This is closed-note, closed-script, and closed-internet exam. That is, students are not allowed to reference their course notes, any previously developed R script, or any internet site while completing the exam. Students should only use the internet to download and upload the midterm documents from the Sakai course site. At any other point during the exam, internet browsers should be closed. The only R script that should be open in R Studio is the one that accompanies the exam. Students are permitted to use the R Help Pages within R Studio during the exam. Failure to comply with these rules will result in a score of zero on the exam and being reported to the B&B MB Program for an Honor Code Violation.

- Show ALL work! Partial credit will be given.
- Please save your responses, R code, and R output in this word document. Save the file using the naming convention below, and upload to your Sakai Drop Box:
  - LastName_FirstName_Midterm.docx
- By typing your name below, you are agreeing to abide by the Duke Honor Code.
  - Name:

---

1. Consider the R object THING printed below:

```
> THING
$n
     [,1] [,2] [,3]
[1,]    2    5    0
[2,]    3   10    7

$s
[1] "aa" "bb" "cc" "dd" "ee"

$b
[1]  TRUE FALSE  TRUE FALSE FALSE
```

   a. What type of object is THING? Explain your choice. (5 pts)

      The object THING is a list because its elements consist of different objects (with different modes).

   b. Suppose the following R commands were submitted to the processor:

      ```
      THING[[2]][1] <- "ta"; THING[[2]]
      ```

      What would be returned to the console? If output, please provide the expected output. If an error, explain what the error would be. Note: If the command would execute, but

may return a warning, you do NOT need to list the warning, just provide the output that would be returned to the console. (5 pts)

Note: The command [[2]] selects the 2nd element of THING; the command [1] selects the 1st element of the 2nd element of THING; the command <- "ta" replaces the 1st element of the 2nd element of THING with the character string "ta". Thus, the following vector would be printed to the console if these R commands were submitted to the processor:

```
[1] "ta" "bb" "cc" "dd" "ee"
```

c. What type of variables are stored in the object named b in THING? If possible, list the mode of the object. If not, explain why there is not enough information to determine the mode of the object. (5 pts)

The object named b in THING is a vector so the elements must be of the same mode. Because the values TRUE and FALSE were not printed with quotes, we know that the elements of b must be logicals.

2. Suppose a wild life biologist is planning to study the different species of song-birds present at a local conservatory. The researcher collects birds by setting a small, no-harm trap that is big enough for a single bird. The researcher is able to check the trap once a day to collect specimens. Suppose there are 4 species of song-birds on the conservatory with the following distribution: 50%, 20%, 20%, and 10%, respectively. The researcher would like to estimate the number of days he will need to set traps in order to obtain 10 birds of each species. It is safe to assume that he will be able to catch a bird each day.
    a. Develop R code that will estimate the number of days the researcher will need to set traps in order to obtain 10 birds of each species. Make sure that the results of the estimation are reproducible each time the R code is submitted. Please submit R code and output below. (20 pnts)
        i. Note: The R command below can be used to simulate the random selection of species that will be present in the trap on a given day: `rmultinom(1,1,prob=c(0.5,0.2,0.2,0.1))`. It creates a p by 1 matrix of zeros and ones where p is the length of the 'prob' vector supplied to the rmultinom() function and the 1 indicates the species of bird that species present in the trap on a given day.

Note: Using the seed below, the number of days needed to trap 10 (at least) of each species is 72.

R Code:

```
set.seed(3214)            # Set the seed so the results are reproducible
trapped <- matrix(0,4,1)  # Create a storage vector to count No. of Species
days <- 0                 # Create a counter to estimate No. of Days set traps

while(sum(trapped>=10)!=4) { # Keep setting traps until have 10 of each species
  days <- days+1
  trapped <- trapped+rmultinom(1,1,prob=c(0.5,0.2,0.2,0.1))
}

trapped          # Print results ...
days
```

