week8 exercise

Xiang Li

2023/12/28

```
library(brolgar)
library(vioplot)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.7: type help(sm) for summary information
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(mvtnorm)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

Exercise 1

2

```
vc = function(v) {
    if (!is.numeric(v)) {
        stop("Function input should be a numeric vector!!!")
    } else if (length(v[is.na(v)]) > 0) {
        warning("The supplied vector contains missing values!!!")
    } else {
        result = v - mean(v)
        return(result)
    }
}
vs = function(v) {
```

```
    if (!is.numeric(v)) {
        stop("Function input should be a numeric vector!!!")
    } else if (length(v[is.na(v)]) > 0) {
        warning("The supplied vector contains missing values!!!")
    } else {
        result = v/sd(v)
        return(result)
    }
}
vn = function(v) {
    if (!is.numeric(v)) {
        stop("Function input should be a numeric vector!!!")
    } else if (length(v[is.na(v)]) > 0) {
        warning("The supplied vector contains missing values!!!")
    } else {
        result = (v - mean(v))/sd(v)
        return(result)
    }
}
```

3

```
obj1 = matrix(1:10, 5, 2)
obj2 = c(5, 7, 10, -25)
obj3 = c(42, NA, 3, 7)
obj4 = c(pi, 42, "apple", sqrt(3))
```

```
vc(obj1)
```

```
##      [,1] [,2]
## [1,] -4.5  0.5
## [2,] -3.5  1.5
## [3,] -2.5  2.5
## [4,] -1.5  3.5
## [5,] -0.5  4.5
```

```
vc(obj2)
```

```
## [1]   5.75   7.75  10.75 -24.25
```

```
vc(obj3)
```

```
## Warning in vc(obj3): The supplied vector contains missing values!!!
```

```
vc(obj4)
```

```
## Error in vc(obj4): Function input should be a numeric vector!!!
```

```
vs(obj1)
```

```
##           [,1]     [,2]
## [1,] 0.3302891 1.981735
## [2,] 0.6605783 2.312024
## [3,] 0.9908674 2.642313
## [4,] 1.3211565 2.972602
## [5,] 1.6514456 3.302891
```

```
vs(obj2)
```

```
## [1]  0.3068101  0.4295341  0.6136201 -1.5340503
```

```
vs(obj3)
```

```
## Warning in vs(obj3): The supplied vector contains missing values!!!
```

```
vs(obj4)
```

```
## Error in vs(obj4): Function input should be a numeric vector!!!
```

```
vn(obj1)
```

```
##            [,1]      [,2]
## [1,] -1.4863011 0.1651446
## [2,] -1.1560120 0.4954337
## [3,] -0.8257228 0.8257228
## [4,] -0.4954337 1.1560120
## [5,] -0.1651446 1.4863011
```

```
vn(obj2)
```

```
## [1]  0.3528316  0.4755556  0.6596416 -1.4880288
```

```
vn(obj3)
```

```
## Warning in vn(obj3): The supplied vector contains missing values!!!
```

```
vn(obj4)
```

```
## Error in vn(obj4): Function input should be a numeric vector!!!
```

Exercise 2

2

```r
sk = function(v) {
    v_bar = mean(v)
    result = mean((v - v_bar)^3)/(var(v)^(3/2))
    return(result)
}
gen_sk = function(v) {
    if (!(is.vector(v) | is.matrix(v))) {
        stop("Wrong type of input!!!")
    } else if (any(is.na(v))) {
        warning("The supplied vector contains missing values!!!")
    } else if (is.matrix(v)) {
        warning("Supplied input is a matrix. Skewness by column returned!!")
        return(apply(v, 2, FUN = sk))
    } else {
        return(sk(v))
    }
}
gen_sk(matrix(rnorm(100), nrow = 20, byrow = TRUE))
```

