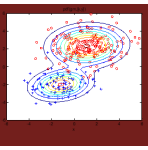


STAT 445/645 Assignment Cover Page

Student Name _____

SFU Student Number _____

SFU email address _____

Assignment Number _____

Due Date _____

Provide references for any data sets used in this assignment

List software used in this assignment.

List **ALL** resources used to complete this assignment, including books, internet sources and people.

☐ I personally completed the computations and wrote the solutions submitted in this document.

A2_Q1

Joohyeok

2024-02-21

```
library(readxl)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

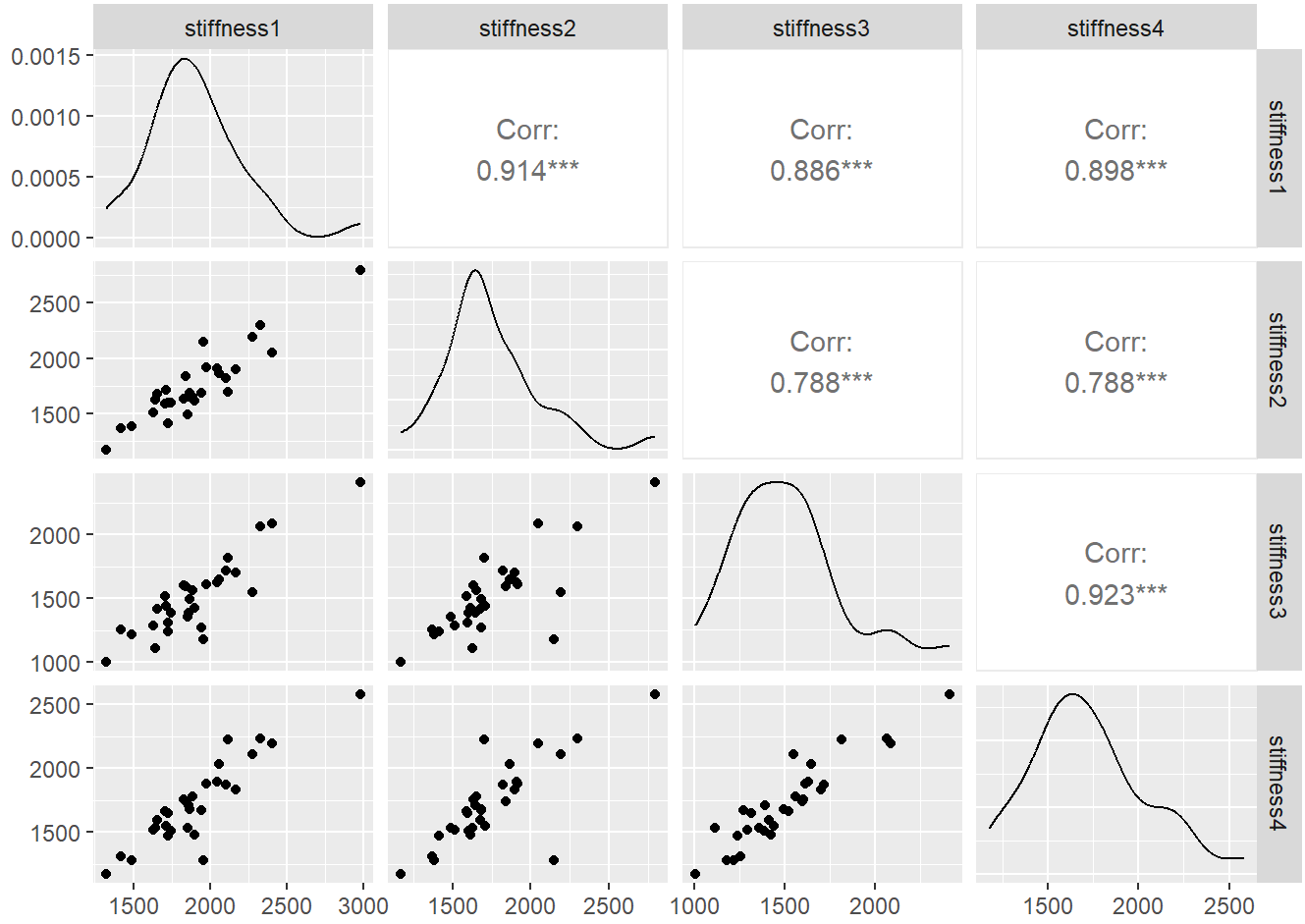
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
## ✓ readr     2.1.5
```

```
## ── 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 becom
e errors
```

```
data=read.table("lumber.txt",header=FALSE)
colnames(data) <- c("stiffness1","stiffness2","stiffness3","stiffness4")
```

a. Display a matrix scatter plot.

```
ggpairs(data[,c("stiffness1","stiffness2","stiffness3","stiffness4")])
```



b. Based on this graph, the correlation coefficients are all very high, above 0.7. These high correlation coefficients indicate that each variable has a strong linear relationship with the other variables, so the data matrix is not full rank.

c. Mean of x

```
colMeans(data)
```

```
## stiffness1 stiffness2 stiffness3 stiffness4
##   1906.100   1749.533   1509.133   1724.967
```

Sx

```
cov(data)
```

```
##            stiffness1 stiffness2 stiffness3 stiffness4
## stiffness1  105616.30   94613.53   87289.71   94230.73
## stiffness2   94613.53  101510.12   76137.10   81064.36
## stiffness3   87289.71   76137.10   91917.09   90352.38
## stiffness4   94230.73   81064.36   90352.38  104227.96
```

