Practice: Data, Distributions, and Correlation

Jay Wei

2023-09-10

Contents

1 Exercise 1.

1.1 Question a.

```
library(mvtnorm)
library(rgl)
# find f(x) at x=mu
mu <- c(15, 20)
sigma <- matrix(data = c(1, 0.5, 0.5, 1.25), nrow = 2, ncol = 2, byrow = TRUE)

p <- 2 # 2 dimensional

fx <- 1/((2*pi)^(p/2)*det(sigma)^(1/2))
fx
```

```
## [1] 0.1591549
```

```
# use mvtnorm to check our result
dmvnorm(x = mu, mean = mu, sigma = sigma)
```

```
## [1] 0.1591549
```

1.2 Question b.

```
mu <- c(15, 20)
sigma <- matrix(data = c(1, 0.5, 0.5, 1.25), nrow = 2, ncol = 2, byrow = TRUE)

p <- 2

x <- c(14, 19)

fx <- exp(-0.5*(t(x-mu)%*%solve(sigma)%*%(x-mu)))/((2*pi)^(p/2)*det(sigma)^(1/2))

fx
```

```
##            [,1]
## [1,] 0.0851895
```

```
# use mvtnorm to check our result
dmvnorm(x = x, mean = mu, sigma = sigma)
```

```
## [1] 0.0851895
```

1.3 Question c.

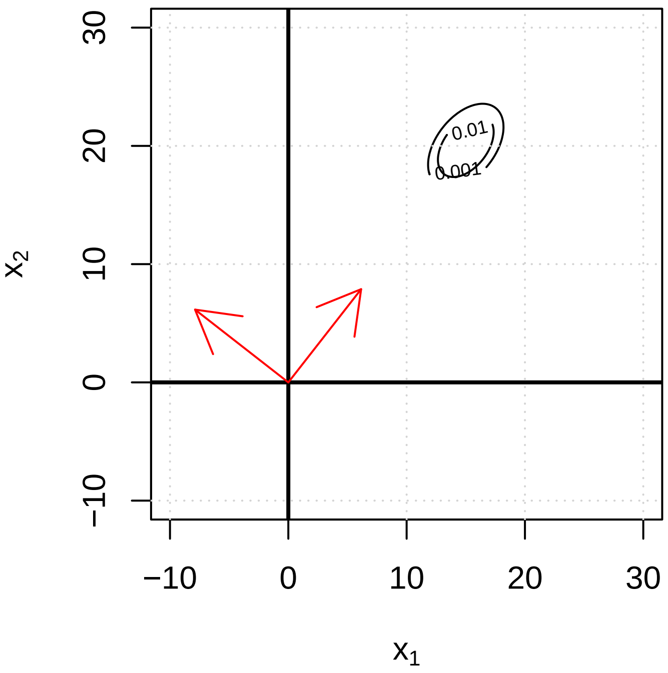
Plot the distribution and explore what happens to the distribution when changes are made to the mean vector and the covariance matrix. Make sure to examine the eigenvalues as well.

Check codes from the course carefully.

1.3.1 Benchmark

```
## Benchmark with mean vector: 15 20
##  covariance matrix:  1 0.5 0.5 1.25
##  and correlation matrix:  1 0.4472136 0.4472136 1
```

