Data Science for Economists

Lecture 7: Introduction to Regression - Math

Drew Van Kuiken

University of North Carolina | ECON 370

Table of contents

1. Introduction

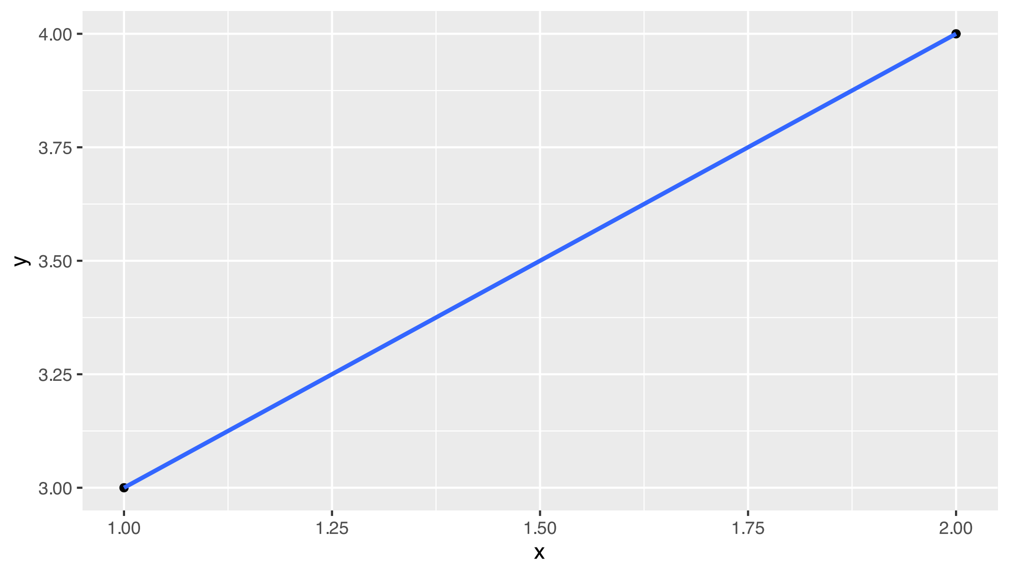
2. Regression: An Intuitive Approach

3. Regression: Some Math

An Easy Question

What's the line of best fit when our data looks like this?

```
d = data.frame(x=1:2, y=3:4)
```



$$Y = mX + B \Rightarrow Y = \beta_0 + \beta_1 X$$

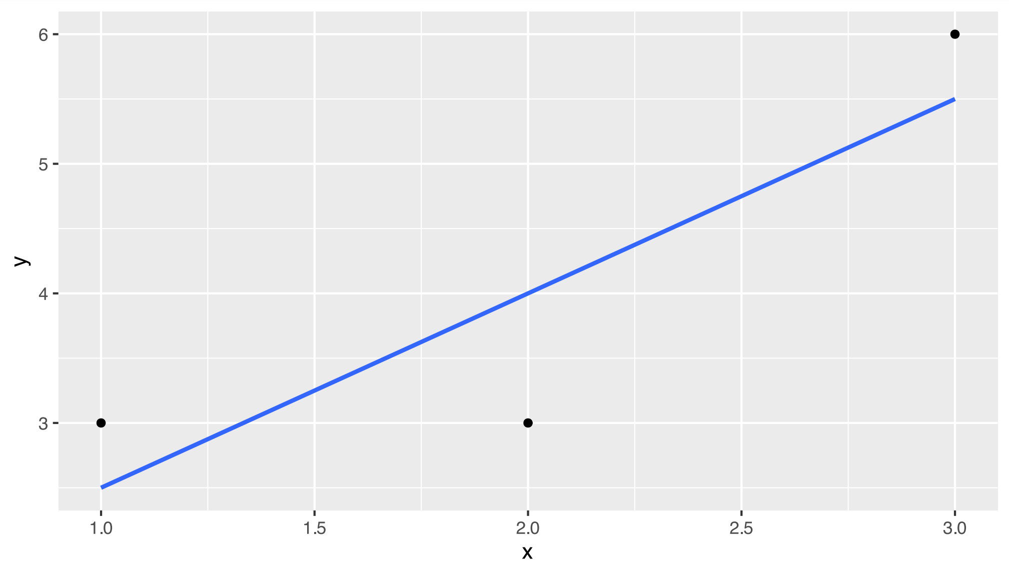
$$\beta_1 = \frac{\text{Rise}}{\text{Run}} = \frac{4-3}{2-1} = 1$$

$$\beta_0 = Y - \beta_1 X = Y - X = 3 - 1 = 2$$

We need a little more notation now:

