**Class Project**

*Note: This is a fluid document so this example might look different then the Class Project Rubric linked in the course – however, the functionality remains the same.*

The Class Project is an ongoing assignment you will complete as you work your way through the class. It is the only submission you are evaluated on. The goal of the Class Project is to apply the skills you learn to your own data (or data we supply to you if you do not have your own data).

This document contains two tables that you will submit at the end of class along with your project script:

1. Project Requirements
   1. All the standards in this table must be met in order to pass the class.
   2. The unshaded cells in the ***How Used*** columns need to be filled out with a brief (2-3 sentences) description of how you applied the requirements in your project.
2. Skills List
   1. You need to earn 75% of the points in the Skills List to pass the class.
   2. In the ***How Used*** column, give a brief (1-2 sentence) description of how you used this skill in your project.
   3. Put the comment ***# SKILL XX*** next to the place in your code where you met the skill – ***XX*** is the skill number.
   4. Adjust the points in ***Your Points*** column.
   5. You can adjust the ***Total Points*** (bottom cell in ***Your Points*** column) by **highlighting the cell** and clicking F9 – this will automatically add all the above cells and put the results in the bottom cell.
   6. Note: You can use the same line or lines of your code to satisfy multiple skills.

Class Project submissions are sent to the instructor's email ([belinsky@msu.edu](mailto:belinsky@msu.edu)). The submission needs to include:

1. This document with the tables filled out
2. All R script files used by your project
3. All data files used by your project
4. A list of R packages you used (if any)

