In-class Lab 4

ECON 425 (Justin Heflin, West Virginia University)

February 3, 2023

The purpose of this in-class lab is to continue to familiarize yourself with RStudio, more specifically today we will be learning about how to run a simple linear regression by taking a theoretical regression equation and estimating it to calculate regression coefficients, fitted values, and residuals. We will also plot our estimated regression line to the data to visually see how close our estimated ($\hat{Y}$'s) are to the actual data points. The lab may be completed as a group, but each student should turn in their own work. To get credit, upload your .R script to the appropriate place on eCampus ("In-Class Labs'' folder).

For starters

Open up a new R script (named `ICL4_XYZ.R`, where `XYZ` are your initials)

Your first regression (of this class)

Let's load a data-set and run various regressions. This data-set contains year-by-year statistics about counties in the US. It has counts on number of various crimes committed, as well as demographic characteristics about the county.

```
library(wooldridge)
crime_data <- as.data.frame(countymurders)
#View(crime_data)
```

A handy command to get a quick overview of an unfamiliar data-set is `str()`:

```
str(crime_data)
```

`str()` tells you the number of observations, number of variables, and the name and type of each variable (e.g. integer, numeric,...)

Regression syntax

To run a regression of y on x in R, use the following syntax:

```
estimate <- lm(y ~ x, data=data.name)
```

Here, `estimate` is an object where the regression coefficients (and other information about the model) is stored. `lm()` stands for "linear model" and is the function that you call to tell R to compute the coefficients. `y` and `x` are variable names from whatever `data frame` you have stored your data in. The name of the `data frame` is `data.name`

Regress murders on population

Using the `crime_data` data-set we created above, let's run a regression where `murders` is the dependent variable and `popul` is the independent variable:

```
estimate <- lm(murders ~ popul, data = crime_data)
```

$$Murders_i = \beta_0 + \beta_1 Population + \epsilon_i$$

To view the output of the regression in a friendly format, type

```
summary(estimate)
```

```
##
## Call:
## lm(formula = murders ~ popul, data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -284.39   -0.40    3.48    5.24  595.48
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.324e+00  1.235e-01   -51.2   <2e-16 ***
## popul        1.523e-04  4.317e-07   352.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.68 on 37347 degrees of freedom
## Multiple R-squared:  0.7693, Adjusted R-squared:  0.7693
## F-statistic: 1.246e+05 on 1 and 37347 DF,  p-value: < 2.2e-16
```

In the `estimate` column, we can see the estimated coefficients for β_0– `(Intercept)` in this case and β_1 `(popul)`. `estimate` also contains other information that we will use later in the course.

Another easy way to view a regression coefficients is to use the `cofficients()` function:

```
coefficients(estimate)
```

```
##   (Intercept)         popul
## -6.3239142987  0.0001523426
```

To interpret these results let's first start with the intercept coefficient ($\hat{\beta}_0$). Recall that the intercept coefficient tells us the average expected value for the dependent variable when all of the independent variables are equal to zero. In this specific example if population is zero then it stands to reason that there would not be any murders so our intercept coefficient of -6.323 is not really meaningful. It does not tell us much.

To interpret our ($\hat{\beta}_1$) coefficient we can look at in the aspect that, on average, each additional individual (human being) is associated with an increase of 0.0001523 murders. Basically, as the population grows each additional individual is associated with an increase of 0.001523 murders.

We can plug the regression coefficients back into our estimated regression equation:

$$\widehat{Murders}_i = -6.324 + 0.0001523 Population_i + e_i$$

Fitted/Estimated Values & Residuals

We can view our estimated/fitted values of our dependent variable (murders) by using the `fitted.values()` function like so:

```
fitted.values(estimate)
```

To see how close our $\hat{Y}$ (estimated/fitted values of our dependent variable) is to the actual value of Y we can use the `residuals()` function.

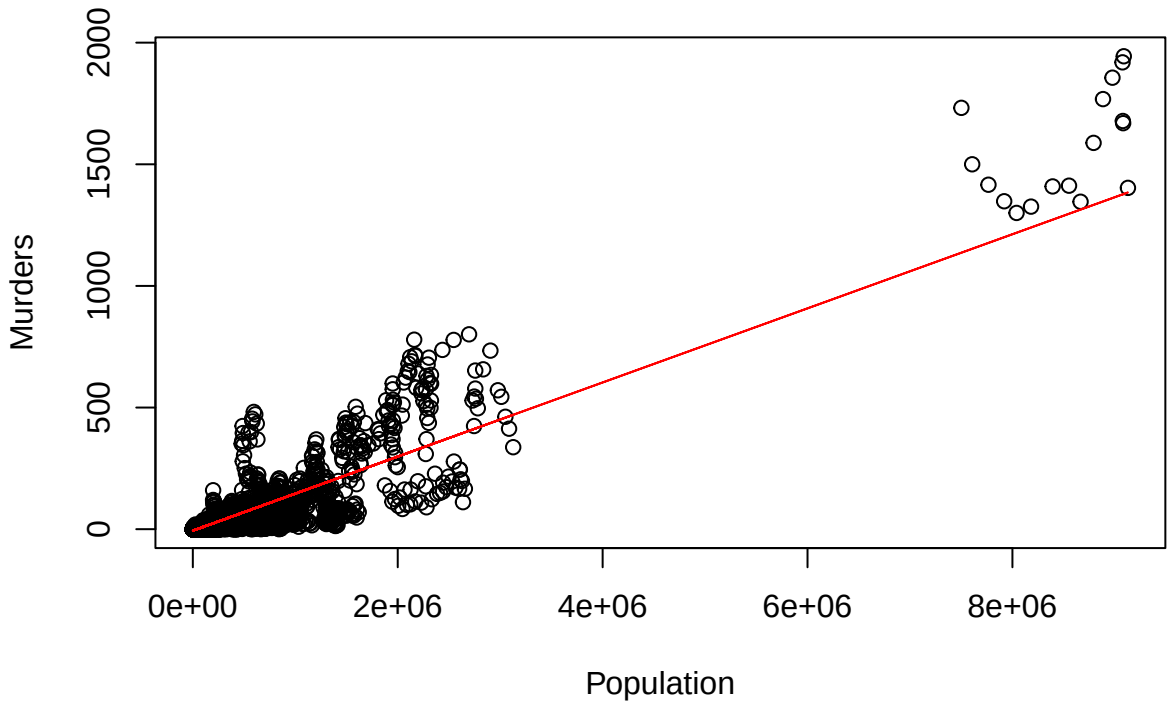
$$Residual: e_i = Y_i - \hat{Y}_i$$

```
residuals(estimate)
```

Plot regression line to the data

We can take our regression results to fit a line to the actual data from our random sample.

```
fit <- crime_data$murders - estimate$residuals
plot(y = crime_data$murders, x = crime_data$popul, ylab = "Murders",
     xlab = "Population") + lines(y = fit, x=crime_data$popul, col = "red")
```




```
## integer(0)
```

The red line is our estimated regression line. The black circles are the actual data points from our sample. The idea of the regression line (estimated line) is to fit a straight line through all the data points that gets as close as possible to as many data points as possible. The difference between a black circle and the red line is a visual representation of the residual for that particular observation.