

# **Data Science**

## **CS301**

### **Modeling: Formal Basics**

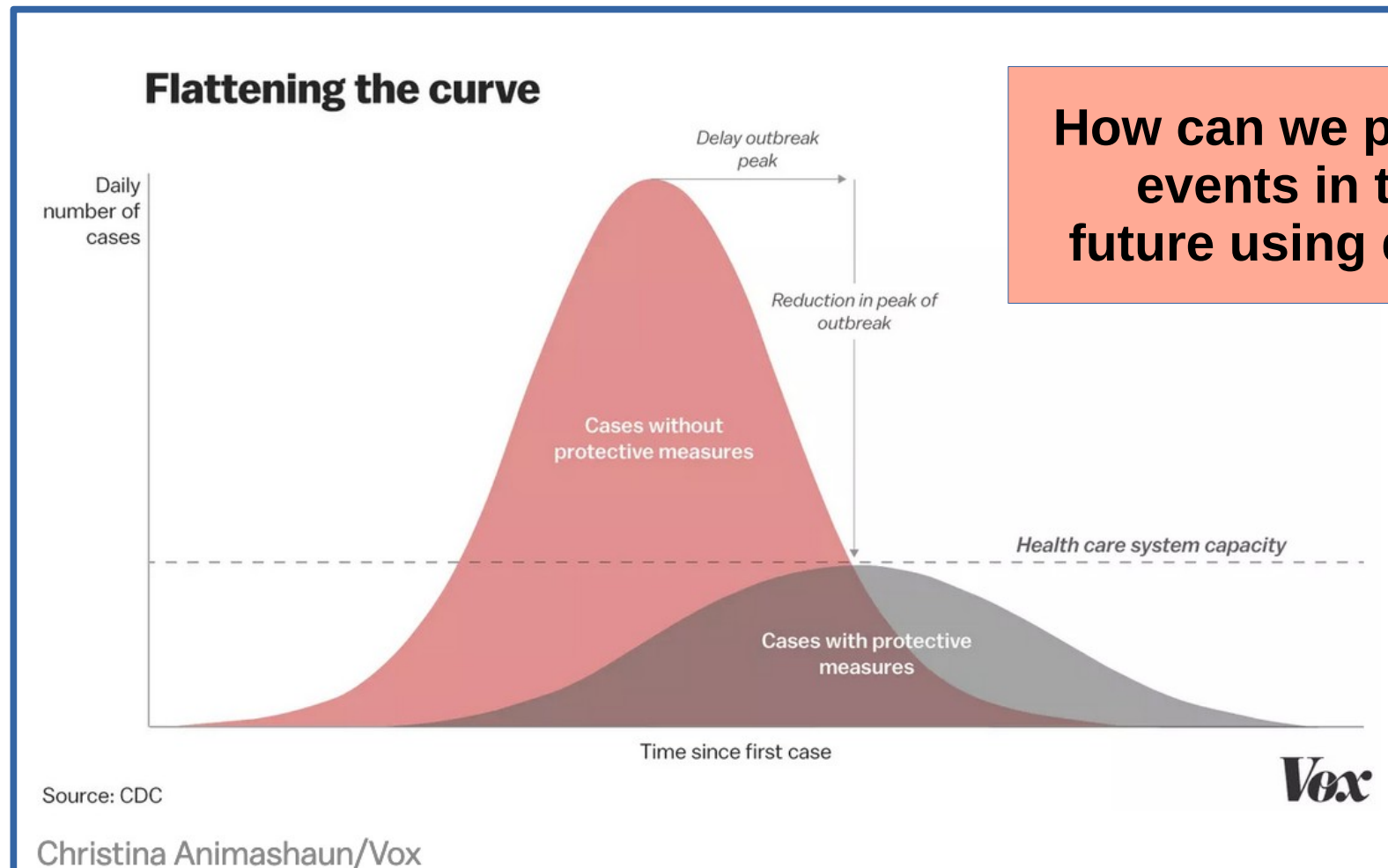
Week 08

Fall 2024

Oliver BONHAM-CARTER

# Good Reads

- How canceled events and self-quarantines save lives, in one chart

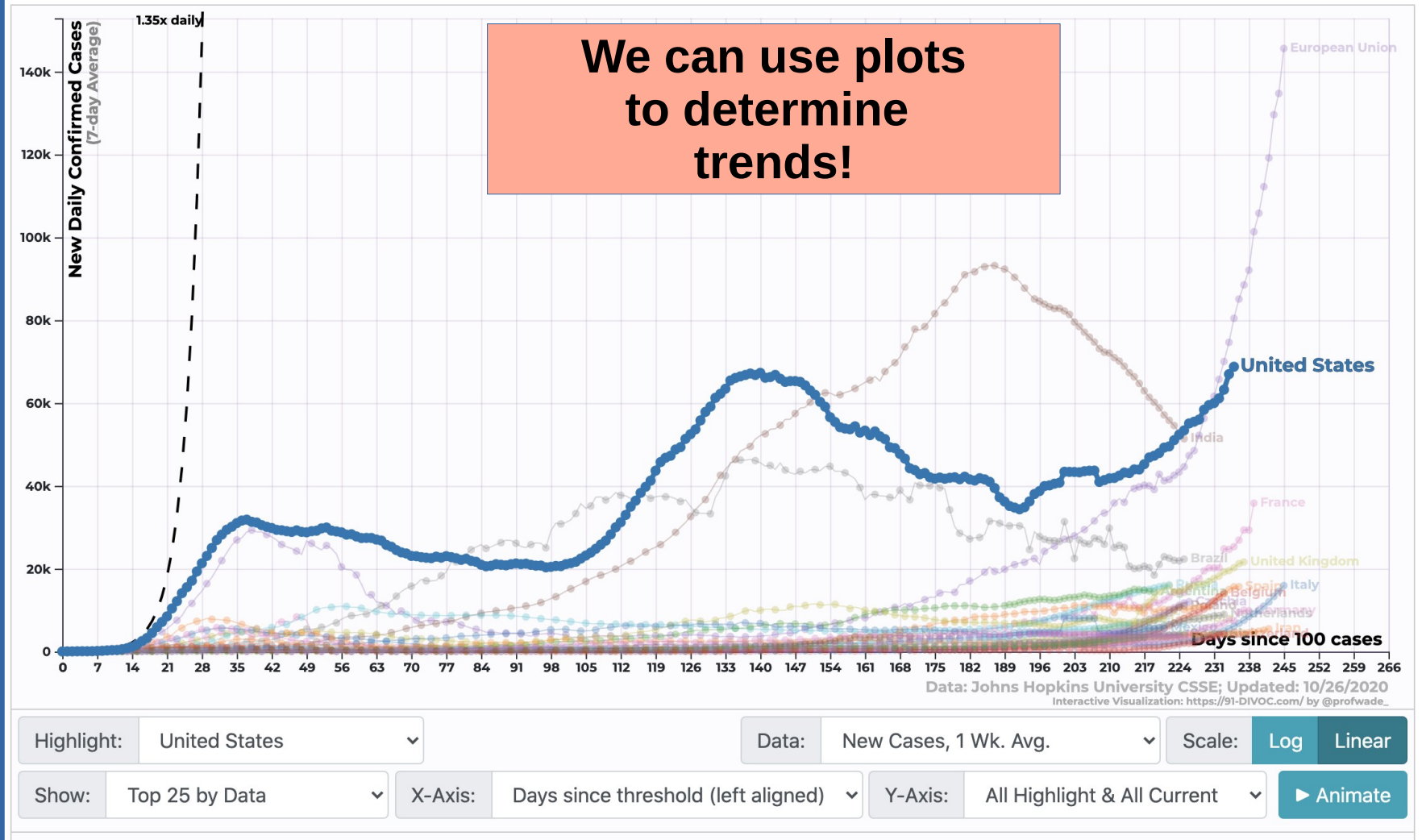


How can we predict events in the future using data?



# Interactive Plots: Covid-19 Cases

## New Confirmed COVID-19 Cases per Day



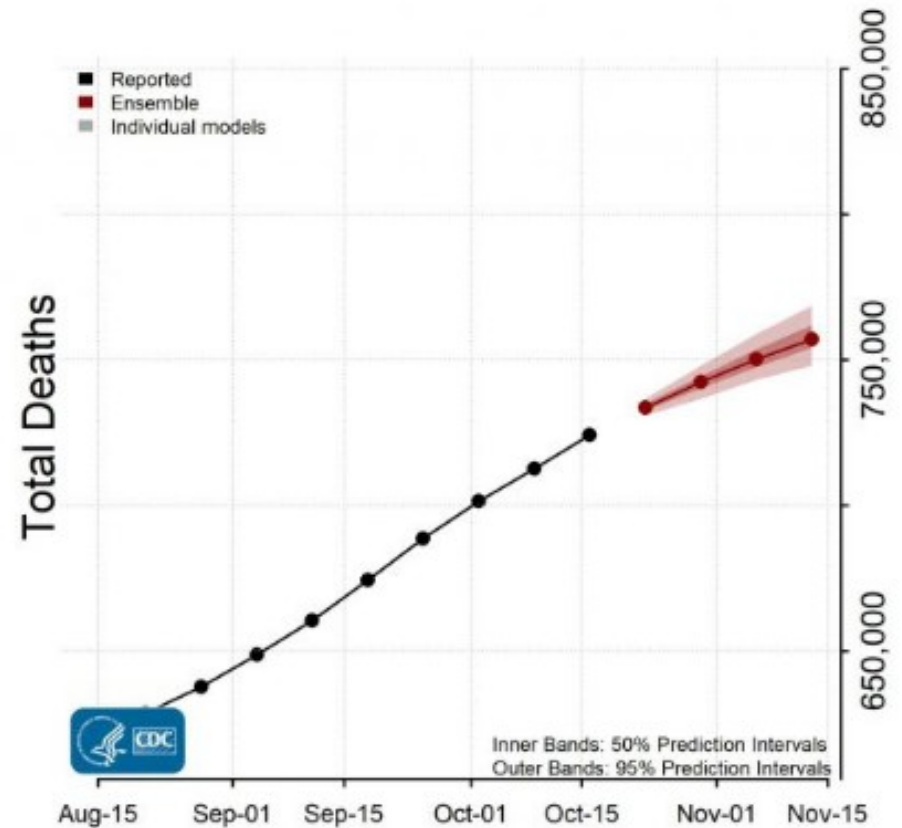
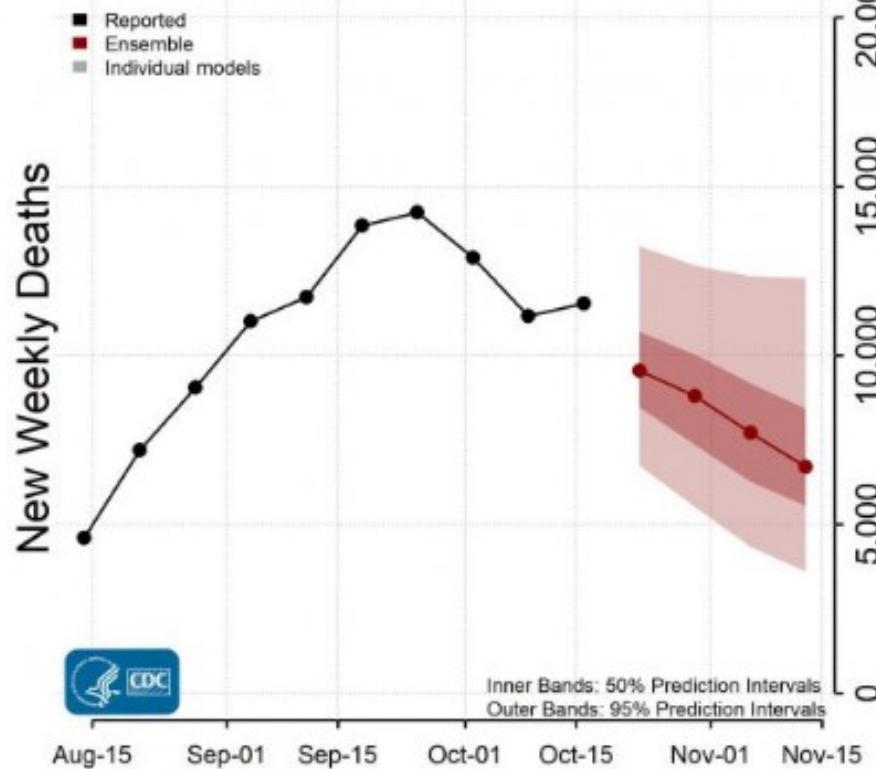


# Predicting With Plots

## National Forecast

More types of trends!

### National Forecast Combined Forecast





# The “Stuff” of Knowledge

- How do we use data to explain or argue something?
- How can we make types of **predictions**?







# 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 based on training by data



# 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 sort-of begun this study already: lines in scatter plots)*

- **LOESS Regression**

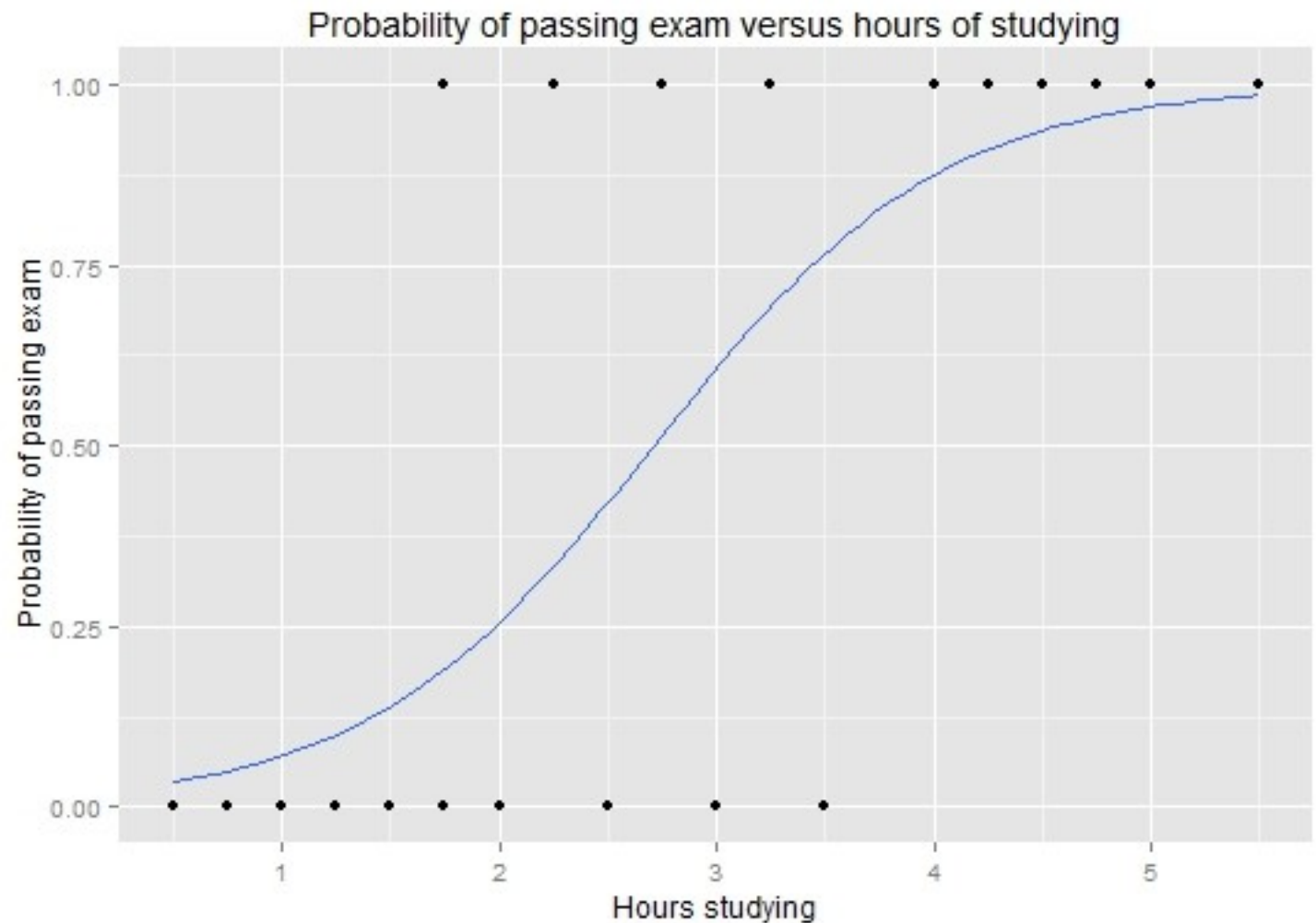
- Combining much of the simplicity of linear least squares regression, but building with the flexibility of nonlinear regression.



# Types of Models (iii)

## Logistic Regression

- Models where the dependent variable is categorical (i.e., 0's or 1's as factors)





# How Do We Answer Complex Questions?

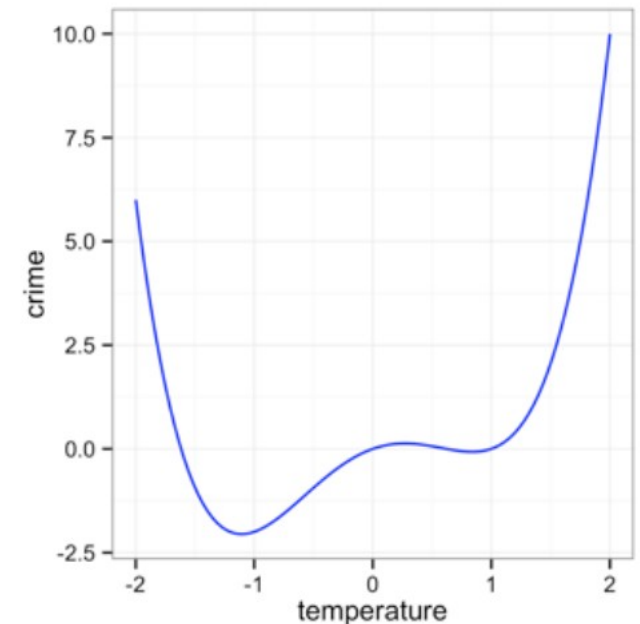
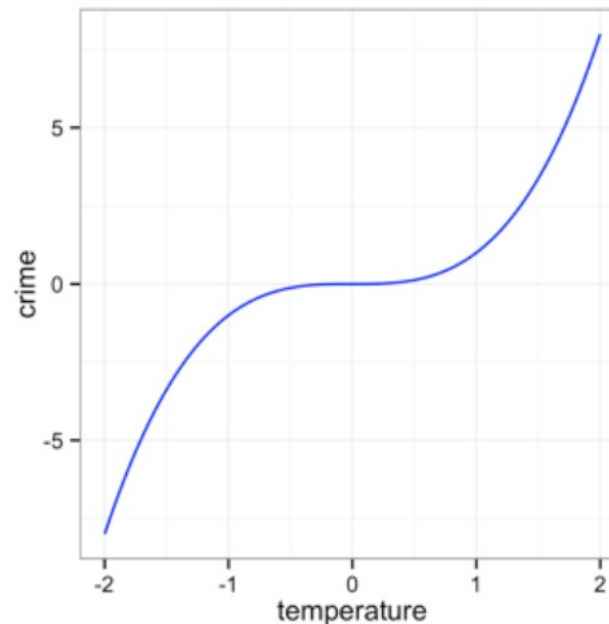
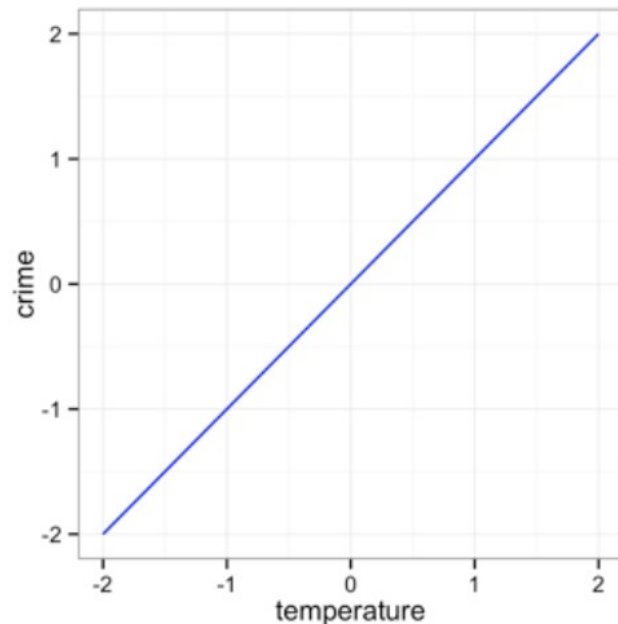
- **Modeling:** We employ a computational framework based-on historical data trends.
- **Prediction:** We play with the framework to see what happens when we apply changes a variable to see what happens ...

*What if...*



# Functions: the *stuff* behind the models

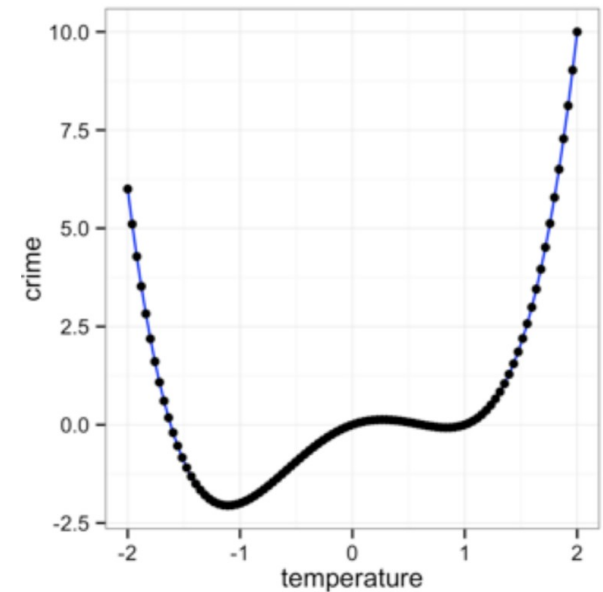
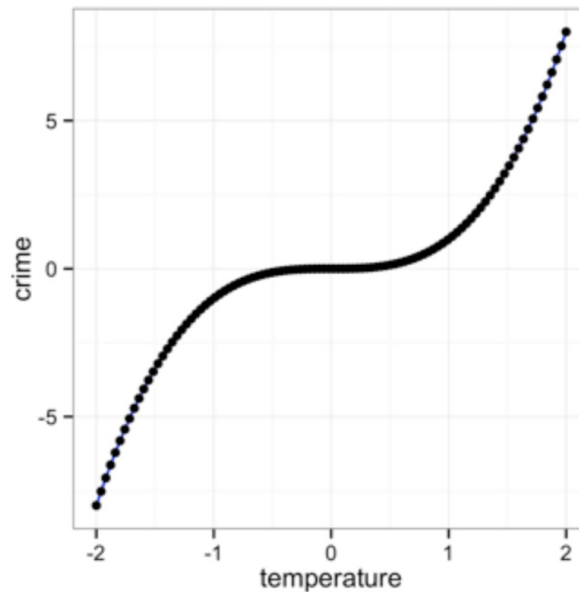
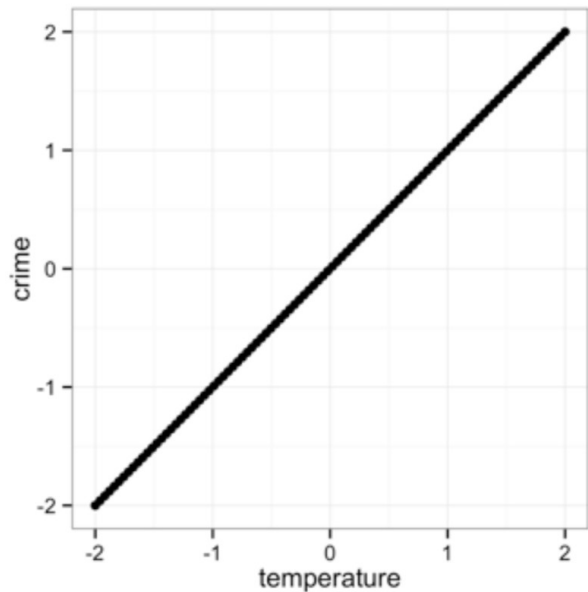
- Ideally, functions are mathematical descriptions of relationships





