

Week 10 Live Session

w203 Instructional Team

March 13, 2017

Announcements

Remember that Lab 3 is due next week.

Simple linear regression

Suppose we have data, represented by $(X_1, Y_1), \dots, (X_n, Y_n)$.

Q1.1: Write a simple regression model for the i^{th} case.

Q1.2: The statistical errors u_i cannot be observed. Why?

Q1.3: What assumptions do we need to make?

Q1.4: Do we want the residuals to be small in magnitude? Why or why not?

Q1.5: To define a regression line, is it sufficient to require $\sum \hat{u}_i = 0$?

Properties of residuals

Q2.1: What are the implications of the following properties?

(1) $\sum \hat{u}_i = 0$.

(2) $\sum X_i \hat{u}_i = 0$

Q2.2: How many different lines through the X-Y plane would fulfill these two conditions?

Q2.3: Using the above conditions, compute $cov(\hat{Y}_i, \hat{u}_i)$.

Regression in R

When a linear pattern is evident from a scatter plot, the relationship between the two variables is often modeled with a straight line. This line is expressed in a linear model between the response (or dependent) variable and the predictor (or independent) variable.

The following functions are useful for running a linear regression in R.

- Fitting a model: `model <- lm(y ~ x)`
- Coefficients: `model$coef` or `coef(model)`
- Fitted values: `model$fitted` or `fitted(model)`
- Residuals: `model$resid` or `resid(model)`

Install and load the BSDA package using the commands `install.packages("BSDA")` and `library(BSDA)`, respectively.

```
library(BSDA)
```

```
## Loading required package: e1071
```

```
## Loading required package: lattice
##
## Attaching package: 'BSDA'
##
## The following object is masked from 'package:datasets':
##
##      Orange
```

We are interested in using the Gpa data frame, which is available in memory once we import the package.

Before we can find the least square regression line, we need to examine the explanatory and response variables. Briefly examine college GPA (CollGPA) and high school GPA (HSGPA).

Q3.1. Create a scatterplot of CollGPA versus HSGPA and find the correlation between the two variables. What can we infer from the correlation?

Now that we know a few things about the data, we want to find a line that best represents the relationship between the variables. In other words, we want to draw a slope that comes closest to describing the data.

Q3.2. Characterize the equation mathematically. Find the least squares estimates of β_0 and β_1 using equations,

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

To perform the least square regression in R we can use the `lm` command. If you are interested use the `help(lm)` command to learn the different options for using this function. The model we are assuming is defined in the `lm` command using a tilde (“~”) between the response variable and the explanatory variable: `lm(Y ~ x)`.

To see the coefficients in your model, use the `coef` function (i.e. `coef(model)`).

If you would like to know what else is stored in the variable you can use the `attributes()` command.

Q3.3. Find the least squares estimates of β_0 and β_1 using the R function `lm()`.

Q3.4. `abline()` adds one or more straight lines to the current plot. Conveniently, you can pass your linear model object into `abline` as an argument (i.e. `abline(model)`).

Q3.5 Compute the correlation between your x values and your model residuals. What do you learn from this?

OLS Goodness of Fit

When building regression models, “goodness-of-fit” explains how closely our model of the data (i.e. the predictor variables) fits the outcome data. In other words, how much of the variation in an outcome can we explain with a particular model?

R-Squared

R-squared is a measure commonly used for assessing model fit. It can be understood as the proportion of variance in the outcome that can be accounted for by the model.

Looking at our simple bivariate model, we can extract R-squared as a measure of model fit in a number of ways. The easiest is simply to extract it from the `lm` object using `summary(model)$r.squared`.

Warning: We normally discourage students from using the summary command with lm objects. The reason, as we will see later, is that summary makes a strong assumption called homoskedasticity, which is usually not justified. However, it is ok to use the command in order to extract R-squared.

But we can also calculate R-squared from our data in a number of ways. Take a couple of minutes to manually calculate R-squared.

1. By squaring the correlation between X and Y.
2. By taking the ratio of the variance of the fitted values to the variance of Y.
3. By weighting the slope coefficient: $R^2 = \beta_1^2 \frac{\text{var}(X)}{\text{var}(Y)}$

Adjusted R Square

The “Adjusted R-squared” is commonly used in place of the “regular” R-squared, which is sensitive to the number of independent variables in the model. In other words, as we put more variables into the model, R-squared increases even if those variables are unrelated to the outcome.

Adjusted R-squared attempts to correct for this by deflating R-squared by the expected amount of increase from including irrelevant additional predictors.

We can see this property of R-squared and Adjusted R-squared by adding a completely random variables unrelated to our other covariates or the outcome into our model and examine the impact on R-squared and Adjusted R-squared.

```
tmp1 <- rnorm(10, 0, 10)
```

Add this variable to your simple regression model, creating a new lm object, then observe what happens to R-squared.

Now extract the adjusted R-squared from both models using `lm$adj.r.squared`. It may also go down, but by less than regular r-squared.