```
## Warning in gen_sk(matrix(rnorm(100), nrow = 20, byrow = TRUE)): Supplied input
## is a matrix. Skewness by column returned!!
```

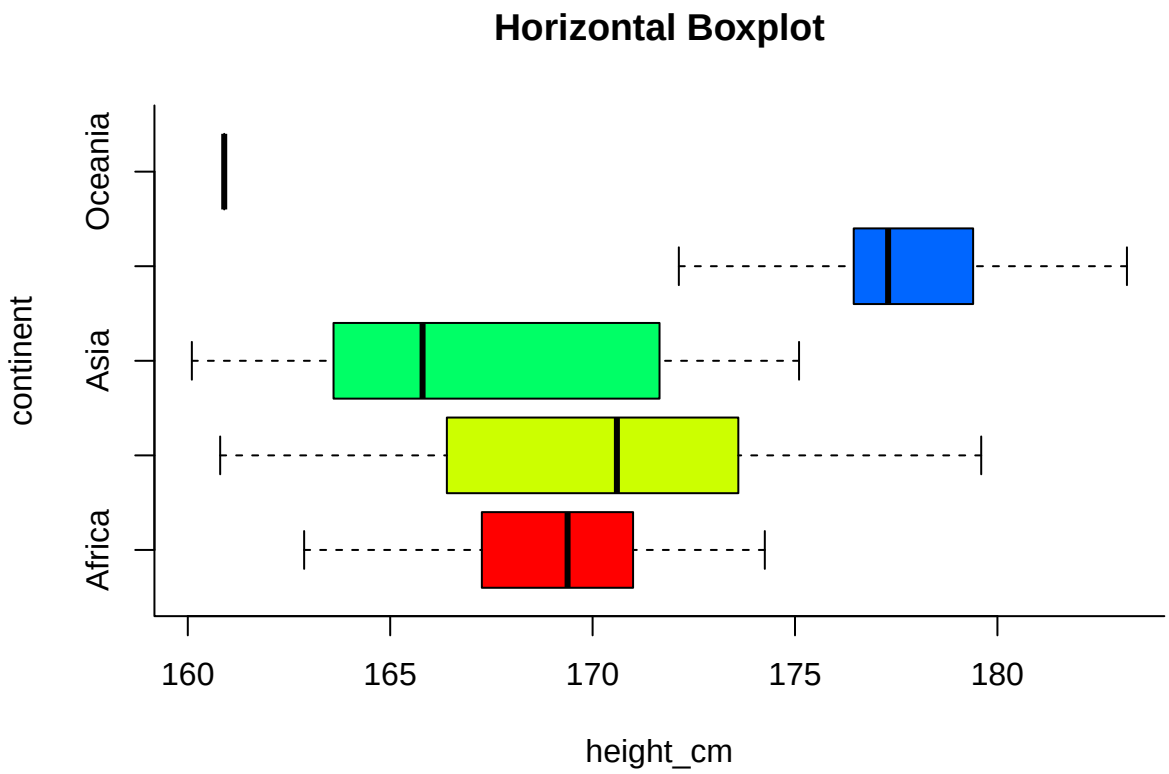
```
## [1]  0.4392015  0.3943714  0.5064618 -0.2647116  0.5440833
```

Exercise 3

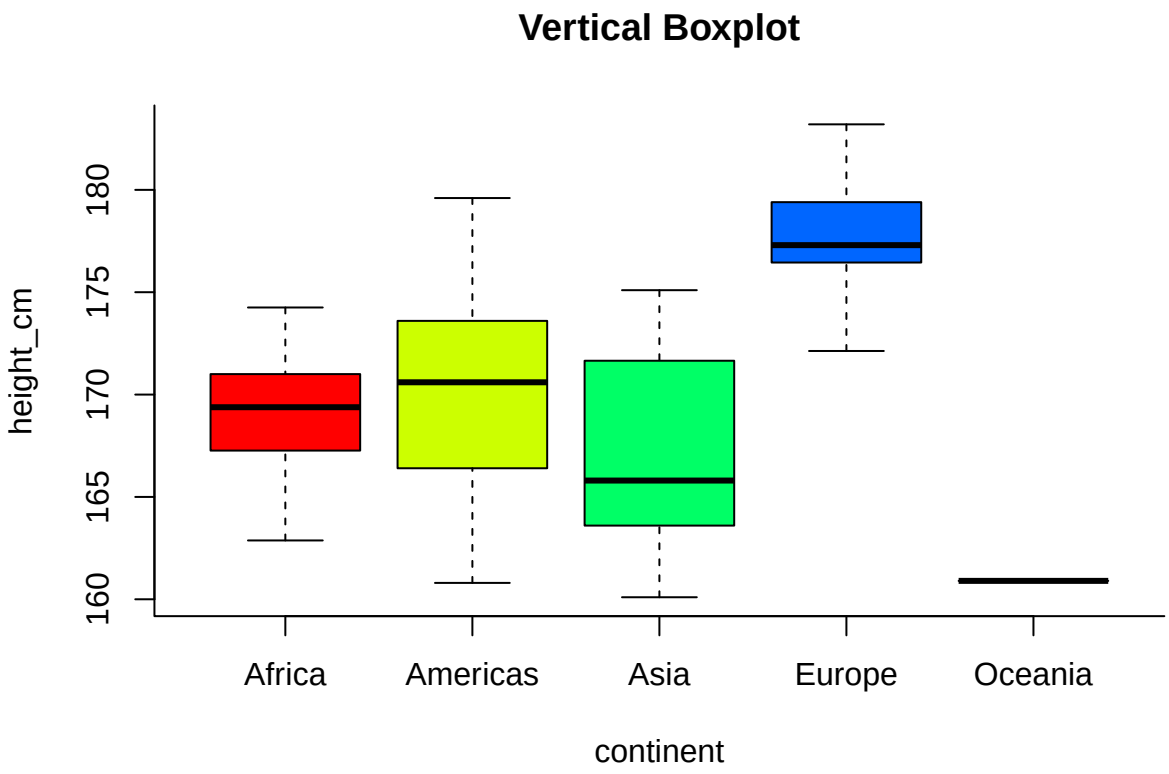
```r
heights = as.data.frame(heights)
wages = as.data.frame(wages)
```

1

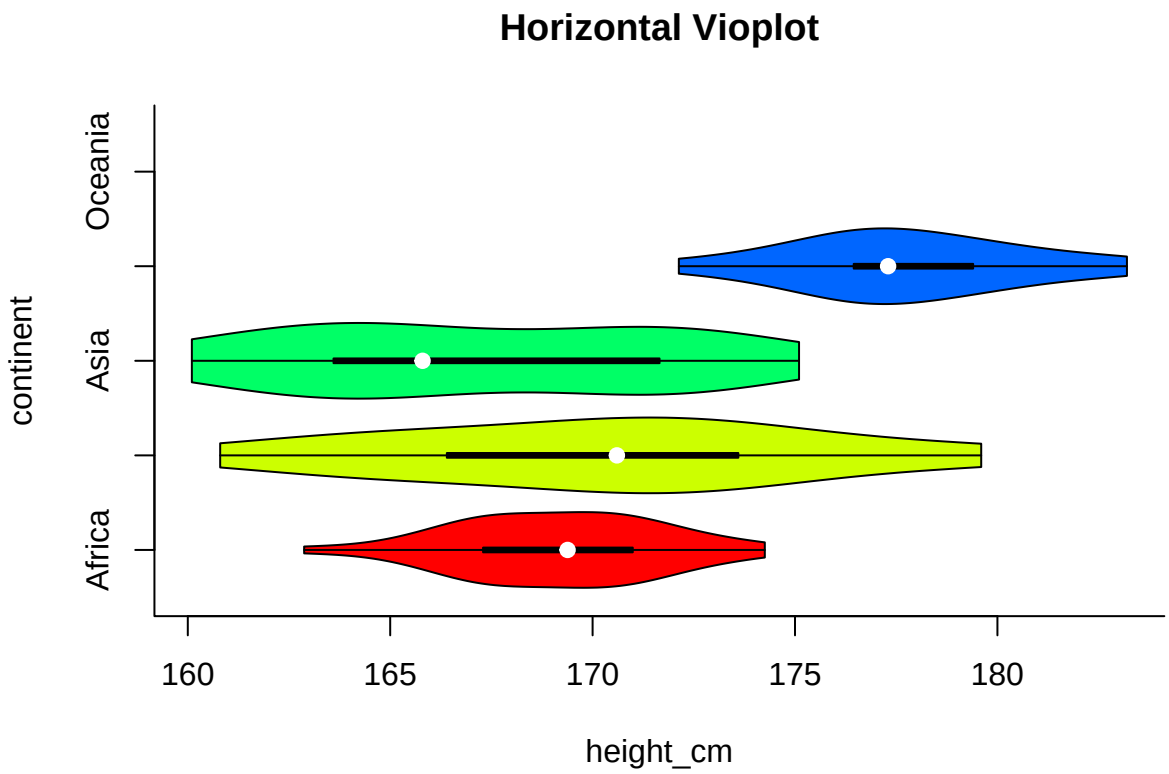
```r
height1980_df = heights[heights$year == 1980, ]
