Data Analytics

CS301

Modeling: Formal Basics

Fall 2018

Oliver Bonham-Carter

Modeling Basics

- What are models?
  - Data does not provide much insight unless something can be learned from it.
  - The ability to use data to extract meaning and extra value (the learning)
- Let’s talk about...
  - How to extract some meaning from your data
  - How to make predictions using your data as training

Modeling Basics

- Topics include
  - Modeling
  - Linear regression
  - Multivariate regression
  - Interaction terms

Types of Models (i)

- **Support Vector Machines**
  - Supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
- **Generalized Linear Models**
  - Flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution
- **Generalized additive models**
  - Generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions

Types of Models (ii)

- **Linear Regression**
  - Linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X
  - *(we have begun this study)*
- **LOESS Regression**
  - Combining much of the simplicity of linear least squares regression, but building with the flexibility of nonlinear regression.
- **Logistic Regression**
  - Models where the dependent variable is categorical (i.e., 0's or 1's as factors)

Let’s Begin Our Discussion...

- Working with models begins with a basic question to answer from the analysis of data.
- We will walk through each of these with a formal discussion

Q1: Do taller people make more money?

Q2: Do hotter places have more crime?

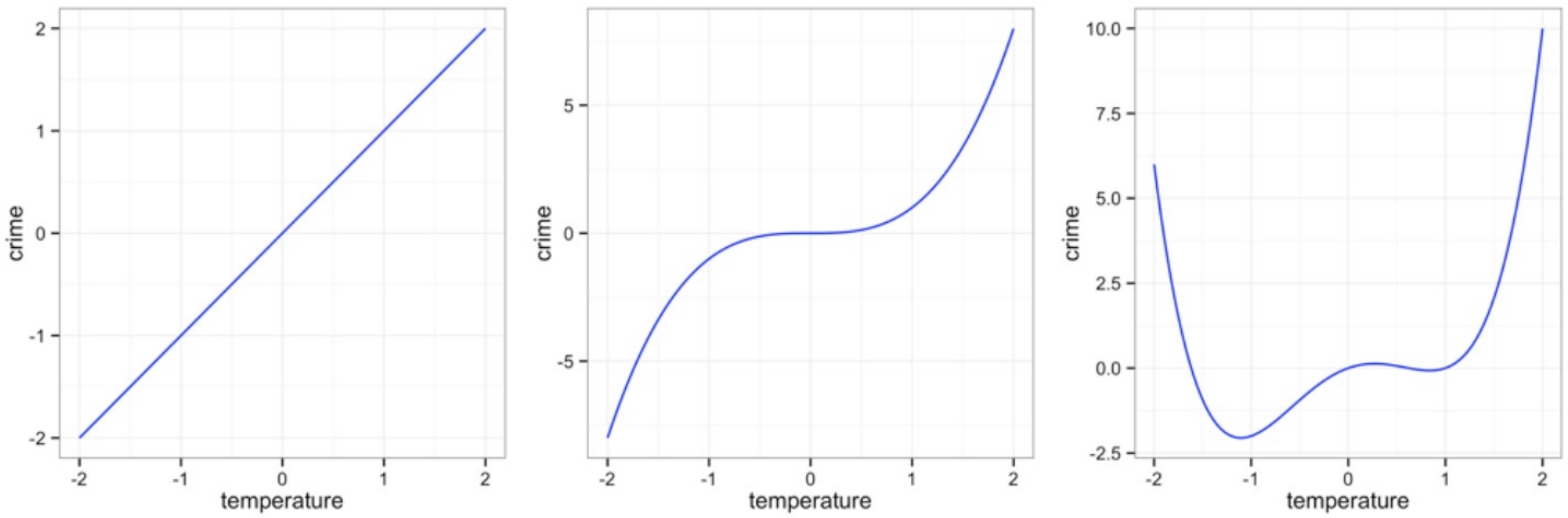
How Do we
Answer The Question?

- Modeling: We employ a computational framework which we used data to build (for training).
- Play with the model to see what happens when we change a part of the data ...

What if...

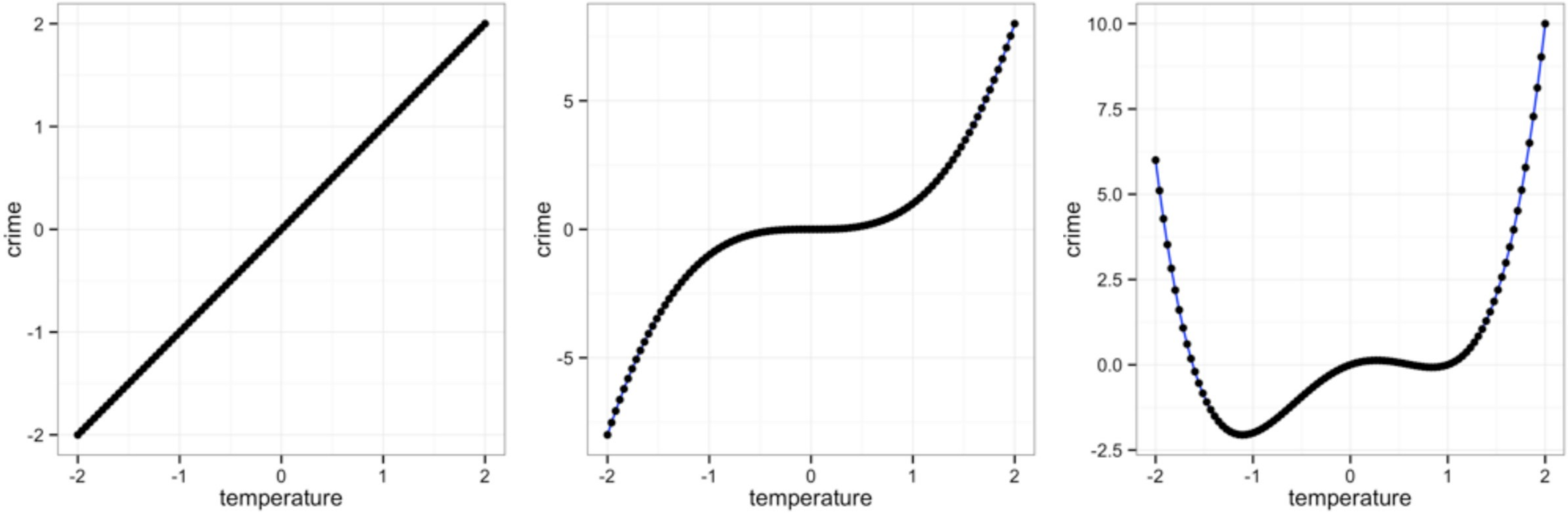
Functions:
the *stuff* behind the models

- A function is a mathematical description of a relationship.



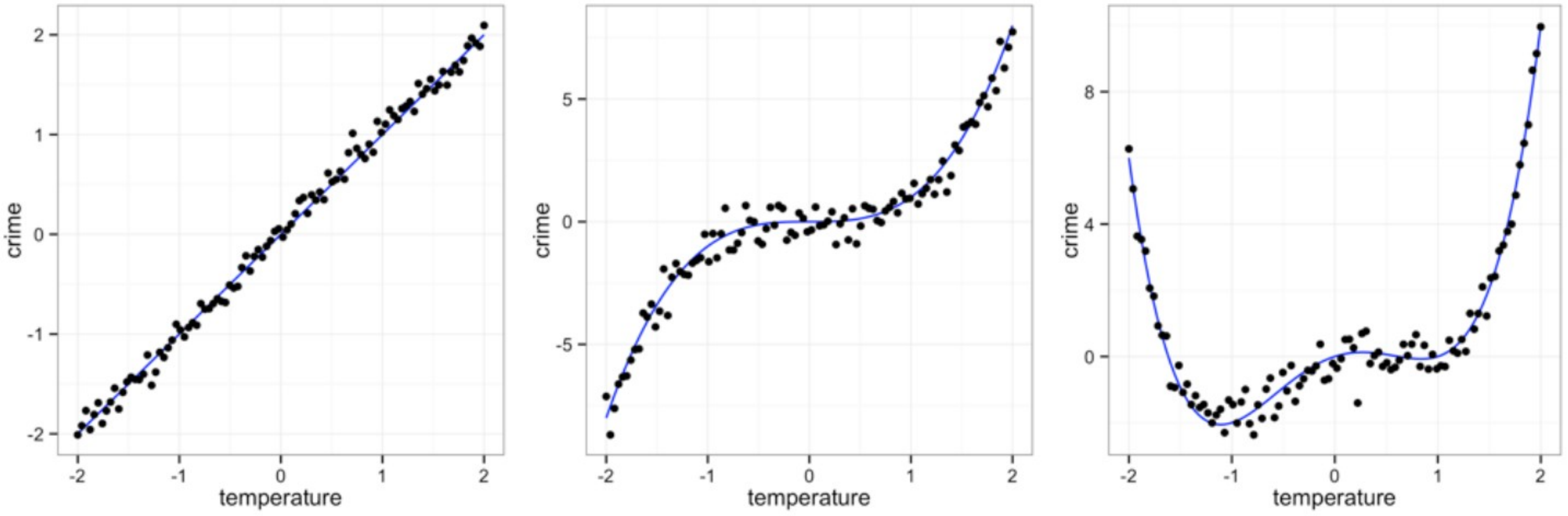
Functions:
the *stuff* behind the models

- If one variable completely determines another, every (x, y) data point will fall on the **function** line.



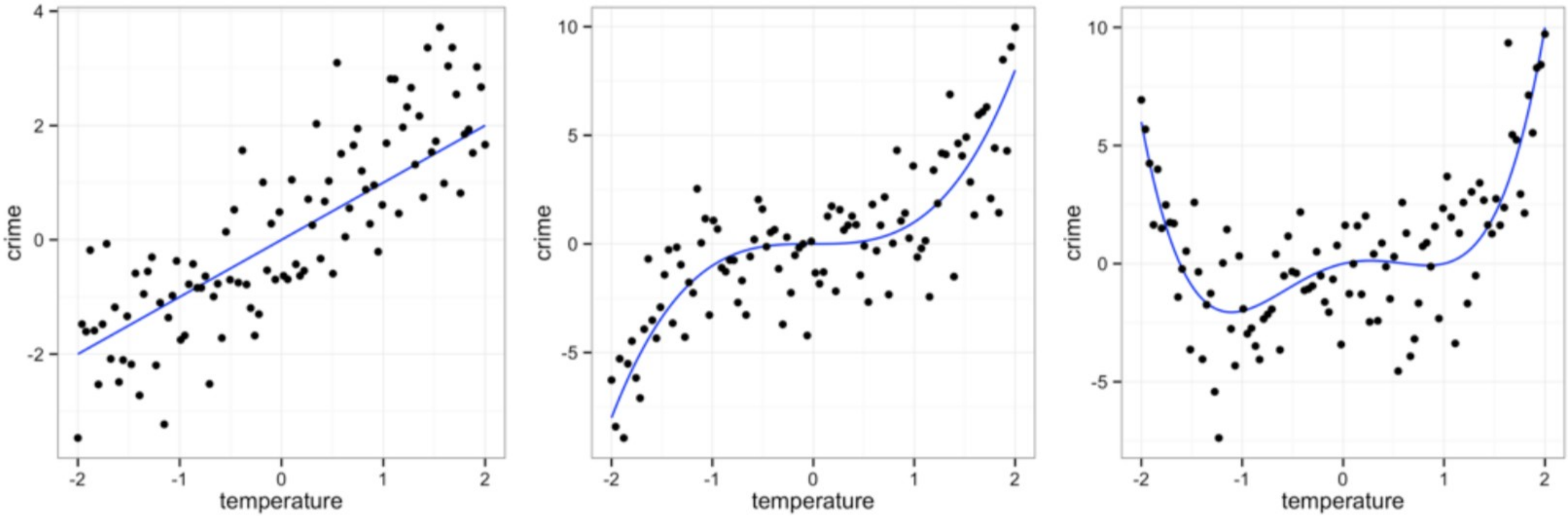
Relationships Between Variables Is Messy

- This is what real data looks like on a good day!



