Important Note on what Category C Means

It has become clear to me that several of you are not aware of what a Category C assignment means. In particular, it means that you may not copy another student's code, nor may you allow another student to copy your code. This includes both copy/paste and visually copying another student's code. You also may not divide the work up with others, and you cannot provide the R code for any part of a question to another student, digitally, verbally, or visually.

Here's what you are allowed to do that you cannot do with a Category A assignment: You may work with others on the assignment, verbally sharing overall approaches like whether to use training and test sets, how to index through a vector during a loop, the name of a function or package that might be used, or what parameters might be appropriate in a particular case. In this Assignment, for example, there are several questions that require you to comment on what you see. The answers to these may be discussed freely, but each student is expected to respond in their own words.

There were several instances in Assignment 1 where these rules were obviously not followed. I am assuming that this was due to a lack of precision on my part about my expectations. However, having now made my expectations explicit, I will hereafter assign an assignment grade of 0 on a first offense, and refer the matter to the Honor Council on a second offense.

If you have questions about what is/is not allowed, please ask the question in class. I will be happy to clarify these rules.

David M.

Assignment 2: Linear Regression

BUAD 5072 – Fall 2018

---

1. Objectives

The purpose of this assignment is to provide you with some experience working with the lm() function and several of its supporting and extractor functions.

2. What You Will Need

- Access to a Windows computer with R

3. What You Will Hand In

Submit your script file as Assignment2.R via Blackboard - Assignment 2.

4. Due Date

Monday November 26th, just before midnight.

5. Note on Collaboration

This is a Category C assignment. Specifically, you may work with others or receive help from the instructor on this assignment. You must, however, turn in your own paper. You may not divide the work with others or copy another student's work. **It would be an honor code offense to do so**.

6. Preliminaries:

To get set up for the assignment, follow these steps:

1. As the first statement in your script file, enter rm(list=ls())
2. Each part of each question in the assignment should begin with the following three comment lines, where *n* is the question number:

   ##########################
   #### QUESTION *n* Part *m* ####
   ##########################

   Note: This is a new requirement

3. I should be able to run your script on my computer without errors or interruptions. For this to happen, you must:
   a. Avoid entering file path information…my files will be located in a different location that yours, and so your code will fail on my machine. Instead, always refer only to files in your working directory.
   b. Do not use functions like file.choose(), fix(),edit(), or q()
4. Do not create console output other than what is asked of you explicitly. For example, in your final script, remove any statements that you used to verify the contents or structure of data.

7. Assignment Tasks:

Problem 1: Simple Linear Regression (20%)

In this problem you will create several simulated data sets and will fit simple linear regression models to them.

a) Set the random seed to 5072.
b) Using the rnorm() function, create a vector named x, containing 100 observations drawn from a $N(0, 1)$ distribution. This represents a feature space (or set of predictors), X.
c) Using the rnorm() function, create a vector named eps, containing 100 observations drawn from a $N(0,0.25)$ distribution that is, a normal distribution with mean zero and **variance** 0.25.
d) Using x and eps, generate a vector named y according to the formula

$$y = -1 + 0.5x + \text{eps}$$

representing some linear relationship between y and x in a population with irreducible error equal to var(eps)
   a. Don't overthink this – it just requires some very simple vector arithmetic.
e) Display the length of y.
f) On a single comment line, indicate what the population parameter values of β_0 and β_1 are in this linear model.
g) Create a plot displaying the relationship between x and y, with x on the x-axis.

h) Again on a single comment line, comment briefly on the type of relationship you observe (positive or negative), the degree of linearity what you observe, and the amount of variability you observe.
i) Fit a least squares linear model to predict *y* using *x*.
j) Answer these questions, each on a single comment line:
    a. What are $\hat{\beta}_0$ and $\hat{\beta}_1$ (recall the coef() extractor function)?
    b. How do they compare with β_0 and β_1?
k) Display the least squares line (in black) on the plot obtained above.
l) Display the population regression line (in red) on the plot.
m) Use the legend() command to create an appropriate legend.
n) Fit a polynomial regression model that predicts *y* using x and x^2.
o) Comment on whether or not there is there evidence that the quadratic term improves the model fit, and briefly explain your answer
    a. Recall that the anova() function (pp. 116 of the text) can be used to evaluate whether or not one model is superior to another.
p) Repeat **c**)–m) after modifying the data generation process in such a way that there is *less* noise in the data. Do this by changing the variance of the model ε in d) to 0.1. Otherwise, the model should remain the same.
q) Repeat **c**)–m) after modifying the data generation process in such a way that there is *more* noise in the data. Do this by changing the variance of the model ε in d) to 0.5. Otherwise, the model should remain the same.
r) On a comment line, contrast the closeness of the fit to the population regression line among all three levels of population variance ε.
s) Display the 95% confidence intervals for β_0 and β_1 based on the original data set, the noisier data set, and the less noisy data set.
    a. Recall the confint() extractor function.
t) Comment on the reason why the widths of the confidence intervals are as observed.

