Friday, September 9, 2022, 6PM, BbLearn (58 points scaled to 100)

You may discuss this assignment with whomever you wish, but please prepare and submit work in groups of ***ONE to THREE*** students, no more and no fewer. ***Each group*** will submit a copy of their group's completed assignment, via BbLearn, including the ***names and student ID numbers of all group members*** who participated on the assignment. If you discover a mistake, you may submit another version before the deadline. The last submitted (on-time) version will be graded, with all team members receiving the same score, which will be recorded in BbLearn.

GROUP MEMBERS WHO DO NOT CONTRIBUTE SUBSTANTIALLY TO AN ASSIGNMENT MAY BE REQUIRED TO WORK IN THEIR OWN GROUP OF ONE FOR THE REMAINDER OF THE SEMESTER.

While you are permitted to discuss the assignment with other groups, please prepare your own group's code/output and written answers. GROUPS WHOSE CODE AND SOLUTIONS APPEAR SUBSTANTIALLY SIMILAR MAY BE SUBJECT TO A 10% PENALTY.

Please prepare solutions in a ***neat, organized and concise fashion***! I prefer typeset presentations (e.g., cut and paste code/output into MS Word with added exposition when appropriate; knitr via EMACS and ESS; knitr or R Markdown via RStudio, the latter being the method preferred by students in recent years). At the very least, you need to ensure code and output are presented with a fixed-width font. Neatly handwritten presentations may also be appropriate for some problems. Sloppily prepared or disorganized solutions will not receive full credit.

To complete the items below, I expect you to find and use material in our lecture notes, including code/output, possibly after some modification. Remember, you may use the help functionality in `R`, as briefly introduced in Lecture 1 of our notes, or search online. Some questions may be answered with code and output alone, but some exposition may be required beyond code and output for other questions. It's up to you to communicate concisely!

1. Use the following vector to complete the following items **using `R` code/output** (perhaps with short exposition if necessary to communicate concisely). Be sure to use `set.seed` as indicated, or you may not get the same results.

```
> set.seed(24601)
> (myvec<- round(rnorm(n=30), digits=1))

 [1] -0.3  0.6  0.5 -1.9 -0.3 -0.6  0.3 -0.3 -1.2 -3.2 -0.6 -0.3 -1.0
[14] -0.2  1.3 -0.4  0.0 -0.7  0.0  0.7  1.0 -0.1 -1.0  0.6  0.5  0.8
[27] -0.7  0.7  0.2  0.4
```

   (a) Compute the sum of the elements of the vector. (1 point)

   (b) Compute the product of the elements of the vector. (1 point)

   (c) Compute the square of each element of the vector. (1 point)

(d) Compute the exponential (base e) of each element of the vector. (I may have skipped over this in my recorded lecture of Appendix A, but this material is in Appendix A.) (1 point)

(e) Compute the natural log of each element of the vector, when possible. Is there a problem? What is it? (1 point)

(f) Compute the log, base 10, of each element of the vector, when possible. (1 point)

(g) Compute the average, variance, standard deviation, median and the three quartiles of the of the elements in the vector. (2 points)

2. Use the `graphics::curve` function `R` to plot the standard normal pdf, $\phi(z)$, from $z = -3.5$ to $z = 3.5$. (The `graphics` package is loaded onto the search path by default. There is no need to use the package scoping operator, '::'.) Be sure to annotate axes in a reasonable manner and add a main title to the plot. (Note that `graphics::curve` accepts some `graphics::plot` function arguments; see `help(curve)` or `help(plot)`. (And see our note appendices for examples, of course.) (5 points)

3. Use `R` to compute $\Phi(2.576)$, where we use Φ to denote the standard normal cdf, i.e., $\Phi(2.576) = P(Z \leq 2.576)$, where Z is a standard normal random variable. This is `R`'s "z-table." (1 point)

4. Use `R`'s `qnorm` to compute $\Phi^{-1}(0.995)$. (1 point)

5. Use `R` to plot the binomial pmf with $n = 7$ and $p = .75$. Be sure to annotate plot axes in a reasonable manner and add a main title to the plot. (5 points)

6. Use `R` to compute $P(Y \geq 2)$ for $Y \sim binom(7, 0.75)$. Note that the "`p`" functions in `R` give $P(Y \leq y)$ by default. You may use the "`lower.tail=FALSE`" option to get $P(Y > y)$. Note the presence/absence of "=" in "$\leq$" and ">": it may make a difference for discrete random variables. You may want to look at `help(pbinom)`. (1 point)