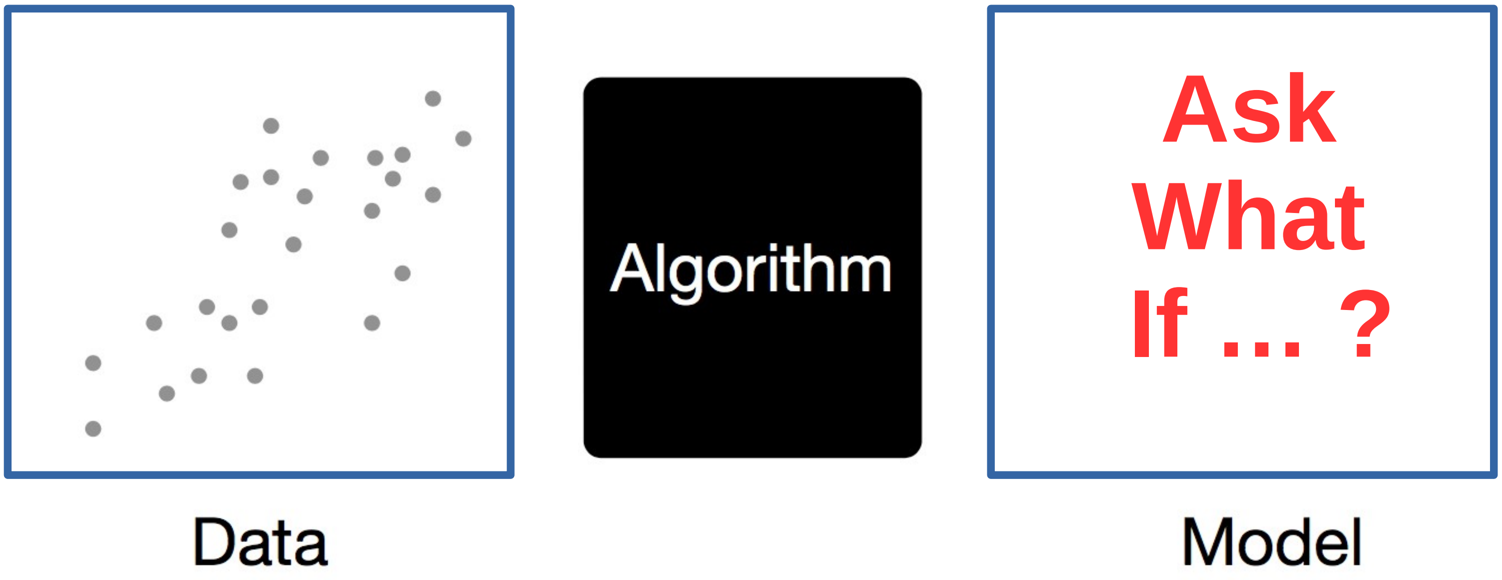
Relationships Between Variables

- If the actual relationship is affected by other variables, data points may not fall directly on the function line.
- **Noise**: The greater the effect of other variables, the weaker the relationship. This is normally the situation with real data.



So, A Model, Then?

- Noise is what we get in data when not every point does *what it is supposed to do.*
- **Modeling *attempts* to *more*-correctly identify relationships in noisy data.**

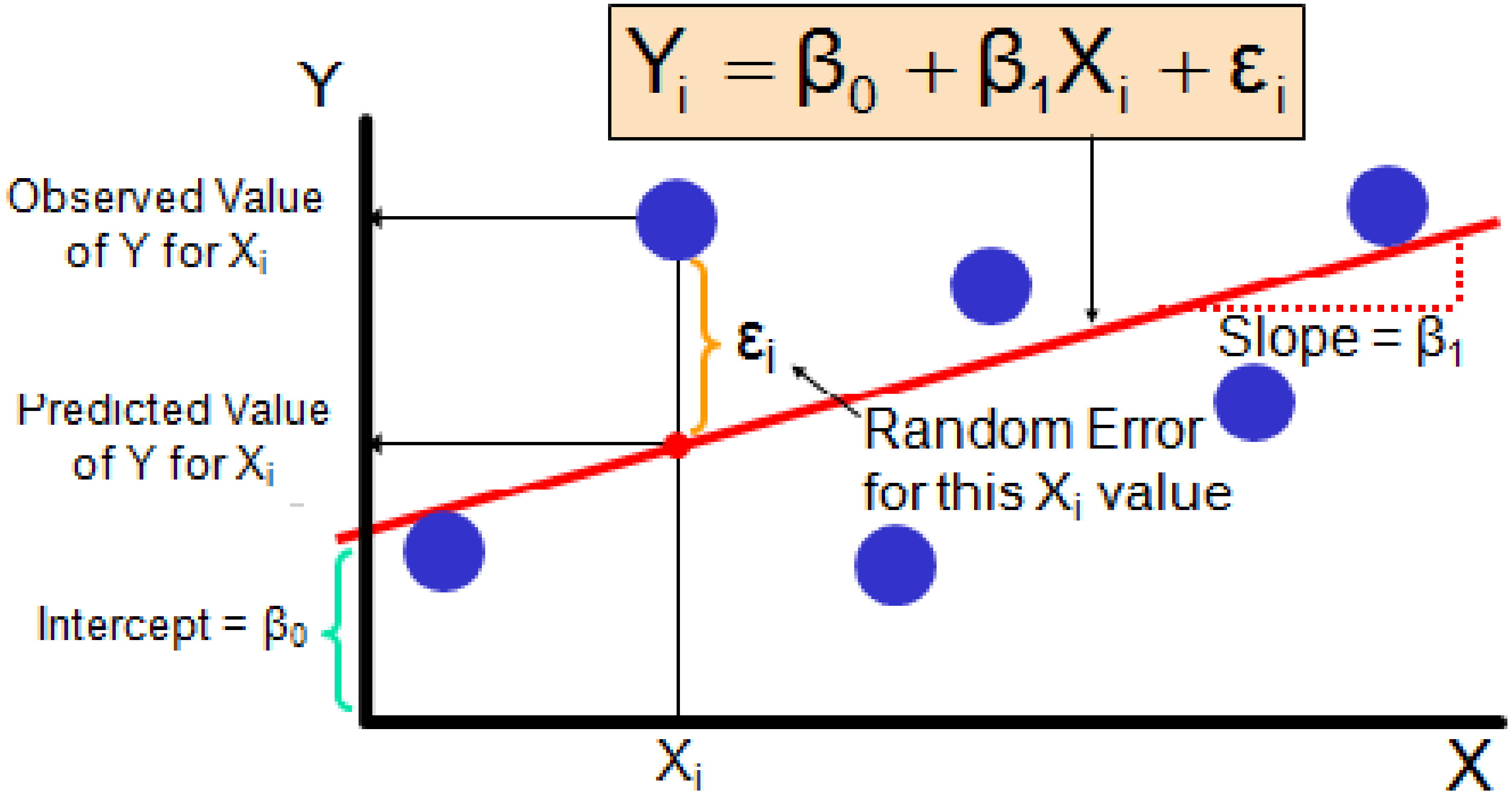


Ask What If … ?

Let’s Talk Linear Models

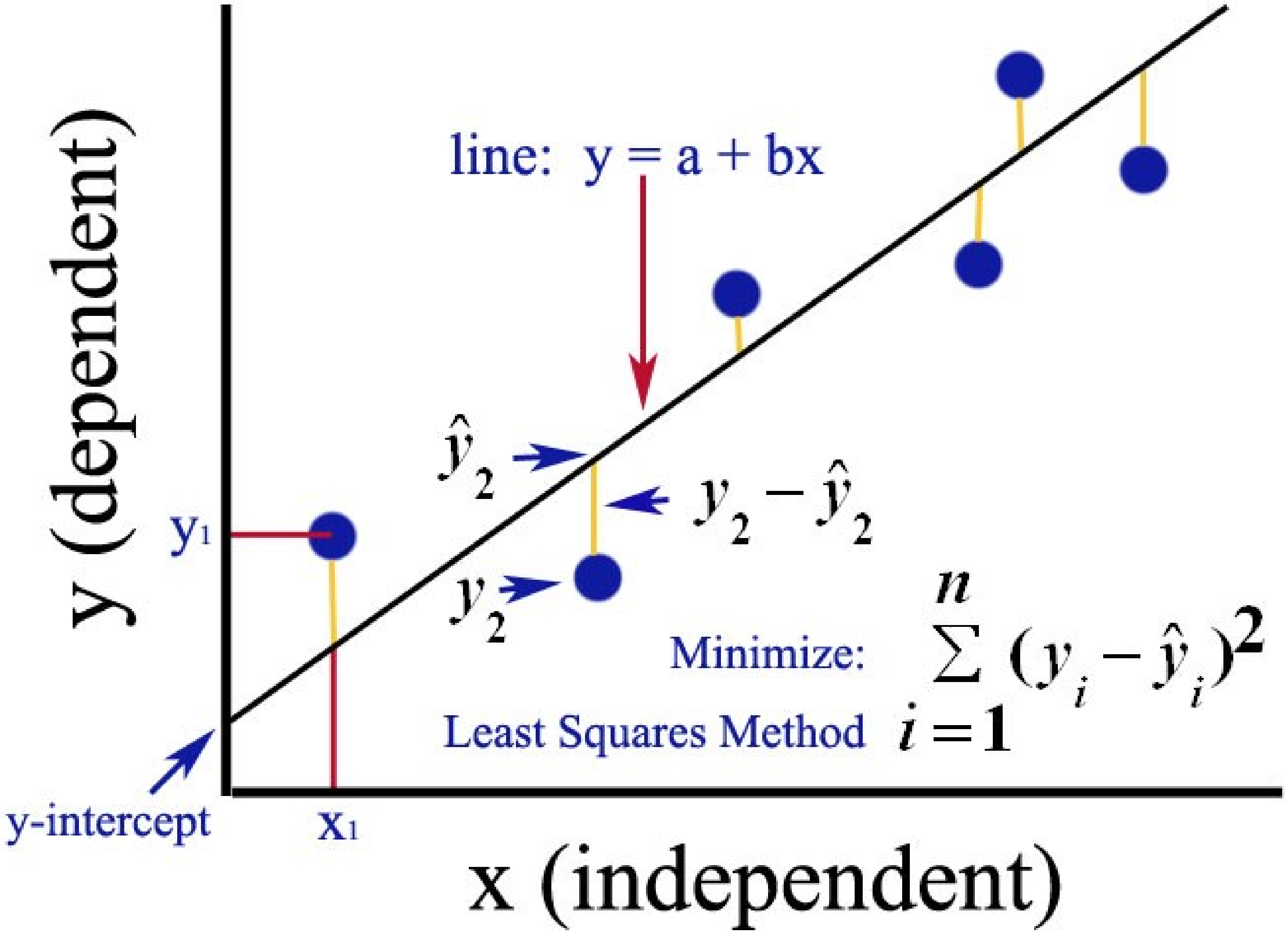
- Linear regression: How much do/does my **independent variable(s)** influence my **dependent variables**?
- As one variable climbs, does the other also climb (decline) at some *predicable* rate?
- Can I impose some value into my model to determine a ***what-if*** type of question which is firmly based on my data?

Let’s Talk Linear Models



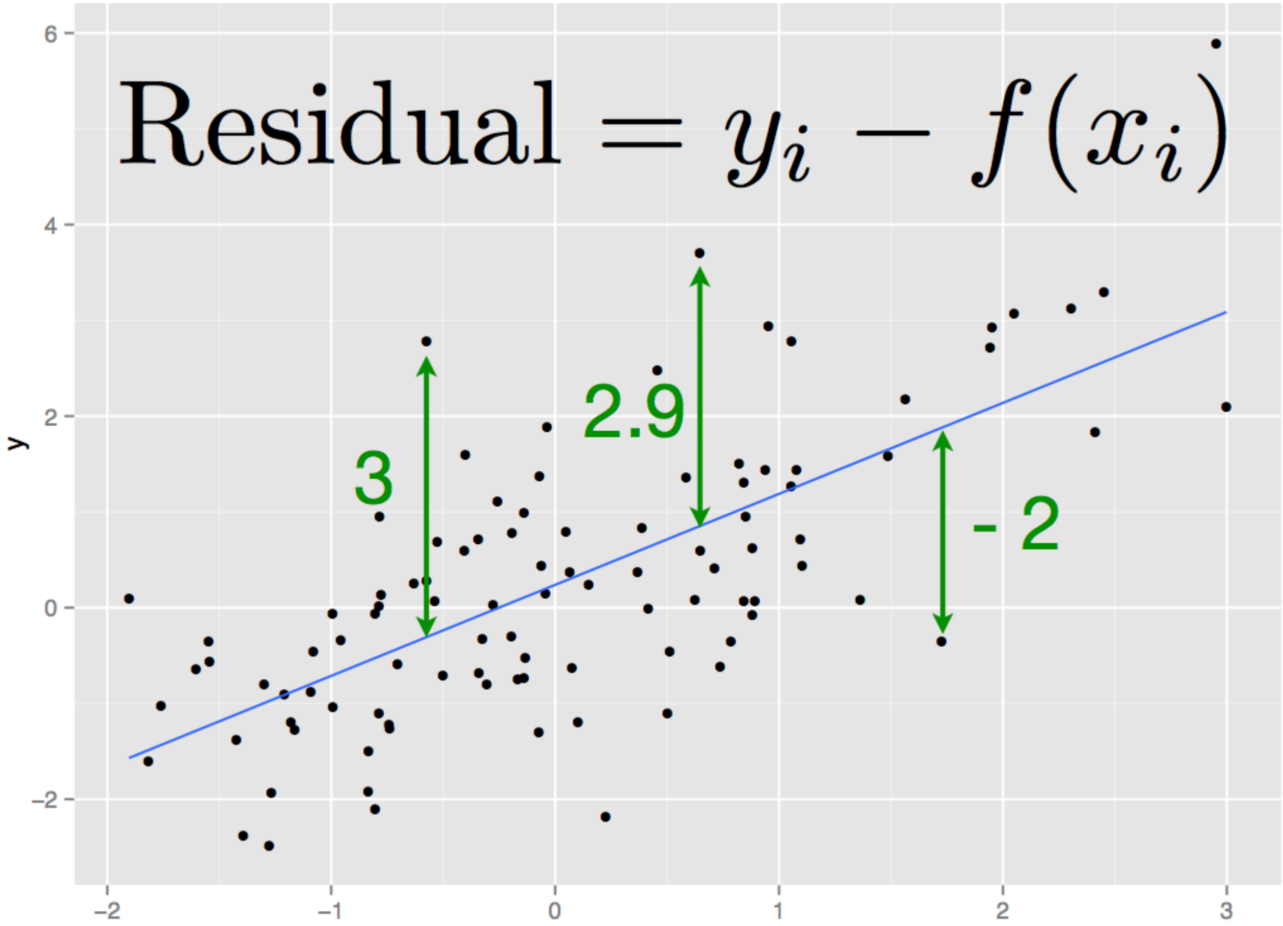
$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
Y
Observed Value of Y for X_i
ε_i
Slope = β_1
Predicted Value of Y for X_i
Random Error for this X_i value
Intercept = β_0
X_i
X