c_n = length(unique(height1980_df$continent))
par(bty = "l")
boxplot(height_cm ~ continent, data = height1980_df, horizontal = TRUE,
    main = "Horizontal Boxplot", col = rainbow(c_n))
```



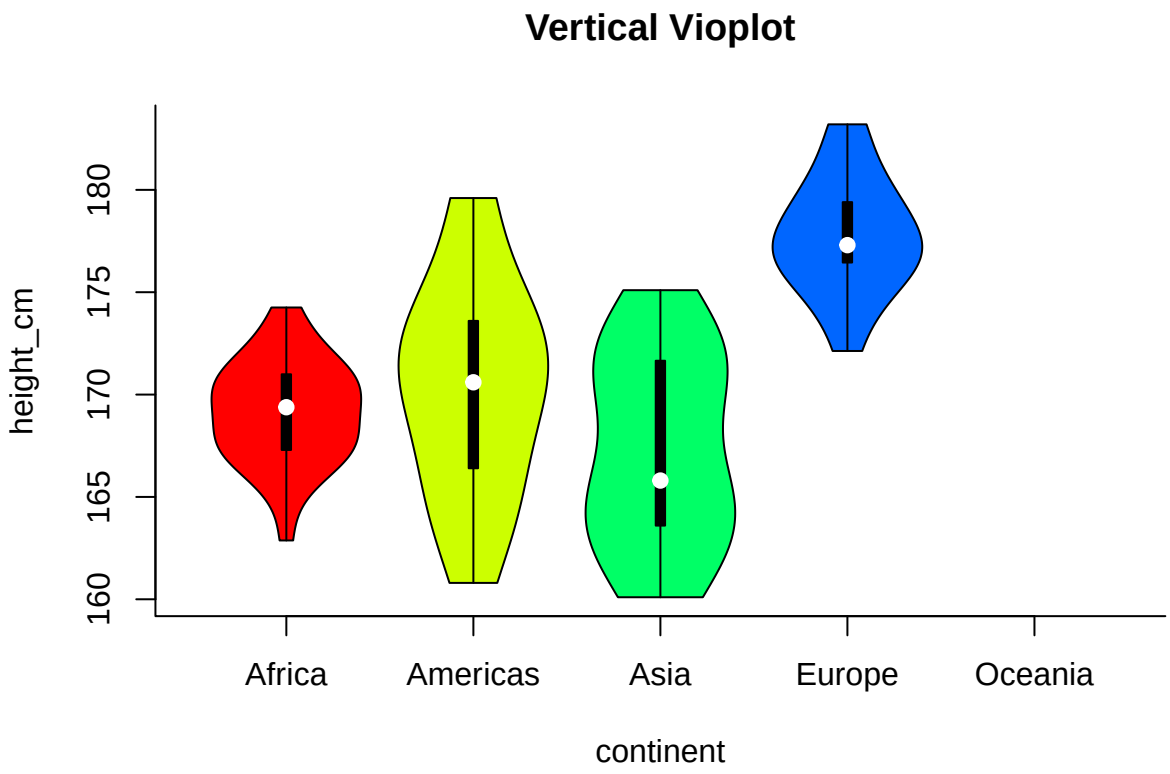

```
par(bty = "l")
boxplot(height_cm ~ continent, data = height1980_df, horizontal = FALSE,
    main = "Vertical Boxplot", col = rainbow(c_n))
```




```
par(bty = "l")
vioplot(height_cm ~ continent, data = height1980_df, horizontal = TRUE,
    main = "Horizontal Vioplot", col = rainbow(c_n))
```

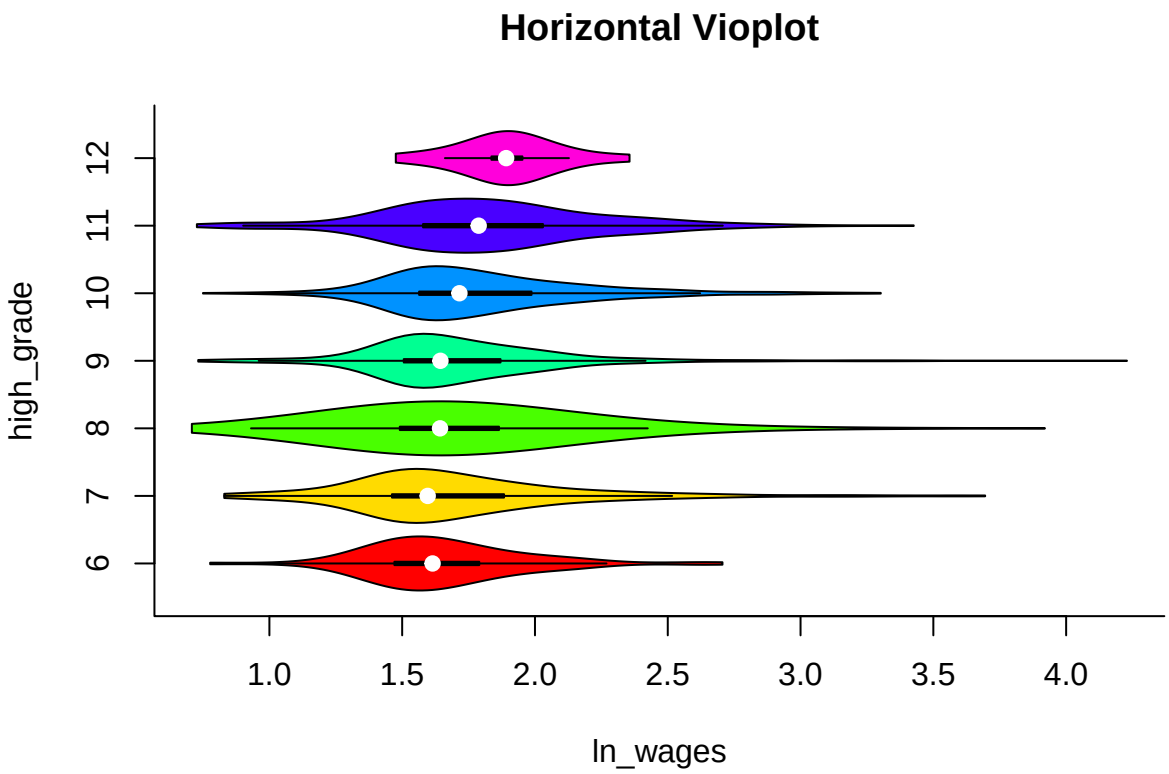