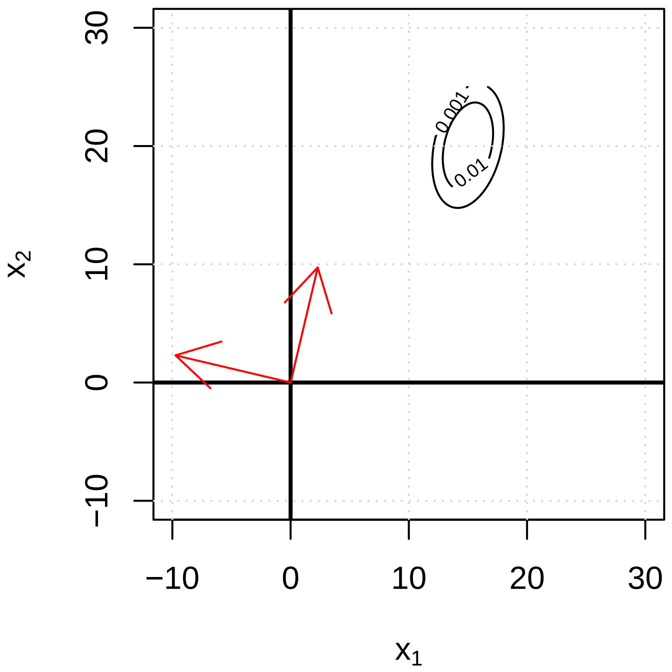
Multivariate normal with $\mu_1 = 15$, $\mu_2 = 20$, $\rho_{12} = 0.45$, and eigenvts for Σ



1.3.2 Increase the variance of x2

```
##  Mean vector: 15 20
##  covariance matrix:  1 0.5 0.5 3
##  and correlation matrix:  1 0.2886751 0.2886751 1
```

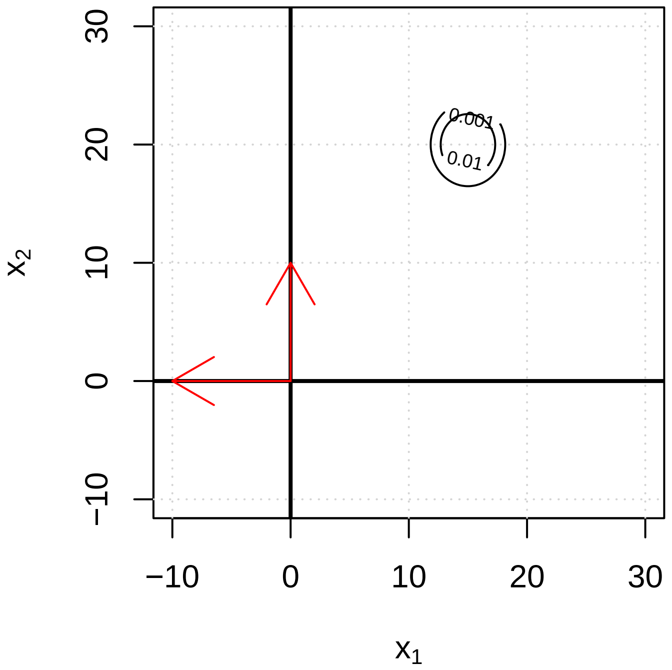
Multivariate normal with $\mu_1 = 15$, $\mu_2 = 20$, $\rho_{12} = 0.29$, and eigenvts for Σ



