**Least squares and parameter estimates**

Suppose we have a sample of size n where we obtain (x1, y1), (x2, y2), …, (xn, yn). The formulas for  and  are found using the least squares method (more on this later):



where Σ assumes summing from i=1,…,n

 =  – 

Some books will use the terminology:

* 
* 

where “SS” stands for sums of squares.

Example: Sales and advertising (sales\_advertising.R)

What is the relationship between sales and advertising for a company?

Let x = advertising ($100,000) and y = sales units (10,000)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **x** | **y** | **x2** | **y2** | **x⋅y** |
|  | x1=1 | y1=1 | 1 | 1 | 1 |
|  | x2=2 | y2=1 | 4 | 1 | 2 |
|  | 3 | 2 | 9 | 4 | 6 |
|  | 4 | 2 | 16 | 4 | 8 |
|  | 5 | 4 | 25 | 16 | 20 |
| **Σ** | 15 | 10 | 55 | 26 | 37 |

(e.g., Σx2=55)



 =  –  = 10/5 – (0.70)⋅15/5 = 2 – 2.1 = -0.1

 = -0.1 + 0.7x

Scatter plot:

# 

We will discuss the R code later. Below is the same plot with the sample regression model:



|  |  |  |  |
| --- | --- | --- | --- |
| x | y |  | y- |
| 1 | 1 | 0.6 | 0.4 |
| 2 | 1 | 1.3 | -0.3 |
| 3 | 2 | 2 | 0 |
| 4 | 2 | 2.7 | -0.7 |
| 5 | 4 | 3.4 | 0.6 |

For example:

 = -0.10 + 0.70×1 = 0.6 when x = 1

What does the sales and advertising sample regression model mean? Remember advertising is measured in $100,000 units and sales is measured in 10,000 units.

1. Estimated slope  = 0.7:

Sales volume is estimated to increase by 0.7×10,000 = 7,000 units for each 1×100,000 = $100,000 increase in advertising.

1. Use model for prediction:

Estimate sales when advertising is $100,000:

 = -0.10 + 0.70×1 = 0.60

Estimated sales are 6,000 units.

Estimate sales when advertising is $250,000:

 = -0.10 + 0.70\*2.5 = 1.65

Estimated sales are 16,500 units.

#### Residual

#### ei = yi – = observed value – predicted value

#### This gives a measurement of how far the predicted value is from the sampled value.

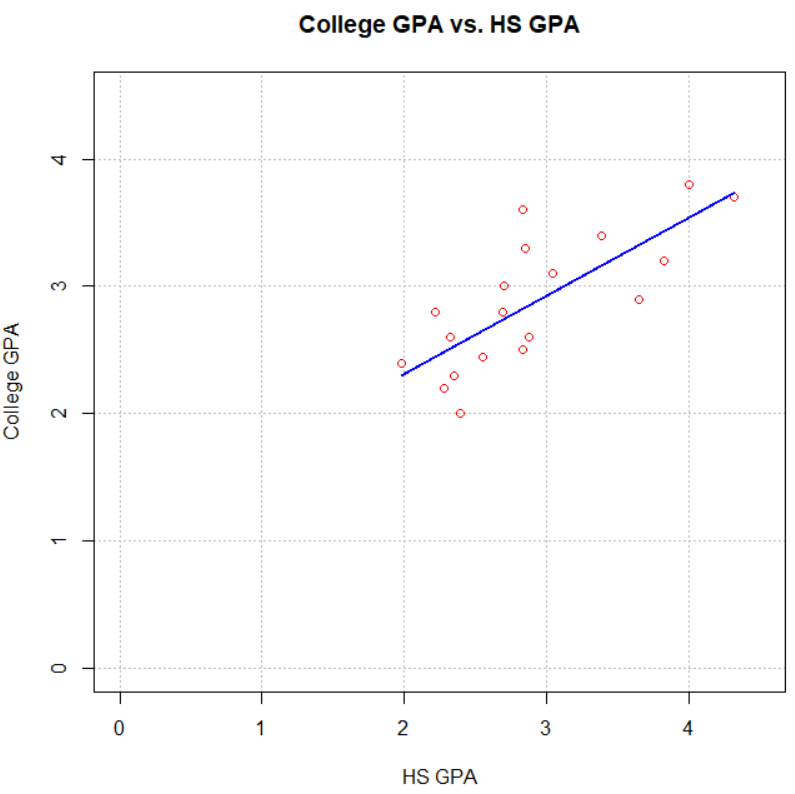
#### We want these to be small!

