**Introduction to Regression Analysis in R**

Ladies and gentlemen, the story you are about to see is true. The names have been changed to protect the innocent.

This is the city, Easton, Pennsylvania. I work here. I’m a statistician.

-------------------

*Um, Jeff?*

*Yes, Trent?*

*No one is going to get the reference. Just get on with the computer lab.*

*….. Okay…..*

--------------------

It was Thursday. It was cloudy in Easton. We were working the fall semester out of the mathematics department. The chair is Gary Gordon. My partner’s Trent Gaugler. My name’s Liebner.

-----------------

*Um, Jeff?*

*Yes, Trent?*

*Stop it already.*

*….. Okay…..*

--------------------

This week we will be examining the grades.txt file that we worked with in the introductory lab. This data is real data from one of Professor Liebner’s previous Math 186 classes, but of course the names have been changed to protect the innocent (and the guilty). Recall that the file contains the first three test scores and final exam grade of 25 students in the class. Our goal will be to determine the relationship between the individual test scores and the final exam grade. We will see if we are able to predict the final exam grade based on our knowledge of how a student performed earlier in the semester.

To start, download the file grades.txt from Moodle to your computer. Change the active directory to wherever you saved the file. Then, use the command:

grades=read.table("grades.txt", header = T)

to import the data into R. For simplicity, now use the attach command to make the column variables into variable names:

attach(grades)

This will create 5 variables: Name, Test1, Test2, Test3, and Final.

**Exploratory Data Analysis**

Before we start worrying about fitting any regression lines, the first step is to always explore our data. There’s always a temptation to fit a regression line immediately, but you should examine your data to anticipate any issues that may arise.

While other plots and summary statistics would be appropriate in this case, let’s restrict our exploratory data analysis to scatterplots for now. Our goal is to predict the final exam score using the earlier tests. Thus, it is natural that Final should be our response variable and the three variables Test1, Test2, and Test3 should be our explanatory variables. I will guide you through how to use R to examine the relationship between Test1 and Final. You’ll be asked to do the others on your own.

To create our scatterplot, recall that we can simply use the **plot** command.

plot(Test1, Final)

This will give us a general idea about the relationship between the two variables.

**Your turn:**

1. Describe the scatterplot. What is the relationship between the variables Test1 and Final?

2. Guess the value of the correlation between Test1 and Final.

3. Use the command **cor**(Test1, Final) to determine the actual correlation. How close was your guess?

4. Create a scatterplot for the variables Test2 and Final and find the correlation between the two variables. Summarize your results.

5. Create a scatterplot for the variables Test3 and Final and find the correlation between the two variables. Summarize your results.

**Finding the Regression Line with lm**

Once we have decided that we do wish to find the line of best fit for our two variables, we need to ask R to perform the calculation. Fortunately, there is a simple command in R to find regression lines, namely **lm** (which stands for **l**inear **m**odel). The general notation to use lm is:

lm(dependent variable ~ independent variable)

Thus, to find the line of best fit where we explain the Final exam grades using the scores from Test1 would be:

lm(Final~Test1)

Generally, I like to save the linear model results as a variable so that I can return to them.

model = lm(Final~Test1) # You can call your linear model anything you want.

If I now type in the name of my model (which I cleverly called “model”) I get the result:

> model

Call:

lm(formula = Final ~ Test1)

Coefficients:

(Intercept) Test1

39.1171 0.4326

You can see that R has returned the intercept and slope (given below the independent variable name Test1) of my regression line.

I can now add my fitted regression line to my scatterplot by using the command **abline**. All I need to tell R are the values of my intercept (the **a** value) and slope (the **b** value) and it plots a **line**. (That’s at least how I remember the name of the command.)

plot(Test1, Final)

abline(39.1171, 0.4326)

**Your turn:**

1. Interpret the values of the slope and intercept in the context of the problem.

2. How well does the line describe your data?

**Digging Deeper – What else does lm give you?**

If all that R gave you was the intercept and slope of your regression line, that wouldn’t be anything special. (By the way, it’s pretty much all that Excel gives you.) Fortunately, there’s a lot more hidden beneath the surface. Use the **summary** command to access this hidden information.

> summary(model)

Call:

lm(formula = Final ~ Test1)

Residuals:

Min 1Q Median 3Q Max

-26.351 -6.245 3.677 9.322 18.053

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) **39.1171** 38.7331 1.01 0.323

Test1 **0.4326** 0.4122 1.05 0.305

Residual standard error: 12.92 on 23 degrees of freedom

**Multiple R-squared: 0.04571**, Adjusted R-squared: 0.004216

F-statistic: 1.102 on 1 and 23 DF, p-value: 0.3048

*Okaaaaaaaaay.* That looks a bit daunting at first. However, all of this information is very important. Most of it we won’t cover until later in this class, if at all. Some of you may encounter this summary screen (or a similar one) in detail in a class such as econometrics. For now, let’s focus on the most important values, which I’ve bolded for you.