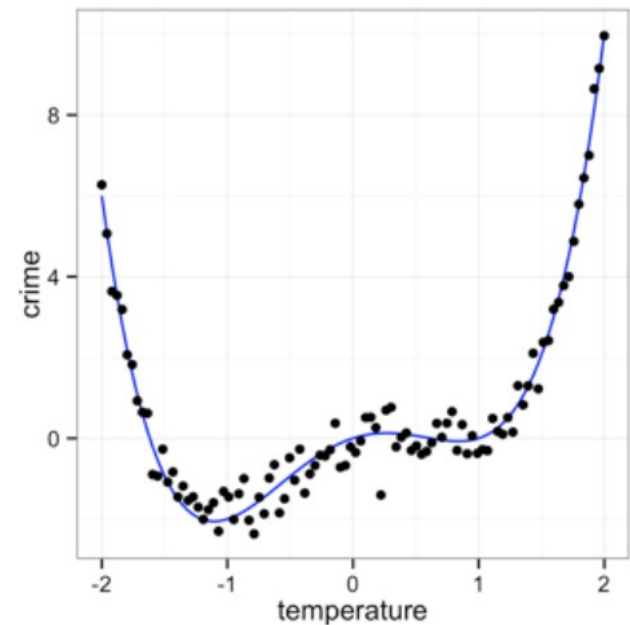
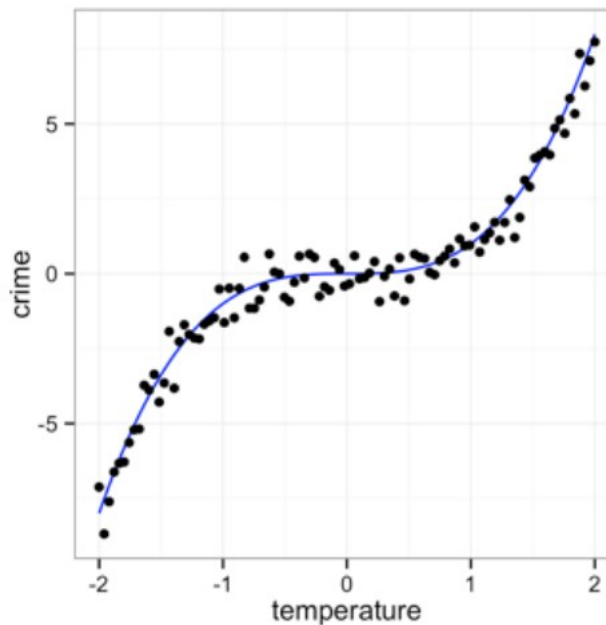
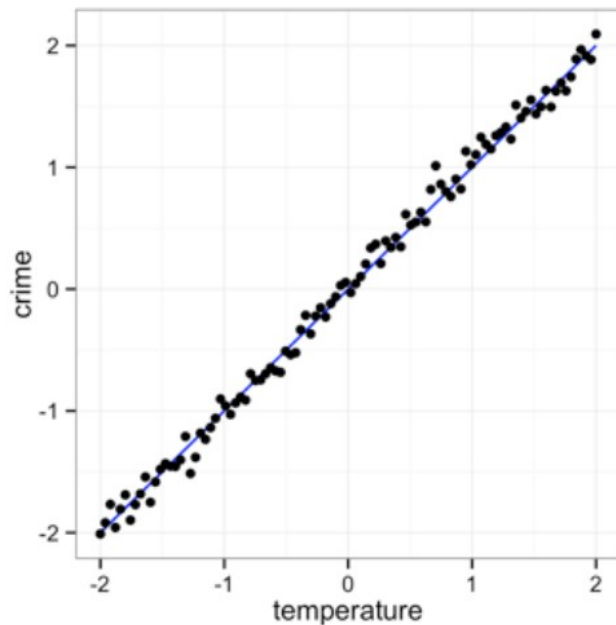
# Functions: the *stuff* behind the models

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



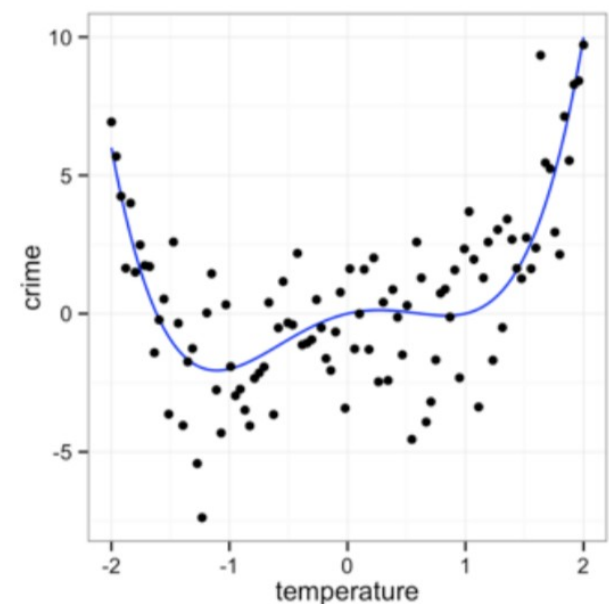
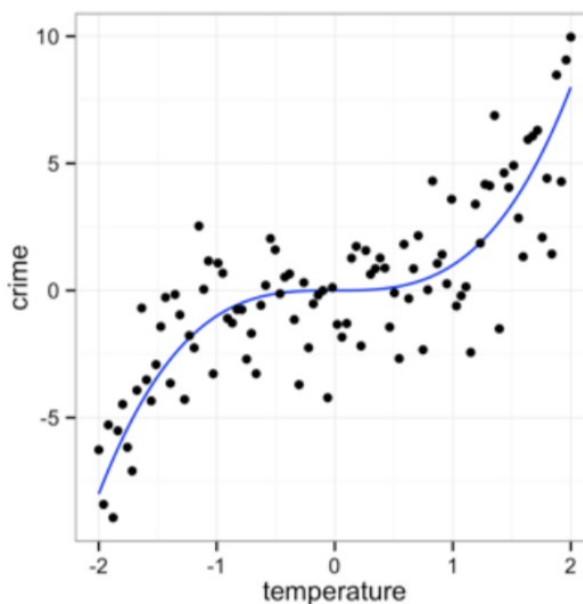
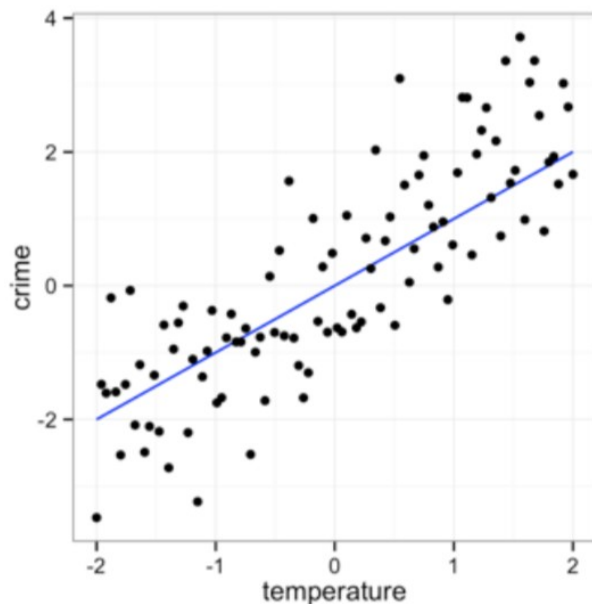
# Relationships Between Variables are Messy

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



# Relationships Between Variables

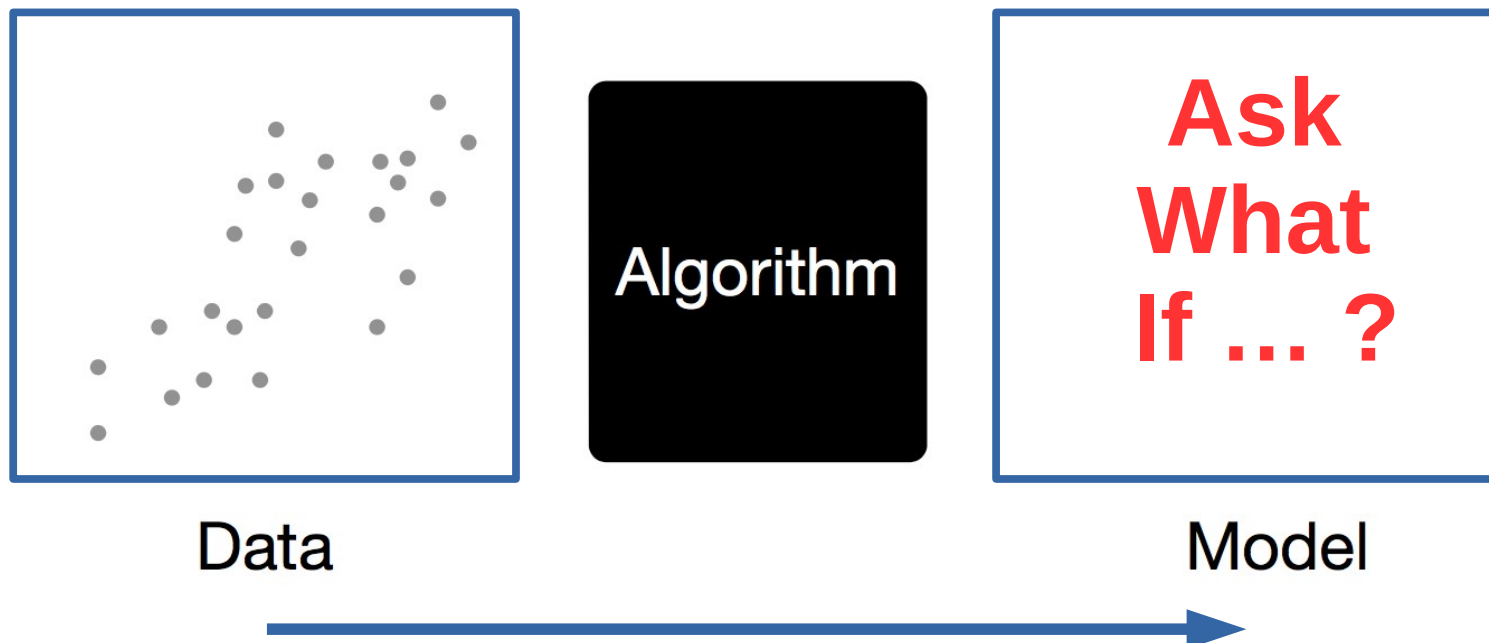
- If the actual relationship is affected by other variables, data points may not fall directly on the function-line.
- **Noise:** The greater the effect of other variables, the weaker the relationship. This is normally the situation with real data.