R Output:

```
> trapped          # Print results ...
[,1]
[1,]   33
[2,]   11
[3,]   18
[4,]   10

> days
[1] 72
```

b. The researcher is planning to conduct this study at several conservatories around North Carolina. The number of song-bird species and their distribution may vary across the conservatories. Using the R code developed in part (a), create a function that will estimate the number of days the researcher will have to set traps in order to obtain N birds of each species for any number of species, with any distribution, for any value of N. Call this function Trap_Days. Make sure that the results of the estimation are reproducible each time the function is run for the same set of parameters. Demonstrate that the function can produce the same results given in part (a) and can produce an estimate for a new value of N, a new number of species, and new distribution of species (the distribution of proportions must add to 1). Please submit R code and output below. (15 pnts)

Note: To create this function, just need to generalize the code developed in part (a) so that it works for any species count (i.e. replace the 10 with a function input) and for any distribution of species proportions (i.e. replace the c(0.5,0.2,0.2,0.1) with a function input). To make sure that the results are reproducible, need to include the seed number as a function input.

R Code:

```
# - Create function Trap_Days using the code developed in part (a)
Trap_Days <- function(N,probs,seed) {
set.seed(seed)              # Set the seed so the results are reproducible
no.s <- length(probs)       # Determine the No. of Species under study
trapped <- matrix(0,no.s,1) # Create a storage vector to count No. of Species
days <- 0                 # Create a counter to estimate No. of Days set traps

while(sum(trapped>=N)!=no.s) { # Keep setting traps until have N of each
    days <- days+1
    trapped <- trapped+rmultinom(1,1,prob=probs)
   }

return(list(days=days,trapped=trapped)) }
```

R Output:

```
> # - Show that the function works for scenario in part (a)
> est1 <- Trap_Days(10,probs=c(0.5,0.2,0.2,0.1),seed=3214)
> est1$days
[1] 72
>
> # - Show that the function works for another scenario
> est2 <- Trap_Days(7,probs=c(0.3,0.2,0.2,0.1,0.15,0.05),seed=555)
> est2$days
[1] 170
```

3. Suppose the data frame DATA contains weekly biomarker measures for subjects enrolled in a clinical study along with their gender.

```
set.seed(3214)
Week1 <- rnorm(100,300,5)
Week2 <- rnorm(100,290,5)
Week3 <- c(rnorm(88,250,8),rep(NA,12))
Week4 <- c(rnorm(73,225,8),rep(NA,27))
Gender <- sample(c('Female','Male'),100,replace=TRUE)
DATA <- data.frame(Week1,Week2,Week3,Week4,Gender)
```

Use this data frame to answer the questions below.

a. Create a data frame that contains the observations for all male subjects enrolled in the study. Call the data frame MALES. Please submit R code below. (5 pts)

R Code:

```
MALES <- DATA[which(DATA$Gender=="Male"),]
```

b. Create a 2 by 4 matrix that contains the weekly means in the first row and the weekly standard deviations in the second row for non-missing values among males. Call the

matrix StatsM. Name the rows Mean and StdDev, respectively. Name the columns Week 1 … Week 4, respectively. Please submit R code and output below. (10 pnts)

R Code:

```
# - Compute summary measures by column using the apply() fucntion
#   with dimension set to 2
Means <- apply(MALES[,1:4],2,mean,na.rm=TRUE)
StdDev <- apply(MALES[,1:4],2,sd,na.rm=TRUE)
StatsM <- rbind(Means,StdDev)
StatsM
```

R Output:

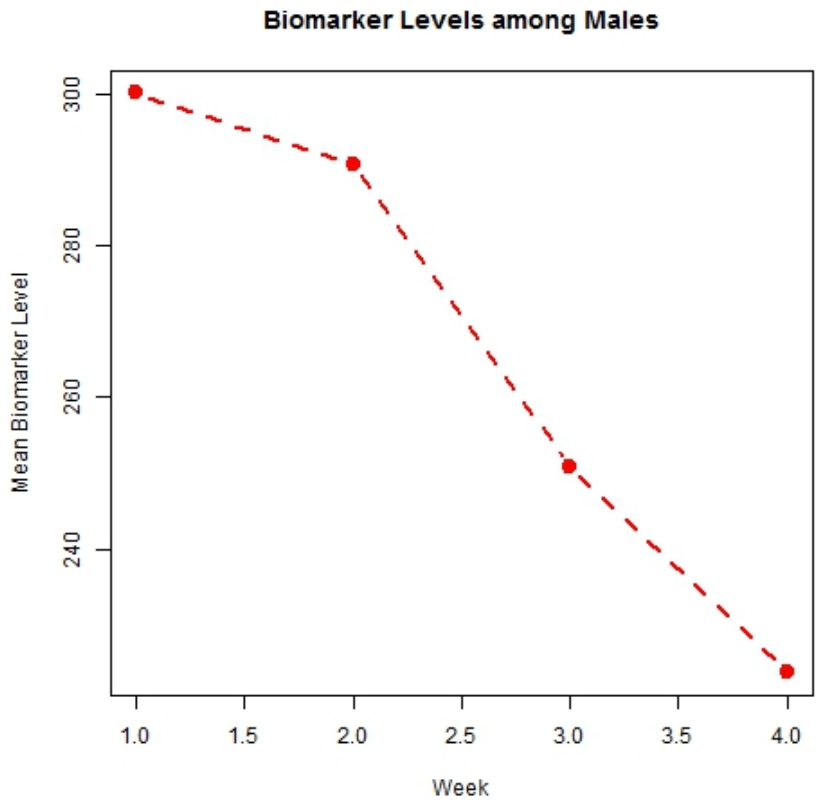
```
> StatsM
            Week1      Week2      Week3      Week4
Means  300.038033 290.710878 250.730195 223.735505
StdDev   5.595378   5.049375   8.107972   7.017781
```

c. Using the matrix created in part (b), plot the mean biomarker level among males across the follow-up period. That is, recreate the figure given below. Please submit R code below. (25 pnts)
   i. Note: For the purposes of this problem, ignore the fact that R automatically creates a finer grid of value for the Week axis (i.e. presents options like Week 1.5). Also, there is NO NEED to get the limits of the y-axis exactly the same – you only need to ensure that no plotting data is cut off.

R Code:

```
x <- 1:4            # Create vector for x-coordinate points (can skip)
plot(x,StatsM[1,],  # Pull off means to plot
     type='b',      # Plot both points and lines
     col='red',     # Color the points / lines red
     pch=19,        # Select shaded circle for plotting character
     cex=1.5,       # Increase size of plotting character
     lty=2,         # Select dashed line for plotting line
     lwd=2,         # Increase width of plotting line
     main='Biomarker Levels among Males',
     ylab='Mean biomarker level',
     xlab='Week')
```

R Output:



d. Using the figure created in part (c), the matrix created in part (b), and the segments() function in R, add vertical bars around each mean that represents ± one standard deviation for each week. That is, recreate the figure given below. Please submit R cope below (10 pnts)
    i. Note: For the purposes of this problem, ignore the fact that R automatically creates a finer grid of value for the Week axis (i.e. presents options like Week 1.5). Also, there is NO NEED to get the limits of the y-axis exactly the same – you only need to ensure that no plotting data is cut off.

R Code:

```
x <- 1:4              # Create vector x-coordinate points (can skip)
plot(x,StatsM[1,],    # Pull off means to plot
     type='b',        # Plot both points and lines
     col='red',       # Color the points / lines red
     pch=19,          # Select shaded circle for plotting character
     cex=1.5,         # Increase size of plotting character
     lty=2,           # Select dashed line for plotting line
     lwd=2,           # Increase width of plotting line
     ylim=c(210,310), # Set limits of y-xis so no data is cutoff
     main='Biomarker Levels among Males',
     ylab='Mean biomarker level',
     xlab='Week')
# - Add vertical bars around each plotted mean to represent 1SD
segments(1,StatsM[1,1]-StatsM[2,1],1,StatsM[1,1]+StatsM[2,1])
segments(2,StatsM[1,2]-StatsM[2,2],2,StatsM[1,2]+StatsM[2,2])
segments(3,StatsM[1,3]-StatsM[2,3],3,StatsM[1,3]+StatsM[2,3])
segments(4,StatsM[1,4]-StatsM[2,4],4,StatsM[1,4]+StatsM[2,4])
```

R Output: