Weekly Summary Template

Brady Miller

Table of contents

Tuesday, Jan 31

> **!** TIL
>
> Include a *very brief* summary of what you learnt in this class here.
> Today, I learnt the following concepts in class:
>
> 1. What statistical learning is and its goals
> 2. What simple linear regression is and how to create plots using beta values
> 3. What the least squares estimator is and how to create a plot that shows this value/residuals

```r
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0     v purrr   1.0.1
v tibble  3.1.8     v dplyr   1.1.0
v tidyr   1.3.0     v stringr 1.5.0
v readr   2.1.3     v forcats 1.0.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
library(knitr)
library(ISLR2)
library(cowplot)
library(kableExtra)
```

```
Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
library(htmlwidgets)
```

Statistical learning

- Statistical learning = finding patterns in the data
- Learn to make and quantify predictions
- Different types of statistical learning

    1. supervised learning
        - regression (has quantitative responses)
        - classification (categorical responses)
    2. unsupervised learning (there is no y)
    3. semi-supervised learning (case when we have y - both supervised & unsupervised)
    4. unsupervised learning (reinforcement learning - x & y variables can change)

Simple Linear Regression

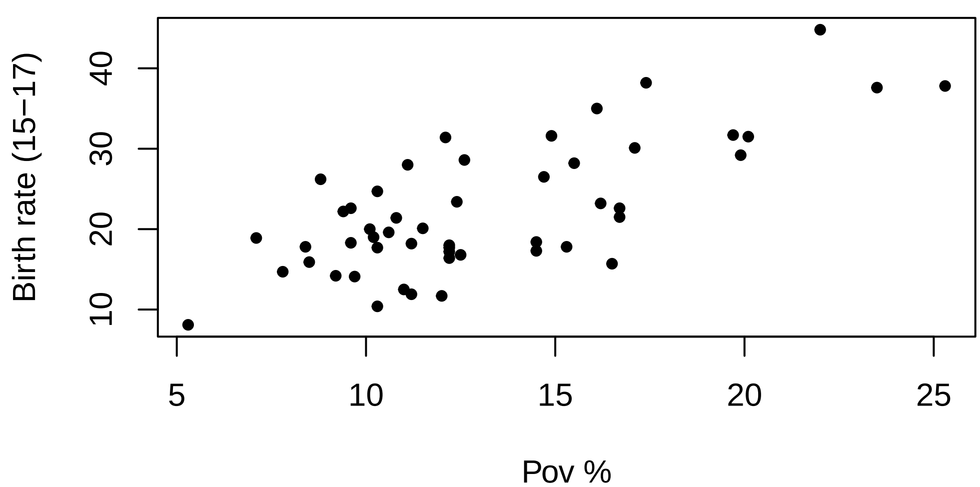
Creating a basic scatterplot that displays poverty rate vs birth rate

```r
url <- "https://online.stat.psu.edu/stat462/sites/onlinecourses.science.psu.edu.stat462/fi

df <- read_tsv(url)
```

```
Rows: 51 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr (1): Location
dbl (5): PovPct, Brth15to17, Brth18to19, ViolCrime, TeenBrth
```

```
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

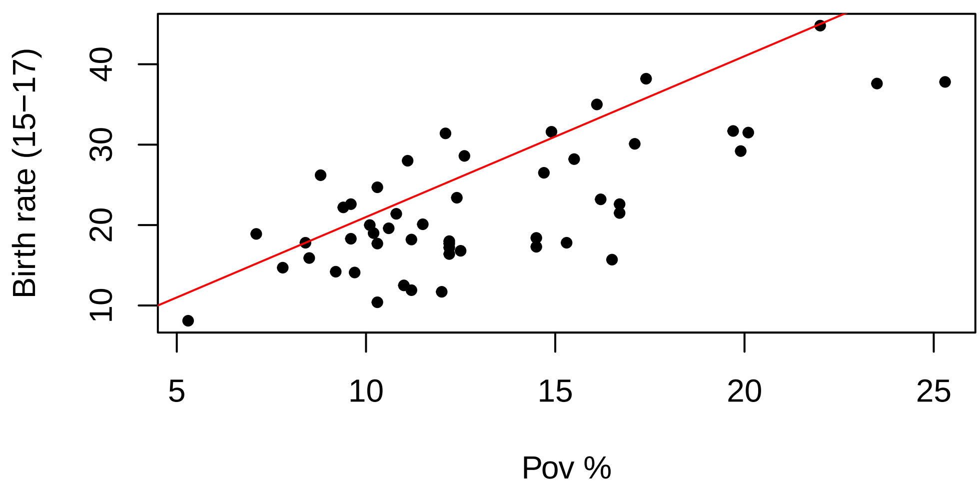
```
colnames(df) <- tolower(colnames(df))
x <- df$povpct
y <- df$brth15to17