# 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 functional relationships in noisy data.





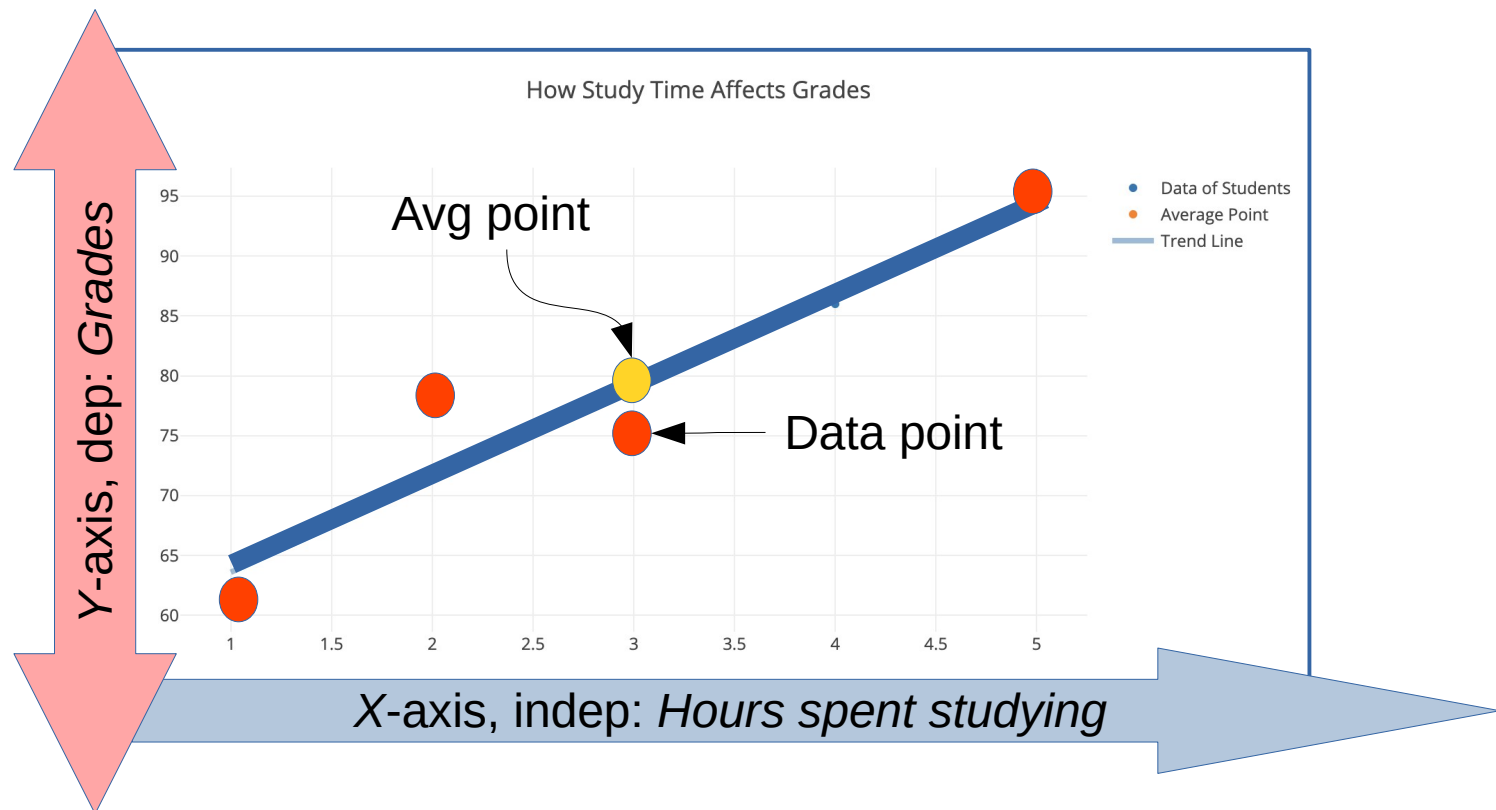
# Let's Talk Linear Models

- **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?



# Variables?

- **Independent variable:** a variable (often denoted by  $x$ ) whose variation does not depend on that of another (i.e., time).
- **Dependent variables:** a variables (often denoted by  $y$ ) that depends, by some law or rule (e.g., by a mathematical function), on the values of other variables (i.e., grades).
- Example: <https://chart-studio.plotly.com/~bchapman27/73.embed>





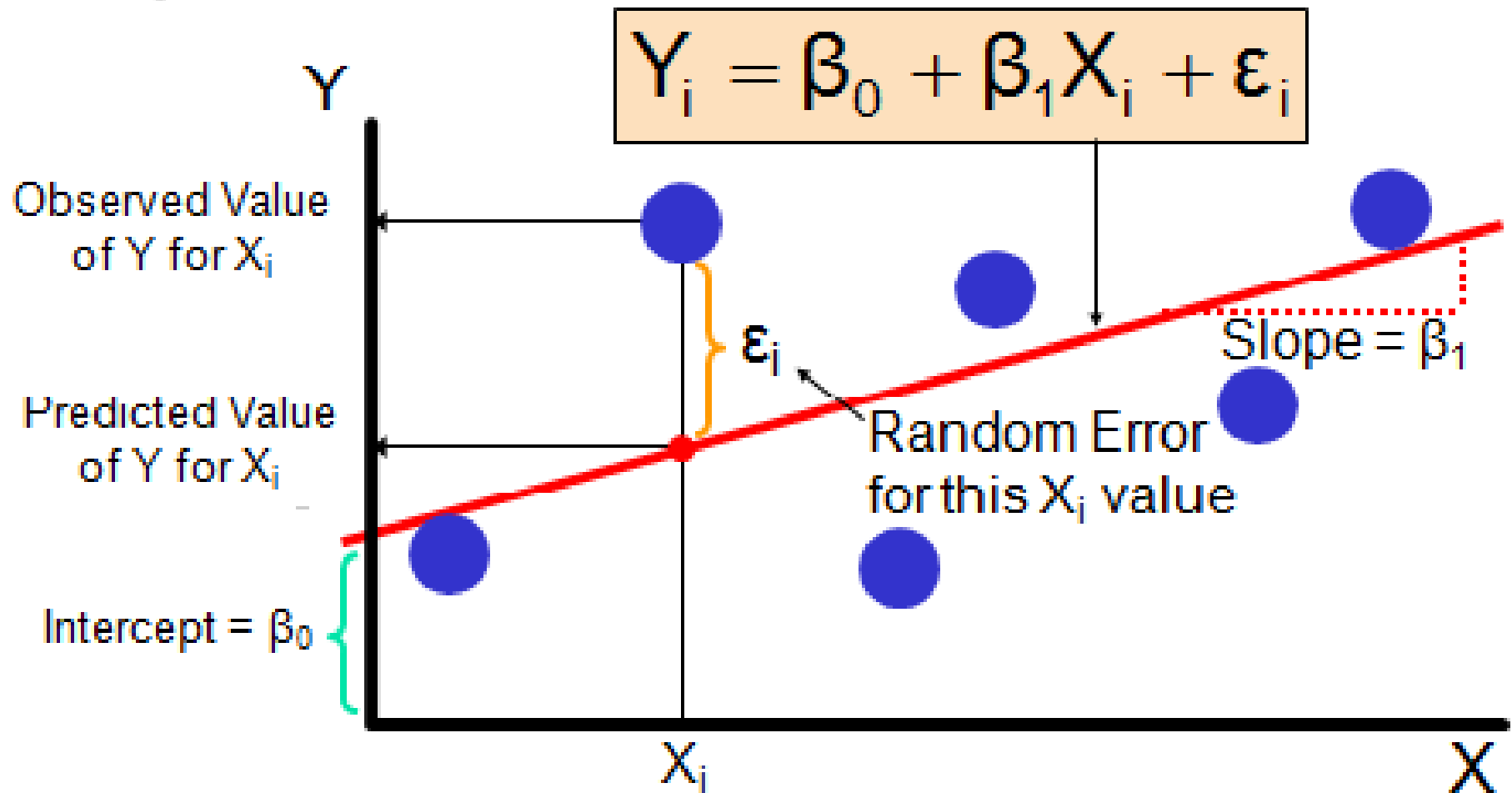
# Let's Talk Linear Models

- Linear regression (formally) is a function that draws a line from data
- The function  $f(x)$  has the form:

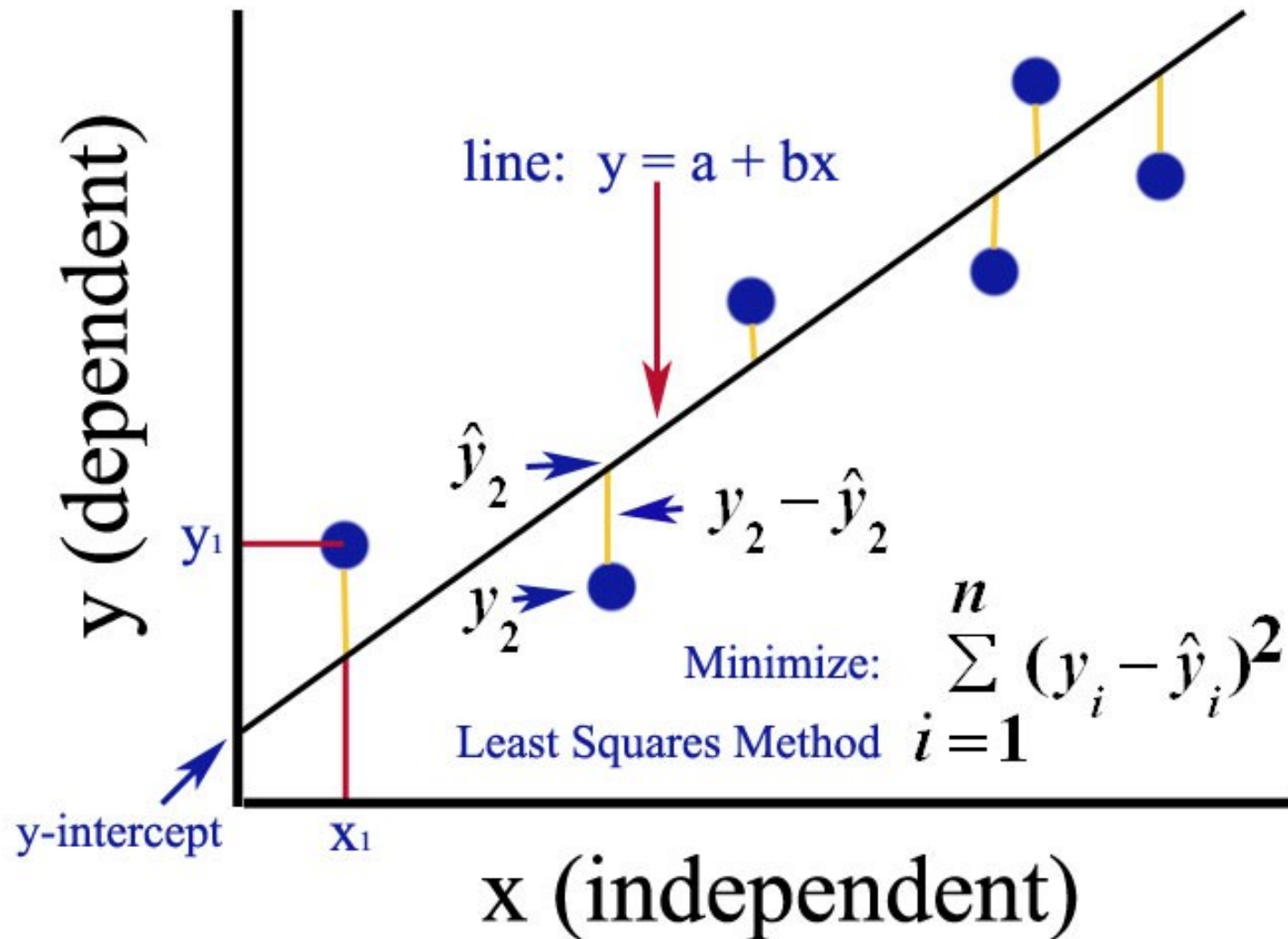
$$f(x) = \alpha + \beta x + \epsilon$$

- **alpha: intercept**
  - **Beta: a weighted slope**
  - **Epsilon: account for the error**
- *Note: This  $f(x)$  will be a straight line for  $x$*