```
par(bty = "l")
vioplot(height_cm ~ continent, data = height1980_df, horizontal = FALSE,
    main = "Vertical Vioplot", col = rainbow(c_n))
```



2

```
wages12_df = wages[wages$xp <= 2, ]
c_n = length(unique(wages12_df$high_grade))
par(bty = "l")
vioplot(ln_wages ~ high_grade, data = wages12_df, horizontal = TRUE,
    main = "Horizontal Vioplot", col = rainbow(c_n))
```



Exercise 4

1

```
set.seed(8)
Sigma1 = matrix(c(1, 0.5, -0.3, 0.5, 1, -0.6, -0.3, -0.6,
    1), 3, 3)
m1 = rmvnorm(500, mean = rep(0, 3), sigma = Sigma1)
Sigma2 = matrix(c(1, 0.4, 0.4, 1), 2, 2)
m2 = rmvnorm(500, mean = rep(0, 2), sigma = Sigma2)
all.vars = cbind(m1, m2)
colnames(all.vars) = paste0("X", 1:5)
```

The code is generating 5 variables which have 500 samples for each one. The first 3 variables follow 3-variate normal distribution with parameter $\mu = [3, 3, 3]$ and $\sigma = Sigma1$. The last 2 variables follow 2-variate normal distribution with parameter $\mu = [2, 2]$ and $\sigma = Sigma2$.

2

```
cor_tab = cor(all.vars)
cor_tab
```

```
##             X1          X2          X3          X4          X5
## X1  1.00000000  0.49602696 -0.33569496 -0.07637467 -0.08984472
## X2  0.49602696  1.00000000 -0.66982079 -0.07231346 -0.06348094
## X3 -0.33569496 -0.66982079  1.00000000  0.06248103  0.11240236
## X4 -0.07637467 -0.07231346  0.06248103  1.00000000  0.38312350
## X5 -0.08984472 -0.06348094  0.11240236  0.38312350  1.00000000
```

3

```
cor_tab[1:3, 1:3]
```

```
##           X1         X2         X3
## X1  1.000000  0.4960270 -0.3356950
## X2  0.496027  1.0000000 -0.6698208
## X3 -0.335695 -0.6698208  1.0000000
```

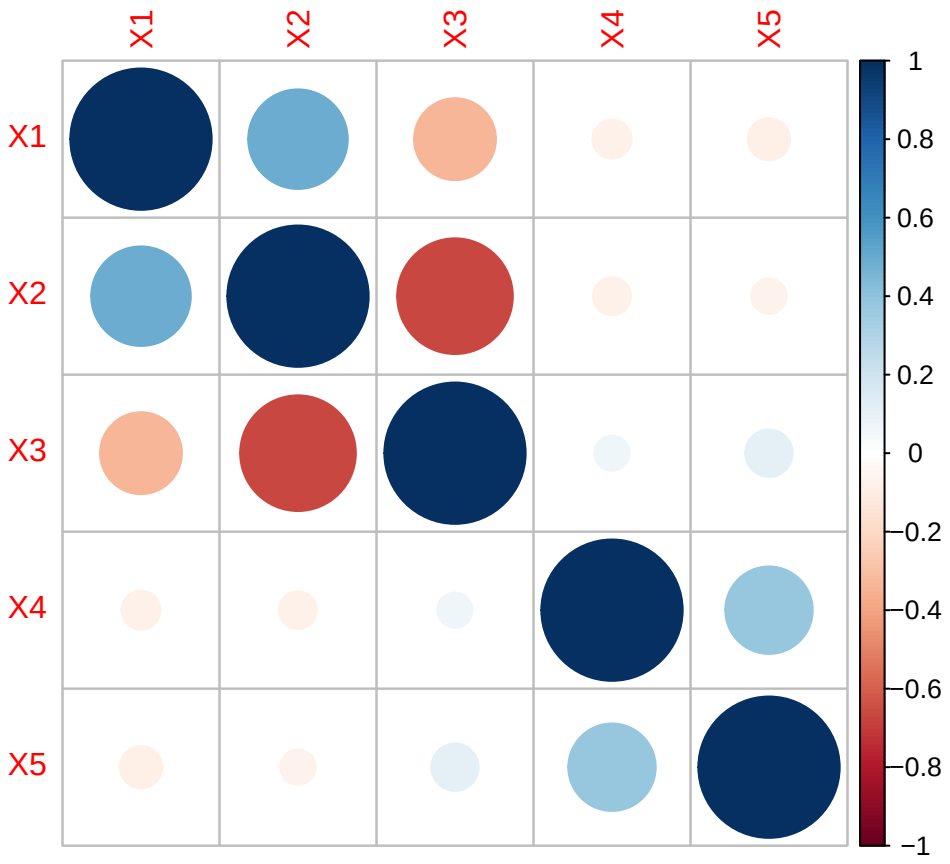
```
cor_tab[4:5, 4:5]
```