Another Linear Model



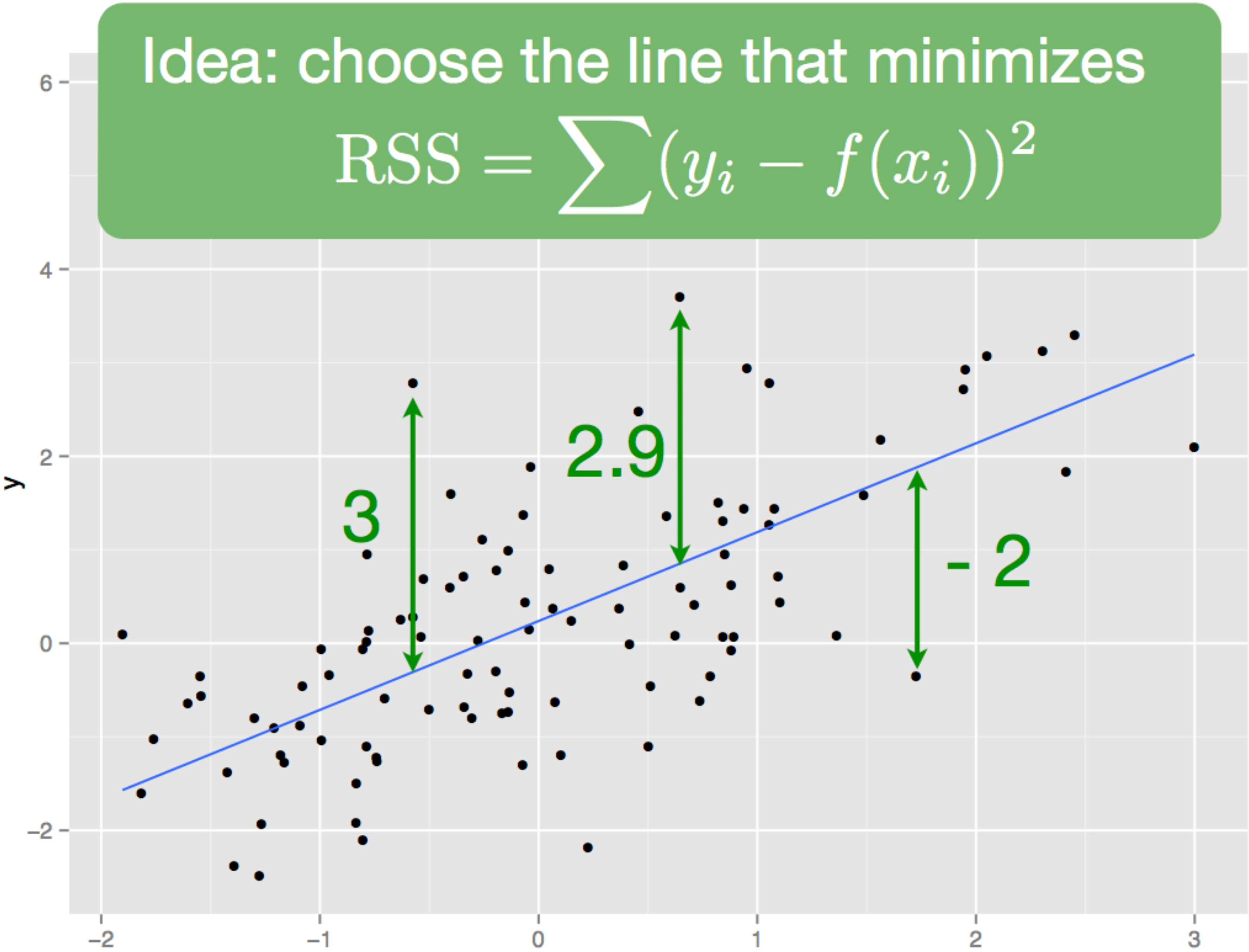
How To Best Draw a Line Through The Data?

- A *residual* of an observed value is the difference between the observed value and the estimated value of the quantity of interest



How To Best Draw a Line Through The Data?

- Residual sum of squares (RSS), also known as the sum of squared residuals (SSR) or the sum of squared errors of prediction (SSE)
- The sum of the squares of residuals (deviations predicted from actual empirical values of data).



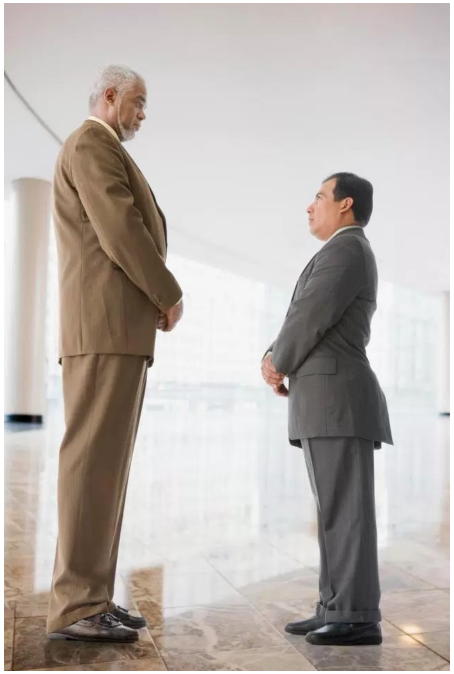
Idea: choose the line that minimizes

$$\mathrm{RSS} = \sum (y_i - f(x_i))^2$$

Types of Questions to Address With Data

Do you think that hotter places have more crime?

File: crime.csv



Do you think that taller people make more money?

File: wages.csv

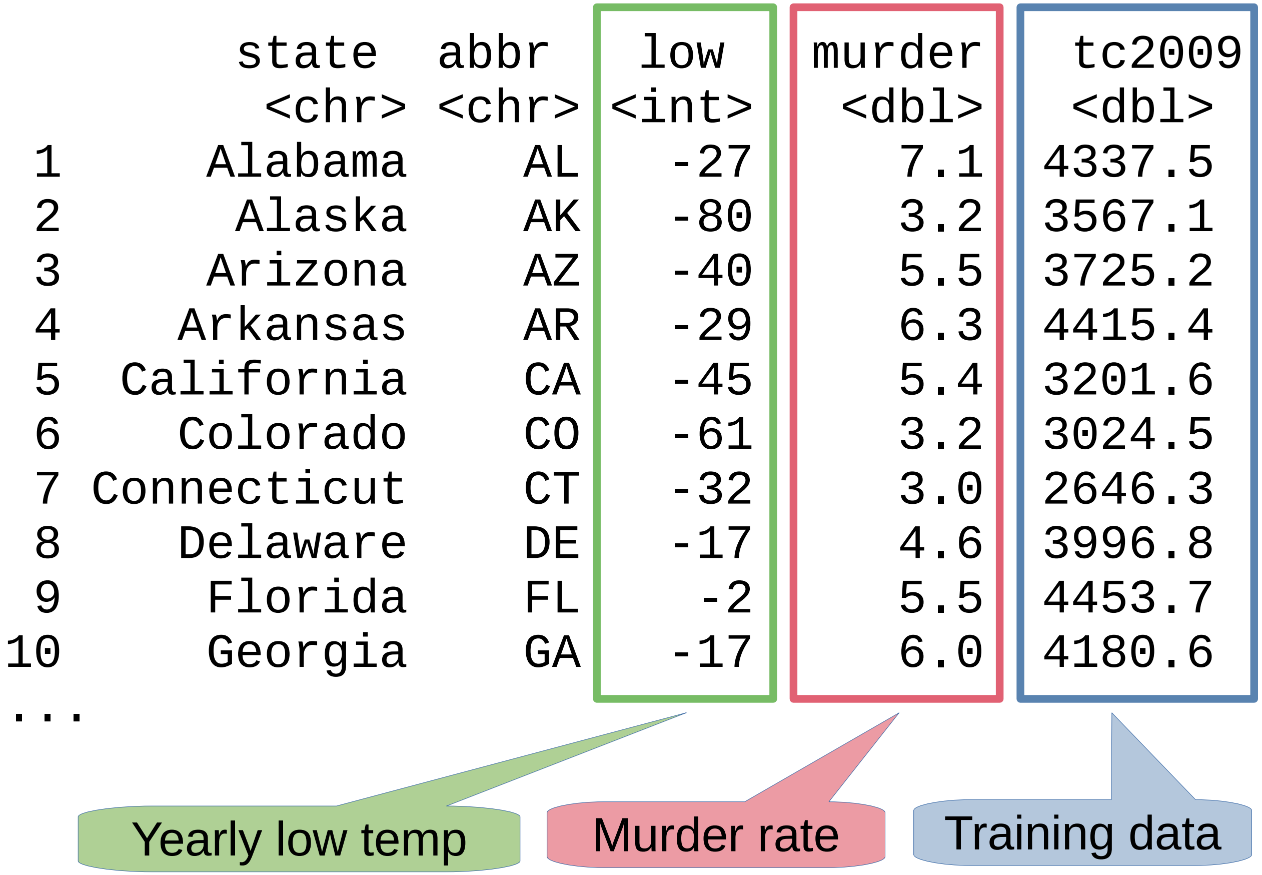
Crime Data Set

- Is there a relationship between crime and temperature? State statistics from 2009.

```
# open the crime dataset from the data.
c <- file.choose() # set the filename
crime <- read.csv(c) # load and read the data.
```