Rx

```
cor(data)
```

```
##            stiffness1 stiffness2 stiffness3 stiffness4
## stiffness1  1.0000000  0.9137620  0.8859301  0.8981212
## stiffness2  0.9137620  1.0000000  0.7882129  0.7881034
## stiffness3  0.8859301  0.7882129  1.0000000  0.9231013
## stiffness4  0.8981212  0.7881034  0.9231013  1.0000000
```

d. Find the matrix A (by hand) such that Y = AX and display

```
A<-matrix(c(2,1,0,0,0,1,0,0,-3,1,4,2,0,0,0,-2),ncol=4,byrow=FALSE)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    0   -3    0
## [2,]    1    1    1    0
## [3,]    0    0    4    0
## [4,]    0    0    2   -2
```

```
X <- as.matrix(data[, c("stiffness1", "stiffness2", "stiffness3", "stiffness4")])

Y<-t(A%*%t(X))
Y
```

```
##        [,1] [,2] [,3]  [,4]
##  [1,]  -905 5101 6244  -434
##  [2,] -1455 6538 8348  -220
##  [3,] -1207 5634 7260  -814
##  [4,]   -40 4382 4440  -846
##  [5,]  -890 5506 6456  -538
##  [6,]  -893 4863 5756  -214
##  [7,]    73 4899 5084  -800
##  [8,]  -943 5641 6868  -314
##  [9,] -1270 8189 9648  -338
## [10,]  -662 4729 5536  -248
## [11,] -1134 4819 6072  -298
## [12,]  -789 5580 6508  -542
## [13,] -1105 5276 6380  -292
## [14,]  -745 5045 5972  -370
## [15,]  -449 4897 5556  -650
## [16,]   368 5283 4720  -202
## [17,]  -356 3497 4008  -348
## [18,]  -918 4042 5008  -112
## [19,] -1150 5064 6408  -306
## [20,]  -489 4632 5252  -666
## [21,]   -89 6012 6188 -1128
## [22,]  -468 4935 5688  -110
## [23,]  -604 4436 5160  -452
## [24,]  -816 5574 6584  -782
## [25,]  -356 4705 5424  -354
## [26,]  -260 4377 4952  -462
## [27,]  -767 5765 6804  -266
## [28,]  -932 4744 5656  -366
## [29,] -1543 6692 8260  -338
## [30,]  -662 4086 4856  -140
```

e. display Y

```
Y
```

```
##        [,1] [,2] [,3]  [,4]
##  [1,]  -905 5101 6244  -434
##  [2,] -1455 6538 8348  -220
##  [3,] -1207 5634 7260  -814
##  [4,]   -40 4382 4440  -846
##  [5,]  -890 5506 6456  -538
##  [6,]  -893 4863 5756  -214
##  [7,]    73 4899 5084  -800
##  [8,]  -943 5641 6868  -314
##  [9,] -1270 8189 9648  -338
## [10,]  -662 4729 5536  -248
## [11,] -1134 4819 6072  -298
## [12,]  -789 5580 6508  -542
## [13,] -1105 5276 6380  -292
## [14,]  -745 5045 5972  -370
## [15,]  -449 4897 5556  -650
## [16,]   368 5283 4720  -202
## [17,]  -356 3497 4008  -348
## [18,]  -918 4042 5008  -112
## [19,] -1150 5064 6408  -306
## [20,]  -489 4632 5252  -666
## [21,]   -89 6012 6188 -1128
## [22,]  -468 4935 5688  -110
## [23,]  -604 4436 5160  -452
## [24,]  -816 5574 6584  -782
## [25,]  -356 4705 5424  -354
## [26,]  -260 4377 4952  -462
## [27,]  -767 5765 6804  -266
## [28,]  -932 4744 5656  -366
## [29,] -1543 6692 8260  -338
## [30,]  -662 4086 4856  -140
```

Mean of Y

```
colMeans(Y)
```

```
## [1] -715.2000 5164.7667 6036.5333 -431.6667
```

Sy

```
cov(Y)
```

```
##           [,1]       [,2]       [,3]      [,4]
## [1,]  202242.44 -190992.60 -404687.34 -37152.28
## [2,] -190992.60  815124.19 1021375.58 -20607.16
## [3,] -404687.34 1021375.58 1470673.36  12517.61
## [4,]  -37152.28  -20607.16   12517.61  61761.13
```

Ry

```
cor(Y)
```

```
##            [,1]        [,2]        [,3]        [,4]
## [1,]  1.0000000 -0.47040129 -0.74203665 -0.33242346
## [2,] -0.4704013  1.00000000  0.93285784 -0.09184355
## [3,] -0.7420366  0.93285784  1.00000000  0.04153418
## [4,] -0.3324235 -0.09184355  0.04153418  1.00000000
```