# Just a Formula For a Line!



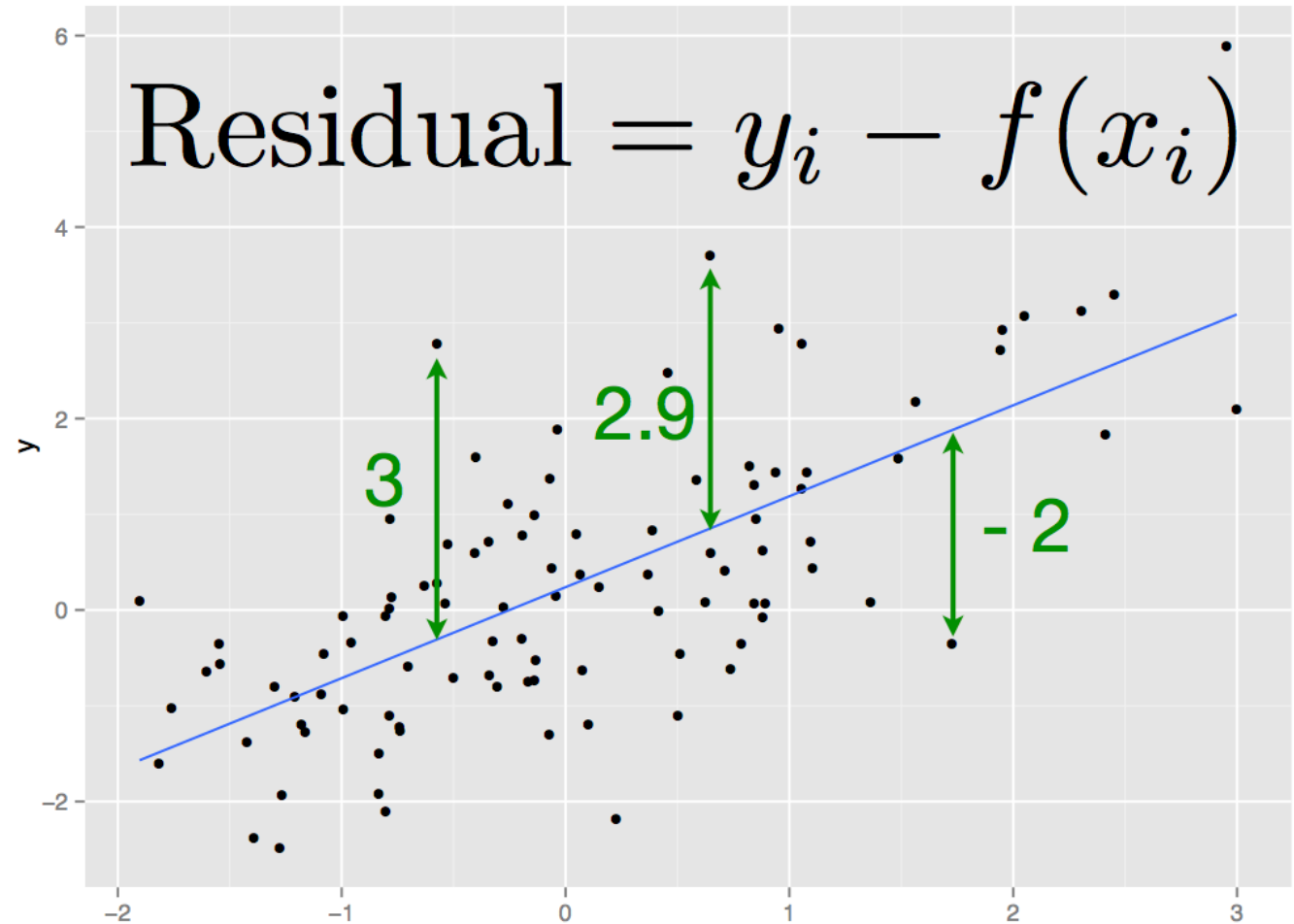
# Another Linear Model





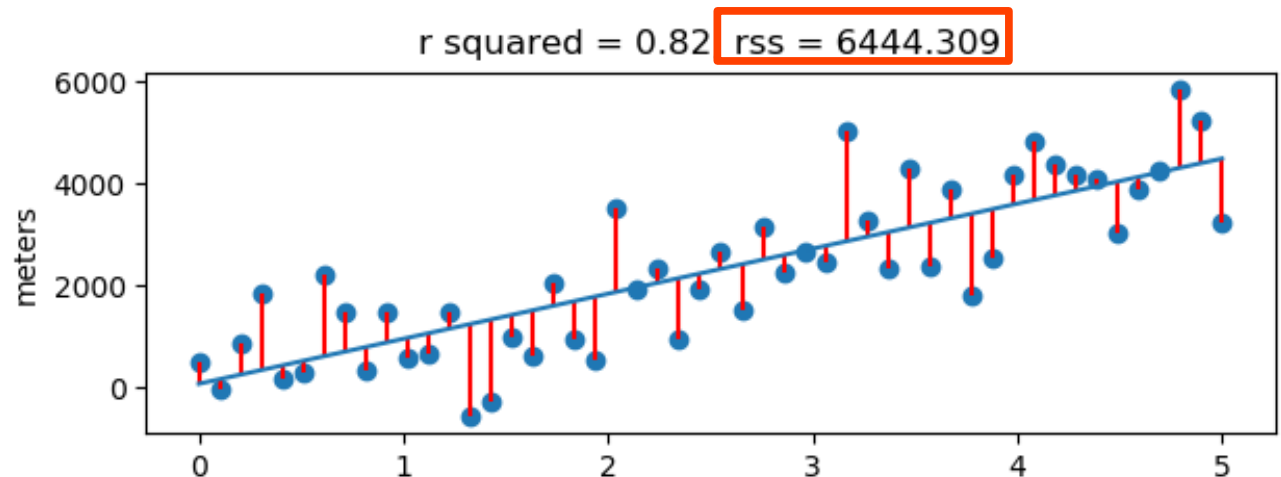
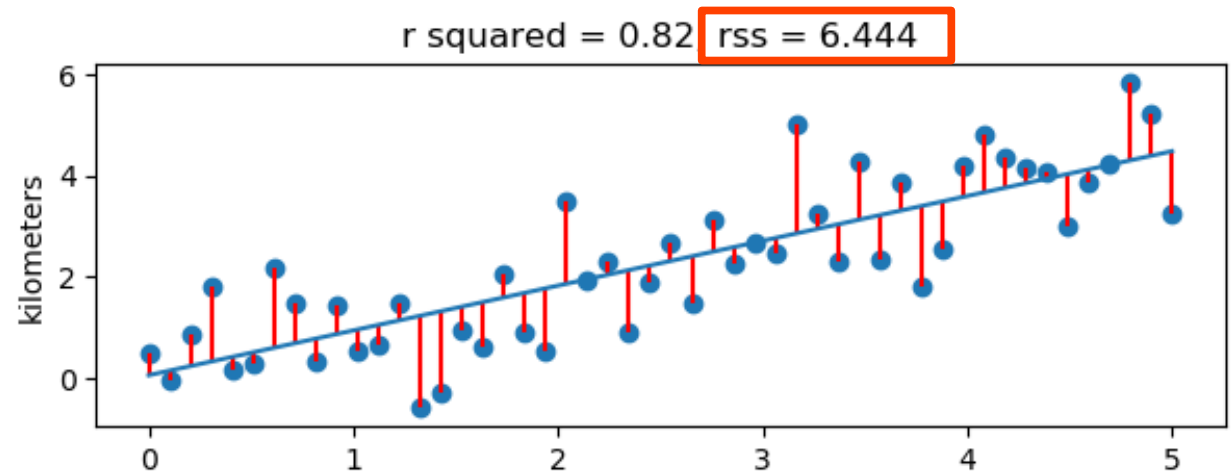
# Residuals

- A residual is the difference between the actual value and the value predicted by the model ( $y - \hat{y}$ ) for any given point.
- A least-squares regression model minimizes the sum of the squared residuals



# A Difference Between the Expected and the Actual Data

- The *Residual Sum of Squares* is a measure of the discrepancy between the data and an estimation model.
- A **small RSS** value indicates a **tight fit** of the model to the data; better prediction





# Types of Questions to Address With Data

**Q1: Is crime influenced by yearly temperature?**

**File: crime.csv**



**Q2: What influence is there on earning potential and personal height?**

**File: wages.csv**

# Crime Data Set



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

```
rm(list = ls()) # remove old vars in memory
library(tidyverse)
# 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
```

```
tibble::as_tibble(crime) # dataframe
```

	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
...					

Yearly low temp

Murder rate

Training data



# Exploratory Plots

#plot with general trend line

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

#plot with linear model line

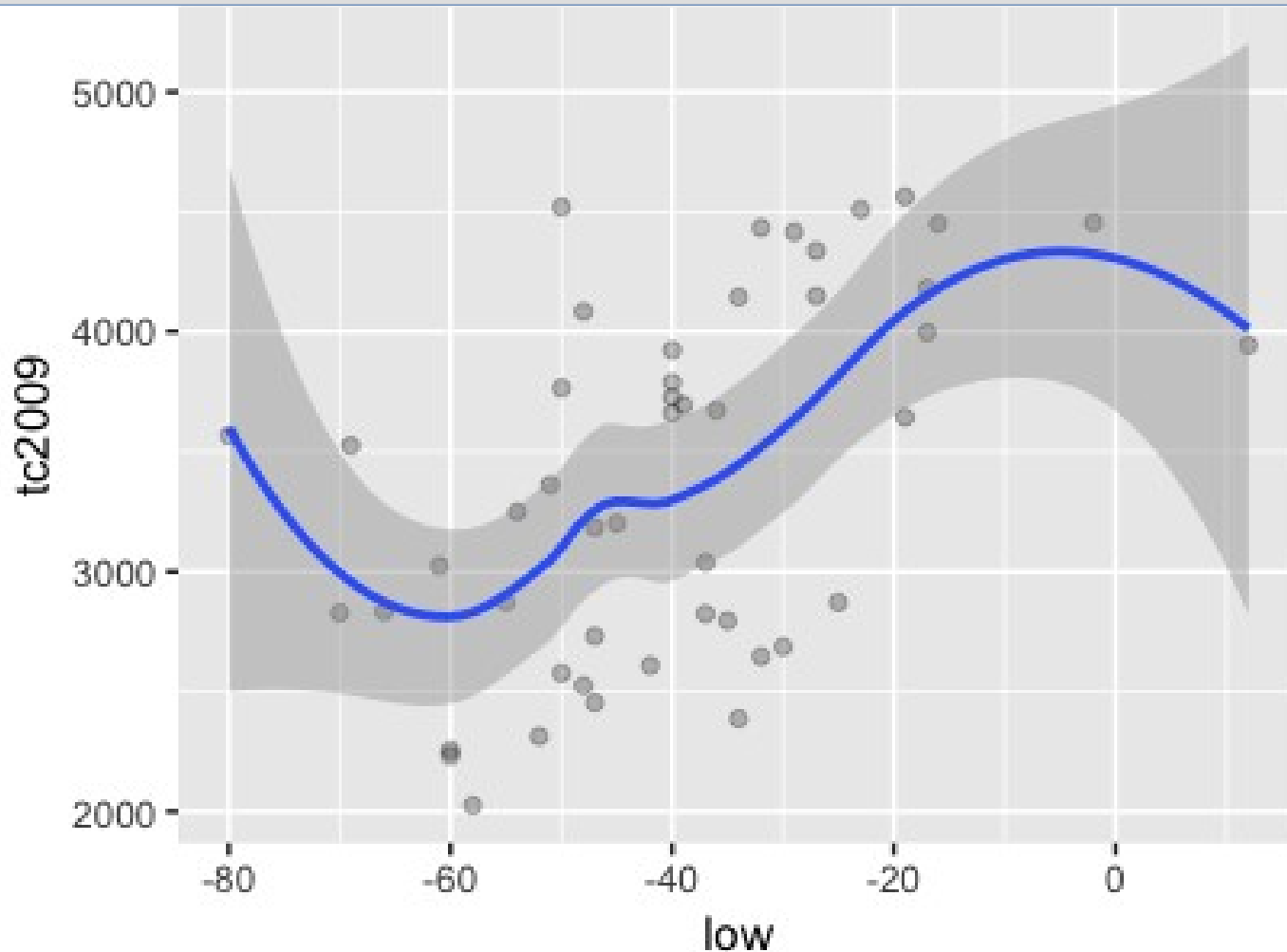
```
crime %>% ggplot(aes(x = low, y = tc2009)) +  
  geom_point(alpha = I(1/4)) +  
  geom_smooth(method = lm)
```