7. Use `R` to compute $P(Y \geq 1.5)$ for $Y \sim binom(7, 0.75)$. (1 point)

8. Let Y_1 and Y_2 be independent random variables with means (expectations) μ_1 and μ_2 and variances σ_1^2 and σ_2^2.

   (a) What is the mean of $5 + Y_1 + 2Y_2$? (Use above symbols!) (1 point)

   (b) What is the covariance between Y_1 and Y_2? (1 point)

   (c) What is the variance of the above linear (affine really) combination? (2 points)

   (d) If Y_1 and Y_2 are normally distributed, what is the distribution of the above linear combination? (1 point)

9. Three matrices, **A**, **B** and **C**, are shown in the R output, below. Again, be sure to use `set.seed` as indicated in the sequence of statements to get the same results as shown.

```
> ## Recall that R operates in column major mode, filling matrices by
> ## columns, by default
> set.seed(5551212)
> (A<- matrix(round(rnorm(n=8,sd=3)),
+             nrow=2,ncol=4, byrow=TRUE))

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

> (B<- matrix(round(rnorm(n=4,sd=3)),
+             nrow=1,ncol=4))

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

> (C<- matrix(round(rnorm(n=8,sd=3)),
+             nrow=2,ncol=4, byrow=TRUE))

     [,1] [,2] [,3] [,4]
[1,]   -1    2    2    3
[2,]   -3    4    4    0
```

Complete the following items using R.

(a) Use R to extract the (2,2) element of matrix **A**. (1 point)

(b) Use **R** to extract the second column of **A**, while keeping the result as a matrix of appropriate size, i.e., do not let R change the result to an R vector that does not have the matrix (or array) class. (2 points)

(c) $\mathbf{A} + \mathbf{C}$ **and** give the size of the result. (2 points)

(d) $\mathbf{A} - \mathbf{C}$ and give the size of the result. (2 points)

(e) $\mathbf{AB}'$ and give the size of the result. (2 points)

(f) $\mathbf{AC}'$ and give the size of the result. (2 points)

(g) $\mathbf{C}'\mathbf{A}$ and give the size of the result. (2 points)

(h) What does `A*C` give? Show the result and explain very briefly. (2 points)

10. Two matrices, $\mathbf{X}$ and $\mathbf{Y}$, for a simple linear regression problem are shown in the R output, below.

```
> set.seed(90620 + 5150)
> X<- matrix(c(rep(1,6),
+              round(runif(n=6,min=0,max=10))), ncol=2)
> Y <- 4.5 + 1.5*X[,2,drop=FALSE] + round(rnorm(6),2)
> ord<- order(Y)
> (X<- X[ord,])

     [,1] [,2]
[1,]    1    2
[2,]    1    3
[3,]    1    6
[4,]    1    6
[5,]    1    6
[6,]    1    8

> (Y<- Y[ord,,drop=FALSE])

      [,1]
[1,]  6.74
[2,]  8.59
[3,] 12.94
[4,] 12.95
[5,] 13.95
[6,] 17.84
```

Complete the following items in R.

(a) $\mathbf{X}'\mathbf{Y}$ (1 point)

(b) $\mathbf{X}'\mathbf{X}$ (1 point)

(c) $(\mathbf{X}'\mathbf{X})^{-1}$ (1 point)

(d) $(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})$ (1 point)

(e) $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ (1 point)

(f) Which of the above matrix results in this problem are symmetric? This should be obvious for those of you who know matrix algebra, but, if not, we can use R to help us. (2 points)

(g) Use `lm` in R to perform the implied simple linear regression (SLR). Report the estimates of the regression model parameters. Note that we typically see the right

hand side of the formula in the call to `lm` refer to variable names in a data frame. You may create a data frame with the suggested response and covariate, and proceed accordingly as we've seen in our notes. Or, the `lm` formula will accept the matrices as is in a natural way, but you will want to tell `lm` not to try to include the column of 1's as $\mathbf{X}$ already has it. Use `+ 0` or `- 1` in the formula to prevent `lm` from attempting to add its own column of 1's. (Or, don't do this and see what happens, but you'll want to do it before the next item (or use the data frame approach as mentioned). (3 points)

(h) Compute/show $Var(\widehat{\boldsymbol{\beta}})$. See Result B.1 and assume $\sigma^2 = 0.3992$. Obtain the square-roots of the result's diagonal elements and compare these to the summary of the linear model object created previously. (You should have saved the object from the call the `lm`, previously, and use `summary(my.lm)` here, assuming `my.lm` is the name of your object from the call to `lm`.) A bit of context would be nice, but I'll accept blind computation/comparison for the time being. (5 points)

Bibliography