- Individual observation i is referred to as (x_i, y_i)
- Predicted y is given by $\hat{y}$
- Average y is given by $\bar{y}$ (average x is given by $\bar{x}$)

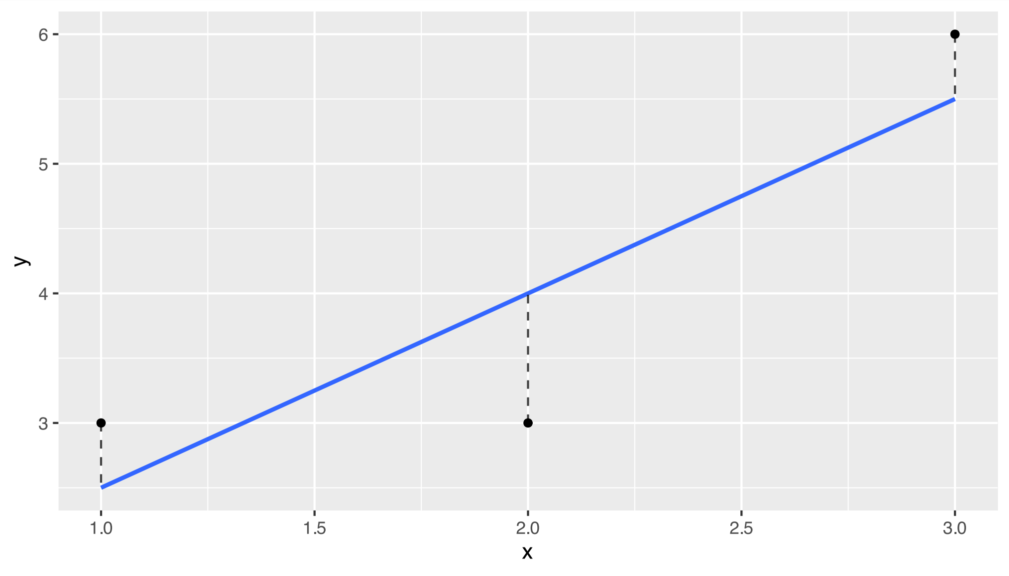
How about three points?



No longer a perfect fit. Is the following correct?

$$y_i = \beta_0 + \beta_1 x_i$$

Error



The dashed lines are ***errors***. Our *line of best fit* is still given by:

$$\hat{y} = \beta_0 + \beta_1 x_i$$

But for any given point, we need to add in those errors: if an observation i is given by (x_i, y_i), then for values of β_0, β_1

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Calculating Errors

$$\varepsilon_i(\beta) = y_i - \beta_0 - \beta_1 x_i$$

Note that this formula implicitly compares *predicted* y, which we call $\hat{y}$ to *observed* y, which is y_i

In this case:

$$\varepsilon_1 = 3 - 1 - 1.5(1) = 0.5$$

$$\varepsilon_2 = 3 - 1 - 1.5(2) = -1$$

$$\varepsilon_3 = 6 - 1 - 1.5(3) = 0.5$$

The sum of our squared residuals:

$$\sum_{i=1}^{3} \varepsilon_i^2 = -0.5^2 + 1^2 + -0.5^2 = 1.5$$

We said a regression **minimizes** errors

Minimizing Error

How can we minimize error?

Choose β_0, β_1 to "minimize" the total (or sum of) squared error,

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{N} \varepsilon_i(\beta_0, \beta_1)^2$$

Why squared?

1. So positive and negative errors don't cancel out
2. So larger errors contain more weight

How to Find β_0, β_1: R Code

Running regressions in `R` is simple. Let's start by downloading a real dataset:

```
url = "https://www.statlearning.com/s/Advertising.csv"
advert_data = read_csv(url)
tv_reg <- lm(sales ~ TV, advert_data)
summary(tv_reg)
```

```
##
## Call:
## lm(formula = sales ~ TV, data = advert_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119,	Adjusted R-squared:  0.6099
```

How to Find β_0, β_1: R Code

General notes on coding regressions. Okay if this doesn't make sense now, should be useful later:

- Anything left of `~` is our dependent variable. To the right is our formula
- `summary()` will return useful information on your regression
- `:` codes interactions. Think of this as: what is the effect of x_1 and x_2 together
    - Conditional on x_1, what is the effect of x_2?
- as.factor(x) will tell R that x is a dummy variable
- use `I(x^2)` to fit higher order polynomials

How to Find β_0, β_1: Algebra

How do we find the minimum of a function? Take the derivative a set it equal to 0! But we'll skip this today.

Instead, take it from me that we end up with the following formulas:

$$\beta_0 = \bar{y} - \hat{\beta}\bar{x}$$

$$\beta_1 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Take 5 minutes and code up our regression coefficients using just algebra for the Sales ~ TV regression

How to Find β_0, β_1: Plug and Chug

We can also write an algorithm to iteratively minimize $f(\beta_0, \beta_1) = \sum_{i=1}^{N} \varepsilon(\beta_0, \beta_1)^2$. Any ideas?

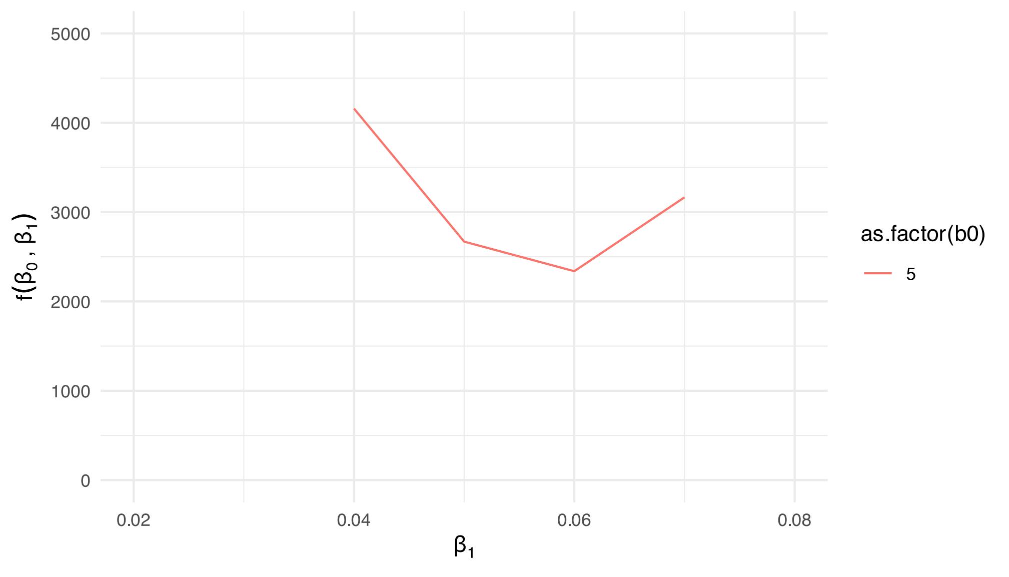
```
# grid search for coefficients
beta = expand.grid(b0 = seq(5,10,0.25), b1 = seq(-0.1,0.1,0.01))
tss = sapply(1:nrow(beta), function(i) {
  sum((advert_data$sales - beta$b0[i] - beta$b1[i] * advert_data$TV)^2)
})
beta[tss==min(tss),]
```

```
##       b0   b1
## 323 6.75 0.05
```

Plot of Objective Function $f(\beta_0, \beta_1)$

We can visualize the results. Let's start with $\beta_0 = \mathbf{5}$:

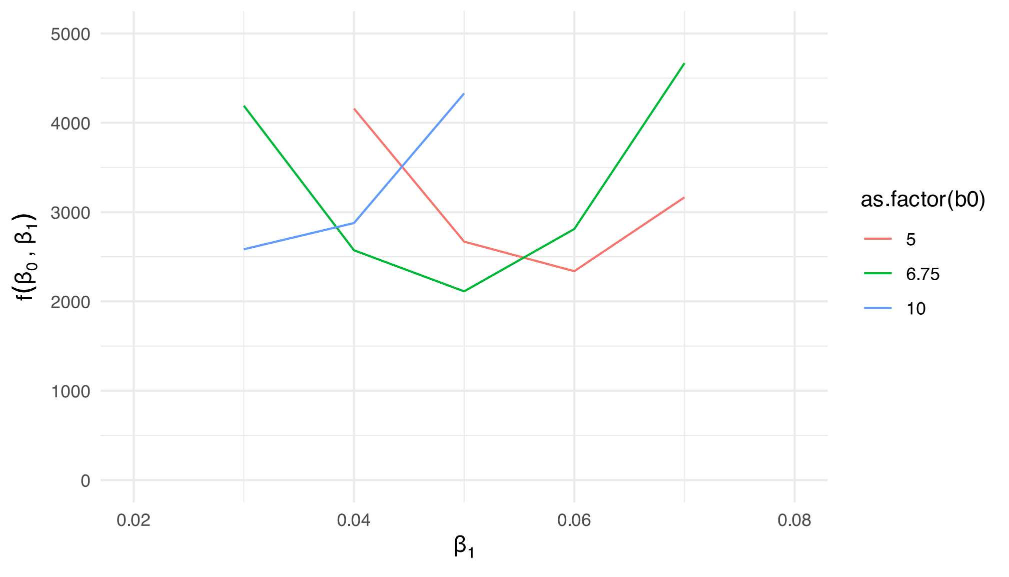
```
g1
```



Plot of Objective Function $f(\beta_0, \beta_1)$

We can visualize the results. Let's add a couple more:

```
g2
```



Notice the green line is lower than the blue and red lines

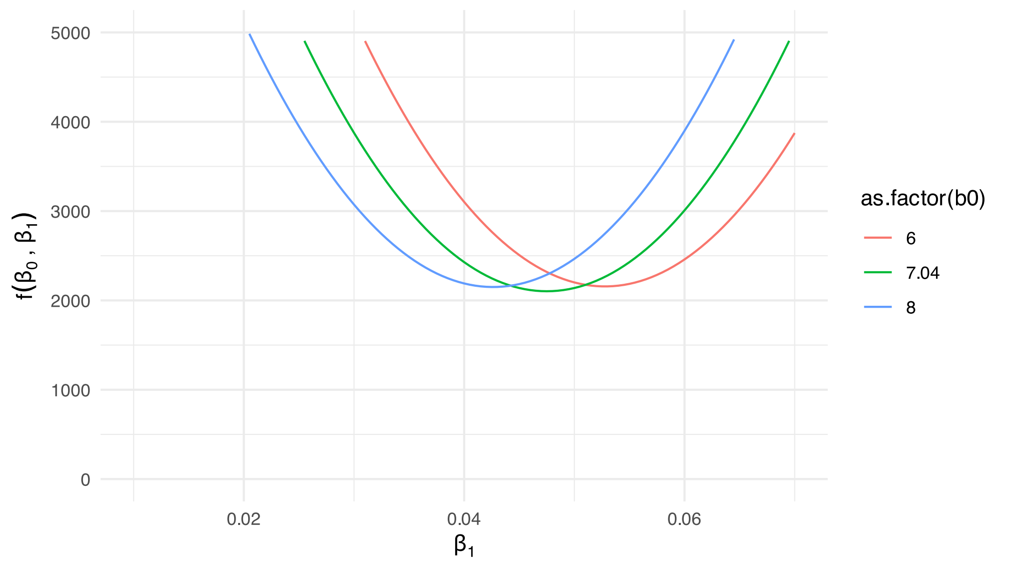
Increasing Our Grid Search

```r
# more precise
beta2 = expand.grid(b0 = seq(6,8,0.01), b1 = seq(-0.1,0.1,0.0005))
tss2 = sapply(1:nrow(beta2), function(i) {
  sum((advert_data$sales - beta2$b0[i] - beta2$b1[i] * advert_data$TV)^2)
})
beta2[tss2==min(tss2),]
```

```
##         b0     b1
## 59400 7.04 0.0475
```

This Also Makes Our Graph Look Better

```
g3
```



This Is Optimization!

```r
# optimization
f = function(theta) sum((advert_data$sales-theta[1]-theta[2]*advert_data$TV)^2) # for

optim(c(mean(advert_data$sales),0),f,method="BFGS")$par # solve numerically with opti
```

```
## [1] 7.03259355 0.04753664
```

```r
coef(lm(sales~TV,data=advert_data))     # solve with OLS
```

```
## (Intercept)          TV
##  7.03259355  0.04753664
```