OLS: Issues to be Aware of

Unfortunately, the pitfalls of applying least squares are not often well understood by many of the people who attempt to apply it. What follows is a list of some of the biggest problems with using least squares regression in practice, along with some brief comments about how these problems may be mitigated or avoided

Outliers

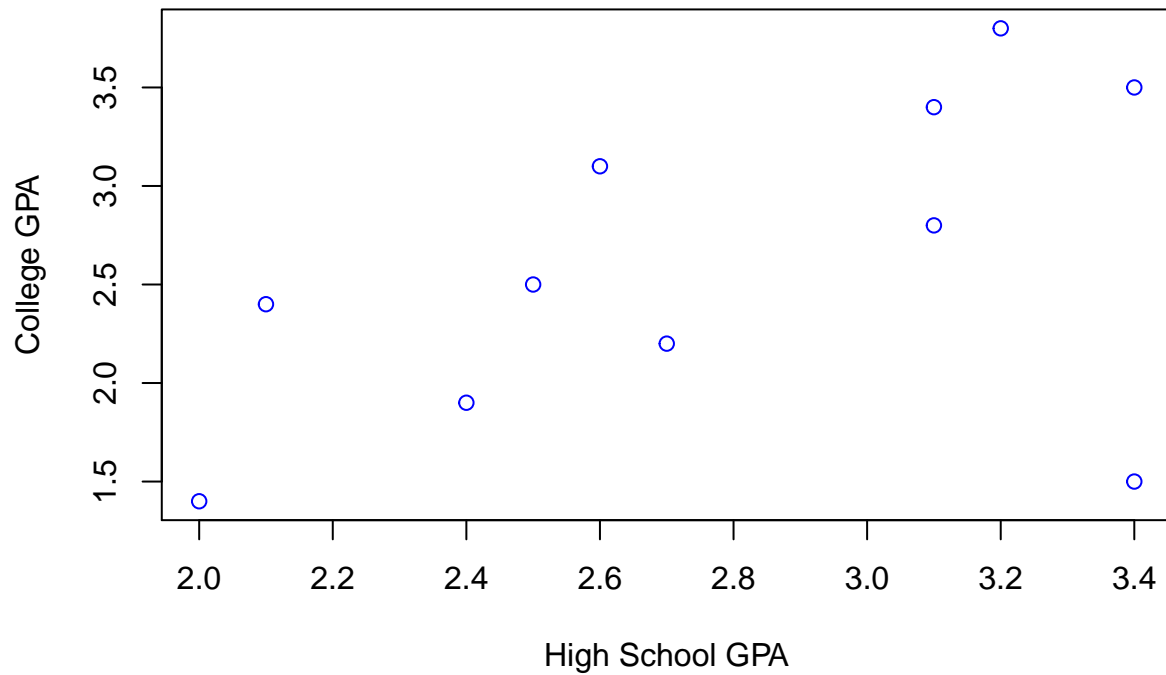
Least squares regression can perform very badly when some points in the training data have excessively large or small values for the dependent variable compared to the rest of the training data. The reason for this is that since the least squares method is concerned with minimizing the sum of the squared error, any training point that has a dependent value that differs a lot from the rest of the data can have a disproportionately large effect on the resulting constants that are being solved for.

WARNING: Do not ever remove an observation just because it’s an outlier.

Returning to our example, let’s add an outlier.

```
y_out <- c(Gpa$CollGPA, 1.5)
x_out <- c(Gpa$HSGPA, 3.4)
plot(x_out, y_out, col="blue", main="Scatterplot of College Versus High School GPA",
      xlab="High School GPA", ylab="College GPA")
```

Scatterplot of College Versus High School GPA



```
cor(x_out,y_out)
```

```
## [1] 0.4986491
```

We can see the outlier pulls the correlation off a lot. Let's see what it does to the linear model.

