Weekly Summary Template

Advait Ashtikar

Table of contents

Tuesday, Jan 30

! TIL

Include a *very brief* summary of what you learnt in this class here.
Today, I learnt the following concepts in class:

1. Intro to Statistical Learning
2. Simple Linear Regression
    1. Motivation
    2. ℓ_2 estimator
    3. Inference
    4. Prediction

Loading Libraries

```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(ISLR2)
library(cowplot)
library(kableExtra)
```

```
Attaching package: 'kableExtra'

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

    group_rows
```

```r
library(htmlwidgets)
```

Statistical Learning

Suppose we have a data set

$\mathbf{X} = [X_1\ , X_2, ... \ X_n]$

- These are called predictor/independent variables

$\mathbf{Y}$

- Th

The goal of statistical learning is to find a function f such that $y = f(x)$

Different flavors: Statistical learning

- Supervised learning (Both y and x)
  - Regression
  - Classification
- Unsupervised learning (There is no y; much harder)
- Semi-supervised learning (The case when you have y but x is something else)
- Reinforcement learning (Corresponds to a case where the model is thought to do the work)

```
## URL for the dataset:
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.
```

```
df %>% head(., 10) %>% knitr::kable()
```

Location	PovPct	Brth15to17	Brth18to19	ViolCrime	TeenBrth
Alabama	20.1	31.5	88.7	11.2	54.5
Alaska	7.1	18.9	73.7	9.1	39.5
Arizona	16.1	35.0	102.5	10.4	61.2
Arkansas	14.9	31.6	101.7	10.4	59.9
California	16.7	22.6	69.1	11.2	41.1
Colorado	8.8	26.2	79.1	5.8	47.0
Connecticut	9.7	14.1	45.1	4.6	25.8
Delaware	10.3	24.7	77.8	3.5	46.3
District_of_Columbia	22.0	44.8	101.5	65.0	69.1
Florida	16.2	23.2	78.4	7.3	44.5

Goal

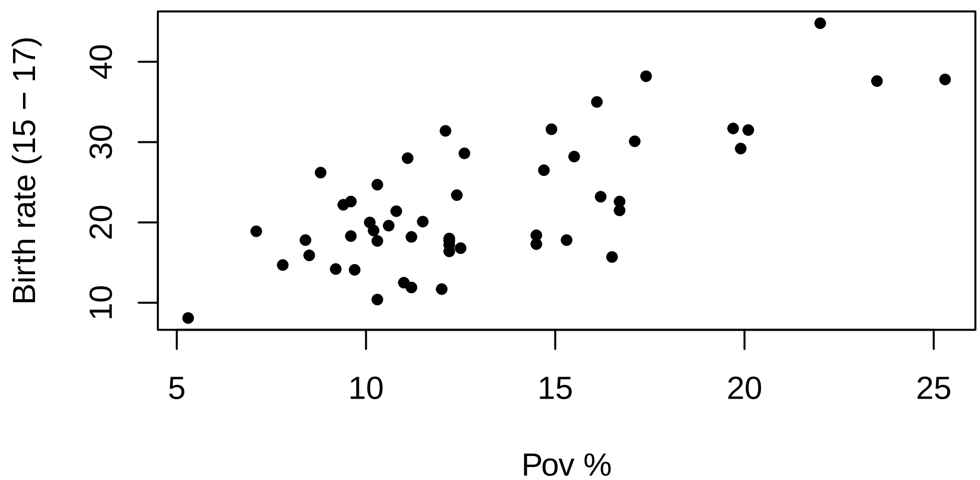
Prdict the birth rate as a function of the poverty rate

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

Scatterplot

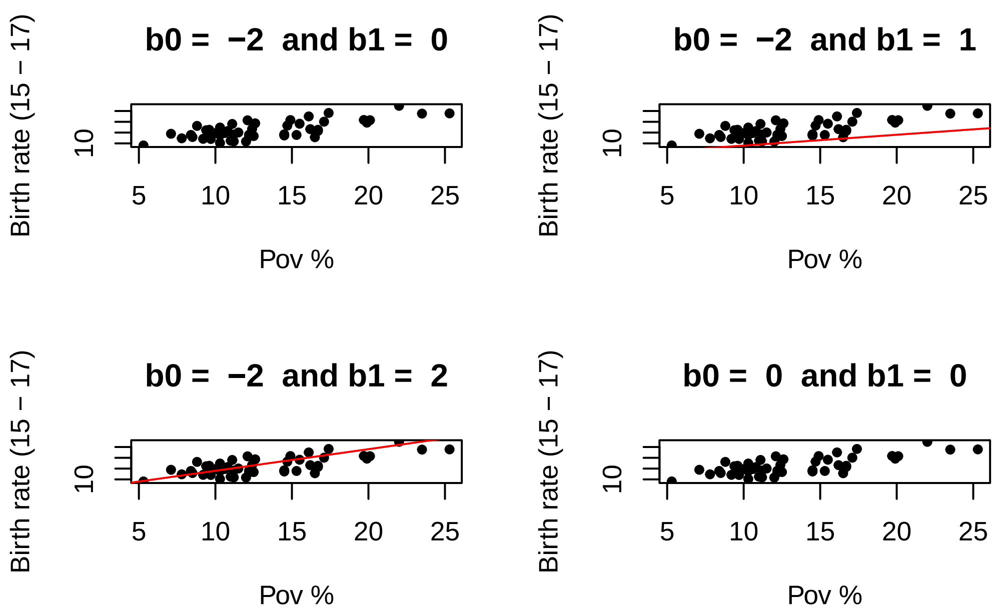
Visualize the relationship between the x and y variables

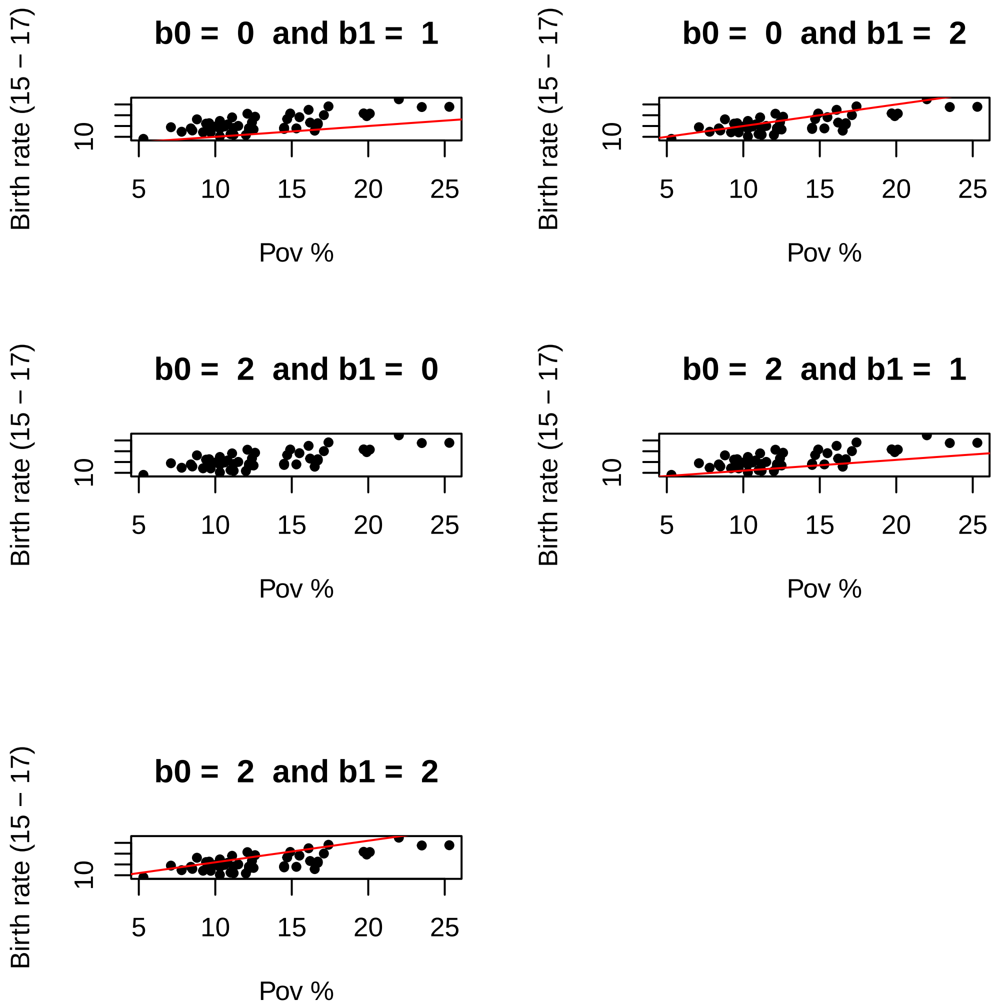
```
plt <- function(){
 plot(
  x,
  y,
  pch=20,
  xlab = "Pov %",
  ylab = "Birth rate (15 - 17)"
)
}
plt()
```



