1

Exercise ~~1(a)~~ II.8.7(a)

```
> Sigma=array(c(21,26,24,26,34,30,24,30,36),dim=c(3,3))
> Sigma
     [,1] [,2] [,3]
[1,]   21   26   24
[2,]   26   34   30
[3,]   24   30   36
```

$= \Sigma$

```
> e=eigen(Sigma)
> e
eigen() decomposition
$values
[1] 84.6417667  5.6848871  0.6733463
```

eigenvalues of Σ

```
$vectors
           [,1]       [,2]        [,3]
[1,] -0.4855890 -0.3243337  0.81179488
[2,] -0.6164771 -0.5313566 -0.58104748
[3,] -0.6198058  0.7826033 -0.05807662
```

Q columns are eigenvectors of Σ

```
> Q=e$vectors
> Lambda=diag(e$values)
> Q
           [,1]       [,2]        [,3]
[1,] -0.4855890 -0.3243337  0.81179488
[2,] -0.6164771 -0.5313566 -0.58104748
[3,] -0.6198058  0.7826033 -0.05807662
> Lambda
         [,1]     [,2]      [,3]
[1,] 84.64177 0.000000 0.0000000
[2,]  0.00000 5.684887 0.0000000
[3,]  0.00000 0.000000 0.6733463
```

check that I have assigned Q and Λ correctly

```
> Sigmasqrt=Q%*%sqrt(Lambda)%*%t(Q)
> Sigmasqrt%*%Sigmasqrt
     [,1] [,2] [,3]
[1,]   21   26   24
[2,]   26   34   30
[3,]   24   30   36
```

check that I computed $\Sigma^{1/2}$ correctly

```
> Sigmasqrt
         [,1]     [,2]     [,3]
[1,] 2.960932 2.777934 2.125080
[2,] 2.777934 4.446664 2.551521
[3,] 2.125080 2.551521 4.997377
>
```

$= \Sigma^{1/2}$

Exercise VI.8.7(b)

Using the R command chol is much easier than using qr

```
> R=chol(Sigma)
> Q=Sigmasqrt%*%solve(R)
> t(Q)%*%Q
             [,1]         [,2]         [,3]
[1,] 1.000000e+00 4.139050e-15 1.276756e-15
[2,] 4.139050e-15 1.000000e+00 4.315992e-15
[3,] 1.276756e-15 4.315992e-15 1.000000e+00
> R
         [,1]     [,2]      [,3]
[1,] 4.582576 5.673665 5.2372294
[2,] 0.000000 1.345185 0.2123977
[3,] 0.000000 0.000000 2.9199856
> t(R)%*%R
     [,1] [,2] [,3]
[1,]   21   26   24
[2,]   26   34   30
[3,]   24   30   36
```

= cholesky factor

$= \Sigma^{1/2} R^{-1} = Q$ in Q-R decomposition

$= R$

$R'R = \Sigma$ reproduces Σ exactly in this case

Exercise II.8.8

Exercise II.8.8 (a)

```
--------------------
> # create a row vector of mu=(0,1,2)
> mu=array(c(0,1,2),dim=c(1,3))
> # create a column vect of 1000 1's
> one=array(1+0*(1:1000),dim=c(1000,1))
> # creat 1000x3 matrix with each row equal to mu
> Mu=one%*%mu
> # generate 3000 N(0,1) values
> samle=rnorm(3000,0,1)
> # construct 1000 3 dimensional rows of sample vectors from N_3(mu, Sigma)
distribution
> samplevec=Mu + array(samle,dim=c(1000,3))%*%Sigmasqrt
```

using $\underset{\sim}{\mu} + \Sigma^{1/2}\underset{\sim}{Z}$

```
> # create a column vector (1,1,1)'
> one=array(c(1,1,1),dim=c(3,1))
> # square each element in samplevec and sum the squares in each to get squared
length and then tyake square root
> length=sqrt((samplevec*samplevec)%*%one)
>
> # count how many of the lengths are <= 10
> count=0
> for (i in 1:1000) {
+ if (length[i] <= 10){ count= count+1}
+ }
> count
[1] 679
> # get the estimate of the probability
> prop=count/1000
> error=sqrt(prop*(1-prop)/1000)
> # here is the estimate and the interval containing true value with virtual
certainty
> prop
[1] 0.679
> prop-3*error
[1] 0.6347097
> prop+3*error
[1] 0.7232903
>
```

