DUE: Wednesday, September 21, 11:59PM (52 points scaled to 100 percent)

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. (a) In the code, below, I use the `readRDS` function to read a data set (`data.frame`) of tree sap flow measurements into `R`. Use `R` to summarize the data frame, **and**, very briefly, give an interpretation of the output for variable `V2`. (See the `Details` section of `help(summary)`.) Your code/output should suffice. (2 points)

   ```
   > sapflow.df<-  readRDS(file="sapflow.rds")
   ```

   (b) Change the names of the variables (aka, list components) to `light`, `fertilizer`, `temperature`, `moisture` and `sapflow`, in that order, **and** verify that the names have been changed. No exposition required. (2 points)

   (c) The first four variables should be factor (aka classification or categorical) variables, each with 2 levels: 'lo' and 'hi' light levels, 'lo' and 'hi' temperature levels, etc. From the previous summary, we should see that the light variable is not considered by `R` to be a factor variable; numeric and factor variables are summarized differently. (Use `sapply(sapflow.df, data.class)` to verify, if you want.) If you issue the following command (output suppressed)

```
> table(sapflow.df$light)
```

you will see that light has two unique values, 1 and 2, eight cases of each. Change light to a factor variable and change the levels to 'lo' and 'hi', corresponding to the numeric values 1 and 2, respectively. Verify and report your changes. (The last variable, sapflow, is the flow rate of sap in a tree (mL/h) as measured by a specialized instrument, and this should be numeric.) No exposition required. (2 points)

(d) Use the `pairs` function to obtain a 'matrix' or 'grid' of 'y vs x' scatterplots for all pairs of variables. Report your code and plot and briefly say what the pairs functions does with factors. (See the Arguments section of `help(pairs)`.) (3 points)

(e) Consider a regression of sapflow on the remaining variables. (With all inputs/covariates being factors, this linear model often goes by the special name, 'ANOVA,' which we will get to more, later.) That is, you expect the mean sapflow to be a function of the other variables, which we denoted as $E(Y|\mathbf{x})$ in class/notes, where Y is the sapflow variable and $\mathbf{x}$ denotes the remaining factor variables. The assumption of normal errors translates to the assumption of normality for sapflow (output/respons), as we've discussed briefly in class/notes. A colleague suggests that you plot a histgram of sapflow to assess normality. Explain why a single histogram may not be a good way to assess normality of sapflow. (Histogram not required.) (5 points)

2. I use the `model.matrix` function to extract the $\mathbf{X}$ ('design' or 'regression') matrix for a linear model of the mean (or expected) `sapflow` as a function of the $k = 4$ factor inputs, `light, fertilizer, temperature` and `moisture`. Notice `R`'s default numerical coding of factors in $\mathbf{X}$: each factor is given an indicator vector of its 'hi' level, i.e., a 1 for 'hi', and 0 for 'lo' (along with the column of 1's). We will discuss more specialized methods for factor inputs later (analysis of variance (ANOVA)). For now, just consider the $\mathbf{X}$ matrix as if we had numerically coded it ourselves to indicate factor 'hi' levels, and we are ready to do regression on these so-coded numerical inputs.

   If we consider the column of 1's as 'observations' of 'input variable 0,' then we may denote the ith observed input as $\mathbf{x}_i = (x_{i0}, x_{i1}, x_{i2}, x_{i3}, x_{i4})'$, as we have done in our notes/class. For example, the first observed input vector, $\mathbf{x}_1 = (1, 0, 1, 0, 1)'$, the first row of $\mathbf{X}$, indicates low light, high fertilizer, low temperature and high moisture. (You may ignore the 'attributes' that follow the $\mathbf{X}$ matrix; we'll learn about those later.)

   I also put the observed outputs in a vector, $\mathbf{y}$, for later use, along with $\mathbf{X}$.

```
> ## Regression matrix (including column of 1's by default)
> (X<- model.matrix(sapflow ~ light + fertilizer + temperature +
+                           moisture,
+                     data=sapflow.df))

   (Intercept) light fertilizerhi temperaturehi moisturehi
1            1     1            1             0          1
2            1     2            1             0          1
3            1     1            0             0          1
4            1     2            0             0          1
5            1     1            1             1          1
6            1     2            1             1          1
7            1     1            0             1          1
8            1     2            0             1          1
9            1     1            1             0          0
10           1     2            1             0          0
11           1     1            0             0          0
12           1     2            0             0          0
13           1     1            1             1          0
14           1     2            1             1          0
15           1     1            0             1          0
16           1     2            0             1          0
attr(,"assign")
[1] 0 1 2 3 4
attr(,"contrasts")
attr(,"contrasts")$fertilizer
[1] "contr.treatment"

attr(,"contrasts")$temperature
[1] "contr.treatment"

attr(,"contrasts")$moisture
[1] "contr.treatment"

> ## Response/output vector:
> y<- sapflow.df$sapflow
```