# No Model: Just General Trends

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

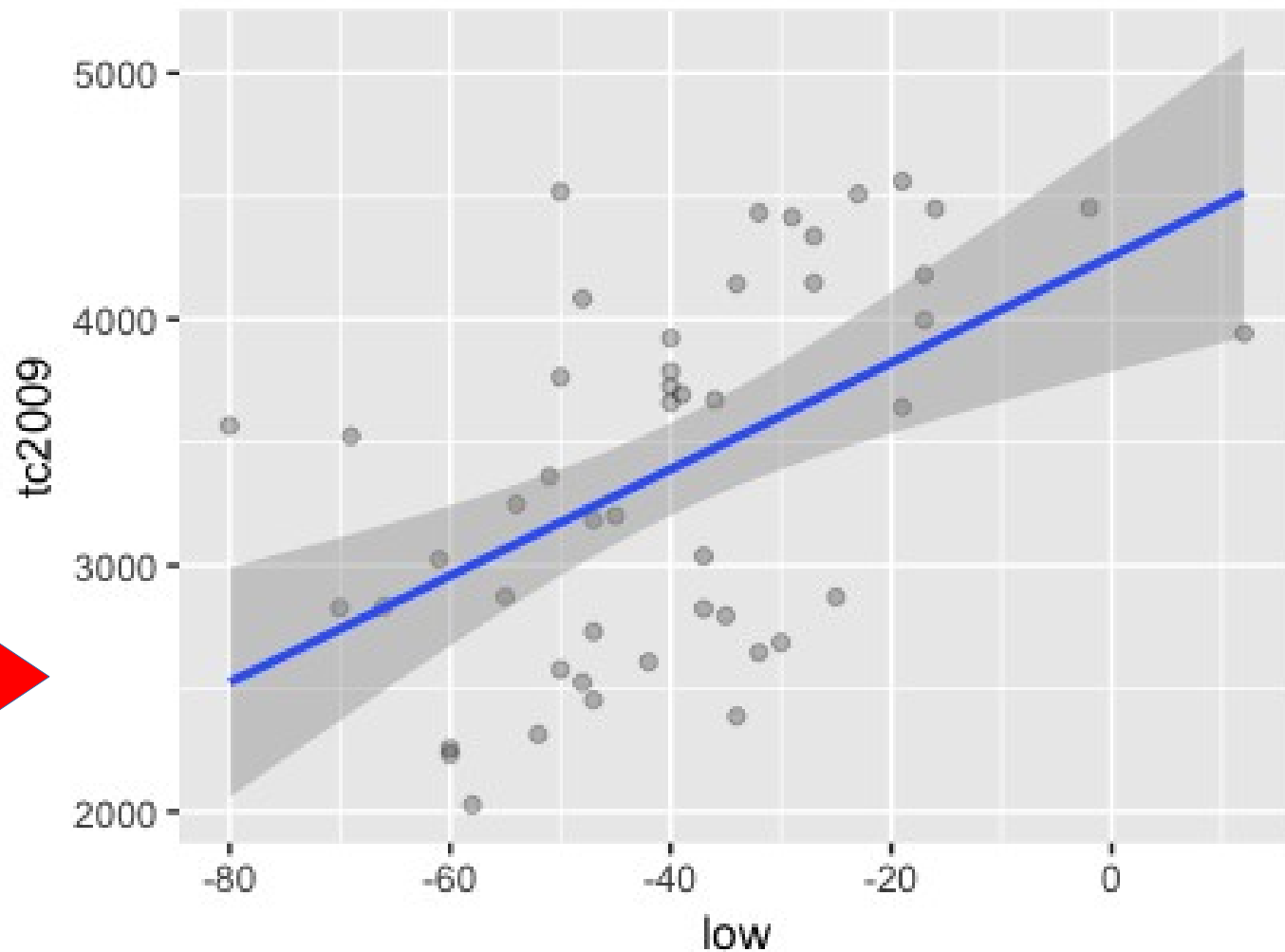


**This is  
the  
data's  
general  
pattern**



# Linear Model: Predictions

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



**The  
linear  
Model:  
used  
for  
predictions**



# The Linear Model

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

lm

Model formula:  
response ~ predictor(s)

data

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



# Syntax

- Formulas only need to include the response and predictor variables

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

Syntax to Build the linear model:

$$y \sim x$$



# Models Use Formulas

- R formulas are expressions built with “” (*tilda*)

```
tc2009 ~ low
```

# Note: **tc2009** is **dependent** variable and **low** is **independent** variable

```
class(tc2009 ~ low)
```

# gives: [1] "formula" meaning that is a line equation

- $tc2009 = f(low) = \beta * low + \epsilon$

# Different Names, Same Meanings

response ~ explanatory

dependent ~ independent

outcome ~ predictors





# Build Your Model!

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

Dependent ~ independent

Call:

```
lm(formula = tc2009 ~ low, data = crime)
```

Coefficients:

(Intercept)	low
4256.86	21.65



# 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)
```

```
# 4256.86158
```

 $\alpha$ 

```
21.64725
```

 $\beta$ 

slope

low



# Interpreting Models

Linear models are very easy to interpret

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

$\alpha$  is the expected value of  $y$  when  $x$  is 0.

$\beta$  is the expected increase in  $y$  associated with a one unit increase in  $x$



# Coefficients

`coef(mod)`

`coefficients(mod)`

# (Intercept) low

# 4256.86158 21.64725

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

$$4256.86 + 21.6 * (-10) = 4040.86$$

$$(x,y) \leftarrow (-10, 4040.86)$$



# Coefficient Calculator

This function is now my data!!

Based on our training using data,  
if  $x = -10$ , then  $y = 4040.86$

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

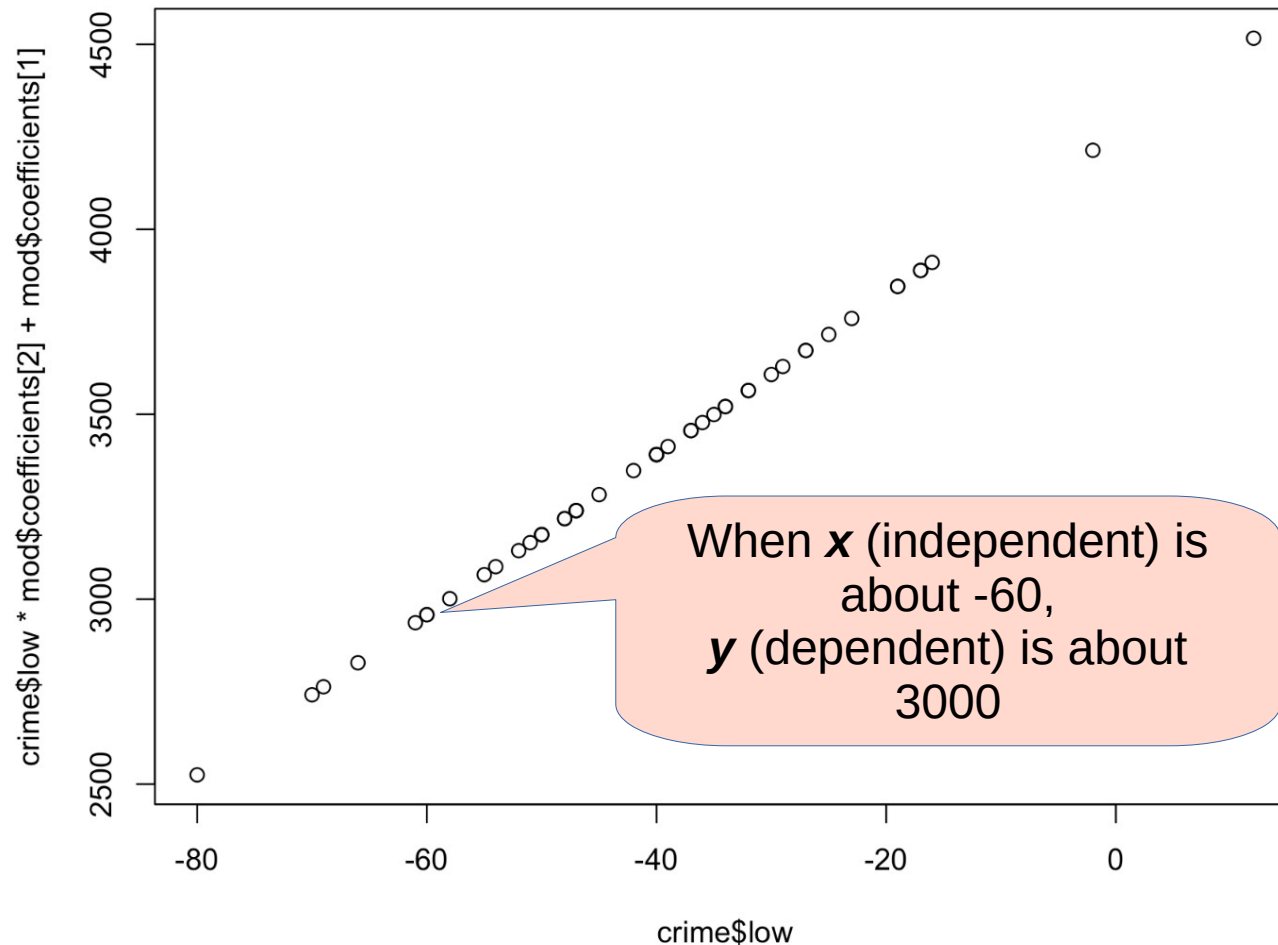
I can predict  $y$ ,  
based on values of  $x$ !

**Due to  
error,  
there  
may be  
a slight  
difference  
between  
expected  
and  
actual  
values**



# Forecasting Trends With a Simple Line

```
plot(crime$low, crime$low*mod$coefficients[2] + mod$coefficients[1] )
```





# Coefficient Calculator Function for *mod*

```
# Function to compute estimated y value for an entered x
value

tellMeY <- function(x_int){

  cat("  intercept :",mod$coefficients[1] )
  cat("\n  slope      :",mod$coefficients[2] )
  y = mod$coefficients[1] + x_int * mod$coefficients[2]
  cat("\n Model predicts y = ",y, "from x = ",x_int)
}