1.3.3 No linear correlation between x1 and x2

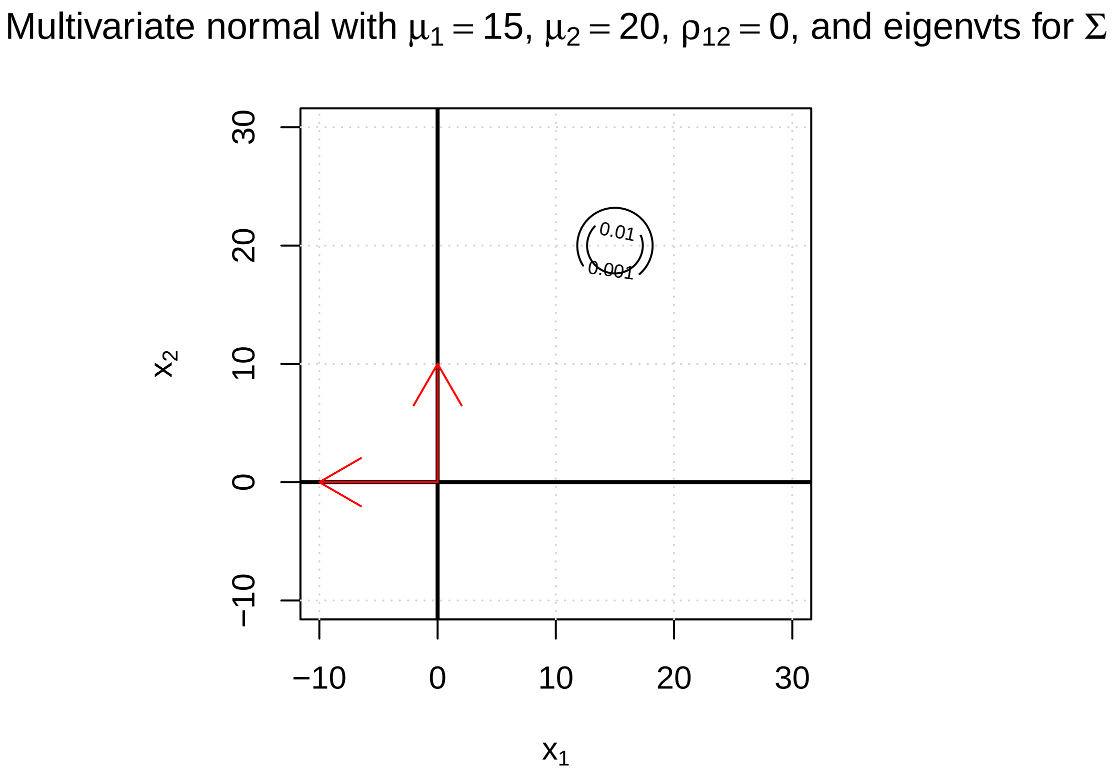
```
##  Mean vector: 15 20
##  covariance matrix:  1 0 0 1.25
##  and correlation matrix:  1 0 0 1
```

Multivariate normal with $\mu_1 = 15$, $\mu_2 = 20$, $\rho_{12} = 0$, and eigenvts for Σ



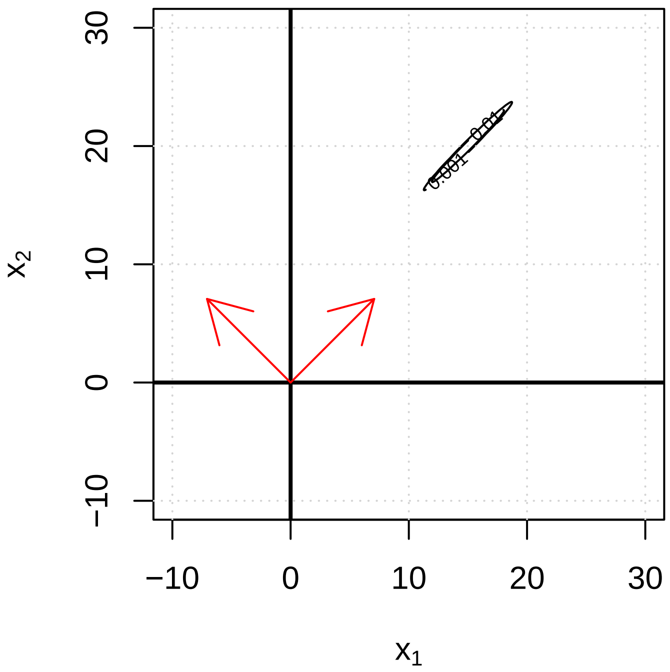
1.3.4 No linear corrleation between x1 and x2 with equal variance for x1 and x2

```
##  Mean vector: 15 20
##  covariance matrix:  1 0 0 1
##  and correlation matrix:  1 0 0 1
```



1.3.5 High linear correlation between x1 and x2

```
##  Mean vector: 15 20
##  covariance matrix:  1 0.99 0.99 1
##  and correlation matrix:  1 0.99 0.99 1
```



