mertgöksel

Question 1

```
#1
a <- seq(20,100,20)
a
```

```
## [1]  20  40  60  80 100
```

```
#2
b <- seq(-10.0, 5.0, .5)
b
```

```
##  [1] -10.0  -9.5  -9.0  -8.5  -8.0  -7.5  -7.0  -6.5  -6.0  -5.5  -5.0  -4.5
## [13]  -4.0  -3.5  -3.0  -2.5  -2.0  -1.5  -1.0  -0.5   0.0   0.5   1.0   1.5
## [25]   2.0   2.5   3.0   3.5   4.0   4.5   5.0
```

```
#3
c <- seq(12, -6, -3)
c
```

```
## [1] 12  9  6  3  0 -3 -6
```

```
#4
d <- seq(.01, .09, .02)
d
```

```
## [1] 0.01 0.03 0.05 0.07 0.09
```

```
#5
e <- seq(17, 23, 6/7)
e
```

```
## [1] 17.00000 17.85714 18.71429 19.57143 20.42857 21.28571 22.14286 23.00000
```

```
#6
f <- c(2:10, 9:2)
f
```

```
##  [1]  2  3  4  5  6  7  8  9 10  9  8  7  6  5  4  3  2
```

```
#7
g <- rep(seq(10,40,10), 4)
g
```

```
##  [1] 10 20 30 40 10 20 30 40 10 20 30 40 10 20 30 40
```

```
#8
h <- rep(c(1:3), each = 7)
h
```

```
##  [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
```

```
#9
i <- rep(seq(10,60,10), each = 2)[seq(-1,-9,-4)]
```

```r
i
```

```
## [1] 10 20 20 30 40 40 50 60 60
```

```r
#10
j <- rep(rep(1:4, each = 2), 3)
j
```

```
##  [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
```

Question 2

```r
# Sigma notations are created with 3 lines;
# 1st being the range,
# 2nd being the formula,
# 3th being the sum() function.

#1
x <- c(1:100)
x <- x*2
sum(x)
```

```
## [1] 10100
```

```r
#2
x <- c(1:100)
x <- 1/(x^2)
sum(x)
```

```
## [1] 1.634984
```

```r
#3
x <- c(1:10)
x <- exp(-.5)*(.5^x)/factorial(x)
sum(x)
```

```
## [1] 0.3934693
```

```r
#4
x <- c(1:20)
x <- ((-1)^x)*x^2
sum(x)
```

```
## [1] 210
```

Question 3

```r
#Enter the column and row index vectors
n_row = c(1:3)
n_col = c(1:4)
```

a (for multiplications)

```r
# "tcrossprod()" lets us to create the desired matrix that
# contains the multiplications of the elements

# "dimnames()" lets us assign a group name to the rows and columns
```

```r
the_matrix <- matrix(tcrossprod(n_col, n_row), nrow = 3, ncol = 4, byrow = TRUE,
                     dimnames = list(Spanish = c("uno", "dos", "tres"),
                                     German = c("eins", "zwei", "drei", "vier")))
the_matrix
```

```
##        German
## Spanish eins zwei drei vier
##    uno     1    2    3    4
##    dos     2    4    6    8
##    tres    3    6    9   12
```

b (change "the_matrix" values to sum of row & column)

```r
sums = outer(n_row, n_col, "+") # "outer()" lets us do algebraic operations with two
# vectors of different sizes as it matches their size first
# We can also use "sapply(n_row, '+', n_col)"
the_matrix[1:12] <- sums
the_matrix
```

```
##        German
## Spanish eins zwei drei vier
##    uno     2    3    4    5
##    dos     3    4    5    6
##    tres    4    5    6    7
```

Question 4

```r
vec1 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
vec2 = c(10, 20, 30, 40, 50, 60)
vec3 = c("Red", "Blue", "Yellow", "Green", "Black")
vec4 = c(-1, -2, 10, 20, -3, -4, 30, 40, -5, 50)
```

```r
#a
q4.a = c(vec1, vec2)
q4.a
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 10 20 30 40 50 60
```

```r
#b
q4.b = q4.a[q4.a%%3 == 0]
q4.b
```

```
## [1]  3  6  9 30 60
```

```r
#c
q4.c = c(vec4[vec4<0], vec3)
q4.c
```

```
##  [1] "-1"     "-2"     "-3"     "-4"     "-5"     "Red"    "Blue"   "Yellow"
##  [9] "Green"  "Black"
```

```r
#d
q4.d = c(vec1[seq(1, length(vec1), 2)], vec3[seq(2, length(vec3), 2)])
q4.d
```

```
## [1] "1"     "3"     "5"     "7"     "9"     "Blue"  "Green"
```

```
#e
sizes = c(length(vec1), length(vec2), length(vec3), length(vec4))
which(sizes == max(sizes)) #gives the indexes of what we searched
```

```
## [1] 1 4
```

Question 5

```
a <- matrix(c(4,3,-5,-1), nrow = 2, ncol = 2)
a
```

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

```
b <- matrix(c(8,1,1,-6), nrow = 2, ncol = 2)
b
```

```
##      [,1] [,2]
## [1,]    8    1
## [2,]    1   -6
```

```
c <- matrix(c(0,-2,1,4), nrow = 2, ncol = 2)
c
```

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

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

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    2    0
```

```
#a
t(a) + b %*% solve(c)
```

```
##      [,1] [,2]
## [1,]   21 -1.0
## [2,]   -9 -1.5
```

```
#b
sizes = c(det(a),det(b),det(c),det(d))
which(sizes == max(sizes))
```

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

```
#c
e = cbind(rbind(a,c), rbind(b,d))
e
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4   -5    8    1
## [2,]    3   -1    1   -6
## [3,]    0    1    2    1
## [4,]   -2    4    2    0
```

```
#d
mean(e[,3])
```

```
## [1] 3.25
```

```
#e
diag(e) = 0 #diagonal elements of the matrix
e
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0   -5    8    1
## [2,]    3    0    1   -6
## [3,]    0    1    0    1
## [4,]   -2    4    2    0
```