# what if x = -10?
tellMeY(-10) # note: x = -10
```

```
> # what if x = -10?
> tellMeY(-10) # note: x = -10
intercept : 4256.862
slope      : 21.64725
Model predicts y = 4040.389 from x = -10
> |
```

$$\begin{aligned} y &= \text{intercept} + \text{slope} * x \\ &= \text{alpha} + \text{beta} * x \\ &= b + mx \end{aligned}$$



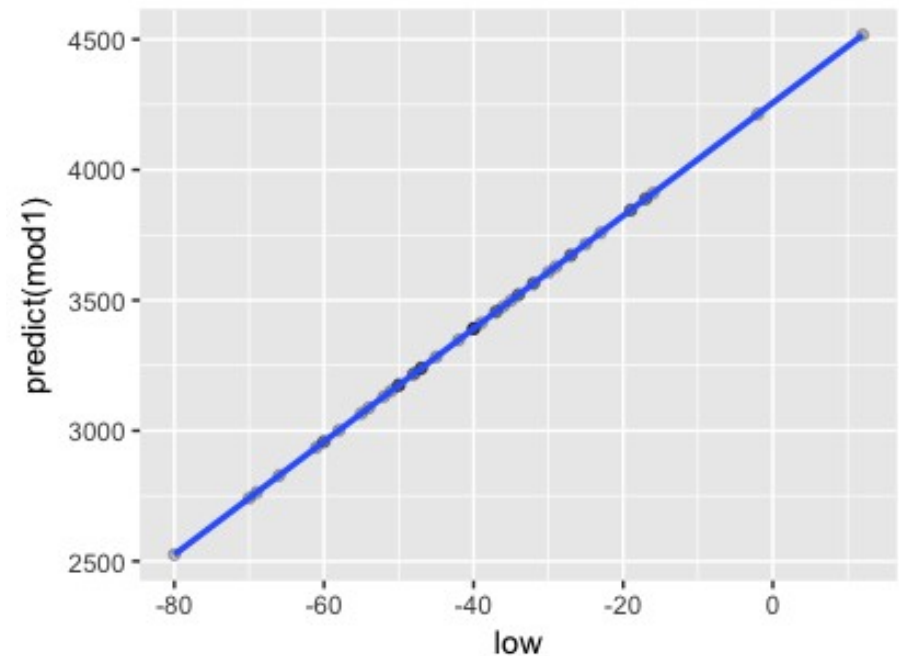
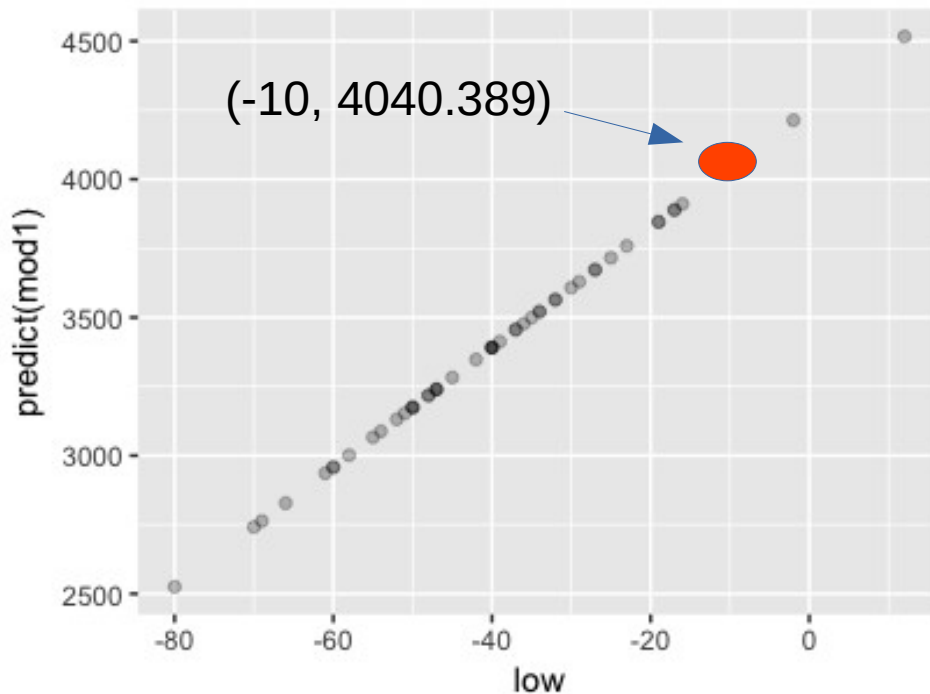


# Forecasting with `predict()`

?predict

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

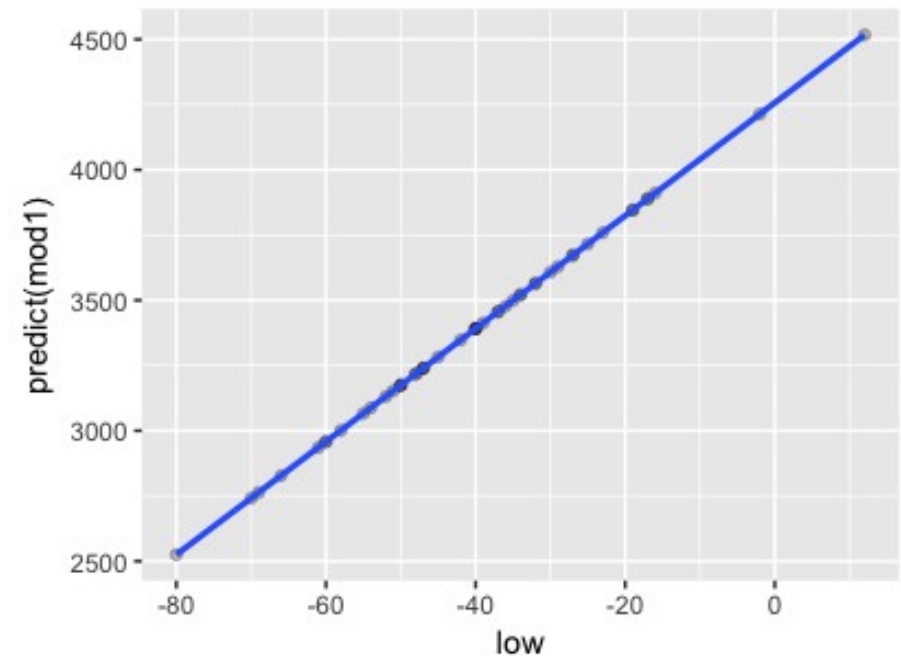
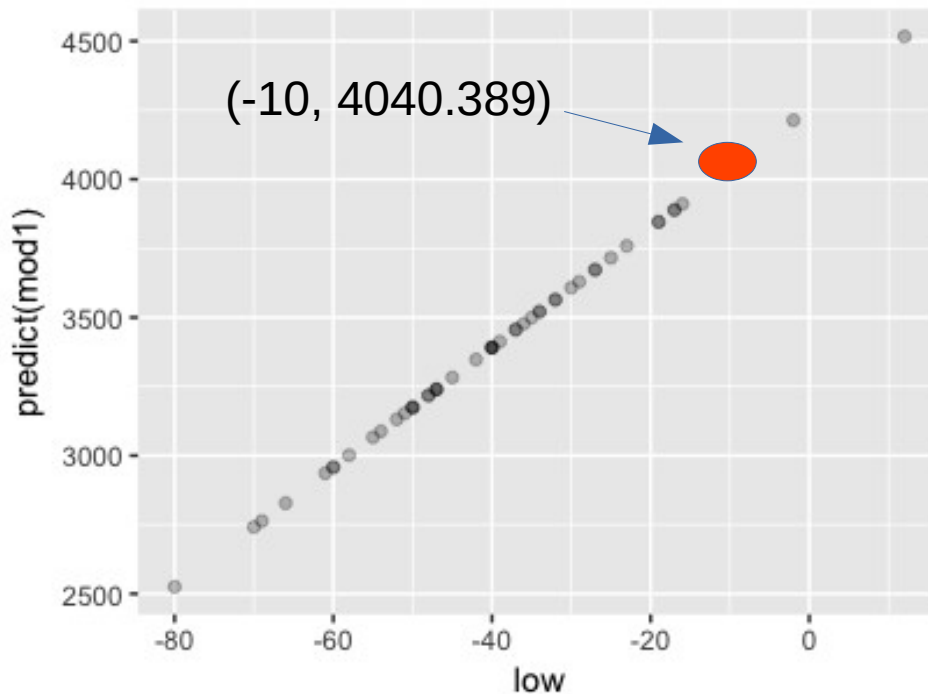
```
crime %>% ggplot(aes(x = low, y = predict(mod))) +  
  geom_point(alpha = I(1/4)) + geom_smooth()
```





# What Conclusions to Draw?

After you have completed the analysis, what does the data indicate?





# So, back to the Question... ?

**Q1: Is crime influenced  
by yearly temperature?**

**File: crime.csv**



*A: The data and its trained model  
suggest that there is a positive  
correlation between crime and  
temperature in the US*



# Aside: intercept terms

R includes an intercept term in each model by default

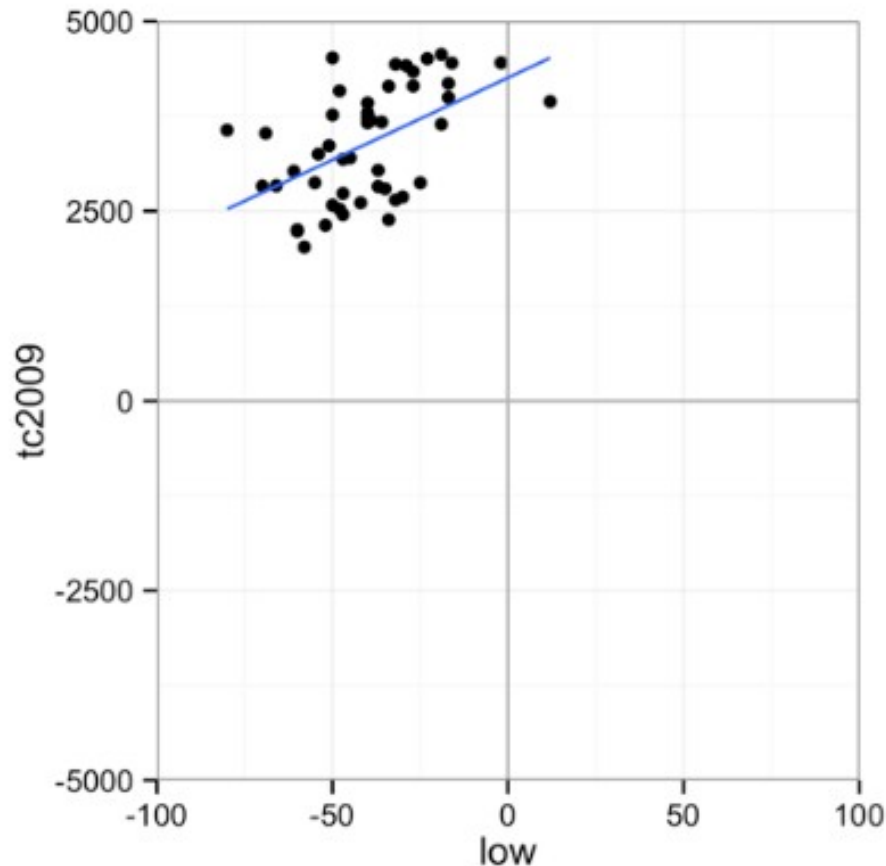
$$y = \alpha + \beta x + \epsilon$$

$$y \sim x$$

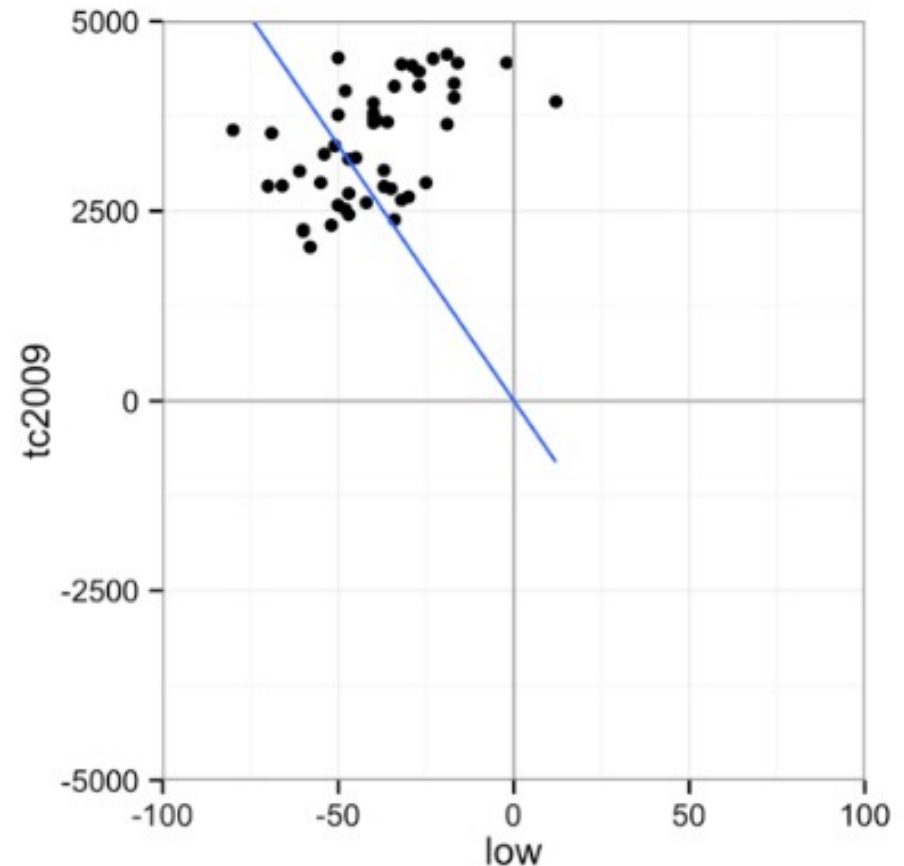


# Study at $x = 0$ ?

(Does  $x = 0$  make sense here?)



**With  $\alpha$**



**Without  $\alpha$**

Every linear model has a y intercept. Including  $\alpha$  lets this term vary. Not including  $\alpha$  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 is relevant (and sometimes it is 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)

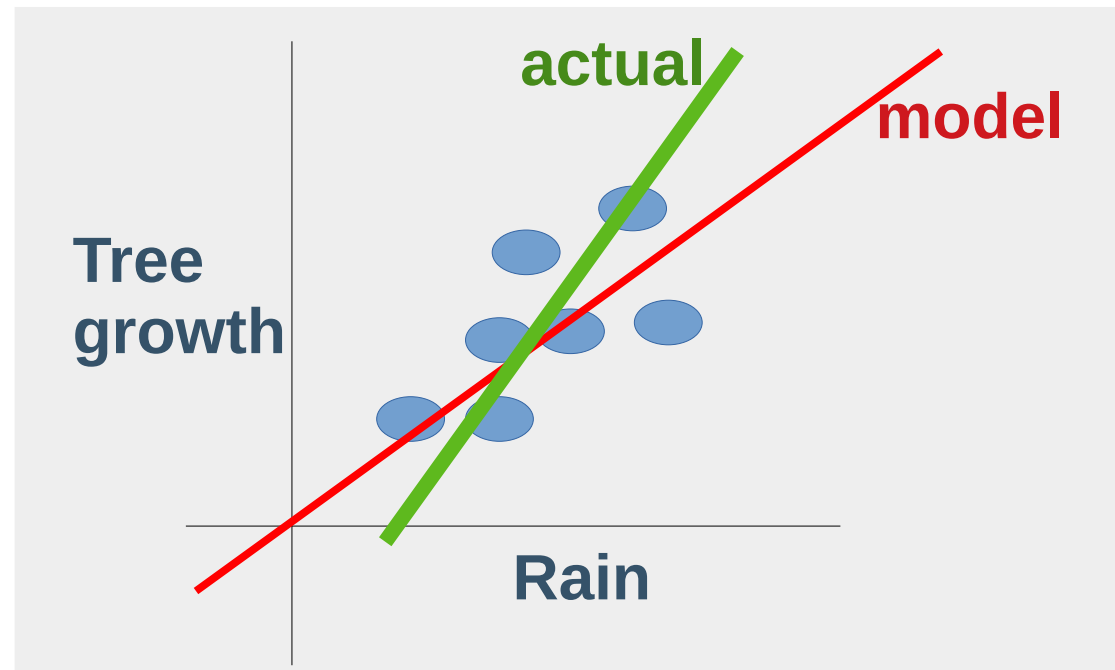


# Study at $x = 0$ ? (Does $x = 0$ make sense here?)

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

If *rain* = 0, then does *tree\_growth* = 0?  
Always? Trees may still grow when sunny out

Intercept not relevant:  
The regression line may cross  $y$ -axis at some other point (other than zero)





# An Intercept Term: To Use or Not?

FYI: 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)
```