#### 

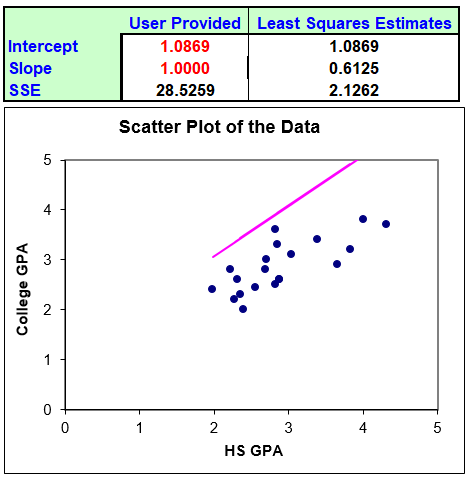
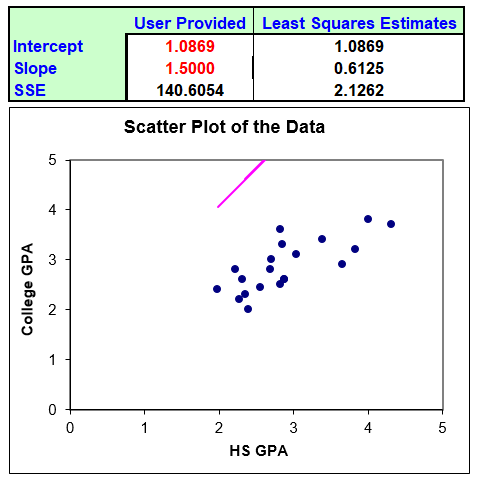
#### Example: Least squares method explanation (least\_squares\_demo.xlsx)

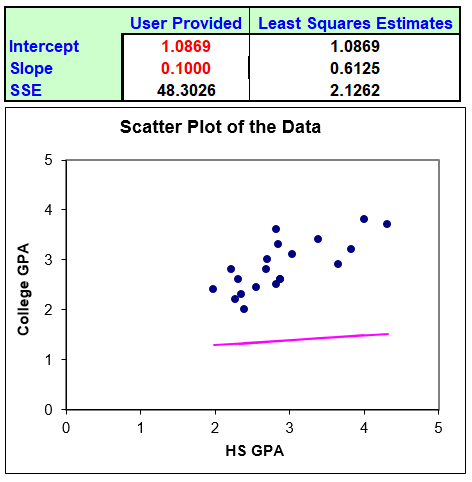
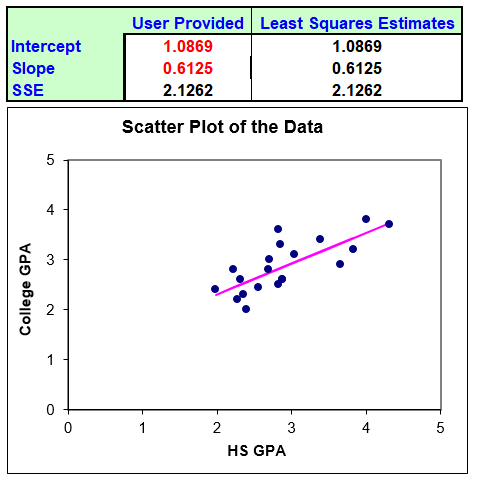
The least squares method is used to find the equations for  and . Below is the explanation of the least squares method relative to the HS and College GPA example:

* Notice how the sample (estimated) regression model seems to go through the “middle” of the points on the scatter plot. For this to happen,  and  must be 1.09 and 0.61, respectively. This provides the “best fit” line through the points.