Problem 2: Collinearity (40%)

This problem focuses on collinearity.

a) Execute the following commands in R:

```
set .seed (5072)
x1 = runif (100)
x2 = 0.5 *  x1 + rnorm (100) /10
y = 2 + 2* x1 + 0.3* x2 + rnorm (100)
```

b) The last line corresponds to creating a linear model in which y is a function of x1 and x2. On a single comment line, print (using the print() function) what the population parameter values of β_0, β_1 and β_2 are in this linear model.
c) Display the Pearson correlation coefficients of y, x1 and x2.
    a. Recall the cor() function.
d) Using a single R statement, create scatterplots displaying the relationship between y, x1 and x2 (recall the pairs() function)
e) Use a print() function to comment on the correlations among these variables,
f) Using this data, fit a least squares regression model called lm.fit.both to predict y using x1 and x2.
g) Display the values of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$.
h) Use a print() function to comment on the statistical significance of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$.

i) Can you reject the null hypothesis H_0: $\beta_1 = 0$? How about the null hypothesis H_0: $\beta_2 = 0$? Use a print() function to display your answer and how you arrived at this conclusion.
j) Fit a least squares regression model called lm.fit.justx1 to predict y using only x1.
k) Use a print() function to comment on your results. Can you reject the null hypothesis H_0: $\beta_1 = 0$? Explain how you arrived at this conclusion.
l) Fit a least squares regression model called lm.fit.justx2 to predict y using only x2.
m) Can you reject the null hypothesis H_0: $\beta_2 = 0$? Use a print() function to display your answer and how you arrived at this conclusion.
n) Do the results obtained in j)-m) contradict the results obtained in f)–i)? Again, explain your answer using a print() function.
o) Now suppose we obtain one additional observation, which was unfortunately mis-measured. Add this point to our x1, x2 and y values using the following R statements:

```
x1=c(x1 , 0.1)
x2=c(x2 , 0.8)
y=c(y,6)
```

p) Re-fit the linear models from f)–m) using this new data.
q) Use a print() function to comment on the effects this new observation has on the each of the models.
r) In each model, is this new observation (point 101) an outlier? A high-leverage point? Both? Neither? Explain your answers using a print() function.
    a. Recall that when you use the plot() function to plot the model itself (e.g. plot(lm.fit.both), four diagnostic plots are produced (use par(mfrow=c(2,2)) to plot all four in the same graphics window). The Scale Location plot can be used to identify outliers and the Residuals vs Leverage plot can be used to identify high-leverage points (Cook's Distance greater than 1). Remember to reset the graphics window to 1x1 after plotting all three models.

<u>Problem 3</u>: Simple vs Multiple Regression, Practice with Extraction Functions (40%)

This problem involves the Boston data set, which we saw in the lab for this chapter. It is in the MASS package. We will now try to predict per capita crime rate using the other variables in this data set.

a) Set the seed to 5072. For each predictor, fit a simple linear regression model to predict the response. For each of these models, save the following data:
    - the name of the predictor
    - The F-statistic for the fit
        i. Recall that the summary(your_model_name) function produces a named item $fstatistic whose first item is the F-statistic for the fit (the other items are degrees of freedom in the numerator and denominator)
    - The *p*-value for this F-statistic
        i. Recall that the anova(your_model_name) function produces a named item $'Pr(>F)' containing this *p*-value.
    - The value of $\hat{\beta}$.
        i. Recall the coef(your_model_name) function.

    Display these values in a table (one row for each predictor).
b) In which of the models is there a statistically significant association between the predictor and the response at an $\alpha = 0.05$ level?

c) Change the graphics window to a 4x3 grid, then for each significant predictor, plot the x-values on the x-axis and the y values on the y-axis and line produced by the least-squares linear model. Label each chart with the name of the predictor. Return the graphics window to a 1x1 state.

d) Fit a multiple regression model to predict the response using **all** of the predictors.

e) Create a statement in R to display only those predictors which are significant at a level of $\alpha = 0.05$.
   - The *p*-values of the predictors are stored in the output of the summary(your_model_name) function. When you use the coef() function on the summary function's output, it produces a matrix with the names of the coefficients as row names – one of the column names is Pr(>|t|).

f) Compare your results from (a) to your results from (d) as follows:
   - Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (d) on the y-axis.
     i. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.
   - Use this plot to comment on the level of agreement between the simple and multiple regression approaches.
   - Which approach produces the most accurate reflection of the population parameters? Why?

g) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a polynomial model of the form

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

and perform an anova on this model versus the simple regression model without the polynomial terms. For each anova, save the F-statistic and the *p*-value associated with it. Then display the following table, sorted in ascending order of p-value:

predictor	fstat	pvalueofFstat
medv	116.6340058	2.504778e-42
dis	46.4603654	3.071837e-19
nox	42.7581707	7.122383e-18
indus	31.9869602	8.408754e-14
age	15.1400633	4.125056e-07
tax	11.6400227	1.144238e-05
ptratio	8.4155300	2.541647e-04
rm	5.3088168	5.229427e-03
zn	4.8118205	8.511995e-03
rad	3.6732699	2.607832e-02
lstat	3.3190437	3.698322e-02
black	0.4622222	6.301501e-01

For which of these can we reject the null hypothesis that there is no difference between the fit of the two models at an $\alpha = 0.05$ level?