

```
scatterplot3d(x = c(2,1), y = c(3,2), z = c(4,4))
```



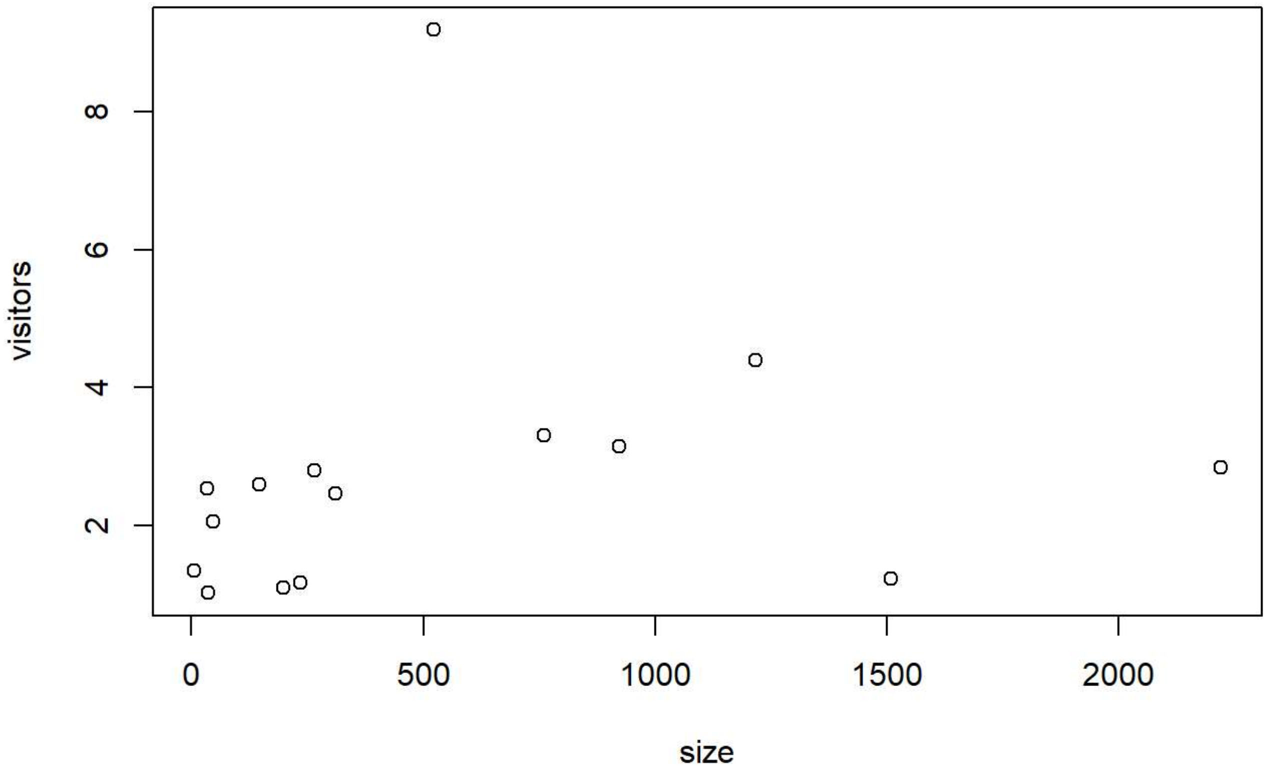
1.27 (p. 46) Table 1.11 presents the 2005 attendence (millions) at the fifteen most visisted national parks and their size (acres)

```
parks_df <- read.table("T1-11.dat")
colnames(parks_df)[1] <- "size"
colnames(parks_df)[2] <- "visitors"
print(parks_df)
```

```
##      size visitors
## 1    47.4     2.05
## 2    35.8     1.02
## 3    32.9     2.53
## 4  1508.5     1.23
## 5  1217.4     4.40
## 6   310.0     2.46
## 7   521.8     9.19
## 8     5.6     1.34
## 9   922.7     3.14
## 10  235.6     1.17
## 11  265.8     2.80
## 12  199.0     1.09
## 13 2219.8     2.84
## 14  761.3     3.30
## 15  146.6     2.59
```

a) Create a scatter plot and calculate the correlation coeeficient

```
plot(parks_df)
```