Exercise II.8.8 (b)

```
-------------------
> # create a row vector of mu=(0,1,2)
> mu=array(c(0,1,2),dim=c(1,3))
> # create a column vect of 1000 1's
> one=array(1+0*(1:1000),dim=c(1000,1))
> # creat 1000x3 matrix with each row equal to mu
> Mu=one%*%mu
> # generate 3000 N(0,1) values
> samle=rnorm(3000,0,1)
> # construct 1000 3 dimensional rows of sample vectors from N_3(mu, Sigma)
```

```
distribution
> samplevec=Mu + array(samle,dim=c(1000,3))%*%R
```

using $\underset{\sim}{\mu} + R'\underset{\sim}{z}$

```
> # create a column vector (1,1,1)'
> one=array(c(1,1,1),dim=c(3,1))
> # square each element in samplevec and sum the squares in each to get squared
length and then tyake square root
> length=sqrt((samplevec*samplevec)%*%one)
>
> # count how many of the lengths are <= 10
> count=0
> for (i in 1:1000) {
+ if (length[i] <= 10){ count= count+1}
+ }
> count
[1] 680
> # get the estimate of the probability
> prop=count/1000
> error=sqrt(prop*(1-prop)/1000)
> # here is the estimate and the interval containing true value with virtual
certainty
> prop
[1] 0.68
> prop-3*error
[1] 0.6357462
> prop+3*error
[1] 0.7242538
```

estimates are similar

Exercise VI.8.9 $X_2 | X_1 = 2 \sim N\left(\mu_2 + \frac{\sigma_{12}}{\sigma_{11}}(2-\mu_1), \sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}\right)$

$= N\left(2 + \frac{1/2}{5/2}(2-1), 5/2 - (1/2)^2/5/2\right) = N(2^1/5, 2^2/5)$

(b) $P_{X_2|X_1}(A|x_1) = P_{X_2|X_1}(\{x_2 : x_2^2 \le 5 - 2^2 = 1\} | 2)$

$= P_{X_2|X_1}(-1 \le X_2 \le 1 | 2) = P_{X_2|X_1}\left(\frac{-1-2.2}{\sqrt{2.4}} \le \frac{X_2 - 2.2}{\sqrt{2.4}} \le \frac{1-2.2}{\sqrt{2.4}}\right)$

$= P(-2.066 \le Z \le -0.775)$ where $Z \sim N(0,1)$

$= \Phi(-0.775) - \Phi(-2.066) = 0.120$

(c) It is difficult to evaluate $P_{(X_1,X_2)}(A)$ directly so we proceed via Monte Carlo.

```
> mu=c(1,2)
> Sigma=array(c(5/2,1/2,1/2,5/2),dim=c(2,2))
> Sigma
     [,1] [,2]
[1,]  2.5  0.5
[2,]  0.5  2.5
> mu
[1] 1 2
> library(MASS)
> X=mvrnorm(1000, mu, Sigma)
> y=X[,1]**2+X[,2]**2
> mean(y<=5)
[1] 0.346
> X=mvrnorm(10000,mu,Sigma)
> y=X[,1]**2+X[,2]**2
> mean(y<=5)
[1] 0.3619
> X=mvrnorm(100000,mu,Sigma)
> y=X[,1]**2+X[,2]**2
> mean(y<=5)
[1] 0.3659
> X=mvrnorm(1000000,mu,Sigma)
> y=X[,1]**2+X[,2]**2
> mean(y<=5)
[1] 0.365723
```

$\underset{\sim}{\mu}$

$= \Sigma$

— install library containing mvrnorm to generate from $N_2(\underset{\sim}{\mu}, \Sigma)$

— estimate of $P_{(X_1,X_2)}(A)$ based on a sample of $n = 10^3$

— estimate based on $n = 10^4$

— estimate based on $n = 10^5$

— estimate based on $n = 10^6$ (2 decimal places)