Multivariate normal with $\mu_1 = 15$, $\mu_2 = 20$, $\rho_{12} = 0.99$, and eigenvts for Σ

2 Exercise 2.

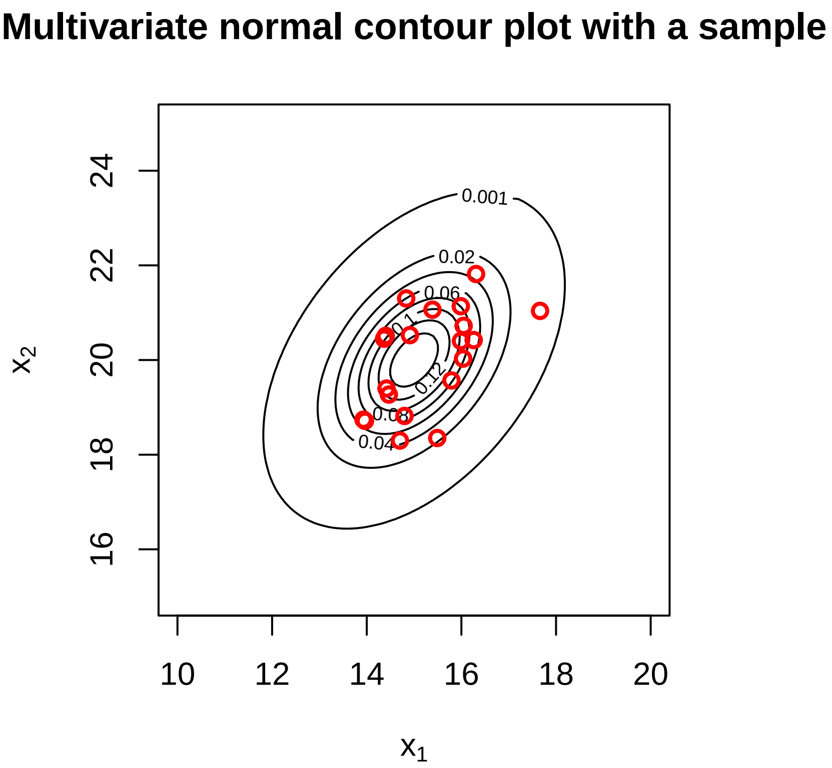
Simulate data from a multivariate normal distribution of your choice.

```
# true distribution

 mu <- c(15, 20) # true mean
  sigma <- matrix(data = c(1, 0.5, 0.5, 1.25), nrow = 2, ncol = 2, byrow = TRUE) # true co

  N <- c(20, 200, 2000, 20000)

  cat("True mean:", mu, "\n", "True covariance matrix:", sigma, "\n", "True correlation ma
```

```
## True mean: 15 20
##  True covariance matrix: 1 0.5 0.5 1.25
##  True correlation matrix: 1 0.4472136 0.4472136 1
```