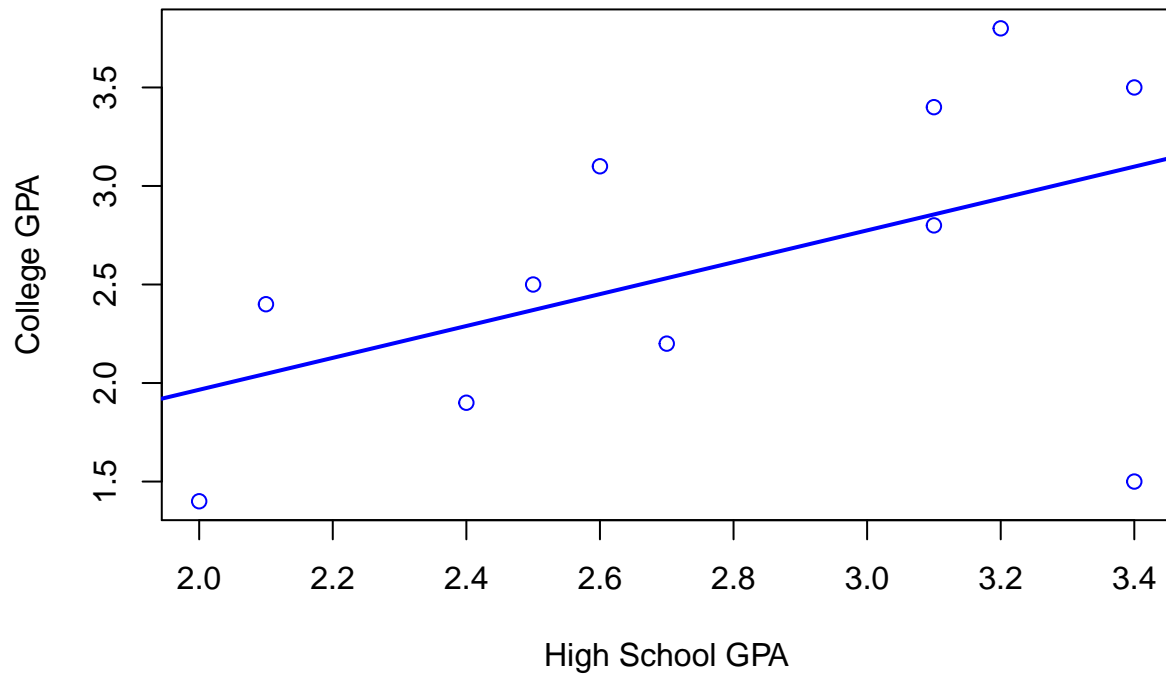
```
model_out <- lm(y_out ~ x_out)
model_out$coef
```

```
## (Intercept)      x_out
##  0.3483516    0.8087912
```

Let's see that scatterplot again with our new regression line.

```
plot(x_out, y_out, col="blue", main="Scatterplot of College Versus High School GPA",
     xlab="High School GPA",ylab="College GPA")
abline(model_out,col="blue",lwd=2)
```

Scatterplot of College Versus High School GPA



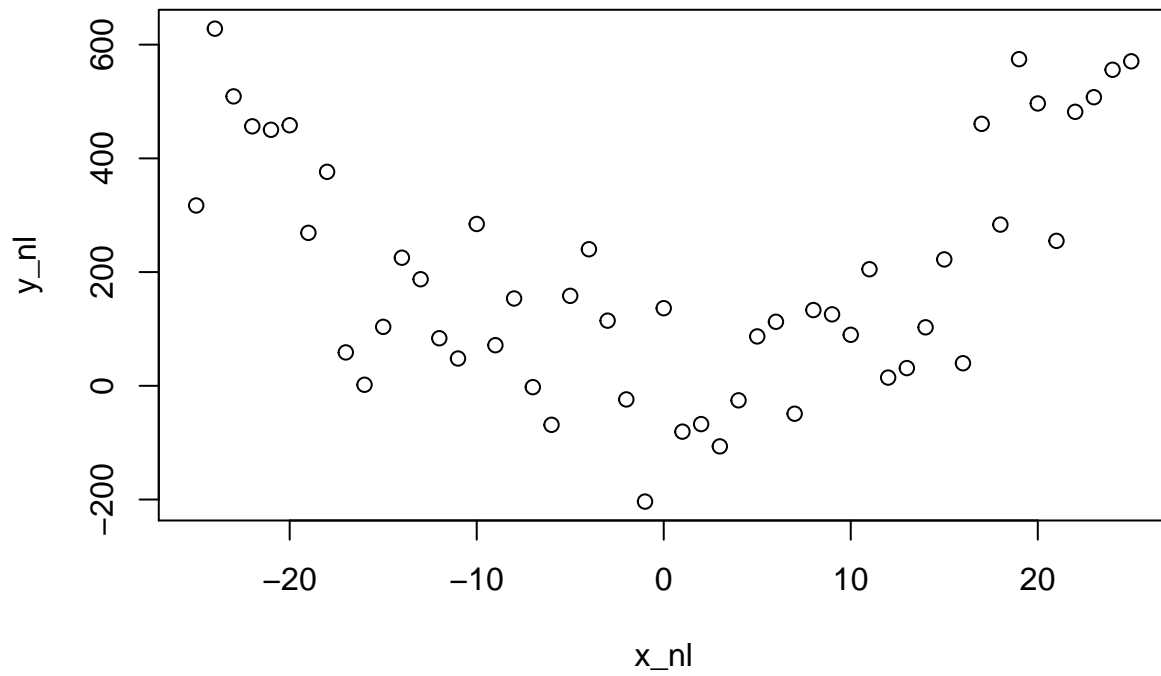
WARNING: Do not ever remove an observation just because it's an outlier.

Non-Linearities

All linear regression methods (including, of course, least squares regression), suffer from the major drawback that in reality most systems are not linear.

Let's take another dataset that is clearly non-linear.

```
x_nl<-seq(-25,25,1)
y_nl<-x_nl^2+rnorm(51,0,100)
plot(x_nl,y_nl)
```



There's definitely a relationship here, but we will need to do a transformation prior to OLS.