(a) Using our typical linear model notation presented in our notes, write (or typeset) the linear model for the first observation, $E(Y_1 \,|\, \mathbf{x}_1)$, simplifying as much as possible given the values of the first observed input vector $\mathbf{x}_1$. (3 points)

(b) Repeat the previous item for observation $i = 5$ and subtract $E(Y_1 \,|\, \mathbf{x}_1)$ from

$E(Y_5 \,|\, \mathbf{x}_5)$ to get the difference of mean sapflow between 'hi' and 'lo' temperature, other inputs held the same. (Right?!) Report $E(Y_5 \,|\, \mathbf{x}_5)$ and the (very simple) difference of means. (3 points)

(c) Use the `lm` function to fit a linear model of the mean (or expected) `sapflow` as a function of the four factor inputs, `light, fertilizer, temperature` and `moisture`. What does our (fitted/estimated/learned) model tell us is the (estimated mean) difference in sapflow to expect when moving from the low temperature level to the high temperature level? In other words, what is the estimate of the difference requested in the previous item? Show your `lm` code and, briefly, any summary code/output to support your answer. (3 points)

(d) Is the so-called 'effect' of (high) temperature, in the previous item, statistically significant? More formally, as you should recall from previous statistics courses (and as discussed with the Galapagos example in class), perform a test of the hypotheses

$$\begin{aligned} H_0{:}\beta_3 &= 0 \\ H_1{:}\beta_3 &\neq 0 \end{aligned}$$

as outlined in the following steps. (We will discuss testing (inference) more formally in a subsequent chapter.)

i. Report the estimated value of β_3. (1 point)

ii. Report the (estimated) standard error of $\widehat{\beta}_3$. (1 point)

iii. Report the $t-$test statistic. (1 point)

iv. Report the degrees of freedom associated with the above statistic. (1 point)

v. Report the p-value. (1 point)

vi. Use the type I error rate, i.e., significance level, $\alpha = 0.05$, to answer the question of significance and report your conclusion in the context of the problem. This should be one or two sentences. (I'm probing your retention of previous statistics training. We'll learn about error rates and significance levels later.) (3 points)

(e) Fit the previous model using matrices and vectors, without the aid of the high level `lm` function. Report your code and resulting estimated regression model parameters. Are they the same as given by `lm` in a previous item? (3 points)

(f) Using matrix-vector computations, obtain the hat matrix $\boldsymbol{H}$ and its trace. Show your code and output. (3 points)

(g) Using your matrix-vector computations, obtain the fitted vector and compare it to that obtained from the `fitted` function applied to the object obtained from `lm` previously. Are the fitted values the same? (Note that (the generic code) `all(round(obj1 - obj2, 10) == 0)` will test if two vectors, `obj1` and `obj2` have the same element values, up to 10 decimals.) (3 points)

(h) Using your matrix-vector computations, obtain the residual vector and compare it to that obtained from `residuals` applied to the object obtained from `lm` previously. Are the residuals from these two approaches the same (up to, say, 10 decimal places)? (3 points)

(i) Using the residual vector, compute the estimate of the standard deviation of the errors, i.e., compute the square root of what we called MSE or what `R` calls residual standard error, i.e., compute $\widehat{\sigma}$! Is this the same as `R` reports in previous output? (3 points)

(j) Obtain the estimate of $Var(\widehat{\boldsymbol{\beta}})$ using `vcov` applied to the object obtained from `lm` previously and compute the same using matrices and vectors; are these the same (up to 10 decimal places)? **Also**, obtain $\widehat{se}(\widehat{\beta}_3)$ using these results; is this the same as you obtained from previous output? (3 points)

(k) Compute the coeficient of determination, R^2, using the empirical correlation between the fitted values and the observed outputs. In our notes, I gave the equation of R^2 as the square of the empirical correlation formula, but you may simply use the `cor` function (squared!) once you obtain the observed output vector and the fitted vector. Show your code and output. Does your computed value of R^2 agree with output from a previous item? (3 points)

(l) (Not to be submitted. I just want to illustrate something here (in my solutions to be posted later of course).) Use the `plot` function to create a scatterplot of the observed responses (vertical axis) vs. the fitted values (horizontal axis); be sure to label your axes appropriately. Add the 1-1 line to your plot using `abline`. Report your code and output. (no points)

(m) (Not to be submitted. I just want to illustrate something here (in my solutions to be posted later of course).) Regress the observed values against the fitted values and create a summary. Do you see a correspondence in the summary output with the correlation or its square, computed previously? Report your code and output. (no points)