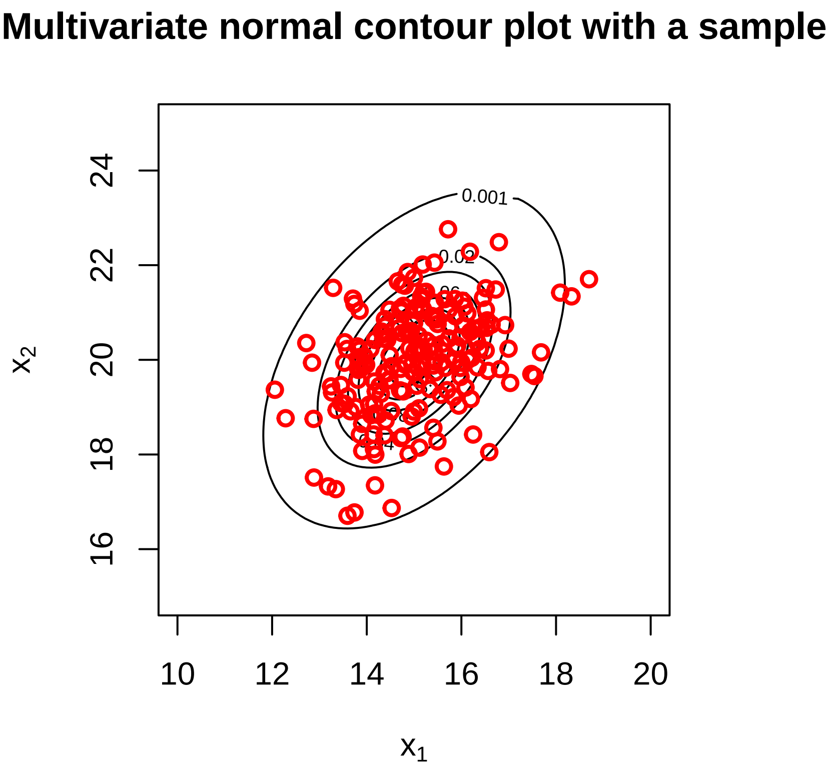
```
###################################################

  set.seed(7812) # Set a "seed number" so that I can reproduce the exact same sample



for (s in 1:length(N) ){
  x <- rmvnorm(n = N[s], mean = mu, sigma = sigma)
  #head(x)

  mu.hat <- colMeans(x) # simulated mean
  sigma.hat <- cov(x) # simulated covariance matrix
  R <- cor(x) # simulated correlation matrix
  cat("Number of observations:", N[s], "\n","Simulated mean:", mu.hat, "\n", "Simulated co


  x1 <- seq(from = 10, to = 25, by = 0.1)
  x2 <- seq(from = 10, to = 25, by = 0.1)
  all.x <- expand.grid(x1, x2)
  eval.fx <- dmvnorm(x = all.x, mean = mu, sigma = sigma)
  fx <- matrix(data = eval.fx, nrow = length(x1), ncol = length(x2), byrow = FALSE)

  par(pty = "s")
  contour(x = x1, y = x2, z = fx, main = "Multivariate normal contour plot with a sample",
    xlab = expression(x[1]), ylab = expression(x[2]), xlim = c(10,20), ylim = c(15, 25),
    levels = c(0.001, seq(from = 0.02, to = 0.14, by = 0.02)))
  points(x = x[,1], y = x[,2], col = "red", lwd = 2)


}
```

```
## Number of observations: 20
##  Simulated mean: 15.28776 20.02926
##  Simulated covariance matrix: 0.9288198 0.5473159 0.5473159 1.126895
##  Simulated correlation matrix 1 0.5349714 0.5349714 1
```



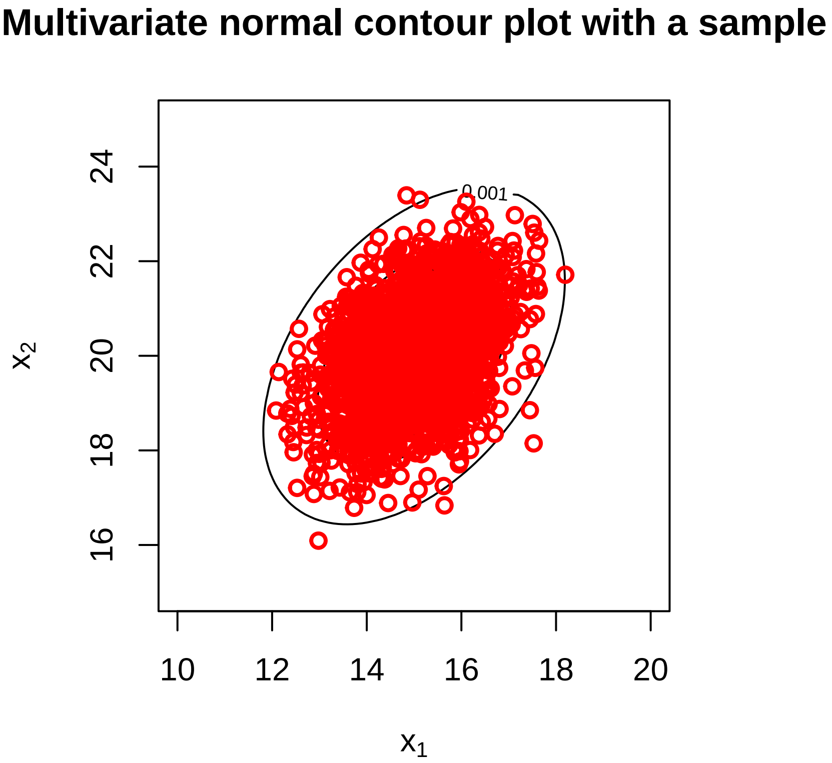
Multivariate normal contour plot with a sample

```
## Number of observations: 200
##  Simulated mean: 15.03036 19.99612
##  Simulated covariance matrix: 1.279074 0.4758293 0.4758293 1.232142
##  Simulated correlation matrix 1 0.3790295 0.3790295 1
```

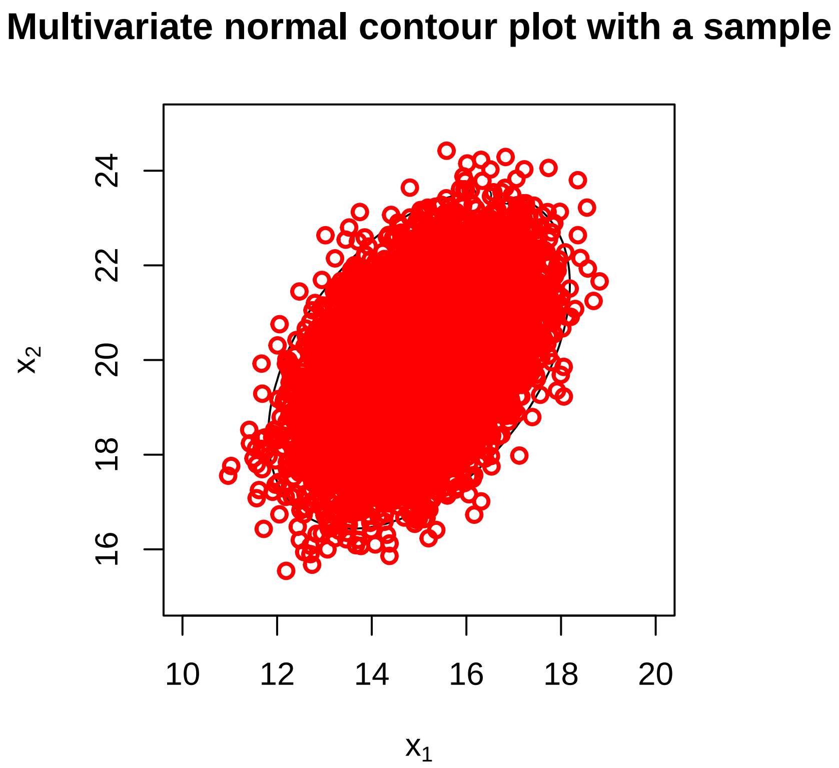


Multivariate normal contour plot with a sample

```
## Number of observations: 2000
##  Simulated mean: 14.98815 20.00306
##  Simulated covariance matrix: 0.9474827 0.4162896 0.4162896 1.213478
##  Simulated correlation matrix 1 0.3882343 0.3882343 1
```



```
## Number of observations: 20000
##  Simulated mean: 14.99788 20.00596
##  Simulated covariance matrix: 0.9899763 0.4951632 0.4951632 1.239669
##  Simulated correlation matrix 1 0.4469748 0.4469748 1
```