# Scatterplot -> visualize relationship between x and y variables
plt <- function(){
  plot(
  x,
  y,
  pch = 20,
  xlab = "Pov %",
  ylab = "Birth rate (15-17)"
)
}
plt()
```



- Using beta0 and beta 1, described below, we can create a regression line on the model as shown below

1. beta0 = intercept -> the value of the y coordinate when x = 0
2. beta1 = slope -> for every 1 increase in x, y increases/decreases by beta1

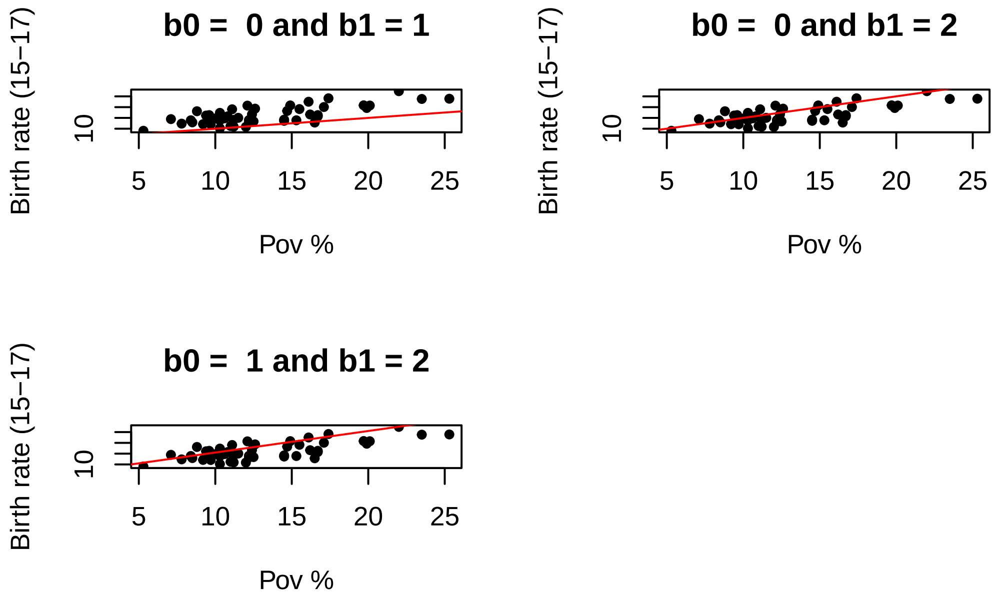
```
# lines through the points
b0 <- 1
b1 <- 2
plt()
curve(b0 + b1 * x, 0, 30, add = T, col = 'red')
```



Can specify various beta values to use in model so you can find the beta values that result in a regression line that best fits the data

```
b0 <- c(0, 1)
b1 <- c(1, 2)
par(mfrow = c(2,2))
for (b0 in b0) {
  for (b1 in b1) {
    plt()
    curve(b0 + b1 * x, 0, 30, add = T, col = 'red')
    title(main = paste("b0 = ", b0, "and b1 =", b1))
  }
```

```
}
```

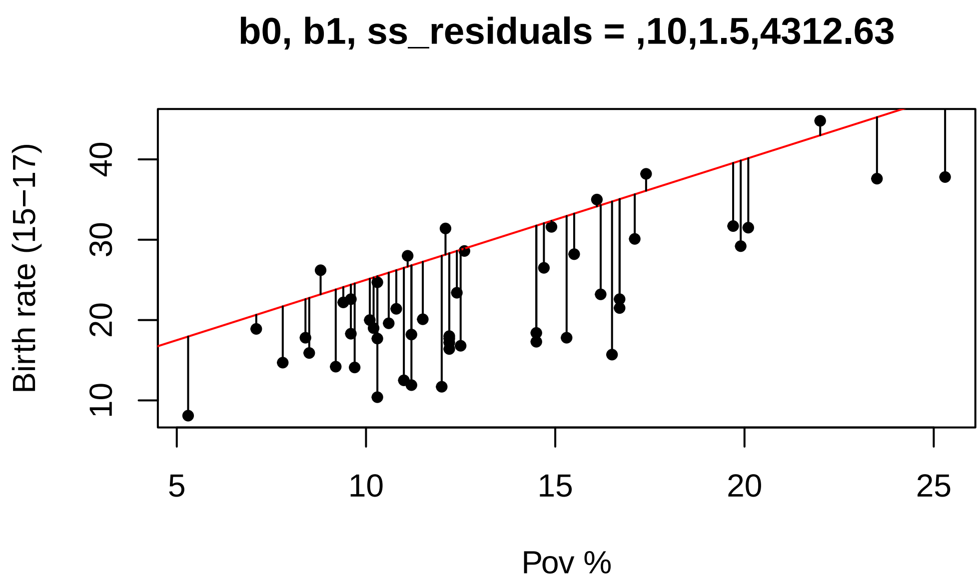


Least sqaures estimator/residuals

- characterizes error in each data point by calculating the distance/projection from a point onto the regression line
- error value = y - y(hat)
- want a line that results in the least amount of total distance from each point to the regression line

```
# can change around beta values to get different slopes/intercepts
b0 <- 10
b1 <- 1.5

yhat <- b0 + b1 * x

plt()
curve(b0 + b1 * x, 0, 30, add = T, col = 'red')
segments(x,y,x,yhat)

resids <- abs(y - yhat)^2
```

```r
ss_resids <- sum(resids)
title(main = paste("b0, b1, ss_residuals = ",b0, b1, ss_resids, sep = ","))
```



Thursday, Feb 2

> **!** TIL
>
> Include a *very brief* summary of what you learnt in this class here.
> Today, I learnt the following concepts in class:
>
> 1. Using and interpreting linear model summaries (using lm() function)
> 2. The null/alternate hypothesis
> 3. How to make a prediction using a graph and place that prediction on the graph

Linear model function/summary

Creating models for 2 different functions

```r
model <- lm(y ~ x)
model
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      4.267        1.373
```

```r
summary(model)
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-11.2275  -3.6554  -0.0407   2.4972  10.5152

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.2673     2.5297   1.687    0.098 .
x             1.3733     0.1835   7.483 1.19e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.551 on 49 degrees of freedom
Multiple R-squared:  0.5333,	Adjusted R-squared:  0.5238
F-statistic:    56 on 1 and 49 DF,  p-value: 1.188e-09
```

```r
x2 <- x^2
model2 <- lm(y ~ x + x2)
model2
```

```
Call:
lm(formula = y ~ x + x2)

Coefficients:
(Intercept)            x           x2
   10.60211      0.43733      0.03128
```

```r
summary(model2)
```

```
Call:
lm(formula = y ~ x + x2)

Residuals:
     Min       1Q   Median       3Q      Max
-10.6341  -3.9590  -0.5538   3.0886  10.9265

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.60211    7.28188   1.456    0.152
x            0.43733    1.02534   0.427    0.672
x2           0.03128    0.03371   0.928    0.358

