Psyc 2111, Jones

Summary of Lab Week 9

Correlation

Load the data in lab9-1.txt on the course website. This dataset contains two variables, `X` and `Y`, that were measured for a single set of subjects. The fact that they're from the same set of subjects means `X[1]` goes with `Y[1]`, etc. Notice that there's a positive relationship between the two variables, i.e. the larger scores on one variable are paired with the larger scores on the other.

Make a scatterplot of the data, using the `plot()` function. Pick out a subject (by having R display all the data) and find that person's point on the plot. Its horizontal position will be that subject's `X` score and its vertical position will be that subject's `Y` score. Also, look at the pattern formed by all the points. The points have an upward slope to them, reflecting the positive relationship between the variables.

```
> plot(X,Y)
```

We can measure how strongly `X` and `Y` are related by computing their correlation. A correlation near 1 means a strong positive relationship. A correlation near -1 means a strong negative relationship. A correlation near 0 means a weak or absent relationship.

The first step in computing the correlation is to convert both sets of scores to z-scores. Make sure you use the mean and standard deviation of `X` for `zX` and the mean and standard deviation of `Y` for `zY`.

```
> zX = (X-MX)/sX
> zY = (Y-MY)/sY
```

The correlation is the average of `zX*zY`. If `X` and `Y` have a positive relationship, then large values of `X` will go with large values of `Y`, and small values of `X` will go with small values of `Y`. This means positive values of `zX` will go with positive values of `zY`, and negative values of `zX` will go with negative values of `zY`. Therefore `zX*zY` will be positive for most subjects (either two positive numbers or two negative numbers), and the correlation will be positive. Conversely, if `X` and `Y` have a negative relationship, then large values of `X` will go with small values of `Y`, and small values of `X` will go with large values of `Y`. This means positive values of `zX` will go with negative values of `zY`, and negative values of `zX` will go with positive values of `zY`. Therefore `zX*zY` will be negative for most subjects (one positive number and one negative number), and the correlation will be negative.

To see what `zX*zY` looks like, and how it relates to the raw scores, let's make a table. The `cbind()` function stands for column-bind; it takes a set of variables and puts them next to each other as columns in a matrix. Here we're just using it as a convenient way to look at everything at once.

```
> cbind(X,Y,zX,zY,zX*zY)
```

Look at this table and think about how the columns relate to each other. Any subject with an `X` score below the mean will have a negative `X` z-score, and any subject with an X score above the mean will have a positive X z-score. The same applies to Y and Y z-scores. Subjects who are above the mean on both variables or below the mean on both variables will have positive values of zX*zY and will thus contribute to a positive correlation. Subjects who are above the mean on one variable and below the mean on the other will have negative values of zX*zY and will thus contribute to a negative correlation. Because of the overall positive relationship between X and Y, most of the subjects have positive values of zX*zY, which means the correlation should be positive.

Now let's actually compute the correlation, using the average of zX*zY. Correlation uses a mean-squares type formula, so to get the average you divide by the degrees of freedom, which for correlation is equal to *n* – 1.

```
> r = sum(zX*zY)/(n-1)
```

The positive value of r indicates the positive relationship. The fact that r is close to 1 means the relationship is a strong one.

Now use the built-in correlation function to verify your result.

```
> cor(X,Y)
```

Prediction

Once you know the correlation between two variables, you can use one to predict the other. Imagine you know the value of X for some subject and want to predict their value on Y. This is easy to do if you use z-scores. The best guess for the z-score of Y, zYhat, equals zX times the correlation. (Statisticians use "hat", as in $\hat{z}_Y$ or $\hat{Y}$, to name predictions or estimates.)

First, get the predictions for all the data we already have. You can do this all at once since zX is a vector.

```
> zYhat = r*zX
```

Next, compare these predictions to the actual data. Start by making a scatterplot of the z-scores, zX and zY. Then add the predictions to the plot. The points() function adds points to an existing plot. Set the color to red for the new points to separate these predictions from your actual data.

```
> points(zX,zYhat,col='red')
```

Notice how the predictions all lie on a line. The actual data (black points) generally lie off the line, but the predictions track the actual data fairly well.

Now draw a line through all the predictions:

```
> lines(zX,zYhat,col='red')
```

This line is called the *regression line*. It corresponds to what's called *regressing Y onto X*, basically because we shrink (regress) all the Y values onto the line that represents the best prediction from X.

Researchers don't usually want to predict data they already have, but the regression line is useful for making predictions about new subjects. Often, the predictor is easy to measure, but the outcome is not. For example, the outcome could be lifespan, and the predictor could be hours of exercise per week. We might gather a sample of recently deceased people to measure both variables (perhaps we'd interview family members to find out how much the dead person had been exercising). After finding the regression line for the sample, we could use it to predict how long other people will live. For example, if you exercise an average of 7 hours per week, I might be able to tell you that you should live to be about 92, without waiting around for you to die before I can give an answer.

With that perspective in mind, try finding predictions for new values of `X` that weren't part of the original sample. (It's best to call your new values something like `X.new`, so that you don't erase your original data.) For each choice of `X.new`, you can predict `zY.new` by multiplying `zX.new` by the correlation, just like you did above. Use `points()` to add the new predictions (`zX.new` & `zYhat.new`) to the scatterplot, using a color other than red or black. Notice how your predictions always end up on the regression line.

Because the correlation between `X` and `Y` is positive, any choice of `X.new` above the mean will lead to a predicted `Y` above the mean (i.e., $z_Y > 0$), and any choice of `X.new` below the mean will lead to a prediction below the mean (i.e., $z_Y < 0$). If the correlation had been negative, then your predictions would follow the opposite pattern.

Now try a prediction based on `X.new = mean(X)`, and notice your prediction is exactly `mean(Y)`, i.e. `zY = 0`. Think about why it always works out this way.