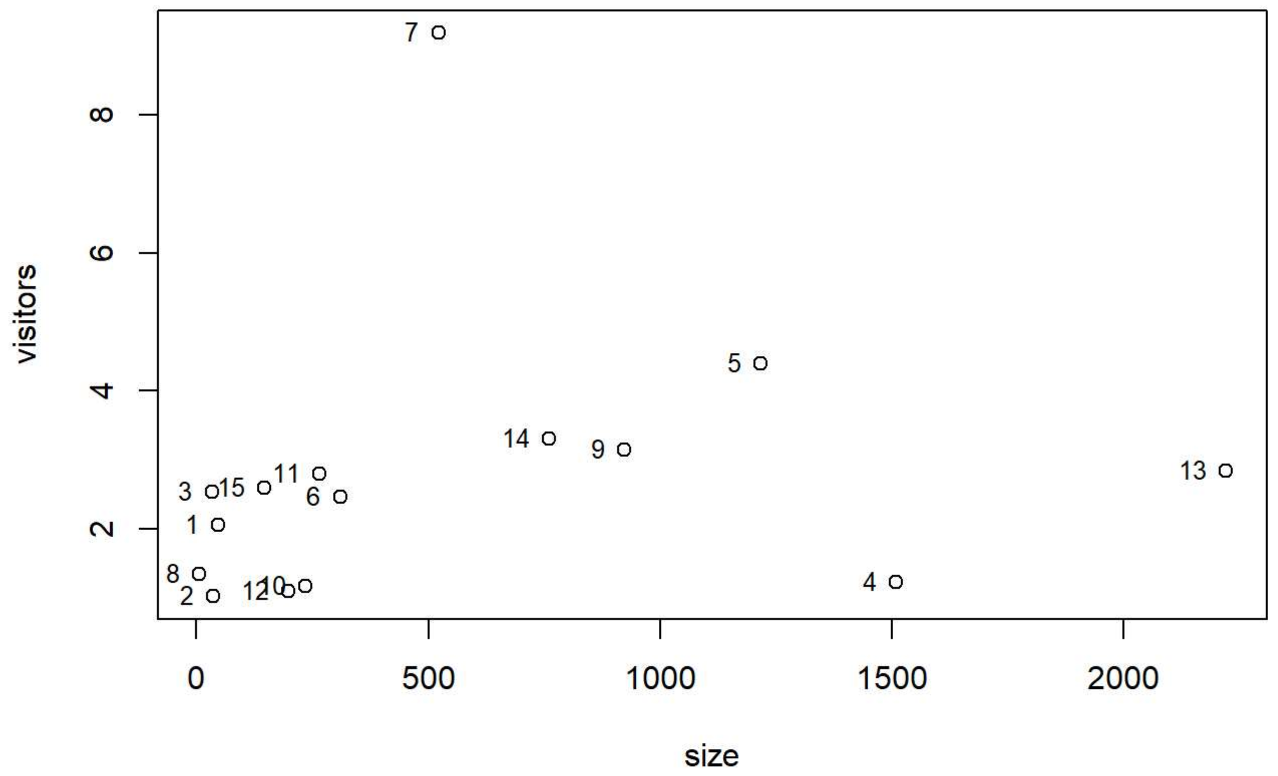

```
cor(parks_df)
```

```
##               size  visitors
## size     1.0000000 0.1725274
## visitors 0.1725274 1.0000000
```

B) Identify the park that is unusual. Drop this point and recalculate the correlation coefficient. Comment on the effect of this point correlation.

We do this by adding names to the plot to detect the outlier

```
plot(parks_df)
text(parks_df,
     labels=rownames(parks_df),
     cex= 0.8, # Font size
     pos=2) # Position
```



It appears to be value 7 or thirteen, I will say that is is number 7 since it has a ridiculousness number of visitors

```
print(parks_df[c(7, 13), ])
```

```
##      size visitors
## 7   521.8     9.19
## 13 2219.8     2.84
```

Both are unsual, I will try removing one at a time and then both

No number 7

```
print("original")
```

```
## [1] "original"
```

```
cor(parks_df)
```

```
##               size  visitors
## size     1.0000000 0.1725274
## visitors 0.1725274 1.0000000
```

```
print("No number 7")
```

```
## [1] "No number 7"
```

```
cor(parks_df[-7, ])
```

```
##               size  visitors
## size     1.0000000 0.3907829
## visitors 0.3907829 1.0000000
```

```
print("No number 13")
```

```
## [1] "No number 13"
```

```
cor(parks_df[-13, ])
```

```
##               size  visitors
## size     1.0000000 0.2299564
## visitors 0.2299564 1.0000000
```

```
print("No 13 or 7")
```

```
## [1] "No 13 or 7"
```

```
cor(parks_df[c(-7, -13), ])
```

```
##              size visitors
## size     1.000000 0.398539
## visitors 0.398539 1.000000
```

There is suddenly a moderat positive correlation between size and visitors, and it appears that number thirteen is not really the big outlier, since it has very little effect on the correlation.

c) Would the correlation in part b change if you measure in size in square miles instead of acres? Explain.

Let us test it:

```
parks_df$size <- parks_df$size / 640

print(parks_df)
```

```
##          size visitors
## 1  0.07406250     2.05
## 2  0.05593750     1.02
## 3  0.05140625     2.53
## 4  2.35703125     1.23
## 5  1.90218750     4.40
## 6  0.48437500     2.46
## 7  0.81531250     9.19
## 8  0.00875000     1.34
## 9  1.44171875     3.14
## 10 0.36812500     1.17
## 11 0.41531250     2.80
## 12 0.31093750     1.09
## 13 3.46843750     2.84
## 14 1.18953125     3.30
## 15 0.22906250     2.59
```

```
print("original")
```

```
## [1] "original"
```

```
cor(parks_df)
```

```
##               size  visitors
## size     1.0000000 0.1725274
## visitors 0.1725274 1.0000000
```

```
print("No number 7")
```

```
## [1] "No number 7"
```

```
cor(parks_df[-7, ])
```

```
##               size  visitors
## size     1.0000000 0.3907829
## visitors 0.3907829 1.0000000
```

```
print("No number 13")
```

```
## [1] "No number 13"
```

```
cor(parks_df[-13, ])
```

```
##               size  visitors
## size     1.0000000 0.2299564
## visitors 0.2299564 1.0000000
```

```
print("No 13 or 7")
```

```
## [1] "No 13 or 7"
```

```
cor(parks_df[c(-7, -13), ])
```

```
##              size visitors
## size     1.000000 0.398539
## visitors 0.398539 1.000000
```

No the measurement does not have any impact on the correlation. That is because correlation is a statistical measurement of how two variables are related ie. if one value changes by one unit, how does the other change. Therefore, they change the same.