Lines through the points

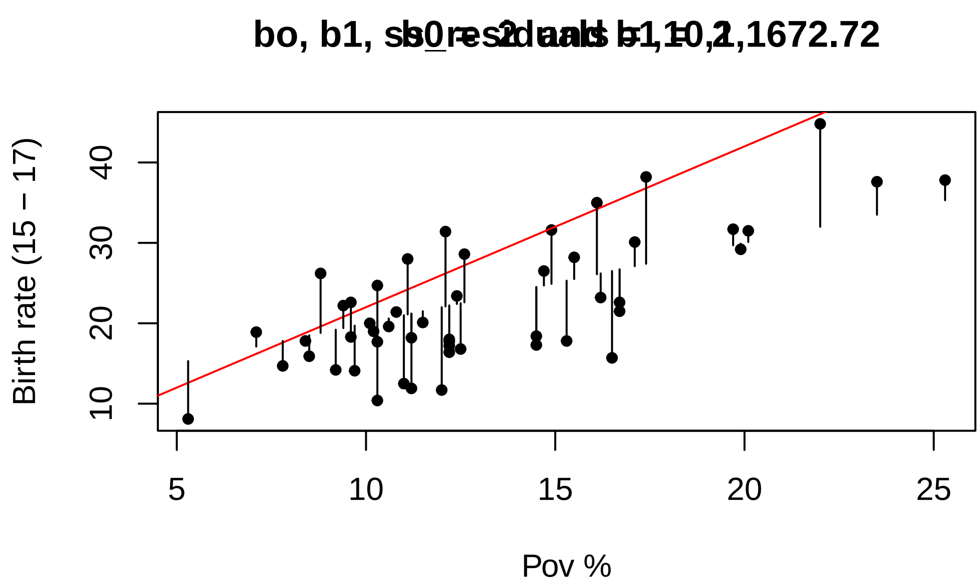
```
b0 <- c(-2, 0, 2)
b1 <- c(0, 1, 2)

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))
  }
}
```




b0 = 0 and b1 = 1
b0 = 0 and b1 = 2
b0 = 2 and b1 = 0
b0 = 2 and b1 = 1
b0 = 2 and b1 = 2
Birth rate (15 – 17)
10
5
10
15
20
25
Pov %

Least squares estimator

```r
b0 <- 10
b1 <- 1

yhat <- b0 + b1 * x

plt()
curve( B0 + B1 * x, 0, 30, add=T, col="red")
title(main = paste("b0 = ", B0," and b1 = ",B1))
segments(x, y, x, yhat)

resids <- abs(y - yhat)^2
ss_resids <- sum(resids)
title(main = paste("bo, b1, ss_residuals = ", b0, b1, ss_resids, sep = ","))
```



The best fit line minimizes residuals

```r
model <- lm(y ~ x)
sum(residuals(model)^2)
```

```
[1] 1509.635
```

```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
```

The summary for the model contains the optimal slope.

Thursday, Jan 19

> **! TIL**
>
> Include a *very brief* summary of what you learnt in this class here.
> Today, I learnt the following concepts in class:
>
> 1. Linear Regression
> 2. Multiple Regression
>   1. Extension from simple linear regression

Model Formulae

In our case we want to model y as a function of x. In 'R' the formula for this looks like:

```r
typeof(formula(y~x))
```

```
[1] "language"
```

A linear regression model in 'R' is called using the **L**inear **M**odel, i.e., 'lm()'

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

Q. What are the null and alternate hypotheses for a regression model?

Objective: We want to find the best linear model to fit $y \sin x$

Null Hypotheses: There is no linear relationship between y and x.

- What does this mean in terms of β_0 and β_1

Alternate Hypotheses: $\beta_1 \neq 0$

To summarize

$$H_0 : \beta_1 = 0 \qquad\qquad H_1 : \beta_1 \neq 0 \tag{1}$$

When we see a small p-value, then we reject the null hypothesis in favor of the alternate hypothesis. What is the implication of this w.r.t. the original model objective?

> ** There is a significant relationship between y and x. Or, in more mathematical terms, there is significant evidence in favor of a correlation between x and y **

This is what the p-value in the model output are capturing. We can also use the 'kable' function to print the results nicely:

```r
library(broom)

summary(model) %>%
  broom::tidy() %>%
  knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	4.267293	2.529747	1.686846	0.0979904
x	1.373345	0.183523	7.483234	0.0000000

Regression Models

1. Independent variable x

```r
head(x)
```

```
[1] 20.1  7.1 16.1 14.9 16.7  8.8
```

2. Response y

```r
head(y)
```

```
[1] 31.5 18.9 35.0 31.6 22.6 26.2
```

3. Fitted values $\hat{y}$

```r
yhat <- fitted(model)
head(yhat)
```

```
       1        2        3        4        5        6 
31.87154 14.01805 26.37815 24.73014 27.20216 16.35273 
```

4. Residuals: $e = y - \hat{y}$

```r
res <- residuals(model)
head(res)
```

```
         1          2          3          4          5          6 
-0.3715352  4.8819549  8.6218464  6.8698609 -4.6021608  9.8472677 
```

Some other important terms are the following:

1. Sum of squares for residuals:

$SS_{Res} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i^2)$

2. Sum of squares for regression:

$SS_{Reg} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

3. Sum of squares total:

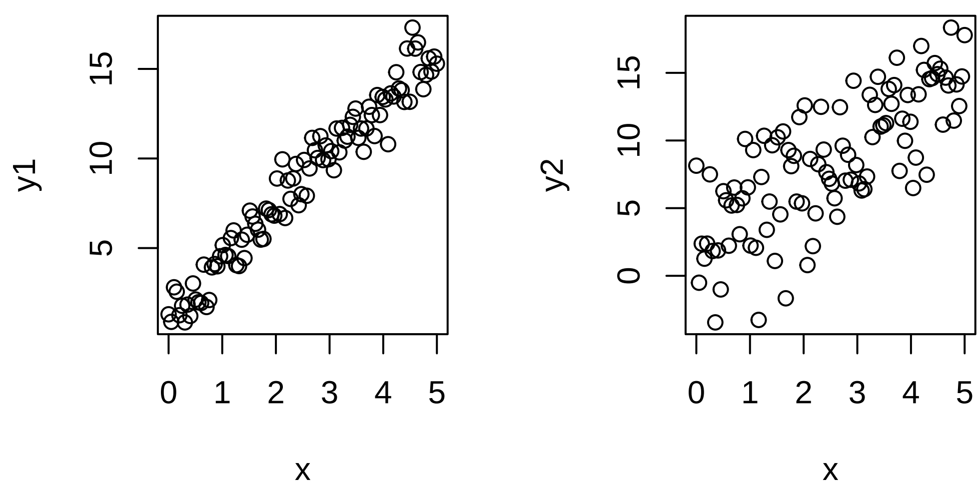
$SS_{Tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2$

Another important summary in the model output is the R^2 value, which is given as follow:

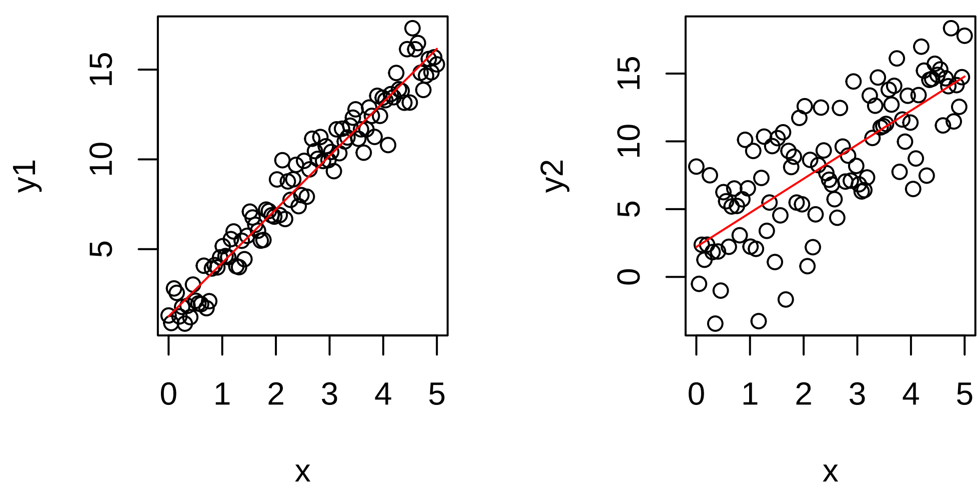
$$R^2 = \frac{SS_{Reg}}{SS_{Tot}}$$

Lets have a look at what this means in the following example.

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

b0 <- 1
b1 <- 3

y1 <- b0 + b1 * x + rnorm(100)
y2 <- b0 + b1 * x + rnorm(100) * 3

par(mfrow=c(1,2))

plot(x, y1)
plot(x, y2)
```



```
model1 <- lm(y1 ~ x)
model2 <- lm(y2 ~ x)

par(mfrow=c(1,2))

plot(x, y1)
curve(
  coef(model1)[1] + coef(model1)[2] * x,
  add=T, col="red"
)

plot(x, y2)
curve(
  coef(model2)[1] + coef(model2)[2] * x,
  add=T, col="red"
)
```



The summary of model 1 is:

```
summary(model1)
```

```
Call:
lm(formula = y1 ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.65341 -0.64050 -0.00907  0.75881  2.50087 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.25635    0.19318   6.504 3.32e-09 ***
x            2.98093    0.06675  44.658  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9731 on 98 degrees of freedom
Multiple R-squared:  0.9532,    Adjusted R-squared:  0.9527 
F-statistic:  1994 on 1 and 98 DF,  p-value: < 2.2e-16
```

The summary for model2:

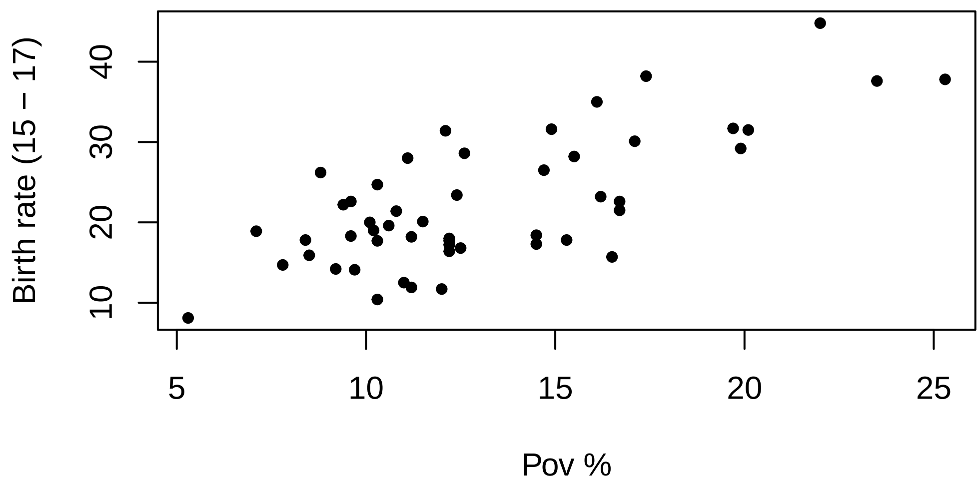
```r
summary(model2)
```

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

Residuals:
    Min      1Q  Median      3Q     Max 
-8.3955 -2.1291  0.1141  2.3341  5.9274 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.2124     0.6428   3.442 0.000851 ***
x             2.5147     0.2221  11.322  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.238 on 98 degrees of freedom
Multiple R-squared:  0.5667,    Adjusted R-squared:  0.5623 
F-statistic: 128.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

The last thing we're going to talk about in simple linear regression is **prediction**. It's the ability of a model to predict values for "unseen" data.

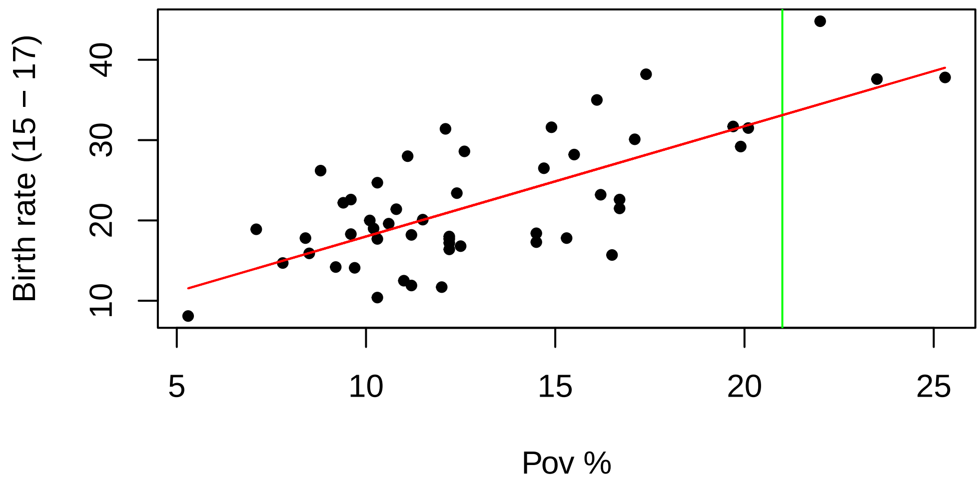
Let's go back to the poverty dataset.

```
x <- df$povpct
y <- df$brth15to17
plt()
```



Suppose we have a "new" state formed whose 'povpct' value is 22.

