

# AMS 162: Lab 4

## Simple Linear Regression

**Objective:** To introduce simple linear regression in R

Note that there is a text file of all the R commands on the class web page, so that if you wish, you can cut and paste those in instead of retyping them.

### Linear Regression

Here we will practice with simple linear regression in R (which your book sometimes refers to as “least-squares”). We will use a built-in dataset, from F. J. Anscombe, “Graphs in Statistical Analysis,” *The American Statistician*, 27: 17-21(1973). To access the data, enter

```
> data(anscombe)
```

You can now access this dataset, and can look at the data by typing the name of the data object:

```
> anscombe
```

Listing out the file, you should see that the data are in a table (technically it is a `data.frame`). You can see that there are four pairs of x-y variables, labeled x1-x4 and y1-y4. Recall that in order to be able to use these variable names, we need to tell R to look in this particular `data.frame`, and we do this with the `attach` command.

```
> attach(anscombe)
```

Now we are ready to fit some linear regressions. When you do a regression in R, R puts the result in something called a linear model object. For you to find out about the regression, you ask the object. The `lm` command stands for “linear model” and will do the regression. The format of the command is `lm( y ~ x )` where `y` is the response variable, `x` is the explanatory variable, and they are connected by a tilde (which stands for “is modelled as”).

```
> mod1=lm(y1 ~ x1)
```

I named the linear model object “mod1” for model 1, although you could name it whatever you like. Now we can get the results from this object.

```
> summary(mod1)
```

The command `summary` gives us the important summary information of the linear regression. It gives back the equation, information on the residuals, information on the coefficients, information on the fit, and the correlation of the coefficients. For now, we are primarily interested in the coefficients table and the R-Squared. How good is this fit? Note that R-Squared gives the percent of the variability in Y that is explained by X. The closer the R-Squared is to 1, the better the fit. In this case, it is 0.6665, which is not bad.

Is the x variable significant? We judge this by looking at the  $t$  statistic for the slope coefficient. In this case the estimated value of the slope coefficient is  $b = 0.5$  with standard error  $s_b = 0.118$ ; it has a  $t$ -statistic of 4.24 (with 9 degrees of freedom) which has a  $p$ -value of 0.00217, which R has flagged with `**` meaning “quite significant”. The regression equation is  $y_1 = 0.5 \cdot x_1 + 3$ . The standard error of the estimate (what your text calls  $s$ , the standard deviation of the errors in the individual responses) is 1.237. Now take a look at the second pair of variables:

```
> mod2=lm(y2 ~ x2)
> summary(mod2)
```

The regression results look pretty similar, don't they? Compare these with the last two pairs of variables.

```
> mod3=lm(y3 ~ x3)
> summary(mod3)
> mod4=lm(y4 ~ x4)
> summary(mod4)
```

So what's going on here? The variables are different, but the regressions seem to be the same. What we should have done at the beginning was to look at a scatterplot of the data. (For this particular data set, it would have ruined the fun.) So let's take a look now. (If necessary, first open a graphical device with `x11()`.)

```
> par(mfrow=c(2,2))      #make four plots on the screen
> plot(x1,y1)
> plot(x2,y2)
> plot(x3,y3)
> plot(x4,y4)
```

These four datasets all have the same regression line, but they are very different datasets. The first one looks like standard data for a linear regression. The second is a quadratic. The third is a perfect fit with an outlier. The fourth is almost a vertical

line. A linear regression should really only be run on the first set. The important lesson here is that you should always look at your data before you plunge into an analysis.

We can also see that something is wrong by looking at residual plots (the residuals are defined as the differences between the actual  $y$  values and the fitted values). The command `resid` will extract the residuals from a regression object. In general, we plot the residuals against either the  $x$  values or the fitted values to check the assumptions of the model, particularly linearity, normality, and equivariance. A good residual plot has no trends (linear or curvilinear) and is evenly scattered around zero. If there is a trend, or if the scatter is not uniform along the  $x$ -axis (for example, the points are tightly clustered around zero for small  $x$  values, but are much more variable for large  $x$  values), then you should probably do a transformation of the data to rectify the problem.

```
> plot(x1,resid(mod1))
> plot(x2,resid(mod2))
> plot(x3,resid(mod3))
> plot(x4,resid(mod4))
```

Only the first plot has good-looking residuals. The other residual plots tell you that you shouldn't be doing a linear regression on that data. For the first data set, we should go further and check that the residuals are approximately normally distributed.

```
> par(mfrow=c(1,1))      #return to one plot per screen
> qqnorm(resid(mod1))
```

We may have some concerns about the normality assumption because the data aren't really along a straight line.