* The least squares method tries to find the  and  such that SSE =  = is minimized. These formulas are derived through using calculus. Note that SSE – sum of squares for error – is equivalent to SSE discussed earlier in the course!
* Least squares method demonstration with least\_squares\_demo.xlsx:
  + Uses the GPA example data set with  = 1.09 and  = 0.61
  + The demo examines what happens to the SSE and the sample regression line plot if values other than  and  are used as the y-intercept and slope in the sample regression model.
  + Below are a few cases:





Notice that as the y-intercept and slope get closer to  and , SSE becomes smaller and the line better approximates the relationship between x and y!

The actual formulas for  and  can be derived through using calculus. The purpose is to find a  and  such that

SSE =  = 

is minimized. Here’s the process:

* Find the partial derivatives with with respect to  and 
* Set the partial derivatives equal to 0
* Solve for  and !



Setting the derivative equal to 0 produces,





 (1)

And,



Setting the derivative equal to 0 produces,

 (2)

Substituting (1) into (2) results in,







Then  becomes



It can be shown that these values do indeed result in a minimum (not a maximum) for SSE.

### Example: HS and College GPA (gpa.R, gpa.csv)

> gpa <- read.csv(file = "gpa.csv")

> head(gpa)

HS.GPA College.GPA

1 3.04 3.1

2 2.35 2.3

3 2.70 3.0

4 2.55 2.45

5 2.83 2.5

6 4.32 3.7

> #Scatter plot

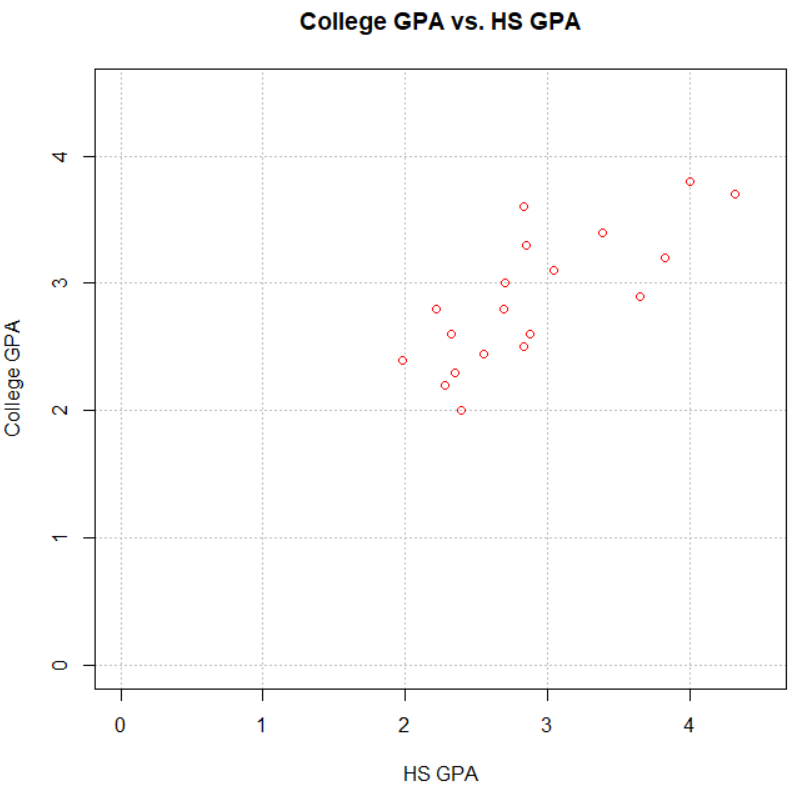
> plot(x = gpa$HS.GPA, y = gpa$College.GPA, xlab = "HS