Crime Data Set

```
View(crime) #or

tbl_df(crime)
```



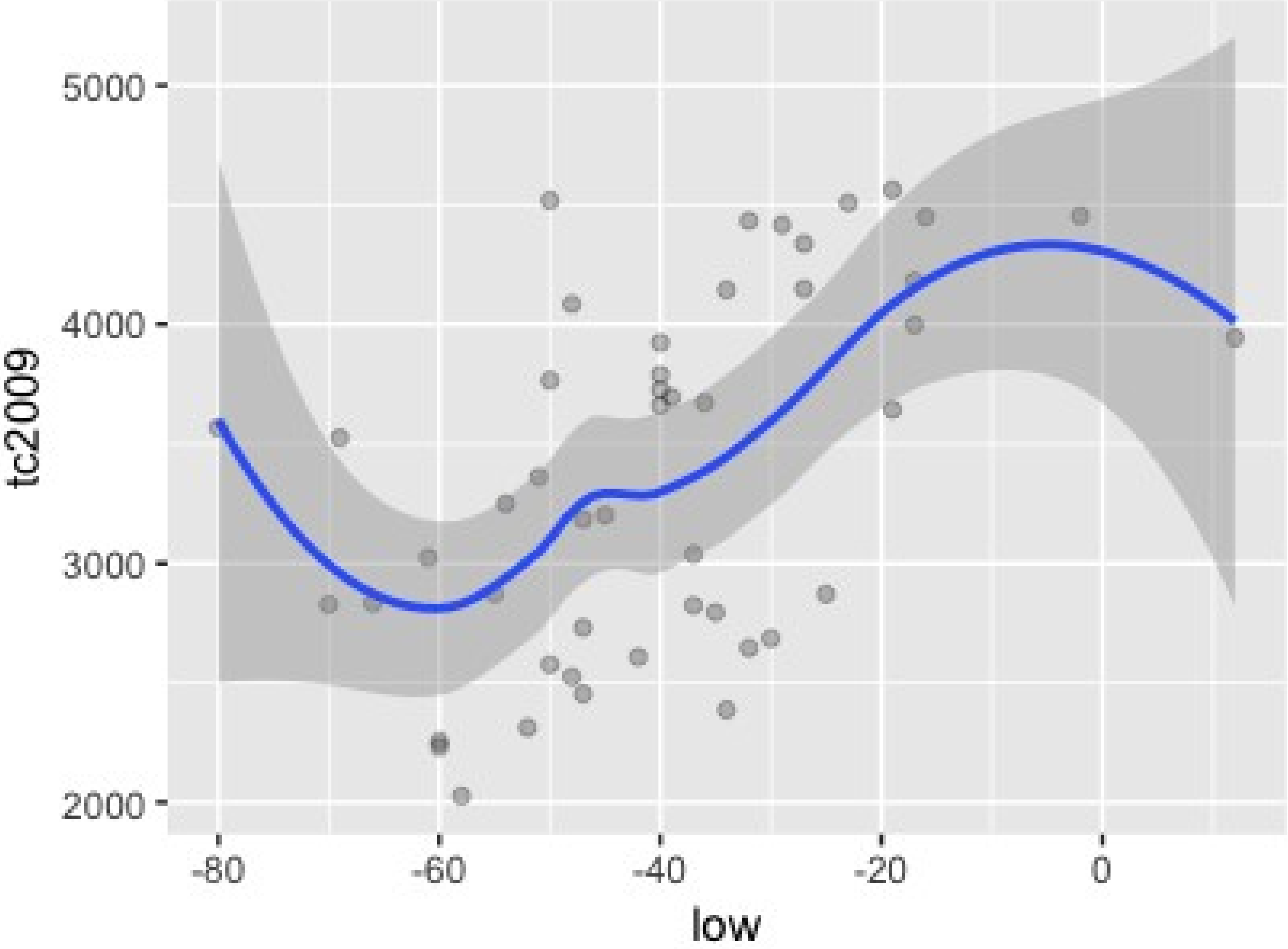
```
          state  abbr   low  murder   tc2009
          <chr> <chr> <int>   <dbl>    <dbl>
1       Alabama    AL   -27     7.1   4337.5
2        Alaska    AK   -80     3.2   3567.1
3       Arizona    AZ   -40     5.5   3725.2
4      Arkansas    AR   -29     6.3   4415.4
5    California    CA   -45     5.4   3201.6
6      Colorado    CO   -61     3.2   3024.5
7   Connecticut    CT   -32     3.0   2646.3
8      Delaware    DE   -17     4.6   3996.8
9       Florida    FL    -2     5.5   4453.7
10      Georgia    GA   -17     6.0   4180.6
...
```

Exploratory Plots

```
#plot the data

crime %>% ggplot(aes(x = low, y =
tc2009)) + geom_point(alpha = I(1/4)) +
geom_smooth()


crime %>% ggplot(aes(x = low, y =
tc2009)) + geom_point(alpha = I(1/4)) +
geom_smooth(method = lm)
```

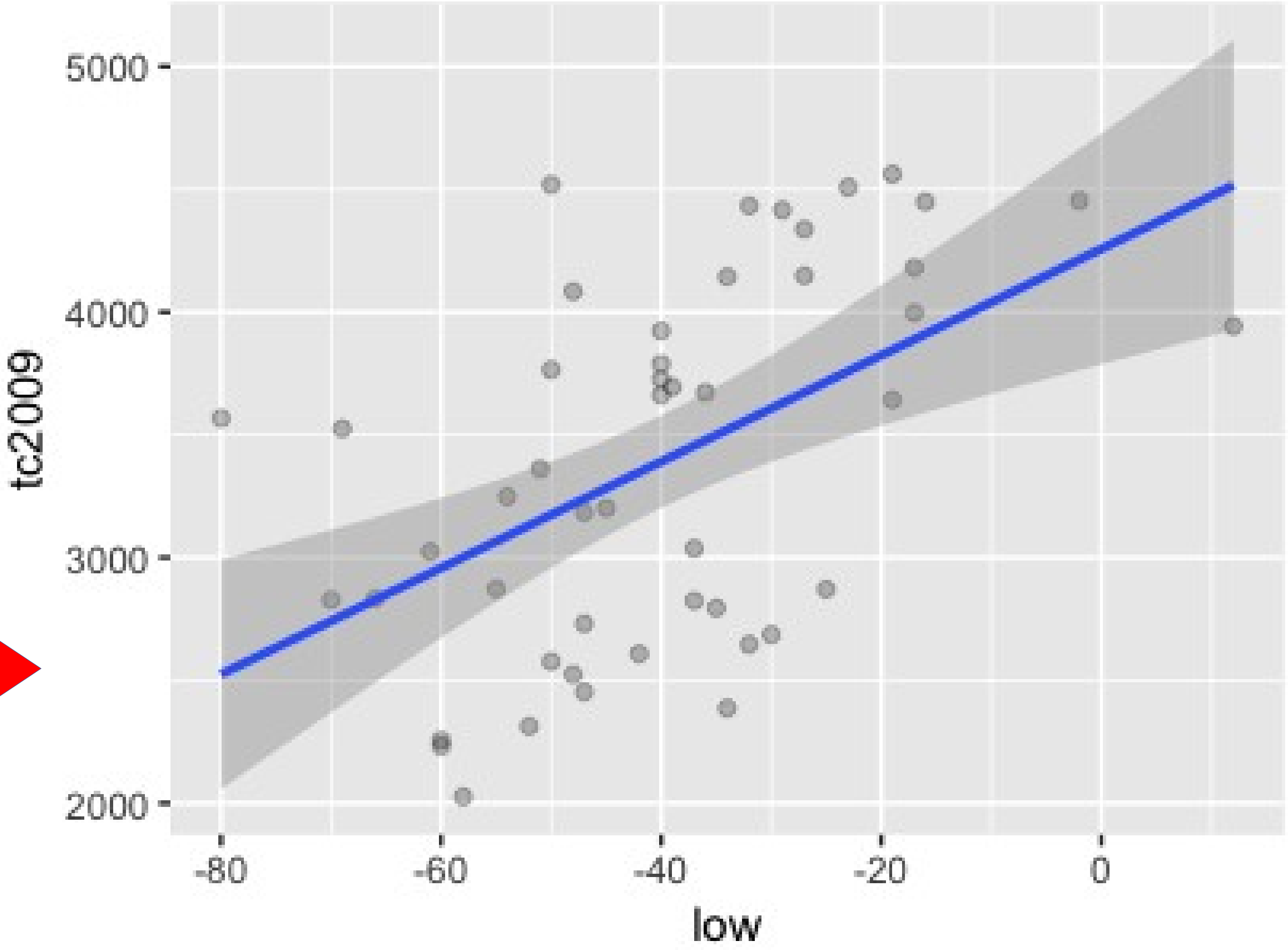
Plots

```
crime %>% ggplot(aes(x = low, y = tc2009)) +
geom_point(alpha = I(1/4)) + geom_smooth()
```



Plots

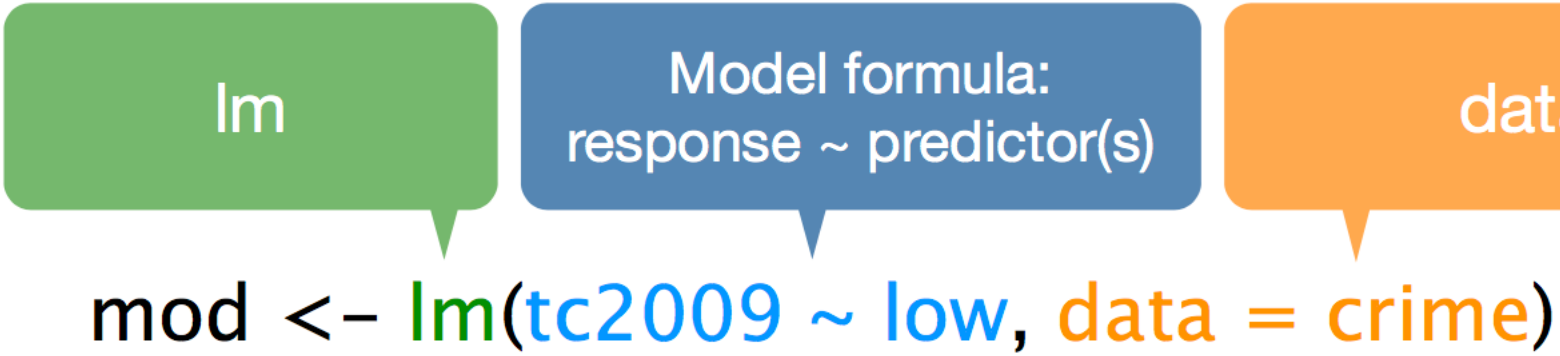
```
crime %>% ggplot(aes(x = low, y = tc2009)) +
geom_point(alpha = I(1/4)) + geom_smooth(method = lm)
```



This is the model's line here!

Build a Linear Model

- How much does ***low (indep)*** influence ***tc2009 (dep)***
- Linear model syntax



lm

Model formula:
response ~ predictor(s)

data

```
mod <- lm(tc2009 ~ low, data = crime)
```

Models Use Formulas

- R formulas are expressions built with ~ (tilda)

```
tc2009 ~ low
# gives:  tc2009 ~ low


class(tc2009 ~ low)
# gives: [1] "formula"
```

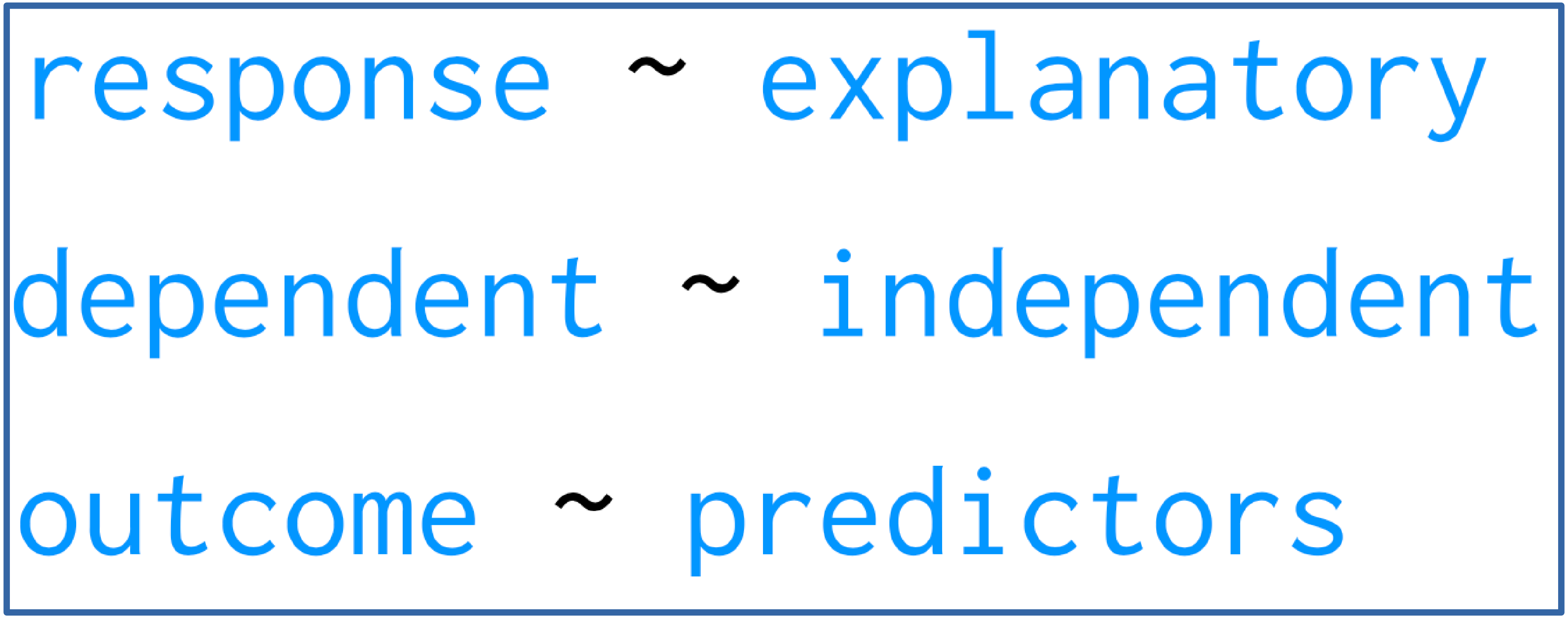
Models Use Formulas

- Formulas only need to include the response and predictor variables

$$y = f(x) = \alpha + \beta x + \epsilon$$

```
#Syntax to Build the linear model:
y ~ x
```