Multivariate normal contour plot with a sample
x_2
24
22
20
18
16
10
12
14
16
18
20
x_1

3 Exercise 3.

With respect to the goblet data described in the notes (data is in goblet.csv of Section 2), complete the following:

3.1 Question a.

Subject-matter researchers are interested in grouping goblets that have the same shape although they may have different sizes. One way suggested by Manly and Alberto (2016) is to adjust the data by dividing each measurement by X3 (height) and then complete additional analyses (shown later in our course). Below is how the data adjustment is done:

```
goblet <- read.csv("goblet.csv")
head(goblet)
```

```
##   goblet x1 x2 x3 x4 x5 x6
## 1      1 13 21 23 14  7  8
## 2      2 14 14 24 19  5  9
## 3      3 19 23 24 20  6 12
## 4      4 17 18 16 16 11  8
## 5      5 19 20 16 16 10  7
## 6      6 12 20 24 17  6  9
```

```
goblet2 <- data.frame(ID = goblet$goblet, w1 = goblet$x1/goblet$x3,
                                       w2 = goblet$x2/goblet$x3,
                                       w4 = goblet$x4/goblet$x3,
                                       w5 = goblet$x5/goblet$x3,
                                       w6 = goblet$x6/goblet$x3)
head(goblet2)
```

```
##   ID        w1        w2        w4        w5        w6
## 1  1 0.5652174 0.9130435 0.6086957 0.3043478 0.3478261
## 2  2 0.5833333 0.5833333 0.7916667 0.2083333 0.3750000
## 3  3 0.7916667 0.9583333 0.8333333 0.2500000 0.5000000
## 4  4 1.0625000 1.1250000 1.0000000 0.6875000 0.5000000
## 5  5 1.1875000 1.2500000 1.0000000 0.6250000 0.4375000
## 6  6 0.5000000 0.8333333 0.7083333 0.2500000 0.3750000
```

3.2 Question b.

Exhibit the adjusted data in a matrix

```
goblet2.mtx <- as.matrix(goblet2[,2:6])
goblet2.mtx
```

```
##              w1        w2        w4        w5        w6
##  [1,] 0.5652174 0.9130435 0.6086957 0.3043478 0.3478261
##  [2,] 0.5833333 0.5833333 0.7916667 0.2083333 0.3750000
##  [3,] 0.7916667 0.9583333 0.8333333 0.2500000 0.5000000
##  [4,] 1.0625000 1.1250000 1.0000000 0.6875000 0.5000000
##  [5,] 1.1875000 1.2500000 1.0000000 0.6250000 0.4375000
##  [6,] 0.5000000 0.8333333 0.7083333 0.2500000 0.3750000
##  [7,] 0.5454545 0.8636364 0.7272727 0.2727273 0.4545455
##  [8,] 0.4800000 0.8800000 0.6000000 0.2800000 0.2800000
##  [9,] 0.6470588 0.8823529 0.6470588 0.3529412 0.2941176
## [10,] 0.7857143 0.9285714 0.7857143 0.5000000 0.2857143
## [11,] 0.4800000 0.8000000 0.7200000 0.2000000 0.4800000
## [12,] 0.5652174 0.9130435 0.6521739 0.3913043 0.3478261
## [13,] 0.6315789 0.7894737 0.6315789 0.2631579 0.3157895
## [14,] 0.5000000 0.8461538 0.6538462 0.2692308 0.3846154
## [15,] 0.5384615 0.8461538 0.5769231 0.2692308 0.3461538
## [16,] 0.7000000 0.9500000 0.8500000 0.2500000 0.5000000
## [17,] 1.0000000 1.0666667 1.0000000 0.6000000 0.4666667
## [18,] 0.9500000 1.0500000 0.8000000 0.4500000 0.5000000
## [19,] 0.4615385 0.7692308 0.6153846 0.2692308 0.3846154
## [20,] 0.6296296 0.7407407 0.6666667 0.2222222 0.5185185
## [21,] 0.4814815 0.7407407 0.6296296 0.2222222 0.3333333
## [22,] 0.9000000 0.9000000 0.7000000 0.4000000 0.3000000
## [23,] 1.1428571 1.1428571 0.7142857 0.2857143 0.2857143
## [24,] 1.1250000 1.1250000 0.5000000 0.2500000 0.2500000
## [25,] 0.4444444 0.7037037 0.6666667 0.1851852 0.4444444
```