Assessing Model Fit

Coefficients have standard errors

- Calculated as $\sqrt{\frac{\sigma_\varepsilon^2}{\sum_1^n (x_i - \bar{x})^2}}$
- Unexplained variation in y as a share of variation in x
- More variation in x $\Rightarrow$ more leverage to estimate β_1
- Can use standard errors to construct a confidence interval for β_1: the range of values such that, with 95% probability, the range will contain the true unknown value of β_1
- 95% CI: $[\beta_1 - 2SE(\beta_1), \beta_1 + 2SE(\beta_1)]$
- P-values assess the likelihood that your coefficient is different than 0 due to random chance
- Small p-values $\Rightarrow$ can infer there is a relationship between independent variable and dependent variable

Assessing Model Fit

Models have R-squared values

- Calculated as $1 - \frac{\sum_1^n (y_i - \hat{y})^2}{\sum_1^n (y_i - \bar{y})^2}$
- Variation in y unexplained by predictors divided by total variance in y
- I.e., how much of the total variation in y does your model explain?
- In OLS, R-Squared = Corr(X,Y) squared!
- Runs from 0 to 1, 0 is explains nothing, 1 is a perfect fit
- What's a good r-squared? Depends
    - Cryptocurrency transaction data? 0.00008 was great
    - Testing lab data from physics experiment? Should be ~1

Multivariate Regression

In our gapminder regression, we added an indicator variable for "defender in a war." This was our first example of multivariable regression.

This extension is fairly simple. Each explanatory variable gets its own slope. Our regression model becomes:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \varepsilon$$

Interpretation: what's the average effect of a one-unit increase in x_1 on y holding all other x variables fixed?

Estimating the coefficients (the βs) is a little trickier. Need matrix algebra to represent the closed-form solution. We are still minimizing squared errors though! Now it's just in p dimensions.

Dummy Variables

```r
library(gapminder)
gp_subset <- gapminder[gapminder$continent=="Asia"&gapminder$year==2007,]
gp_subset$long_and_brutal_war = gp_subset[,"country"] == "Afghanistan"
mv_reg <- lm(lifeExp ~ gdpPercap + long_and_brutal_war, data = gp_subset)
summary(mv_reg)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap + long_and_brutal_war, data = gp_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1358 -3.2854  0.7948  3.4495  6.2692
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.714e+01  1.012e+00  66.340  < 2e-16 ***
## gdpPercap               3.454e-04  5.328e-05   6.483 3.64e-07 ***
## long_and_brutal_warTRUE -2.365e+01  4.332e+00  -5.458 6.40e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.221 on 30 degrees of freedom
## Multiple R-squared:  0.7367     Adjusted R-squared:  0.7191
```

Dummy Variables

```
mv_reg2 <- lm(lifeExp ~ gdpPercap + long_and_brutal_war - 1, data = gp_subset)
summary(mv_reg2)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap + long_and_brutal_war - 1, data = gp_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1358 -3.2854  0.7948  3.4495  6.2692
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## gdpPercap                3.454e-04  5.328e-05   6.483 3.64e-07 ***
## long_and_brutal_warFALSE 6.714e+01  1.012e+00  66.340  < 2e-16 ***
## long_and_brutal_warTRUE  4.349e+01  4.221e+00  10.304 2.27e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.221 on 30 degrees of freedom
## Multiple R-squared:  0.9968,    Adjusted R-squared:  0.9965
## F-statistic:  3117 on 3 and 30 DF,  p-value: < 2.2e-16
```

```
mean(gp_subset$lifeExp[gp_subset$long_and_brutal_war < 1])
```

Last note

In class, we discussed what it would mean if having a lower GDP per capita meant that you were more likely be the defender in a war. For this class, that is okay. As long as we can hold "being at war" and "gdp per capita" fixed, we've dealt with omitted variable bias.

As you learn more econometrics, you'll learn more about exogeneity, endogeneity, and causal inference.

We Covered OLS

The reason we had an "algebra" version of these slides is because OLS is really nice. It has a closed form solution!

- If you've taken 400 or 470, you may have seen the formula for OLS

However, not all optimization problems will have a closed form solution.

- Will need numerical methods and optimization techniques to find solutions.
- We'll cover this a little later in the course.

Next lecture(s): Simulation