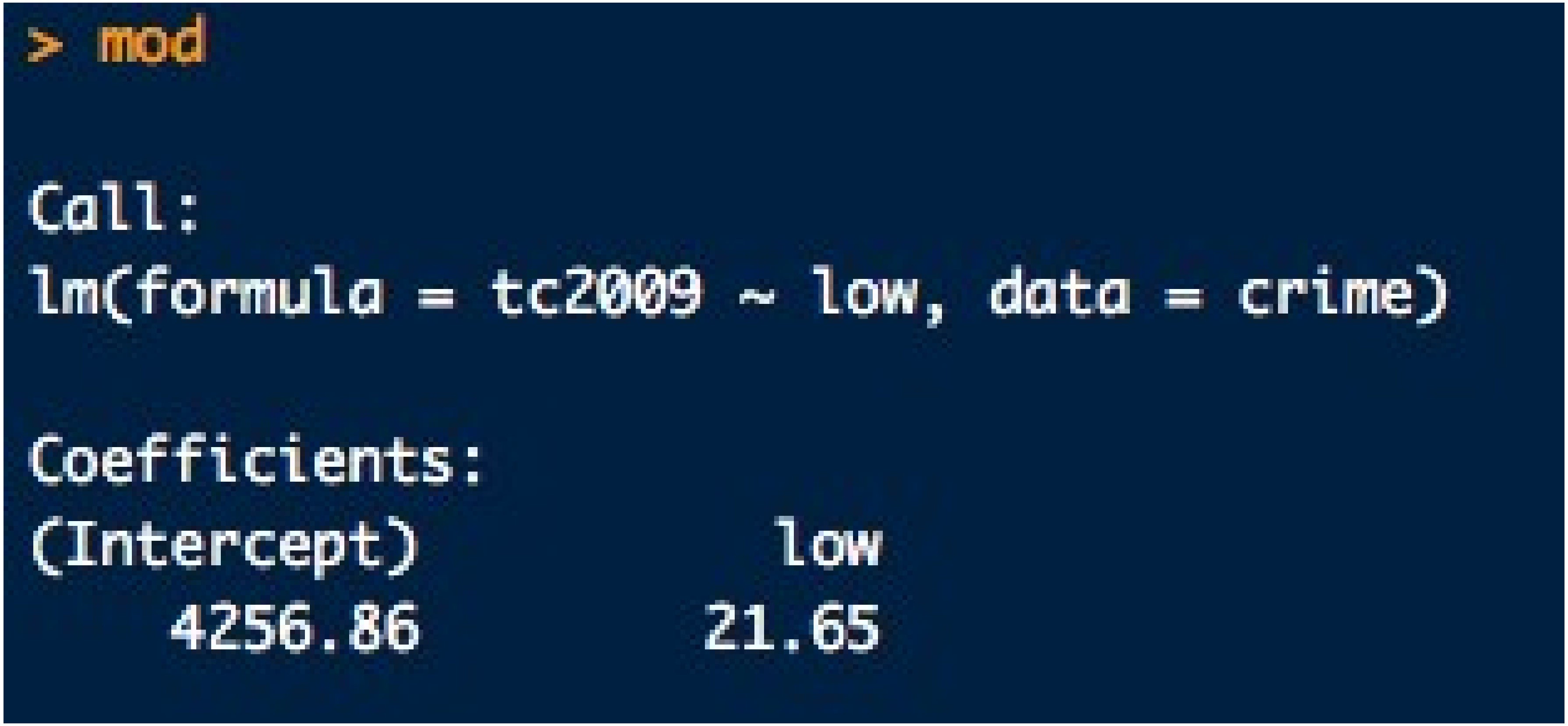
Types of Formulas



```
response ~ explanatory

dependent ~ independent

outcome ~ predictors
```

Intercept and Coefficient

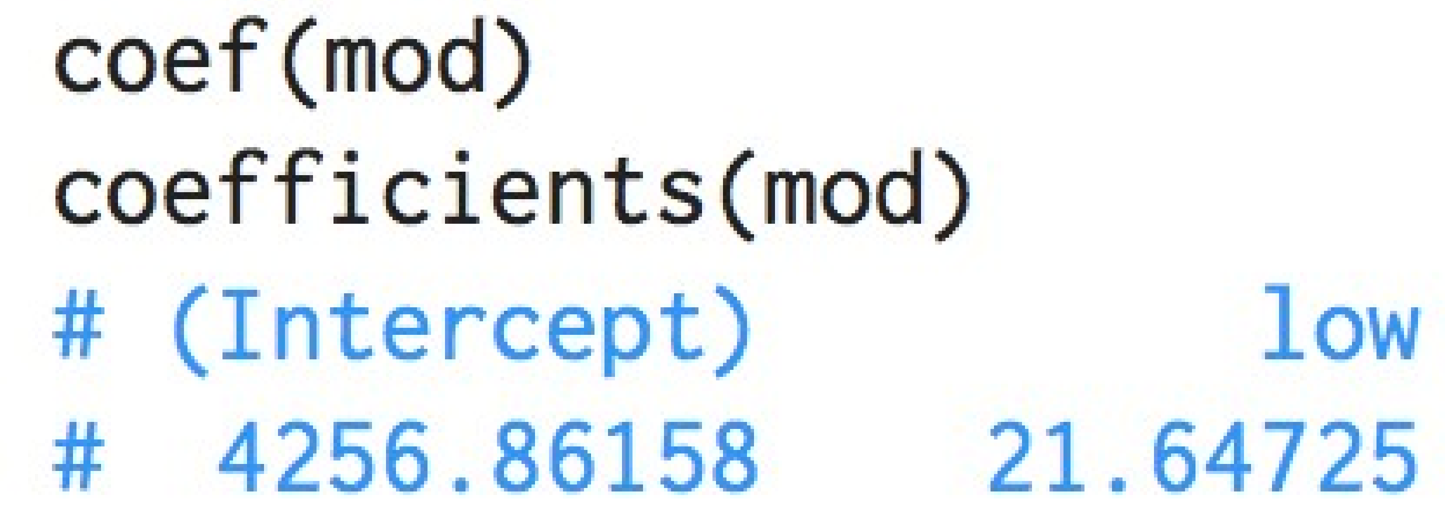
```
mod
```



```
> mod

Call:
lm(formula = tc2009 ~ low, data = crime)

Coefficients:
(Intercept)          low
    4256.86        21.65
```

Coef

- Shows the model’s coefficients (I.e., intercept, slopes)



```
coef(mod)
coefficients(mod)
# (Intercept)           low
#  4256.86158      21.64725
```

α β

Interpreting Models

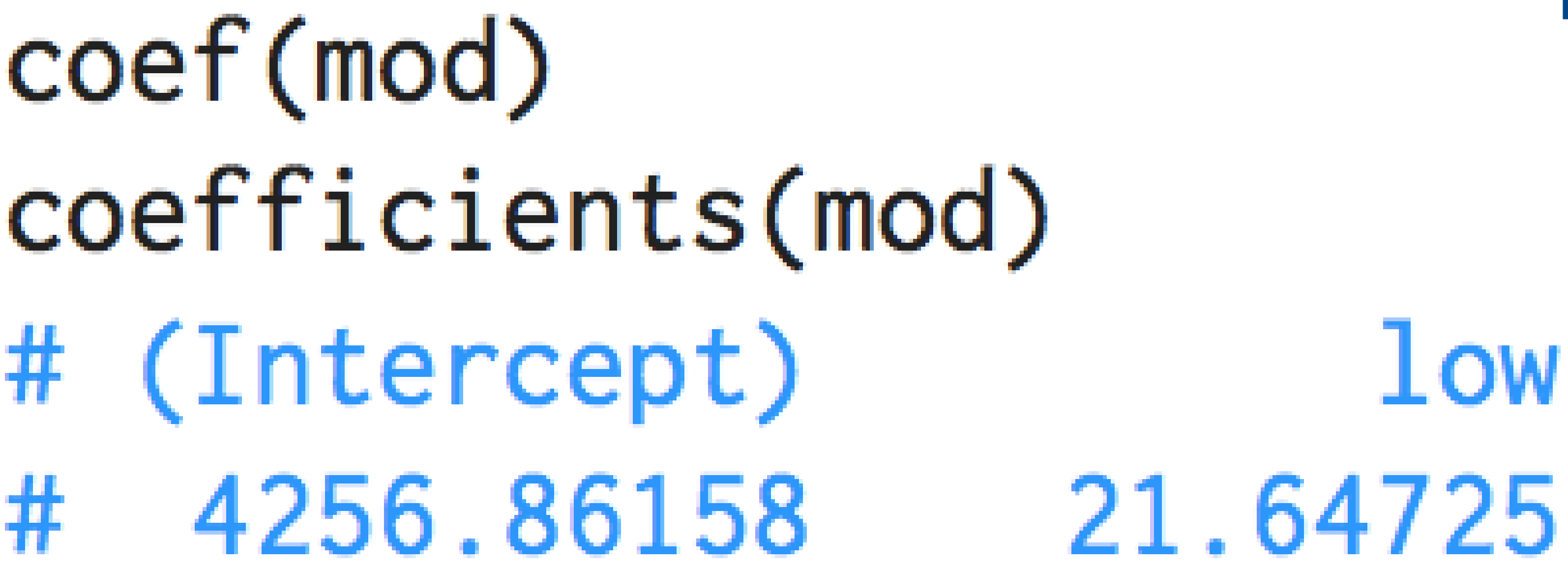
Linear models are very easy to interpret

$$y = \alpha + \beta x + \epsilon$$

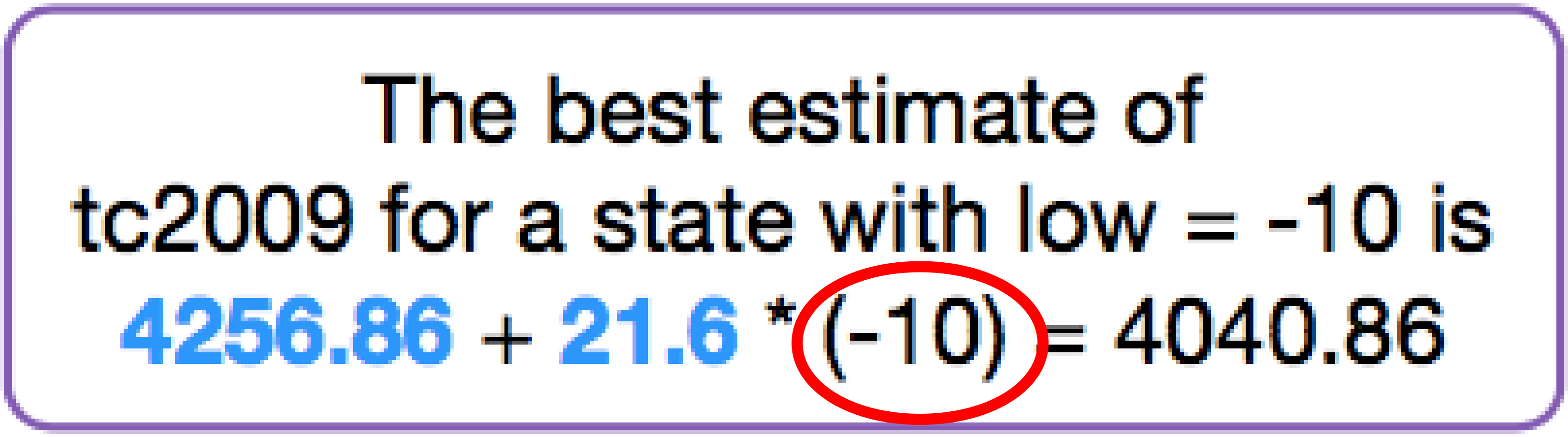
α is the expected value of y when x is 0.

