Homework #1

Carson Crenshaw (cgc8gdt)

Problem 1

Part a

Using the sequence function, data is created and stored in vector1.

```
vector1 <- seq(to=5206,from=5176,by=3)
vector1
```

```
##  [1] 5176 5179 5182 5185 5188 5191 5194 5197 5200 5203 5206
```

Part b

Using the repeat function, character data is created and stored in vector2.

```
vector2 <- rep("STAT", 3)
vector2
```

```
## [1] "STAT" "STAT" "STAT"
```

Part c

Using the syntax a:b creates a vector that starts at a (77) and increases by one until b (93). This vector is stored in vector3.

```
vector3 <- (77:93)
vector3
```

```
##  [1] 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
```

Part d

To repeat each element in the vector a specific number of times, the each option is used. The resulting vector is created and stored in vector4.

```
vector4 <- rep(10:15, each=2)
vector4
```

```
##  [1] 10 10 11 11 12 12 13 13 14 14 15 15
```

Part e

Using the sequence function, data is created and stored in vector5 in descending order.

```
vector5 <- seq(to=68,from=103,by=-7)
vector5
```

```
## [1] 103  96  89  82  75  68
```

Part f

The concatenate function is used to combine multiple components, but not as the main mechanism in storing the values in vector6.

```
vector6 <- c(seq(35,55,5), seq(45,15,-10))
vector6
```

```
## [1] 35 40 45 50 55 45 35 25 15
```

Part g

Combination of the repeat, sequence, and concatenate function to create vector7.

```
vector7 <- rep(seq(to=200,from=100,by=25), c(5,4,3,2,1))
vector7
```

```
##  [1] 100 100 100 100 100 125 125 125 125 150 150 150 175 175 200
```

Part h

Basic use of the repeat function to create and store vector8.

```
vector8 <- rep(6:8, 3)
vector8
```

```
## [1] 6 7 8 6 7 8 6 7 8
```

Part i

Replicated code from part f.

```
vector9 <- c(seq(124,94,-6), seq(92,82,-2))
vector9
```

```
##  [1] 124 118 112 106 100  94  92  90  88  86  84  82
```

Part j

Using the repeat and concatenate function, character data is created and stored in vector10.

```
vector10 <- rep(c("Eggs", "Bacon", "Pancakes"), 3)
vector10
```

```
## [1] "Eggs"     "Bacon"    "Pancakes" "Eggs"     "Bacon"    "Pancakes" "Eggs"
## [8] "Bacon"    "Pancakes"
```

Problem 2

Part a

The probability that a year is drier than 1987 is approximately 3% (2.94%).

```
pnorm(697, mean= 852, sd=82)
```

```
## [1] 0.02936267
```

Part b

Rainfall between 809mm and 895mm is considered typical in India.

```
percentile30 <- qnorm(0.3, mean=852, sd=82)
percentile70 <- qnorm(0.7, mean=852, sd=82)
typicalrainrange <- c(percentile30,percentile70)
typicalrainrange
```

```
## [1] 808.9992 895.0008
```

Part c

Assuming the mean and standard deviation have remained constant over the past 500 years, approximately 18 (17.77) years were expected to have had flooding.

```
percentageofflooding <- pnorm(1000, mean=852, sd=82,lower.tail=FALSE)
percentageofflooding
```

```
## [1] 0.03554688
```

```
percentageofflooding*500
```

```
## [1] 17.77344
```

Problem 3

Part a

The probability that the guest hits the target at least once is 94.37%.

```
# when the lower tail = FALSE, the probabilities are P[x >x], so one has to
# use "0" in the equation to account for one or more successes.
pbinom(0, size=10, prob=0.25, lower.tail = FALSE) # one or more success (hit)
```

```
## [1] 0.9436865
```

Part b

```
probinput <- 0:9
# because the lower tail does not include the value for x, one must start one
# value lower, resulting in the vector holding values 0:9.
roundedprobs <- round(pbinom(probinput, size=10, prob=0.25, lower.tail = FALSE),3)
roundedprobs
```

```
##  [1] 0.944 0.756 0.474 0.224 0.078 0.020 0.004 0.000 0.000 0.000
```

Part c

Regardless of the values used in the new numerical vector, the values in the resulting matrix all round to the same decimal place as the other entries within the vector/the probabilities in the second row. Within each column the decimal places must match between values.

```
minnumberofhits <- 1:10
probmatrix <- rbind(minnumberofhits,roundedprobs)
probmatrix
```

```
##                  [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7] [,8] [,9] [,10]
## minnumberofhits 1.000 2.000 3.000 4.000 5.000 6.00 7.000    8    9    10
## roundedprobs    0.944 0.756 0.474 0.224 0.078 0.02 0.004    0    0     0
```

Part d

```
names(roundedprobs) <- minnumberofhits #descriptive vector created in part (c)
roundedprobs
```

```
##     1     2     3     4     5     6     7     8     9    10
## 0.944 0.756 0.474 0.224 0.078 0.020 0.004 0.000 0.000 0.000
```

Part e

The labeled vector presentation method (d) is easier to read because the values are concise and are not lost as additional values in a matrix display. As a result of the matrix rounding the numeric values of the descriptive vector, it makes it harder for an audience to understand that the values are suppose to be represented the minimum number of throws a person makes.

Part f

```
newdt <- data.frame(minnumberofhits,roundedprobs) # new data frame
newdt
```

```
##    minnumberofhits roundedprobs
## 1                1        0.944
## 2                2        0.756
## 3                3        0.474
## 4                4        0.224
## 5                5        0.078
## 6                6        0.020
## 7                7        0.004
## 8                8        0.000
## 9                9        0.000
## 10              10        0.000
```

Part g

```
newdt2 <- data.frame(roundedprobs) # new data frame with only roundedprobs
newdt2
```

```
##    roundedprobs
## 1         0.944
## 2         0.756
## 3         0.474
## 4         0.224
## 5         0.078
## 6         0.020
## 7         0.004
## 8         0.000
## 9         0.000
## 10        0.000
```

Part h

The first data frame (the data frame containing the descriptive vector and the rounded probabilities as columns) is a better presentation of this information because it gives the additional information of what the percentages are representing in the data frame. The column "minnumberofhits" helps to describe what "roundedprobs" is communicating in the table, while just the index/row numbers alone do not mean anything in the context of this problem.

Problem 4

```
school_name <- c("UVA", "VA Tech")
Conference <- c("ACC")
ranking <- c(2,12)
names(ranking) <- c("UVA","VT")
good_team <- c(TRUE,FALSE)
names(good_team) <- c("UVA","VT")

UVA <- c("Bennett", "Williford")
VT <- c("NA")
best_coaches <- list(UVA, VT)
names(best_coaches) <- c("UVA", "VT")

problemlist <- list(school_name = school_name,
                    Conference=Conference, ranking=ranking,
                    good_team=good_team,best_coaches=best_coaches)

problemlist
```

```
## $school_name
## [1] "UVA"     "VA Tech"
##
## $Conference
## [1] "ACC"
##
## $ranking
## UVA  VT
##   2  12
##
## $good_team
##   UVA    VT
##  TRUE FALSE
##
## $best_coaches
## $best_coaches$UVA
## [1] "Bennett"   "Williford"
##
## $best_coaches$VT
## [1] "NA"
```