|  |  |  |
| --- | --- | --- |
| **Project Requirements** | Lesson | How Used |
| 1. Script is well commented | 1.1 |  |
| 1. Script lines do not go more than 80 characters with exception of:   1) long file names (these cannot be broken down)  2) SKILL comments can extend beyond the 80 characters | 1.6 |  |
| 1. Use semicolons ( ; ) to end commands throughout your scripts. | 1.1 |  |
| 1. Consistent alignment of script inside of curly bracket ( **{ }** ) | 1.7 |  |
| 1. Three uses of if() statements | 1.7 |  |
| 1. Three uses of for() loops | 2.4 |  |
| 1. Two modified plots – data along with some manipulations (colors, legend, points, lines, etc.) | 2.7 |  |
| 1. Three uses of subsetted variables | 2.2 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skills List** | Lesson | Max  Points | Your Points | How Used |
| 1. Use of a standard mathematical operations (+, -, \*, /). Half-point per use/ max of 1 point. | 1.3 | 1 | 0 |  |
| 1. Using parentheses to order mathematical operations | 1.3 | 1 | 0 |  |
| 1. Using powers or roots | 1.3 | 1 | 0 | Length of shell raise to the power of b |
| 1. Get Input from user – save to a variable | 1.5 | 1 | 0 |  |
| 1. Output mixed message (variables and text) to Console Window | 1.6 | 1 | 0 |  |
| 1. Use of newline character | 1.6 | 1 | 0 |  |
| 1. Use of different conditional operators. Half point for each unique operator (6 in all) | 1.7 | 3 | 0 |  |
| 1. Use of if-else structure | 1.8 | 1 | 0 |  |
| 1. Use of if-else-if structure | 1.8 | 1 | 0 |  |
| 1. Error condition in an if-else structure | 1.8 | 1 | 0 |  |
| 1. Use of && operator | 1.9 | 1 | 0 | Combining conditions for shell height (SH) and meat weight (MW) |
| 1. Use of || operator | 1.9 | 1 | 0 |  |
| 1. Using && or || to check conditions on multiple variables | 1.10 | 1 | 0 |  |
| 1. Save data from CSV file to a data frame | 2.1 | 1 | 0 | Save the data from the scaldat.csv file to the data frame sdat. |
| 1. Subset one datum in a data frame – save to variable | 2.2 | 1 | 0 |  |
| 1. Subset column in a data frame – save to vector | 2.2 | 1 | 0 |  |
| 1. Subset individual value in vector | 2.2 | 1 | 0 |  |
| 1. Use of sequence | 2.3 | 1 | 0 | Sequencing through the rows of the data using a for() |
| 1. Use of sequence that increases or decreases by number other than 1 | 2.3 | 1 | 0 |  |
| 1. Sequence to set up iterations in a for() | 2.4 | 1 | 0 | Sequencing through the rows of the data using a for() |
| 1. Three uses of state variable. One point / use | 2.4 | 3 | 0 |  |
| 1. Nested if() inside a for() | 2.4 | 1 | 0 | If() checks for outliers as the for() iterates through the vector |
| 1. Use of Boolean values. | 2.4 | 1 | 0 |  |
| 1. Find max, min, or average value of a vector using for() | 2.5 | 2 | 0 |  |
| 1. Ignoring NA values | 2.5 | 1 | 0 |  |
| 1. Add vector to data frame | 2.6 | 1 | 0 |  |
| 1. Rearrange columns in a data frame | 2.6 | 1 | 0 |  |
| 1. Write a data frame to a CSV file | 2.6 | 1 | 0 |  |
| 1. X-axis, y-axis labels, and title on a plot | 2.7 | 1 | 0 | Added title and labels for the axis to the boxplot. |
| 1. Adding a legend to a plot | 2.7 | 1 | 0 |  |
| 1. Adding colors to a plot | 2.7 | 1 | 0 |  |
| 1. Adding lines or points to a plot | 2.9 | 1 | 0 |  |
| 1. Histogram | 2.7/2.9 | 1 | 0 | Histogram of residuals from the linear models |
| 1. Barplot | 2.7/2.9 | 1 | 0 |  |
| 1. Multi-panel scatterplot using pairs() | 2.8/2.10 | 1 | 0 |  |
| 1. Create and use two of your own functions (2 point/each) | 2.8 | 4 | 0 | Using MWhat function to make predications |
| 1. Create a function with at least 2 parameters | 2.8 | 2 | 0 | a, SH, and b are parameters of MWhat |
| 1. Save the return value from your function to a variable | 2.8 | 1 | 0 |  |
| 1. Create a function with default parameters | 2.10 | 1 | 0 |  |
| 1. Use two functions from another package (1 point/each) | 3.1 | 2 | 0 |  |
| 1. Reshape a data frame | 3.2 | 2 | 0 |  |
| 1. Perform operation on multiple columns of a data frame or matrix | 3.2 | 2 | 0 |  |
| 1. Using substring function | 3.2 | 1 | 0 |  |
| 1. Rounding values | 3.2 | 1 | 0 |  |
| 1. Renaming columns | 3.2 | 1 | 0 |  |
| 1. Create a matrix | 3.3 | 1 | 0 |  |
| 1. Perform operation on rows, columns, and whole matrix | 3.3 | 1 | 0 |  |
| 1. Repeat values using the rep() function | 3.4 | 2 | 0 |  |
| 1. Perform up to two ANOVAs (1 point each), comment on results | 3.4 | 2 | 0 |  |
| 1. Perform up to two t-tests (1 point each), comment on results | 3.4 | 2 | 0 |  |
| 1. Create a boxplots that uses multiple categories | 3.4 | 2 | 0 | Plots shell heights from both the EAST and WEST |
| 1. Randomly sample data | 3.5 | 1 | 0 |  |
| 1. Create pseudo-random values using set.seed() | 3.5 | 1 | 0 |  |
| 1. Sample from a normal distribution | 3.5 | 1 | 0 |  |
| 1. Subset a list | 3.5 | 1 | 0 |  |
| 1. Perform two linear regressions (1 point each), comment on the results | 3.6 | 2 | 0 |  |
| 1. Add regression line to a plot | 3.6 | 1 | 0 | Prediction line for shell height vs meat weight added to plot. |
| 1. Up to four examples of subsetting a vector using which(). (half point each) | 3.7 | 2 | 0 |  |
| 1. Use grep() on a vector | 3.7 | 2 | 0 |  |
| 1. Up to three examples of using subset vector to index another vector (1 point each) | 3.7 | 3 | 0 |  |
| 1. Multiple condition on a subset vector | 3.8 | 2 | 0 |  |
| 1. Multiple conditions in a grep() | 3.8 | 1 | 0 |  |
| 1. Use of union() or intersect() | 3.8 | 1 | 0 |  |
| 1. Up to two examples of plotting subset vectors (1 point each) | 3.7/3.8 | 1 | 0 |  |
| 1. Save results as an .rdata file | 3.5?? | 2 | 0 |  |
| **Total Points** |  | **88** | **0** |  |