β is the expected increase in y associated with a one unit increase in x

Coefficients: For Prediction



```
coef(mod)
coefficients(mod)
# (Intercept)          low
#  4256.86158     21.64725
```



The best estimate of tc2009 for a state with low = -10 is

4256.86 + 21.6 * (-10) = 4040.86

(x,y) ← (-10, 4040.86)

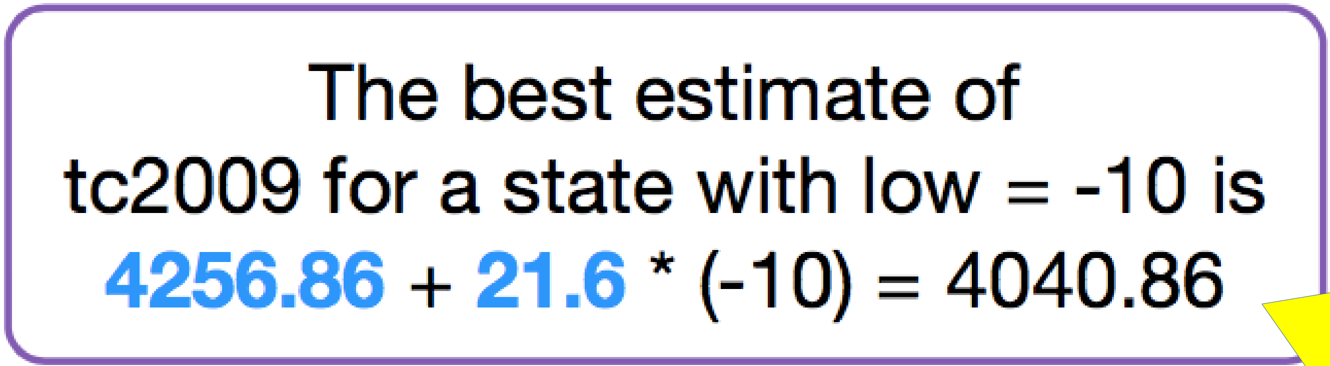
Coefficient Calculator Function

```
# create function to find y for x
tellMeY <- function(x_int){
  #function to get the y value for an entered x
value
  # The best estimate of tc2009 for a state with low
of inputted value x_int
  cat("  intercept :",mod$coefficients[1] )
  cat("\n  slope     :",mod$coefficients[2] )
  y = mod$coefficients[1] + x_int *
mod$coefficients[2]
  cat("\n  y = ",y)
}

tellMeY(-10) # note: x = -10 also, my “what if?”
enabler
```

Coefficient Calculator

This function is now my data!!

Based on our training using data,
If x = -10, my Y will be about 4040.86



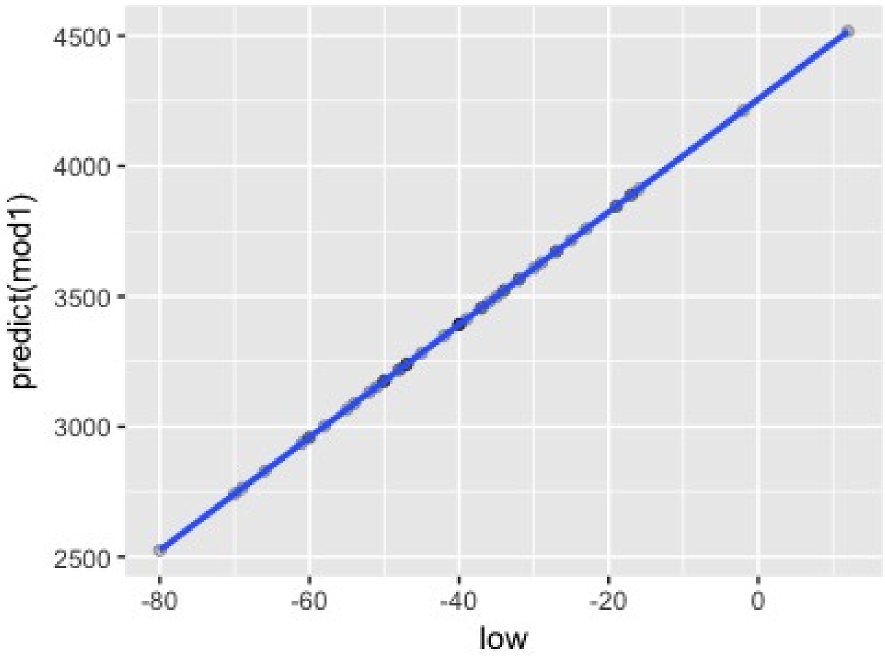
The best estimate of
tc2009 for a state with low = -10 is
4256.86 + 21.6 * (-10) = 4040.86

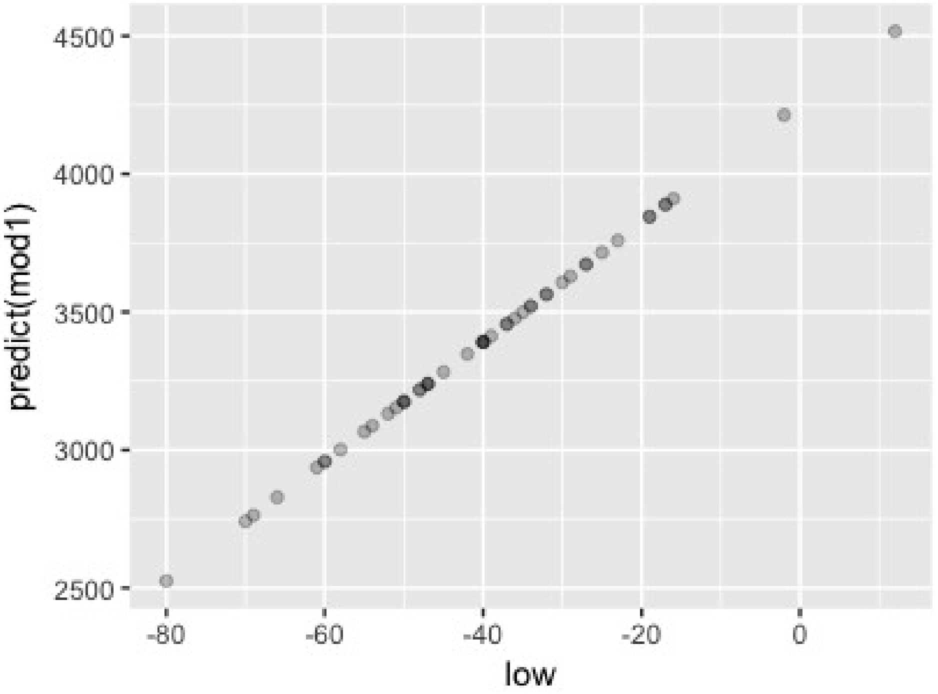
I can even predict *y*,
based on my own values of x!

Due to error, there is a slight difference between This value and our own value.

Forecasting the Data

```
crime %>% ggplot(aes(x = low, y = predict(mod))) +
geom_point(alpha = I(1/4))


crime %>% ggplot(aes(x = low, y = predict(mod))) +
geom_point(alpha = I(1/4)) + geom_smooth()
```





Aside: intercept terms

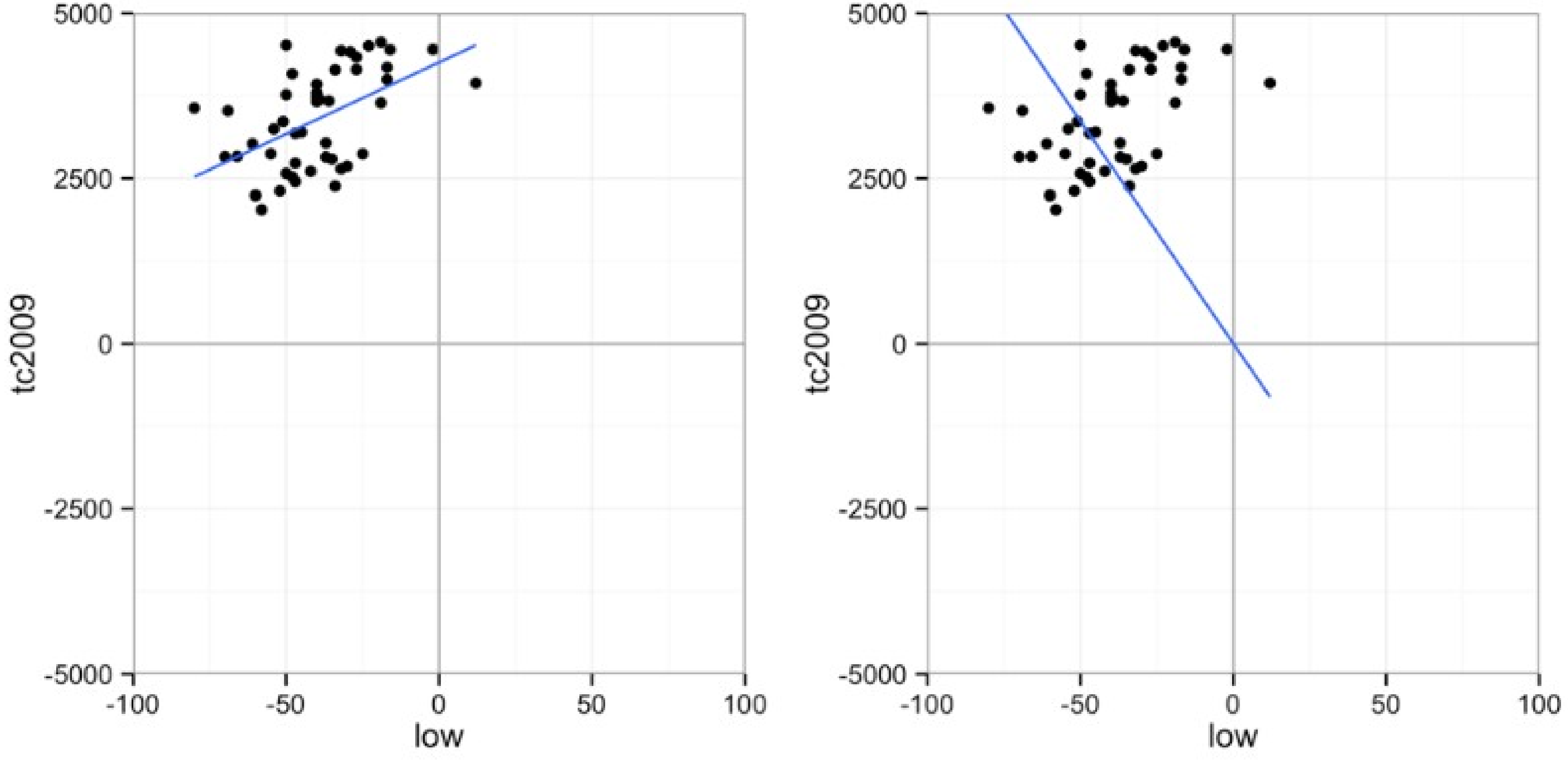
R includes an intercept term in each model by default

$$y = \enclose{circle}{\alpha} + \beta x + \epsilon$$

```
y ~ x
```

Study at x = 0?

(Does x = 0 make sense here?)



With α

Without α

Every linear model has a y intercept. Including α lets this term vary. Not including α forces the intercept to (0, 0).

Study at x = 0?
(Does x = 0 make sense here?)

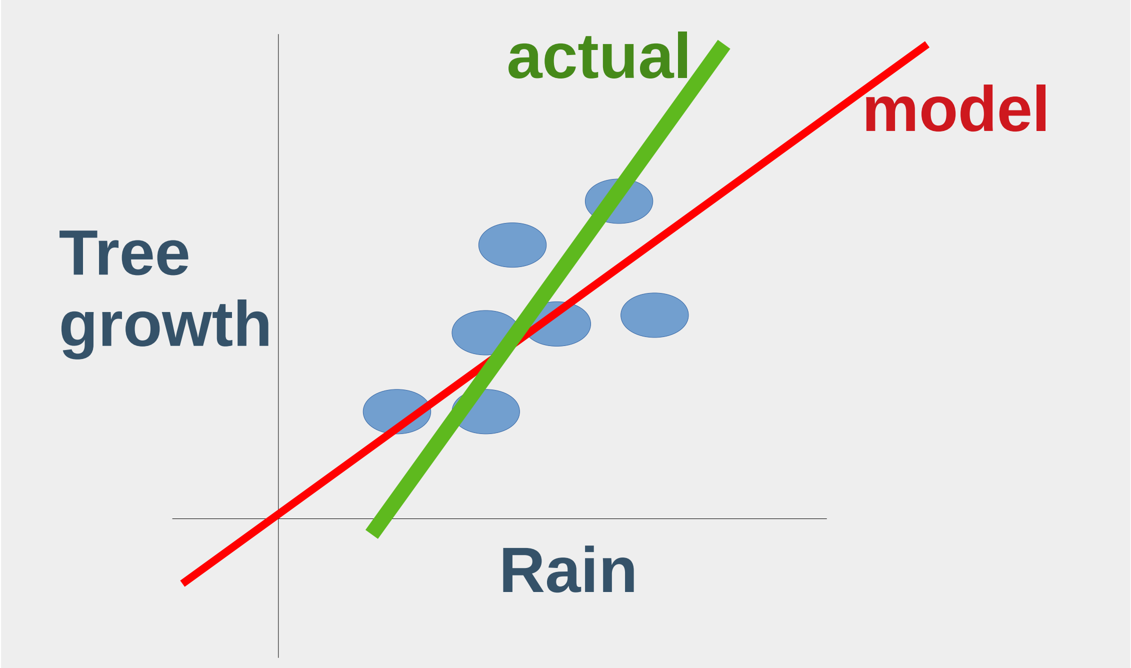
- The y-intercept is the place where the regression line crosses the y-axis (where x = 0), and is denoted by b from $y = mx + b$
- Meaningful interpretation: Sometimes the y-intercept has meaningful interpretion (and sometimes not)
- No meaning for the y-intercept when data is not present near the point where x = 0 (and the model suggests that data is present at this point)

Study at *x* = 0?
(Does *x* = 0 make sense here?)