First, note that the intercept and slope are present in this summary in the “Estimate” column. There’s a lot of (very important) information to the right of these values, but at present we can see that it’s another way to obtain these numbers for our line. The second important piece of information is the “Multiple R-squared” value. This is R2, the important value which shows us how well our line describes our data.

**Your turn:**

1. What is the value of R2? Interpret what this value means in the context of the problem.

2. What does the value of R2 tell us about how well Test1 predicts the Final exam grade? Does this agree with your earlier observations?

Now that I’ve guided you through using lm to find the regression line, it’s your turn to repeat the analysis for the other variables. *Note: use different names for your different models.*

**Your turn:**

1. Find the best fitting line that explains the Final exam grade as a function of the second test. How well does this line explain the relationship? (Use the value of R2 to support your argument.)

2. Find the best fitting line that explains the Final exam grade as a function of the third test. How well does this line explain the relationship?

3. Examine Alice’s three test scores. Use your fitted regression lines to predict her final exam grade using:

a) Her first test score

b) Her second test score

c) Her third test score

Compare the predictive results.

4. **Personal thought challenge!** Based on the results from your regression lines and the knowledge that these are actual scores, what are the implications for you personally this year in Math 186?

**Residual analysis in R**

Recall that residual analysis is a useful tool to determine if there are any issues or problems with the fit of our regression line. The great thing is: R knows this, and it’s already calculated them for you! In addition to the summary data, there are more things hidden in your linear model. If you use the **names** command, you can find what these are:

> names(model)

[1] "coefficients" "residuals" "effects" "rank" "fitted.values"

[6] "assign" "qr" "df.residual" "xlevels" "call"

[11] "terms" "model"

Most of these we won’t worry about, but there are two important ones: residuals and fitted.values, which are exactly what you expect them to be. How do we access this information? Everything here is saved as a list in R. To access a specific part of the list, we will use the dollar symbol. Thus, to access the residuals, we type **model$residuals**.

To create a residual plot, recall that we plot the residuals on the y-axis and either the original independent variable of the fitted values on the x-axis. Thus, to form our plot, we would type:

plot(Test1, model$residuals)

*Note: There’s a neat little trick you can use to avoid excessive typing. When accessing parts of a list in R, as long as you type in enough letters to indicate what part of the list you’re accessing, you don’t have to type in the whole name! Thus, instead of typing* model$residuals *you could type in something like* model$res *and R will still know that you want the residuals!*

**Your turn:**

Create residual plots for your three regression lines. Do you notice any issues with your fitted regression lines? If so, identify the problems.

**Quick and dirty residual plots in R**

There’s a really quick way to create residual plots for your regression line in R. If you’ve saved your regression line (like I’ve named mine “model”), simply type **plot(model)**. This command looks odd at first. How can I plot a fitted regression line? However, in this case, the plot command allows us access to four diagnostic plots. To toggle through these plots, simply hit “enter” or click on the graphics window.

The first is the classic residual plot, with the x-axis being the fitted values of the regression line and the y-axis being the residuals. The second is what is known as a Q-Q plot. Theoretically, the residuals should have a normal distribution. If this is true, the values will fall along the line. The third plot is a variation on the classic residual plot. Note that the y-axis is essentially the square root of the absolute value of the residuals. The last plot is a bit confusing. It determines the leverage of each data point – how much each data point influences the fit of the regression line. Points with high leverage are far from the other points, nearing the red dashed lines in the upper and lower right corners of the plot.

**Your turn:**

1. Explain what problem each of the four plots identify. What do you believe is the purpose of the red line in the first and third plots?

2. Use the built in residual plots to analyze your three regression lines. Do you detect any problems in your fits?

**Bonus challenge! Introduction to Multiple Regression**

Professor Liebner typically puts 10 questions on the final exam. 3 come from the first test, 3 come from the second test, 3 come from the third test, and 1 comes from material after the third test. As a result, one might expect that someone’s final exam grade could be predicted by the following model:



**Your turn:**

Based on what you observed in your earlier analysis, do you think the above equation would be an accurate model to predict the final exam grade?

Fortunately, R has a way to use multiple variables to predict a dependent variable. You simply add the additional predictor variables into the lm command. I’ll save it as “multimodel.”

multimodel = lm(Final ~ Test1 + Test2 + Test3)

**Your turn:**

1. What is the equation for the regression line that uses all three test scores to predict the final exam grade?

2. Are you surprised at all by the results? Explain. (Note that the first test covers numerical and graphical summary statistics, the second test covers probability, and the third test covers confidence intervals and hypothesis tests, which may or may not mean anything to you at this time.)