3.3 Question c.

Find the multivariate summary statistics for the adjusted data.

```
# mean
mu.hat <- colMeans(goblet2[,2:6])
mu.hat
```

```
##        w1        w2        w4        w5        w6
## 0.7079462 0.9040548 0.7231692 0.3303339 0.3882952
```

```
# Alternative way to find mu.hat
apply(X = goblet2[, 2:6], MARGIN = 2, FUN = mean)
```

```
##        w1        w2        w4        w5        w6
## 0.7079462 0.9040548 0.7231692 0.3303339 0.3882952
```

```
# covariance matrix
sigma.hat <- cov(goblet2.mtx)
sigma.hat
```

```
##              w1          w2          w4         w5           w6
## w1 0.0588151251 0.032664837 0.017045208 0.02224044 0.0004540383
## w2 0.0326648368 0.024199433 0.009612116 0.01494699 0.0003982420
## w4 0.0170452077 0.009612116 0.017656026 0.01329360 0.0068352800
## w5 0.0222404418 0.014946989 0.013293604 0.01940272 0.0018132298
## w6 0.0004540383 0.000398242 0.006835280 0.00181323 0.0072358002
```

```r
# correlation matrix
R <- cov2cor(sigma.hat)
R
```

```
##             w1         w2        w4        w5         w6
## w1 1.00000000 0.86583150 0.5289460 0.6583664 0.02200922
## w2 0.86583150 1.00000000 0.4650182 0.6897947 0.03009547
## w4 0.52894596 0.46501816 1.0000000 0.7182324 0.60473700
## w5 0.65836637 0.68979469 0.7182324 1.0000000 0.15303061
## w6 0.02200922 0.03009547 0.6047370 0.1530306 1.00000000
```

```r
# or simply
cor(goblet2[,2:6])
```

```
##             w1         w2        w4        w5         w6
## w1 1.00000000 0.86583150 0.5289460 0.6583664 0.02200922
## w2 0.86583150 1.00000000 0.4650182 0.6897947 0.03009547
## w4 0.52894596 0.46501816 1.0000000 0.7182324 0.60473700
## w5 0.65836637 0.68979469 0.7182324 1.0000000 0.15303061
## w6 0.02200922 0.03009547 0.6047370 0.1530306 1.00000000
```

3.4 Question d.

Compute the covariance of the adjusted X1 and X2 variables without the use of `cov()` or any function that immediately finds the covariance. In other words, program the actual formula into R that computes the covariance for two variables and apply it to data from these variables.

```r
X <- goblet2.mtx
X.tilde <- t(t(X) - mu.hat)
N <- nrow(X)
t(X.tilde)%*%X.tilde/(N-1)
```

```
##              w1          w2          w4         w5           w6
## w1 0.0588151251 0.032664837 0.017045208 0.02224044 0.0004540383
## w2 0.0326648368 0.024199433 0.009612116 0.01494699 0.0003982420
## w4 0.0170452077 0.009612116 0.017656026 0.01329360 0.0068352800
## w5 0.0222404418 0.014946989 0.013293604 0.01940272 0.0018132298
## w6 0.0004540383 0.000398242 0.006835280 0.00181323 0.0072358002
```