A2_Q2

Joohyeok

2024-02-16

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

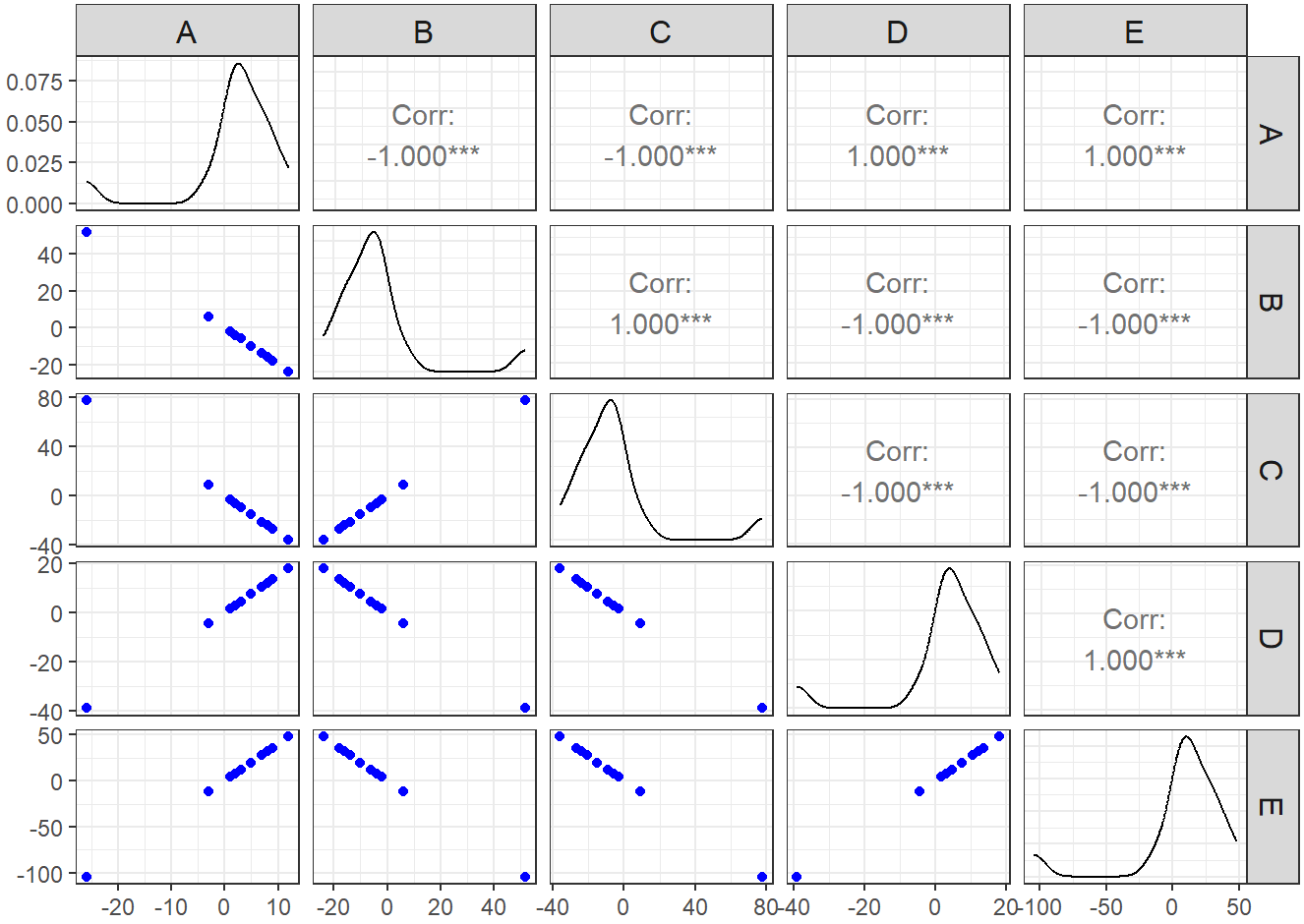
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
R <- read.csv("data_for_pr2-1.txt", header=FALSE, col.names=c("A","B","C","D","E"))
R
```

```
##      A   B   C     D    E
## 1    1  -2  -3   1.5    4
## 2    1  -2  -3   1.5    4
## 3    3  -6  -9   4.5   12
## 4   -3   6   9  -4.5  -12
## 5    5 -10 -15   7.5   20
## 6   12 -24 -36  18.0   48
## 7  -26  52  78 -39.0 -104
## 8    5 -10 -15   7.5   20
## 9    2  -4  -6   3.0    8
## 10   9 -18 -27  13.5   36
## 11   3  -6  -9   4.5   12
## 12   7 -14 -21  10.5   28
## 13   1  -2  -3   1.5    4
## 14   8 -16 -24  12.0   32
```

```
ggpairs(R,
    lower=list(continuous = wrap("points", color = "blue")))+
  theme_bw()+
  theme(strip.text=element_text(size=12))
```



a. Compute and display R

```
cor(R)
```

```
##    A  B  C  D  E
## A  1 -1 -1  1  1
## B -1  1  1 -1 -1
## C -1  1  1 -1 -1
## D  1 -1 -1  1  1
## E  1 -1 -1  1  1
```

b. Explain the values in R

- These correlation all 1 or -1 which means a very strong positive linear relationship or strong negative relationship.
- These plots' correlation of (A and D), (A and E), (B and C) and (D and E) is 1 which means when a value increases, the other increases proportionally.
- These plots' correlation of (A and B), (A and C), (B and D), (C and D), (B and E) and (C and E) is -1 which means when a value increases, the other decreases proportionally.

A2_Q3

Joohyeok

2024-02-21

```
library(readxl)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
## ✓ readr     2.1.5
```

```
## ── 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 becom
e errors
```