Ex: A model where rain (*x*) is used to predict tree growth (y)

If *rain* = 0, then *tree_growth* = 0

As a result, the regression line may cross *y*-axis at some other point (other than zero)

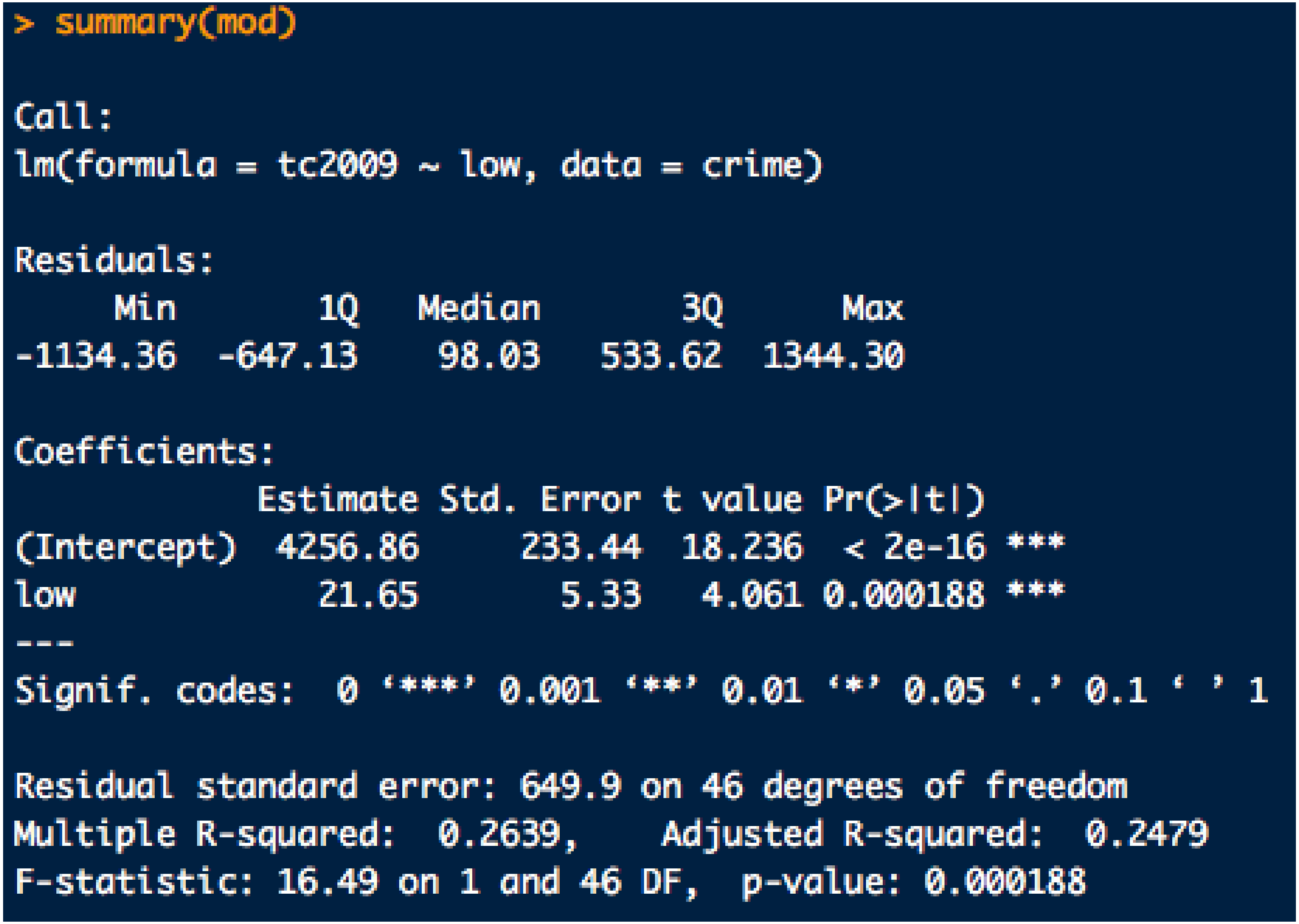


An Intercept Term:
To Use or Not?

You can explicitly ask for an intercept by including the number one, 1, as a formula term. You can remove the intercept by including a zero or negative 1.

```
# equivalent - includes intercept
lm(tc2009 ~ 1 + low, data = crime)
lm(tc2009 ~ low, data = crime)

# equivalent - removes intercept
lm(tc2009 ~ low - 1, data = crime)
lm(tc2009 ~ 0 + low, data = crime)
```

Results: summary(mod)

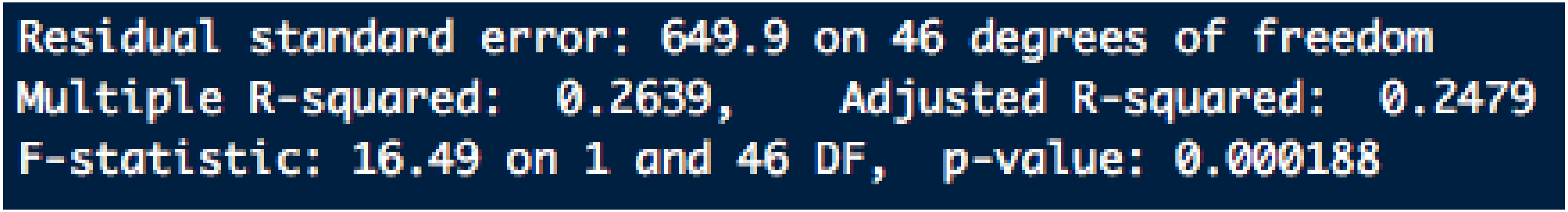


```
> summary(mod)

Call:
lm(formula = tc2009 ~ low, data = crime)

Residuals:
     Min       1Q   Median       3Q      Max
-1134.36  -647.13    98.03   533.62  1344.30

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4256.86     233.44  18.236  < 2e-16 ***
low            21.65       5.33   4.061 0.000188 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 649.9 on 46 degrees of freedom
Multiple R-squared:  0.2639,    Adjusted R-squared:  0.2479
F-statistic: 16.49 on 1 and 46 DF,  p-value: 0.000188
```

R-squared Value

- R2 is a statistic that will give some information about the goodness of fit of a model.
- The R2 coefficient of determination describes how well the regression predictions approximate the real data points.

  An R2 of 1 indicates that the regression predictions perfectly fit the data.
- A measurement of how close the data are to the fitted regression line.



```
Residual standard error: 649.9 on 46 degrees of freedom
Multiple R-squared:  0.2639,	Adjusted R-squared:  0.2479 
F-statistic: 16.49 on 1 and 46 DF,  p-value: 0.000188
```

Extracting Info

- Create model object
- Run functions on model object to get details

Try these commands

```
summary(mod)
predict(mod) # predictions at original vals
resid(mod) # residuals
```

Consider This!

- Fit a linear model to the crime data set.
- Predict **tc2009** (dep) with **low** (ind). What are the model’s ***A*** and **B** variables? Hint: use `coef(mod)`

$$Y = \underline{A} + \underline{B} * x + \epsilon$$

Consider This!

- Try making a model with the other data set to determine whether taller people make more money.

Consider This!

Fit a linear model to the wages data set that predicts ***earn*** with ***height***.

How do you interpret the relationship between **height** and **earnings**?

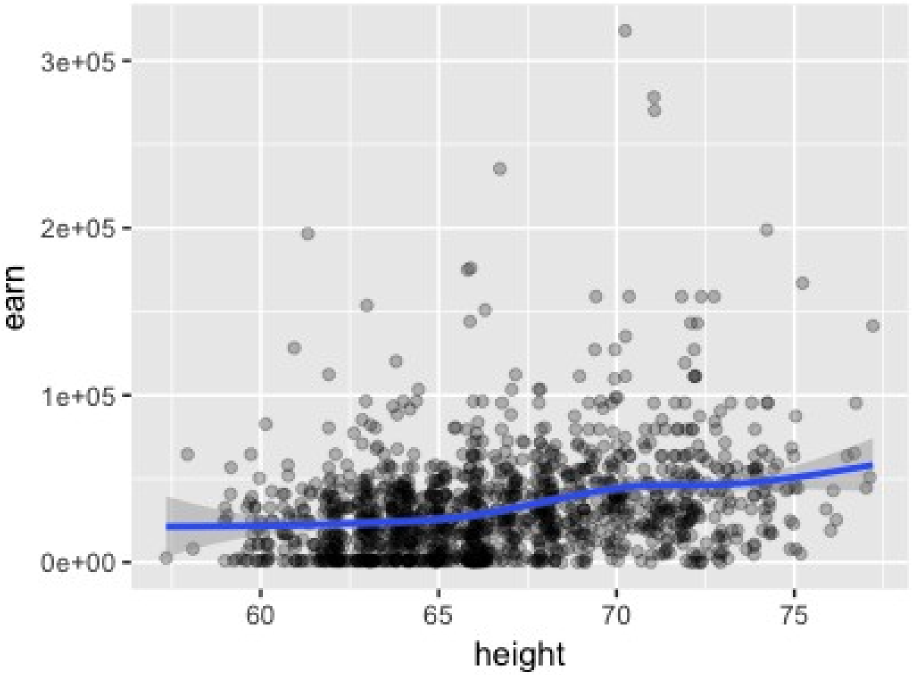
```
wages <- read.csv("wages.csv")
```

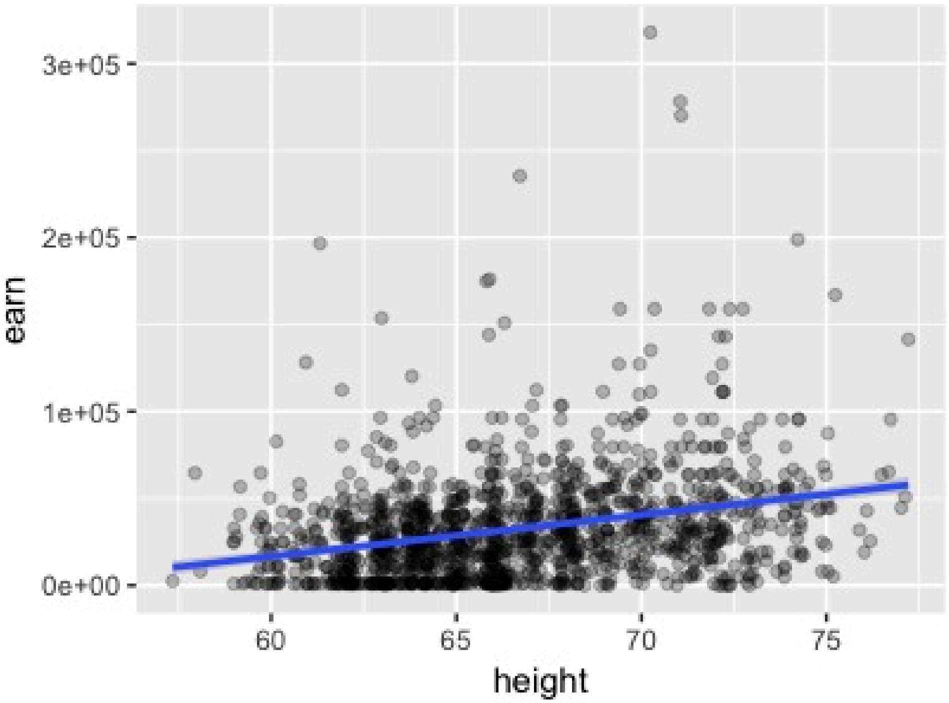
Do Tall People Make More?

```
wages %>% ggplot(aes(x = height, y = earn)) + geom_point(alpha
= I(1/4)) + geom_smooth()
```

```
wages %>% ggplot(aes(x = height, y = earn)) + geom_point(alpha
= I(1/4)) + geom_smooth(method = lm) # regression line
```

Try switching the x’s and y’s for another view.