Residual standard error: 5.558 on 48 degrees of freedom
Multiple R-squared:  0.5416,    Adjusted R-squared:  0.5224
F-statistic: 28.35 on 2 and 48 DF,  p-value: 7.43e-09
```

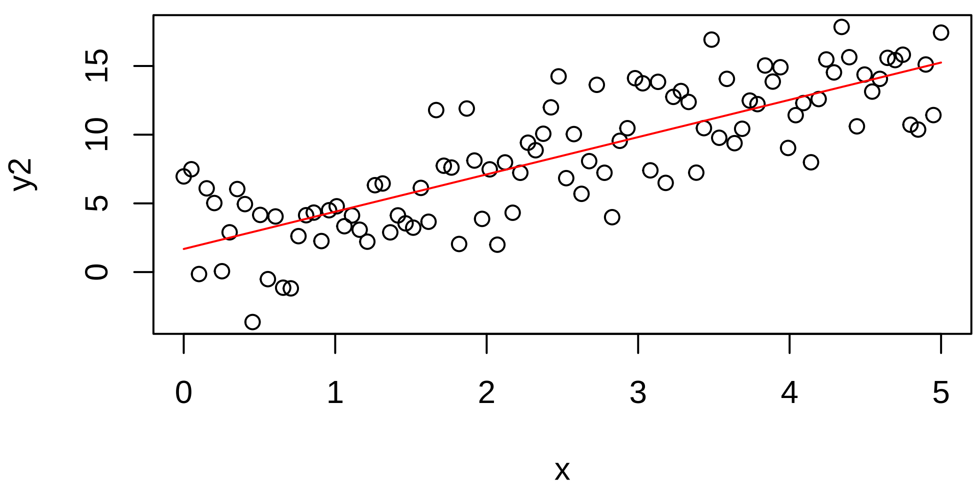
- The R^2 value tells you if the line is a good fit for the plot (if the R^2 value is close to 1, that means the regression line is a very good/perfect fit, and vice versa)
- The *p*-value informs you about how good a certain variable is at predicting the outcome. Is also important in coming to a conclusion about hypotheses (discussed in next section).
- Provides you with the intercept of the line, as well as the standard error, each variables coefficient and other residual values.

Null/alternate hypotheses

- Null model/hypothesis
  1. There is no linear relationship between x and y
     - This means that in terms of β_0 and β_1, that $\beta_0 = 0$ in null hypothesis (H_0)
     - The alternate hypothesis is that $\beta_1 = 0$
- *p*-value helps determine whether we should accept or reject the null hypothesis
  1. When we see a small *p*-value, then we reject the null hypothesis in favor of the alternate hypothesis.
     - This means that **there is a significant relationship between** x **and** y or in more mathematical terms,
     - There is significant evidence in favor of a correlation between x and y.
  2. When we see a large *p*-value, we accept the null hypothesis

The code chunk below is creating a plot and adding a regression line

```r
x <- seq(0,5, length=100)

b0 <- 1
b1 <- 3

y2 <- b0 + b1 * x + rnorm(100) * 3

plot(x,y2)

model2 <- lm(y2 ~ x)

plot(x, y2)
curve(coef(model2)[1] + coef(model2)[2] * x, add = T, col = 'red')
```



By using the summary for the model, which is shown below, we can look at the p-value to help determine to accept/reject the null hypothesis