```
X <- read_excel("w-nat-track-rec.xlsx",col_names=FALSE)
```

```
## New names:
## • `` -> `...1`
## • `` -> `...2`
## • `` -> `...3`
## • `` -> `...4`
## • `` -> `...5`
## • `` -> `...6`
## • `` -> `...7`
## • `` -> `...8`
```

a. Construct Y by eliminating the first column of X and display

```
Y<-X[,-1]

Y
```

```
## # A tibble: 54 × 7
##     ...2  ...3  ...4  ...5  ...6  ...7  ...8
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  11.6  22.9  52.5  2.05  4.25  9.19  150.
##  2  11.1  22.2  48.6  1.98  4.02  8.63  144.
##  3  11.2  22.7  50.6  1.94  4.05  8.78  154.
##  4  11.1  22.5  51.4  1.97  4.08  8.82  143.
##  5  11.5  23.0  53.3  2.07  4.29  9.81  174.
##  6  11.2  22.6  50.6  1.97  4.17  9.04  147.
##  7  11.0  22.6  49.9  1.97  4     8.54  148.
##  8  11.6  23.8  53.7  2     4.22  9.26  152.
##  9  10.8  22.0  49.8  1.93  3.84  8.1   139.
## 10  11.3  22.9  49.6  2.04  4.34  9.37  155.
## # ℹ 44 more rows
```

b. Set R = corr(Y) and display

```
R<-cor(Y)
R
```

```
##           ...2      ...3      ...4      ...5      ...6      ...7      ...8
## ...2 1.0000000 0.9410886 0.8707802 0.8091758 0.7815510 0.7278784 0.6689597
## ...3 0.9410886 1.0000000 0.9088096 0.8198258 0.8013282 0.7318546 0.6799537
## ...4 0.8707802 0.9088096 1.0000000 0.8057904 0.7197996 0.6737991 0.6769384
## ...5 0.8091758 0.8198258 0.8057904 1.0000000 0.9050509 0.8665732 0.8539900
## ...6 0.7815510 0.8013282 0.7197996 0.9050509 1.0000000 0.9733801 0.7905565
## ...7 0.7278784 0.7318546 0.6737991 0.8665732 0.9733801 1.0000000 0.7987302
## ...8 0.6689597 0.6799537 0.6769384 0.8539900 0.7905565 0.7987302 1.0000000
```

c. Compute det(R) and display

```
det_R<-det(R)
det_R
```

```
## [1] 9.011147e-06
```

- det(R) indicates important about R. det(R) is not zero which means invertible.

d.

```
b<-matrix(c(1,3,1,2,-1,1,-2), nrow=7,ncol=1)
b
```

```
##      [,1]
## [1,]    1
## [2,]    3
## [3,]    1
## [4,]    2
## [5,]   -1
## [6,]    1
## [7,]   -2
```

```
x<-solve(R,b)
x
```

```
##            [,1]
## ...2  -23.17426
## ...3   58.14292
## ...4  -26.24643
## ...5   46.35860
## ...6 -120.24705
## ...7   88.85304
## ...8  -23.76193
```

e. Compute the norm

```
Rx <- R %*% x
Rx
```

```
##      [,1]
## ...2    1
## ...3    3
## ...4    1
## ...5    2
## ...6   -1
## ...7    1
## ...8   -2
```

```
norm(Rx - b, type="2")
```

```
## [1] 2.75192e-14
```

- The calculated norm value of 2.75e^(-14)is as expected. It is very small value which means the x is very accurate and that Rx substantially matches the given vector..

f. Compute w

```
z<-matrix(c(1,-1,-1,-1,0,2,1), nrow=7,ncol=1)
z
```

```
##      [,1]
## [1,]    1
## [2,]   -1
## [3,]   -1
## [4,]   -1
## [5,]    0
## [6,]    2
## [7,]    1
```

```
w<-R%*%R%*%R%*%R%*%R%*%z
w
```

```
##          [,1]
## ...2 846.3503
## ...3 858.5484
## ...4 824.5423
## ...5 884.5601
## ...6 872.2149
## ...7 842.7317
## ...8 795.9168
```

g. Compute the projection of z on w

```
projection <- function(a, b) {
  (sum(a * b) / sum(b * b)) * b
}

projection(z,w)
```

```
##           [,1]
## ...2 0.1281418
## ...3 0.1299887
## ...4 0.1248400
## ...5 0.1339270
## ...6 0.1320579
## ...7 0.1275940
## ...8 0.1205059
```

A2_Q4

Joohyeok

2024-02-16

```
library(readxl)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.0
## ✓ readr     2.1.5
```

```
## ── 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 becom
e errors
```

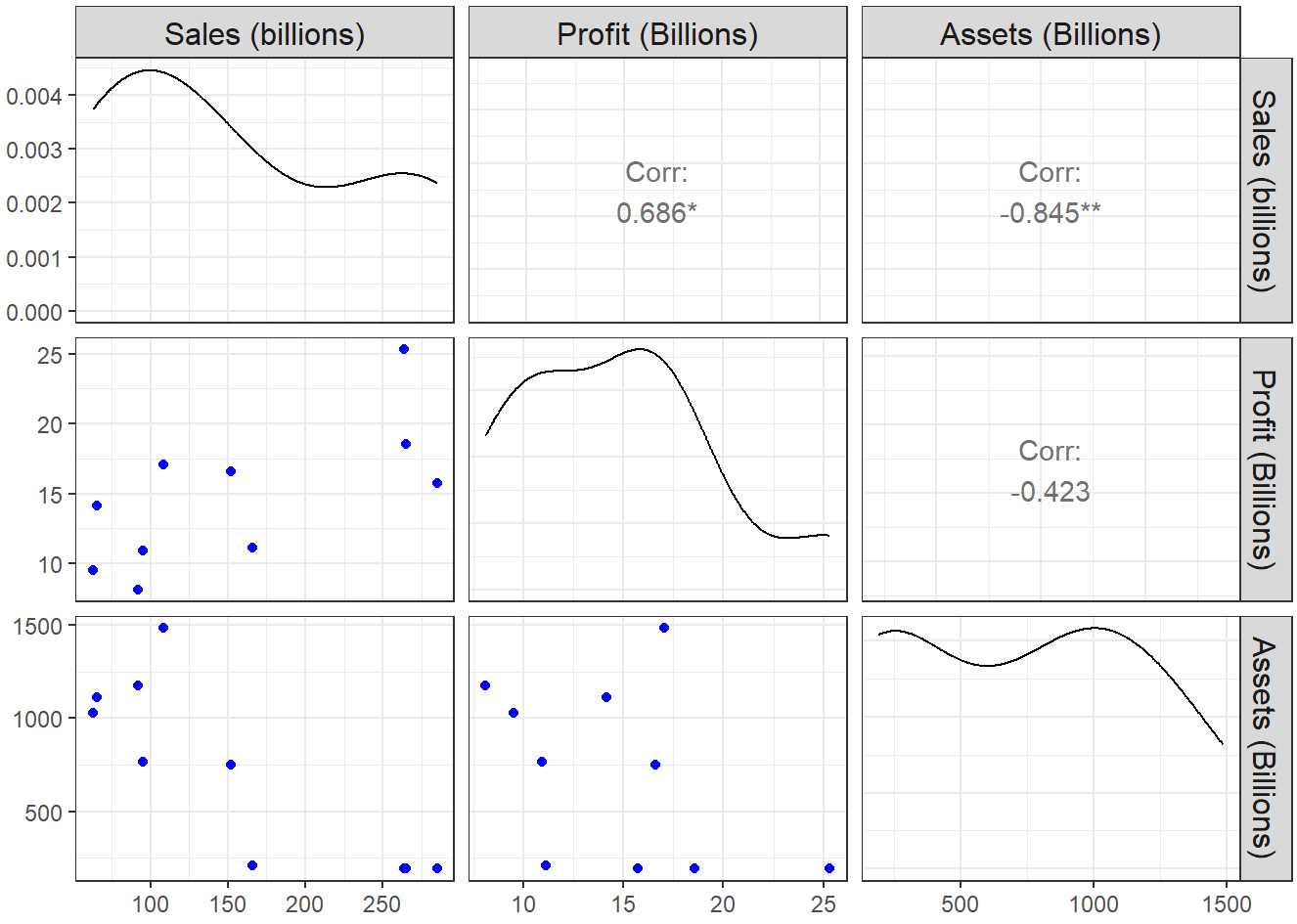
```
X <- read_excel("Forbes-ten-companies.xlsx")
X
```

```
## # A tibble: 10 × 4
##    Company            `Sales (billions)` `Profit (Billions)` `Assets (Billions)`
##    <chr>                           <dbl>               <dbl>               <dbl>
##  1 Citigroup                       108.                17.0                1484.
##  2 General Electric                152.                16.6                 750.
##  3 American Intl Gro…               95.0               10.9                 766.
##  4 Bank of America                  65.4               14.1                1110.
##  5 HSBC Group                       63.0                9.52               1031.
##  6 Exxon Mobil                     264.                25.3                 195.
##  7 Royal Dutch/Shell               265.                18.5                 194.
##  8 BP                              285.                15.7                 191.
##  9 ING Group                        92.0                8.1                1175.
## 10 Toyota Motor                    166.                11.1                 211.
```

a. Produce a matrix scatter plot for X

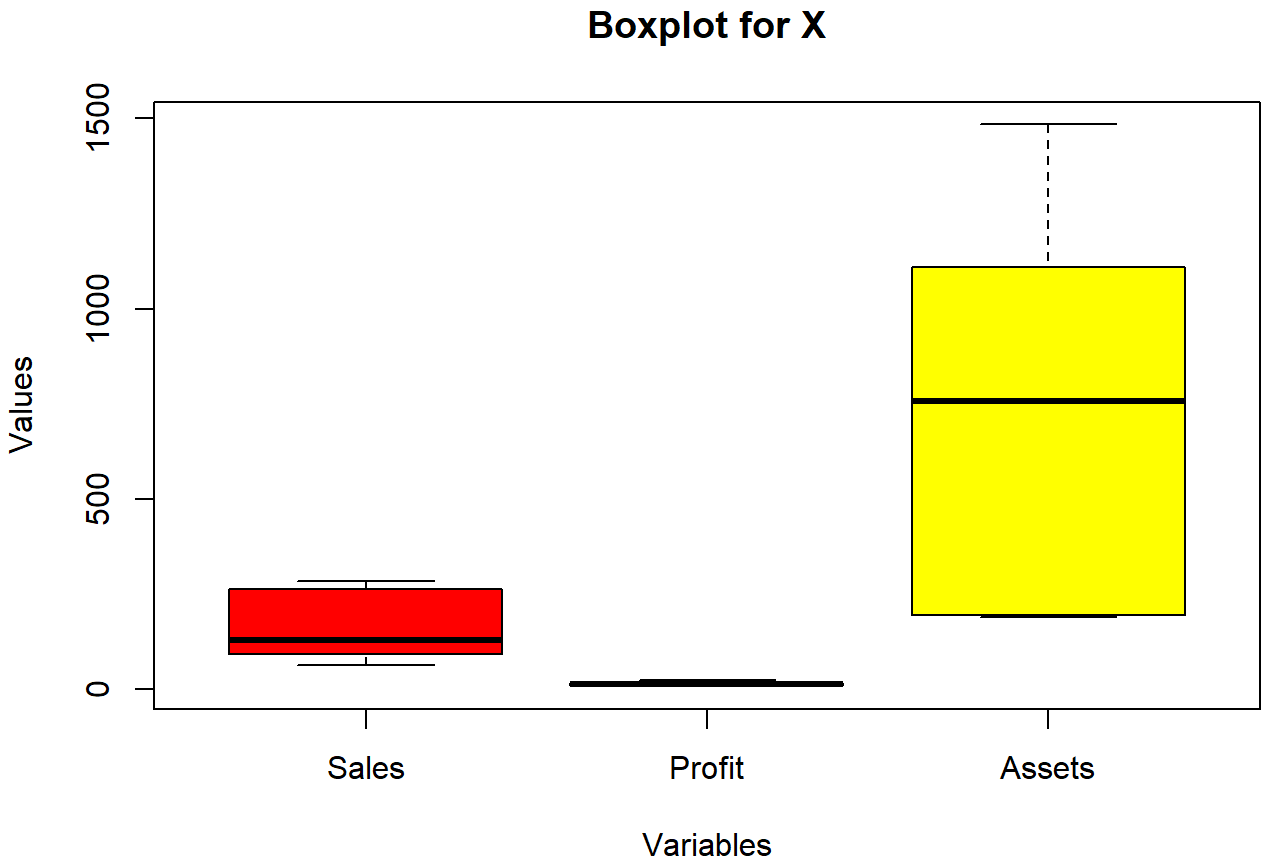
```
ggpairs(X[,-1],colums=c("Sales (billions)","Profit (Billions)","Assets (Billions)"),
        lower=list(continuous = wrap("points", color = "blue")))+
  theme_bw()+
  theme(strip.text=element_text(size=12))
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: "colums" are being
## ignored.  If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



b. Produce a boxplot for X.

```
boxplot(X[, -1], col = c("red","orange","yellow"), main="Boxplot for X", xlab="Variables", ylab
="Values", names=c("Sales", "Profit", "Assets"))
```



c. Scale can be an issue when calculating covariance. Covariance measures the direction and strength of the relationship between two variables, but it depends on the units of the variables, making direct comparison problematic if their scales differ. Therefore, standardizing variables is important when they are measured in different units, allowing for accurate comparison in the same units. In summary, not properly adjusting for scale differences can significantly impact the accuracy and usefulness of analysis results.

d.

```
sapply(X[, c("Sales (billions)", "Profit (Billions)", "Assets (Billions)")], mean)
```

```
##  Sales (billions) Profit (Billions) Assets (Billions)
##           155.603            14.704           710.911
```

```
sapply(X[, c("Sales (billions)", "Profit (Billions)", "Assets (Billions)")], var)
```

```
##  Sales (billions) Profit (Billions) Assets (Billions)
##        7476.45325          26.19032      237054.26983
```

```
cor(X[, c("Sales (billions)", "Profit (Billions)", "Assets (Billions)")])
```

```
##                   Sales (billions) Profit (Billions) Assets (Billions)
## Sales (billions)         1.0000000         0.6861360        -0.8450549
## Profit (Billions)        0.6861360         1.0000000        -0.4229366
## Assets (Billions)       -0.8450549        -0.4229366         1.0000000
```

Mean (x)

- Sales(billions): 155.6
- Profit(billions): 14.7
- Assets(billions): 710.9

Sample Variance (S)

- Sales(billions): 7476.45
- Profit(billions): 26.19
- Assets(billions): 237054.27

Correlation Matrix (R)

- Correlation between Sales and Profit: 0.686
- Correlation between Sales and Assets: -0.845
- Correlation between Assets and Profit: -0.423