```
##           X4        X5
## X4 1.0000000 0.3831235
## X5 0.3831235 1.0000000
```

The correlations between X_1, X_2, X_3 are almost equal to Sigma1. The correlations between X_4, X_5 are almost equal to Sigma2.

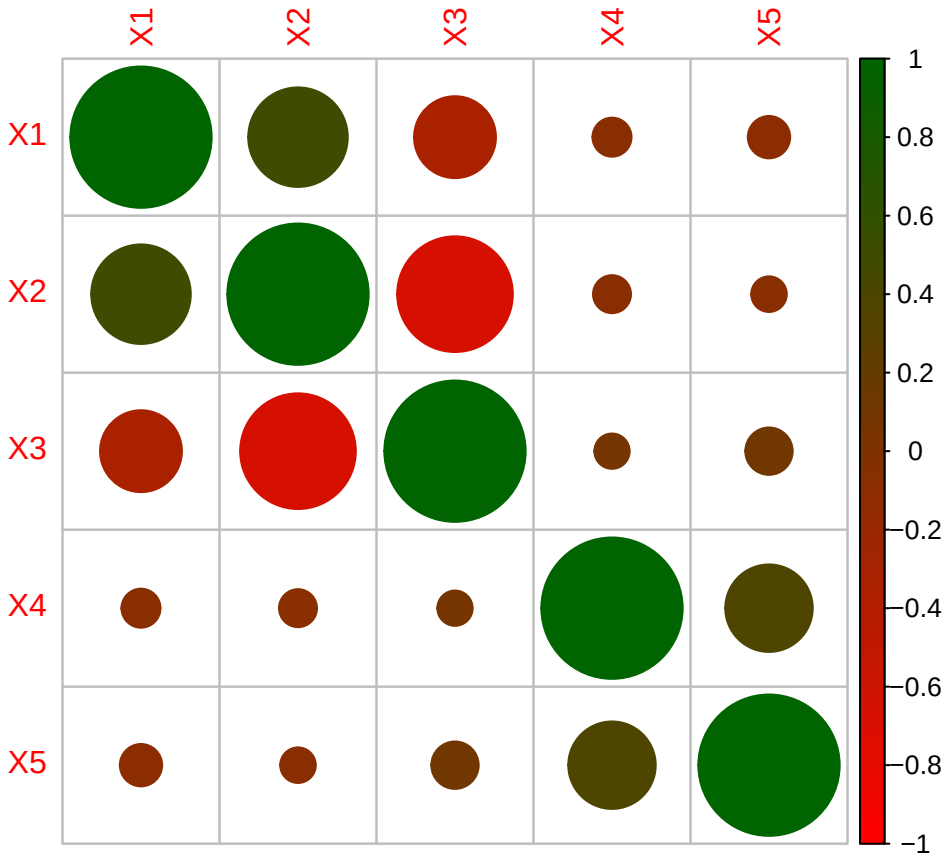
4

```
corrplot(cor_tab, method = "circle")
```



5

```
corrplot(cor_tab, method = "circle", col = colorRampPalette(c("red",
    "darkgreen"))(100))
```


X1
X2
X3
X4
X5
X1
X2
X3
X4
X5
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1