```r
summary(model2)
```

```
Call:
lm(formula = y2 ~ x)
```

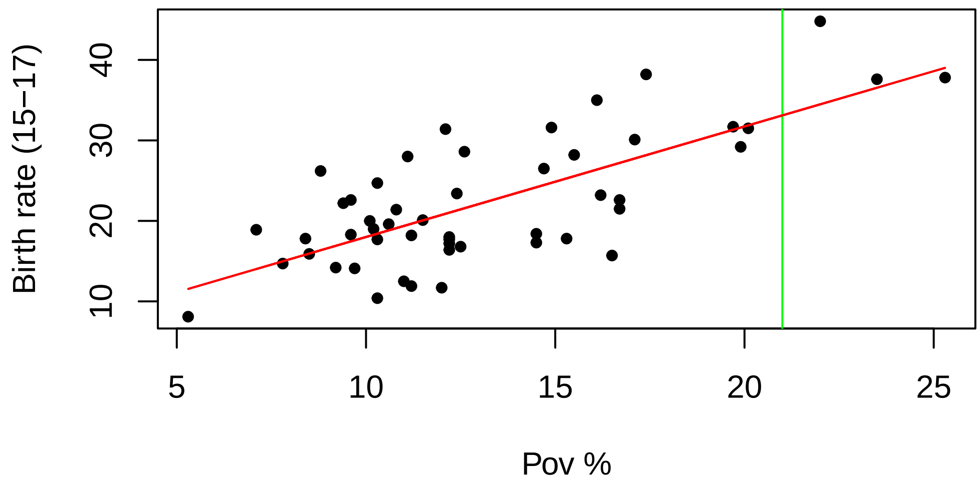
```
Residuals:
    Min      1Q  Median      3Q     Max 
-6.5524 -2.1243  0.2633  1.7268  5.8452 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    1.682      0.573   2.936  0.00415 ** 
x              2.714      0.198  13.706  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.887 on 98 degrees of freedom
Multiple R-squared:  0.6572,    Adjusted R-squared:  0.6537 
F-statistic: 187.8 on 1 and 98 DF,  p-value: < 2.2e-16
```

- Because the p-value is so small, we reject the null hypothesis in favor of the alternate hypothesis. This indicates that there is a relationship between the variables

Prediction

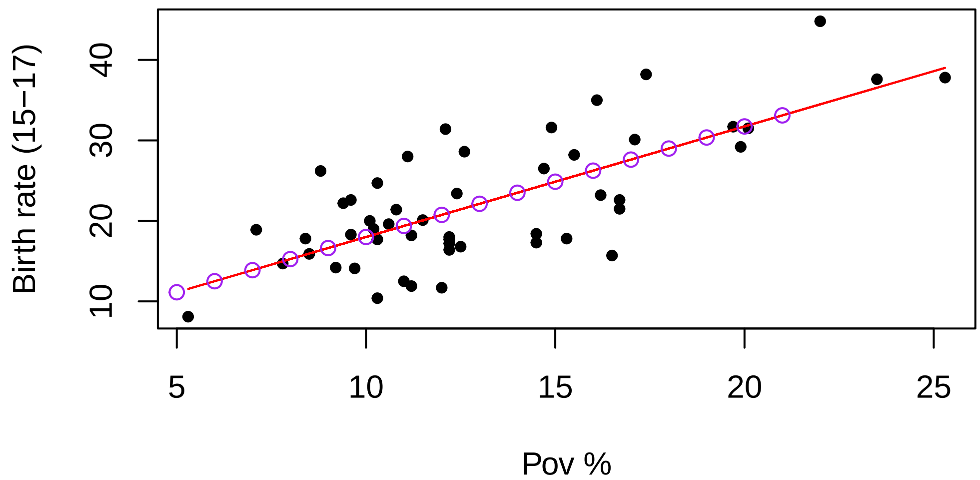
- Adding a line where the poverty rate is 21% to give an indication of what the birth rate might be at this point

```r
x <- df$povpct
y <- df$brth15to17

plt()
abline(v = 21, col = 'green')
lines(x, fitted(lm(y ~ x)), col = 'red')
```



Predicting birth rates for various poverty rates and plotting them on the line

```
new_x <- data.frame(x = c(1:21))
new_y <- predict(model, new_x)

plt()
for (a in new_x) {
abline(v = list(), col = 'green')}
lines(x, fitted(lm(y ~ x)), col = 'red')
points(new_x %>% unlist(), new_y, col = 'purple')
```


Birth rate (15–17)
10
20
30
40
5
10
15
20
25
Pov %