e. Compute and display the standardized data matrix X*

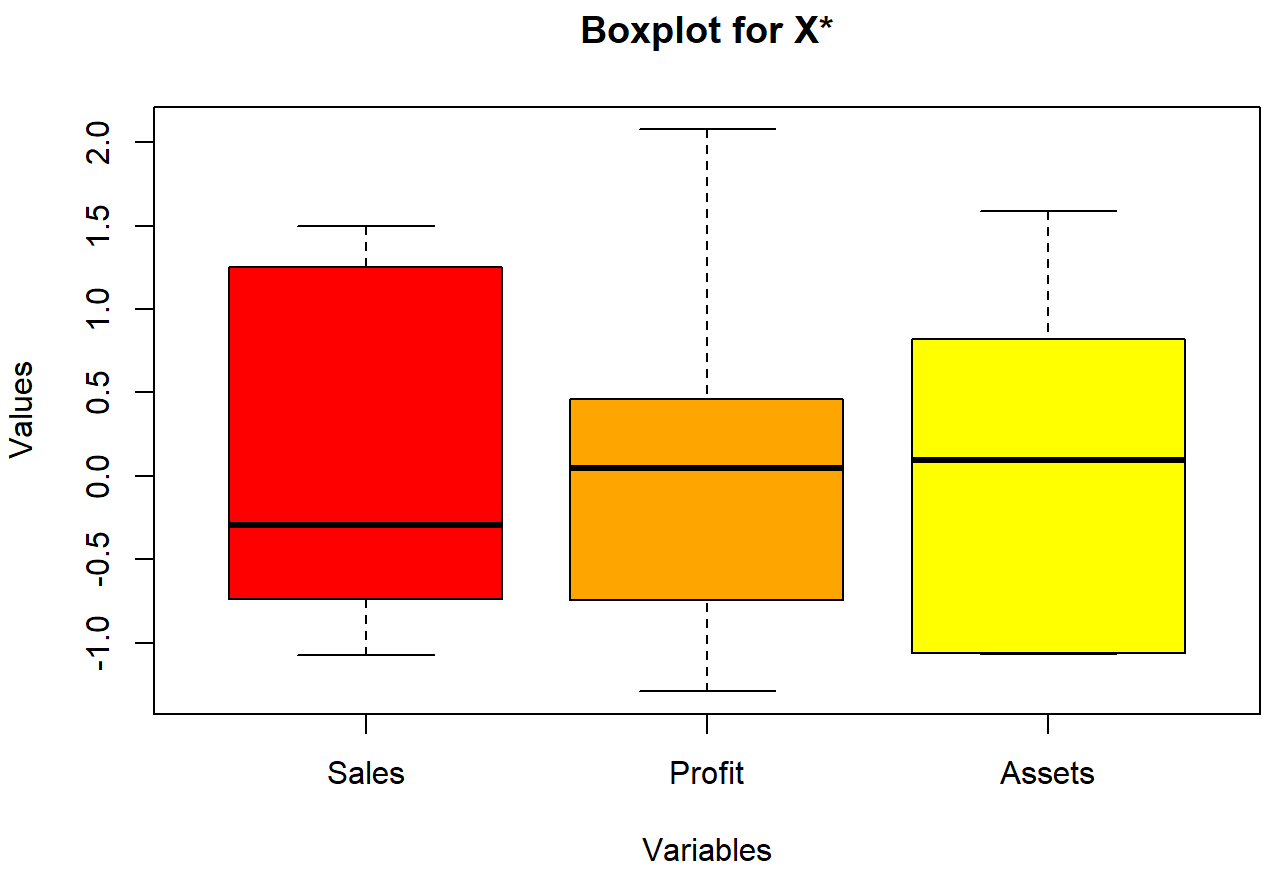
```
stand <- function(x) {
  (x - mean(x)) / sd(x)
}

Xsd<-as.data.frame(lapply(X[, -1], stand))
Xsd
```

```
##    Sales..billions. Profit..Billions. Assets..Billions.
## 1       -0.54729875         0.4584138        1.58804124
## 2       -0.03750586         0.3685287        0.08096209
## 3       -0.70042166        -0.7413563        0.11400910
## 4       -1.04263517        -0.1102069        0.82062767
## 5       -1.07131681        -1.0129654        0.65802160
## 6        1.25351457         2.0763447       -1.05908782
## 7        1.26739278         0.7495632       -1.06202487
## 8        1.49719279         0.2004827       -1.06761144
## 9       -0.73546414        -1.2904367        0.95351403
## 10       0.11654226        -0.6983678       -1.02645159
```

f. Produce a boxplot for X*

```
boxplot(Xsd, col = c("red","orange","yellow"), main="Boxplot for X*", xlab="Variables", ylab="Va
lues", names=c("Sales", "Profit", "Assets"))
```



-The standardization process adjusts the mean of all variables to 0 and their standard deviation to 1. This matches the scale of each variable, facilitating comparison between variables. Therefore, the boxplot for X* shows that all variables are on the same scale, indicating that we have solved the scale problem. In conclusion, the boxplot for X* indicates that the scale problem does not exist for X*. Because all variables are standardized and have the same scale, this allows for fair comparisons between variables in the analysis.

g.

```
sapply(Xsd, mean)
```

```
##  Sales..billions. Profit..Billions. Assets..Billions.
##     -5.968668e-17     -1.387779e-16     -1.276323e-16
```

```
sapply(Xsd, var)
```

```
##  Sales..billions. Profit..Billions. Assets..Billions.
##                 1                 1                 1
```

```
cor(Xsd)
```

```
##                  Sales..billions. Profit..Billions. Assets..Billions.
## Sales..billions.         1.0000000         0.6861360        -0.8450549
## Profit..Billions.        0.6861360         1.0000000        -0.4229366
## Assets..Billions.       -0.8450549        -0.4229366         1.0000000
```

Mean of X*

- Sales(billions): -5.97e^(-17)
- Profit(billions): -1.39e^(-16)
- Assets(billions): -1.28e^(-16) All mean of X* is almost 0.

Sample Variance (S*)

- Sales(billions): 1
- Profit(billions): 1
- Assets(billions): 1 All variance of X* is 1.