GPA", ylab = "College GPA", main = "College GPA vs. HS

GPA", xlim = c(0,4.5), ylim = c(0,4.5), col = "red",

pch = 1, cex = 1.0, panel.first = grid(col = "gray",

lty = "dotted"))



> mod.fit <- lm(formula = College.GPA ~ HS.GPA, data = gpa)

> mod.fit

Call:

lm(formula = College.GPA ~ HS.GPA, data = gpa)

Coefficients:

(Intercept) HS.GPA

1.0869 0.6125

The lm() function performs the regression calculations. The sample regression model is:

 = 1.09 + 0.61x

where y = College GPA and x = HS GPA. It is o.k. to state the model a little less formally with

 = 1.09 + 0.61×(HS GPA),

You can see what is inside of mod.fit by using the names() function:

> names(mod.fit)

[1] "coefficients" "residuals" "effects" "rank"

[5] "fitted.values" "assign" "qr" "df.residual"

[9] "xlevels" "call" "terms" "model"

> mod.fit$coefficients

(Intercept) HS.GPA

1.0868795 0.6124941

> mod.fit$residuals

1 2 3 4 5

0.15113833 -0.22624072 0.25938633 -0.19873954 -0.32023790

6 7 8 9 10

-0.03285418 0.23676538 0.09213411 0.06551128 0.77976210

11 12 13 14 15

-0.55074048 -0.42248310 0.46751221 -0.23273205 0.35338352

16 17 18 19 20

0.10038212 -0.25086261 0.26314395 -0.28336613 -0.25086261

We can put some of this information into one data frame:

> save.fit <- data.frame(gpa, College.GPA.hat =

round(mod.fit$fitted.values,2), residuals =

round(mod.fit$residuals,2))

> head(save.fit)

HS.GPA College.GPA College.GPA.hat residuals

1 3.04 3.10 2.95 0.15

2 2.35 2.30 2.53 -0.23

3 2.70 3.00 2.74 0.26

4 2.55 2.45 2.65 -0.20

5 2.83 2.50 2.82 -0.32

6 4.32 3.70 3.73 -0.03

Also, some of the information in mod.fit can be summarized using the summary() function:

> summary(object = mod.fit)

Call:

lm(formula = College.GPA ~ HS.GPA, data = gpa)

Residuals:

Min 1Q Median 3Q Max

-0.55074 -0.25086 0.01633 0.24242 0.77976

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.0869 0.3666 2.965 0.008299 \*\*

HS.GPA 0.6125 0.1237 4.953 0.000103 \*\*\*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3437 on 18 degrees of freedom

Multiple R-squared: 0.5768, Adjusted R-squared: 0.5533

F-statistic: 24.54 on 1 and 18 DF, p-value: 0.0001027

Notice where  and  are located in the output.

You may remember using summary()elsewhere in the notes to simply find summary measures (like the mean) for a data frame. Why did summary() work differently here?

The answer will be given soon!

Plot of data and model:

> plot(x = gpa$HS.GPA, y = gpa$College.GPA, xlab = "HS

GPA", ylab = "College GPA", main = "College GPA vs. HS

GPA", xlim = c(0,4.5), ylim = c(0,4.5), col = "red",

pch = 1, cex = 1.0, panel.first = grid(col = "gray",

lty = "dotted"))

> curve(expr = mod.fit$coefficients[1] +

mod.fit$coefficients[2]\*x, xlim = c(min(gpa$HS.GPA),

max(gpa$HS.GPA)), col = "blue", add = TRUE, n = 1000,

lwd = 2)



IMPORTANT:

Do not extrapolate beyond the range of data when estimating E(Y).

Example: The GPA data has 1.98 ≤ x ≤ 4.32. Do not try to estimate E(Y) (i.e., find a ) when x is outside of this range.