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



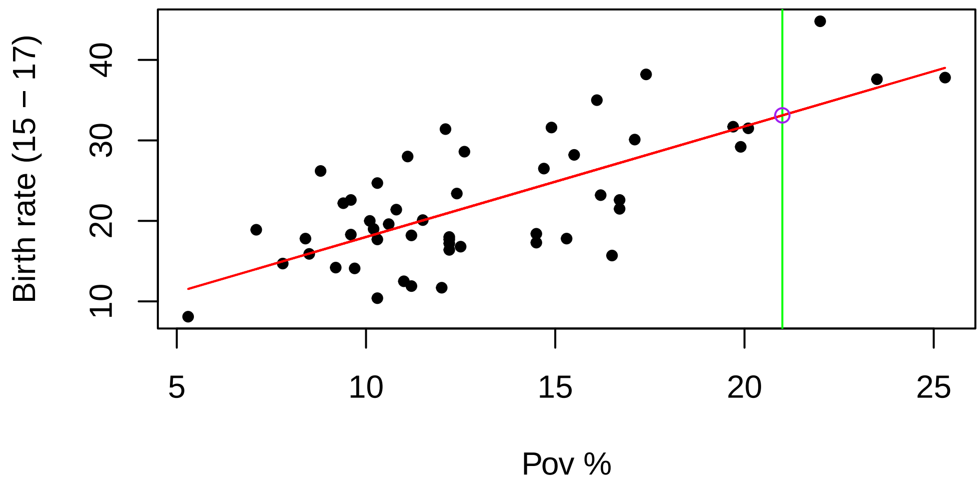
Q. What is the best guess for this prediction going to be? We could consider the graph and our best prediction is going to be the intersection. In *R*, we can use the `predict()` function to do this:

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

new_y
```

```
       1
33.10755
```

If we plot this new point we get

```
plt()
abline(v=21, col="green")
lines(x, fitted(lm(y~x)), col="red")
points(new_x, new_y, col="purple")
```



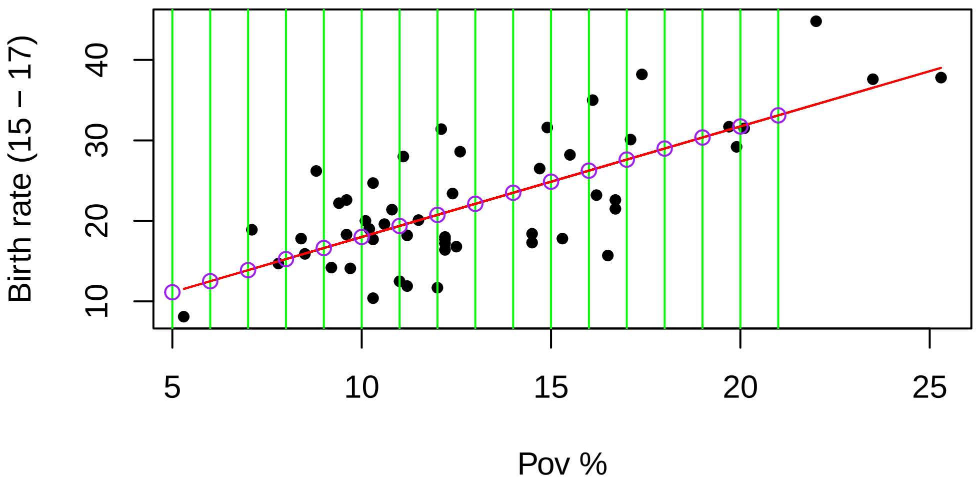
We can make predictions not just for a single observation, but for a whole collection of observations.

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

```
        1         2         3         4         5         6         7         8 
 5.640638  7.013984  8.387329  9.760674 11.134020 12.507365 13.880711 15.254056 
        9        10        11        12        13        14        15        16 
16.627401 18.000747 19.374092 20.747438 22.120783 23.494128 24.867474 26.240819 
       17        18        19        20        21 
27.614164 28.987510 30.360855 31.734201 33.107546 
```

This is what the plot looks like:

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


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