Correlation Matrix (R*)

- Correlation between Sales and Profit: 0.686
- Correlation between Sales and Assets: -0.845
- Correlation between Assets and Profit: -0.423 R and R* is the same.

h. There is an expected relationship between R and S**. The standardization process does not change the correlation between variables, and R is the same before and after standardization. S* will be 1 for all variables, but this will not change the value of R. Therefore, there is no direct relationship between R and S*, and the standardization process does not affect R.

i. R* and S* have the expected relationship. Standardization preserves the correlation between variables, with R* equaling the original correlation matrix, R. S* being 1 for all variables indicates that standardization scales all variables to the same scale. R* and S* reflect the correlation between variables and the standardized scale, respectively, ensuring that the correlation is maintained through standardization.

A2_Q5

Joohyeok

2024-02-16

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

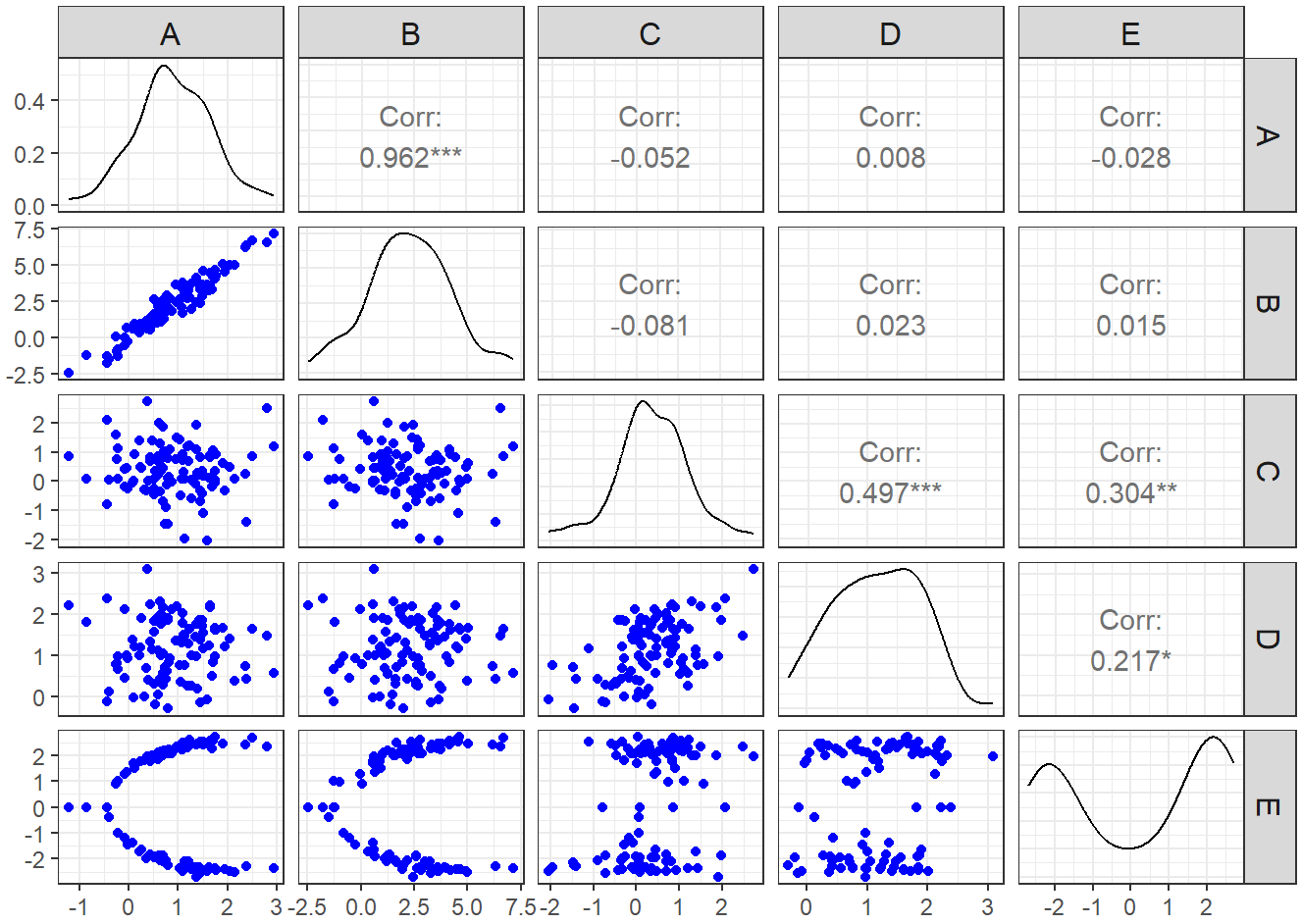
```
data <- read.csv("pr2_second_data_set.txt", header=FALSE)

colnames(data) <- c("A", "B", "C", "D", "E")
```

a. Display a matrix scatter plot

```
ggpairs(data,colums=c("A","B","C","D","E"),
        lower=list(continuous = wrap("points", color = "blue")))+
  theme_bw()+
  theme(strip.text=element_text(size=12))
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: "colums" are being
## ignored.  If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



b. The correlation between A and B is 0.962. It indicates a strong positive linear relationship which is seem in second row first column plot. Since this value is close to 1, there is a strong dependency. The correlation between C and D is 0.497. It indicates positive linear relationship which is seem in fourth row third column plot. We can say this plot is dependency.