Correlations

```
# Find correlations using the "pearson" method

cor(earn, height, method = "pearson")

hmod <- lm(dependent ~ independent)
```

- Where **dependent** var is ***earn***
- And **independent** var is ***height***

Dep And Indep Vars

- #make your model
- hmod <- lm(dependent ~ independent)
- Where **dependent** var is ***earn***
- And **independent** var is ***height***

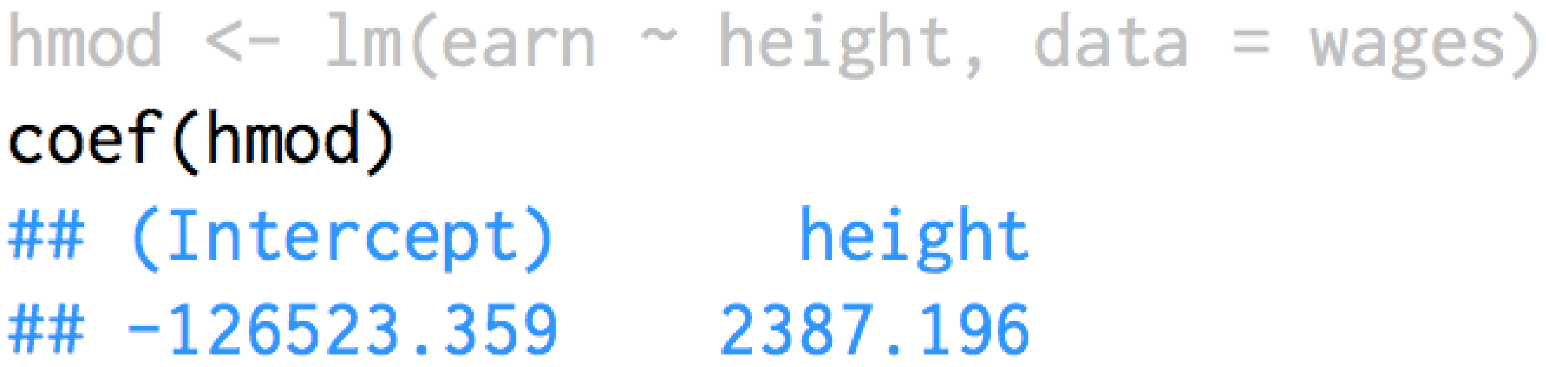
$$y = \alpha + \beta x + \epsilon$$

Earn Regressed Over *height*

- #make your model
- hmod <- lm(earn ~ height)
- Where **dependent** var is ***earn***
- And **independent** var is ***height***

$$earn = \alpha + \beta \times height + \epsilon$$

Earn Regressed Over *height*

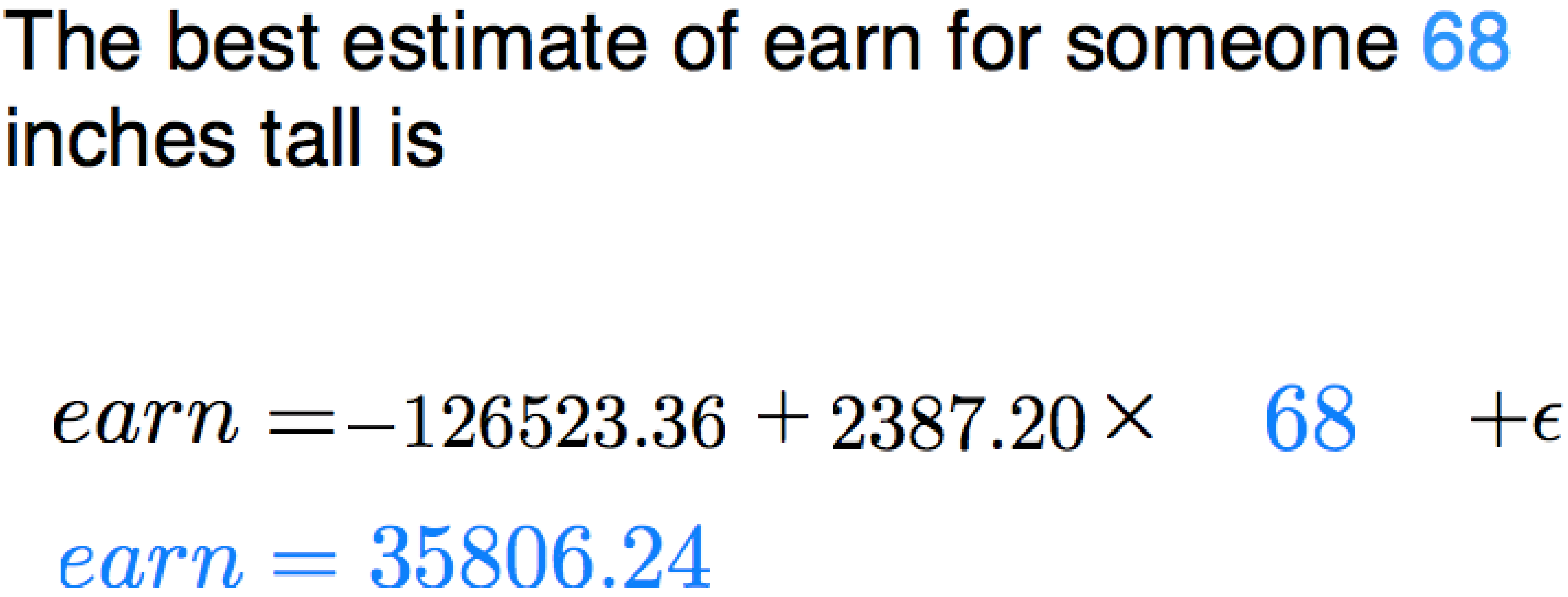


```
hmod <- lm(earn ~ height, data = wages)
coef(hmod)
## (Intercept)      height
## -126523.359    2387.196
```

$$earn = \alpha + \beta \times height + \epsilon$$

$$\boxed{earn = -126523.36 + 2387.20 \times height + \epsilon}$$

Earn Regressed Over *height*



The best estimate of earn for someone 68 inches tall is

$$earn = -126523.36 + 2387.20 \times \quad 68 \quad + \epsilon$$

$$earn = 35806.24$$

Build a model.

- Fit a linear model to the wages data set
- How do we interpret the results?

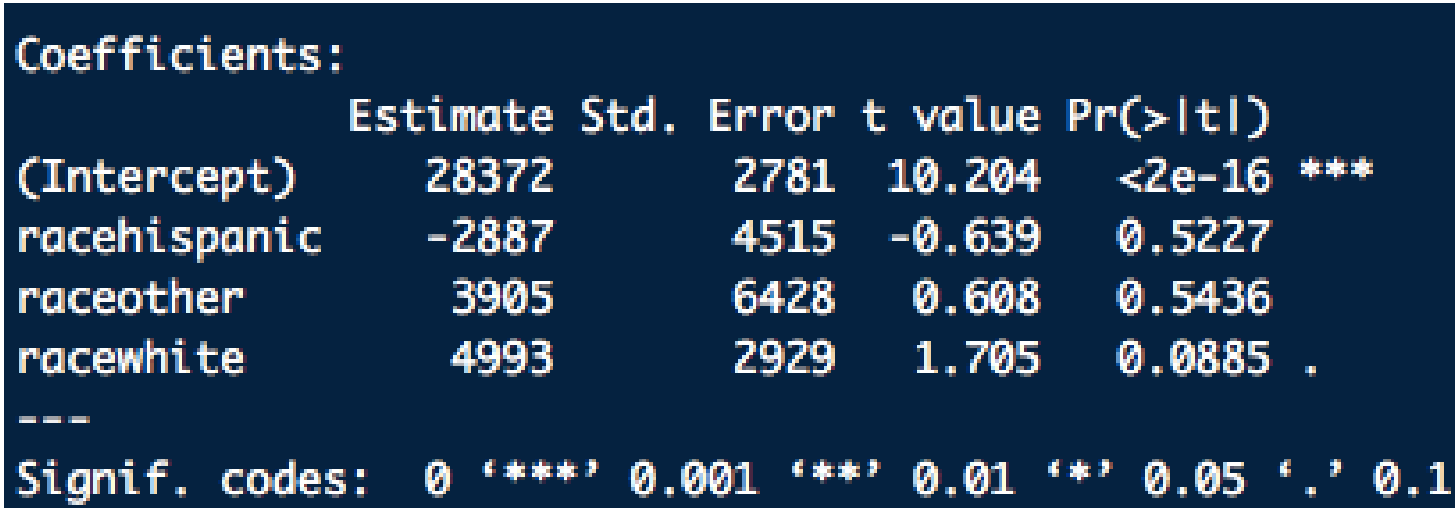
Q: What happens when we regress *earn* over *race*?

Header

```
rmod <- lm(earn ~ race, data = wages)
coef(rmod) # get the model’s y-intercepts and slopes
```

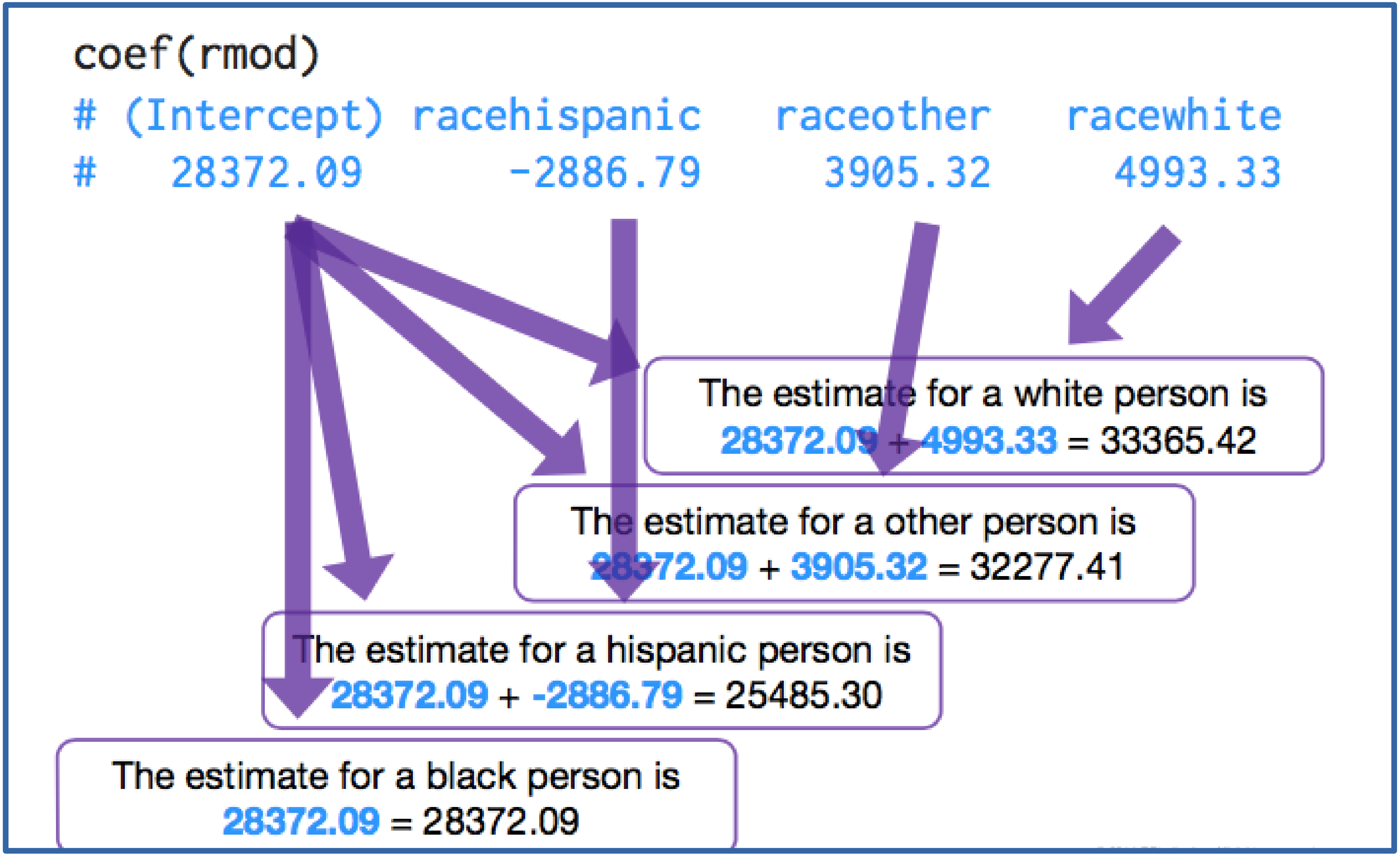
```
coef(rmod)
# (Intercept) racehispanic   raceother   racewhite
#    28372.09      -2886.79     3905.32     4993.33
```

summary(rmod)



```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     28372       2781  10.204   <2e-16 ***
racehispanic    -2887       4515  -0.639   0.5227
raceother        3905       6428   0.608   0.5436
racewhite        4993       2929   1.705   0.0885 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
```

Estimates From Coefficients



```
coef(rmod)
# (Intercept) racehispanic   raceother   racewhite
#    28372.09     -2886.79     3905.32     4993.33
```

The estimate for a white person is
28372.09 + 4993.33 = 33365.42

The estimate for a other person is
28372.09 + 3905.32 = 32277.41

The estimate for a hispanic person is
28372.09 + -2886.79 = 25485.30

The estimate for a black person is
28372.09 = 28372.09

Participation 1
One Check Mark

- Pick a data set to make plots, correlation, linear model(s) over selected columns. Discuss your results.
- **https://classroom.github.com/a/YqgFE8VW**
- Due at end of class today.