# Let's Test An Intercept

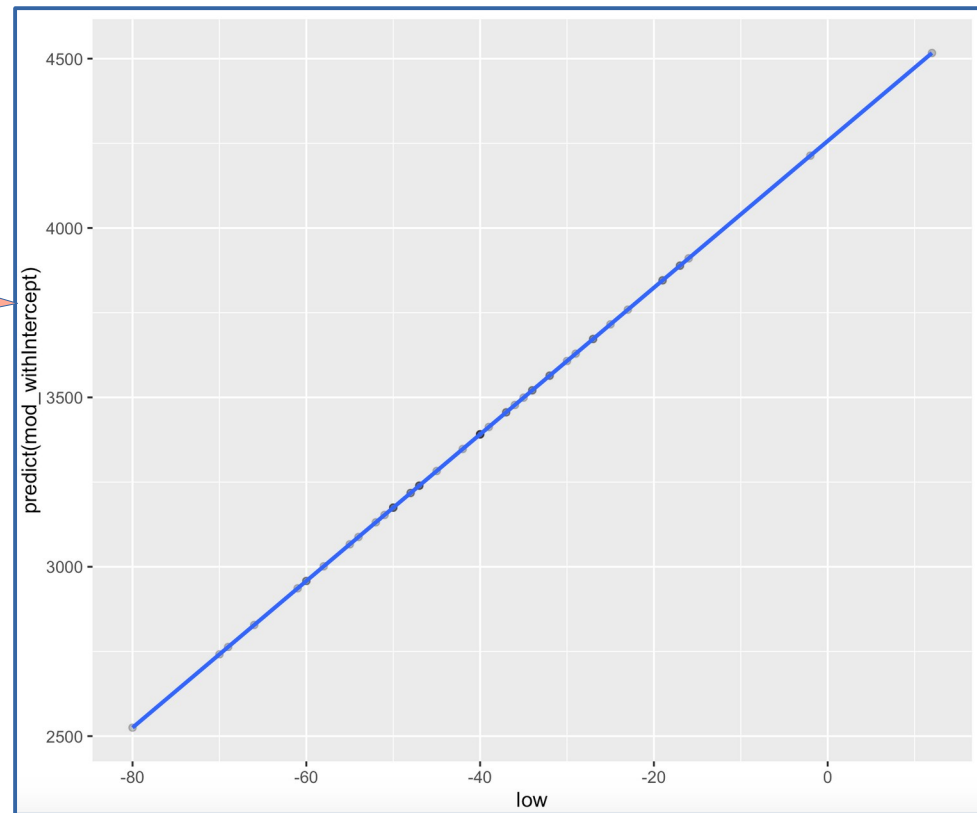
## Add the intercept

```
# equivalent - includes intercept
```

```
mod_withIntercept <- lm(tc2009 ~ 1 + low, data = crime)
```

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

Does this  
represent  
your data?





# Let's Test An Intercept

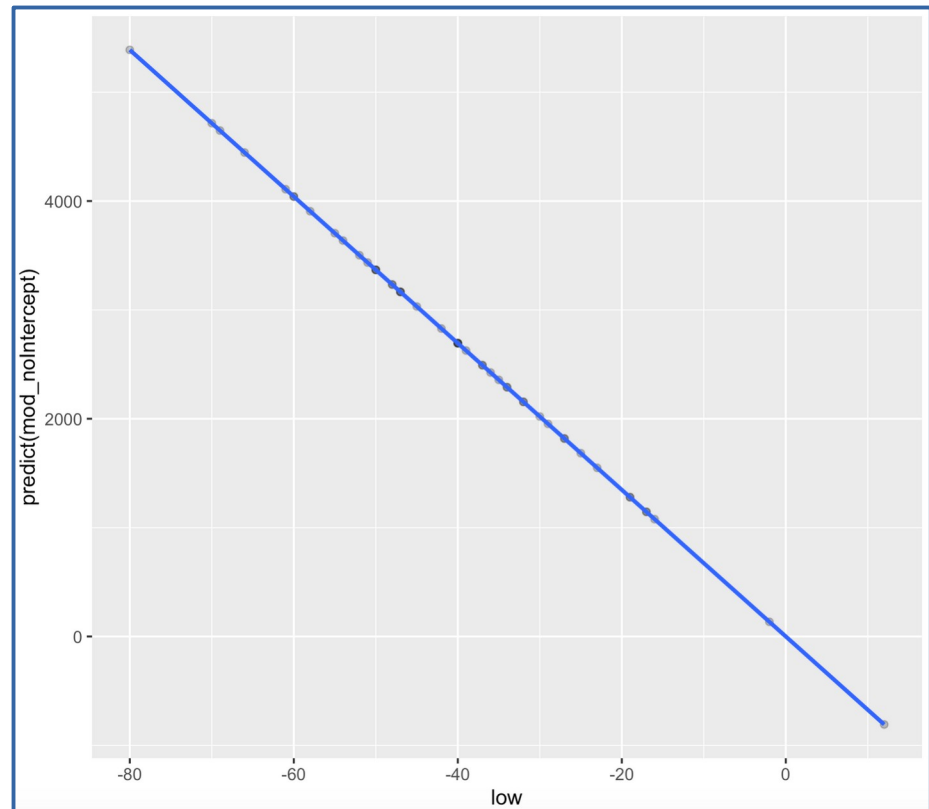
Remove the intercept

```
# equivalent - removes intercept
```

```
mod_noIntercept <- lm(tc2009 ~ low - 1, data = crime)
```

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

Does this  
represent  
your data?





# 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

- Goodness of fit of a model: The  $R^2$  coefficient of determination describes how well the regression predictions approximate the real data points.
- ***How well do the indep vars explain the dep var?***

$R^2 = 1$ , indep variable(s) predict dep variables

$R^2 = 0$ , no prediction

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

# $P$ -Value and Model Significance

- *How do I know if this model is any good?*
- If,
  - $p$ -value is between 0 and 0.01 (or)
  - $p$ -value is between 0 and 0.05
- Then the model is significant
- The closer to zero, the better the model.

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



# Study the p-Value

```
#Create a simple model  
myX <- 0:100  
myY <- myX + 1  
mod <- lm(myY ~ myX)  
summary(mod)
```



# Study the p-Value

```
#Create a simple model  
myX <- 0:100  
myY <- myX + 1  
mod <- lm(myY ~ myX)  
summary(mod)
```

```
Call:  
lm(formula = myY ~ myX)  
  
Residuals:  
      Min       1Q   Median       3Q      Max   
-1.000e-13 -3.420e-16  1.157e-15  2.304e-15  1.242e-14  
  
Coefficients:  
              Estimate Std. Error  t value Pr(>|t|)      
(Intercept)  1.000e+00  2.057e-15  4.862e+14   <2e-16 ***  
myX          1.000e+00  3.553e-17  2.814e+16   <2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 1.041e-14 on 99 degrees of freedom  
Multiple R-squared:  1,    Adjusted R-squared:  1  
F-statistic: 7.921e+32 on 1 and 99 DF, p-value: < 2.2e-16
```

Warning message:

In summary.lm(mod) : essentially perfect fit: summary may be unreliable

# Types of Questions to Address With Data

Q1: Is crime influenced by yearly temperature?

File: [crime.csv](#)



Q2: What influence is there on earning potential and personal height?

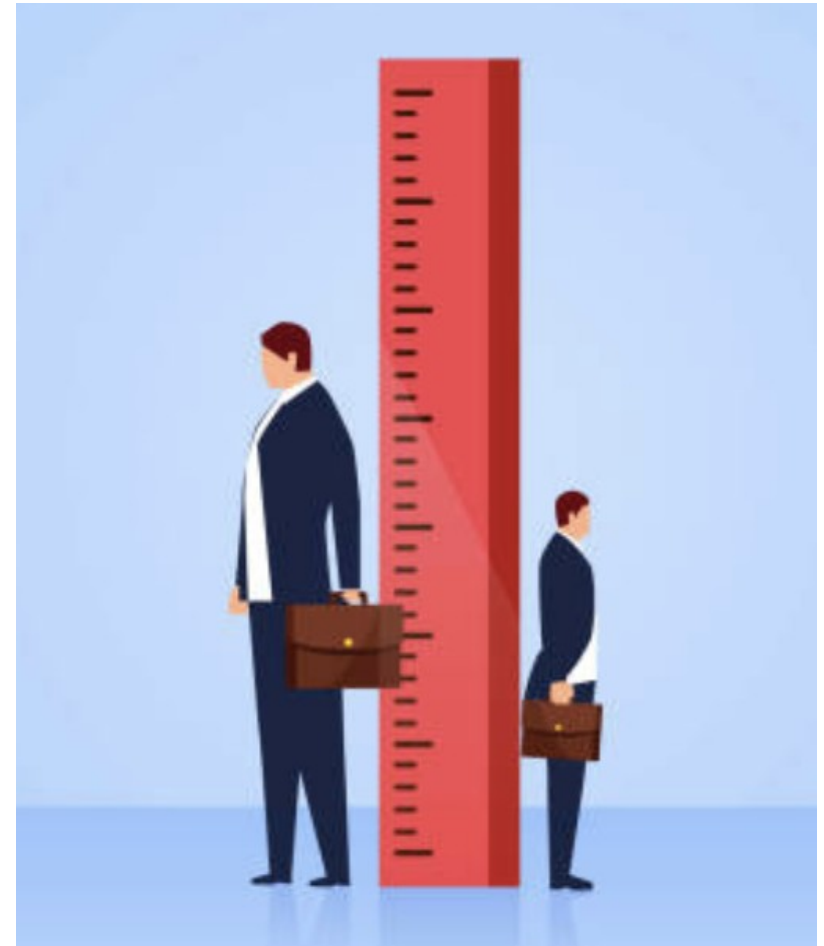
File: [wages.csv](#)





# Consider This!

Let's try making a model with the other data set. Is there a connection between height on earning potential.





# Load the Wages Data

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

```
rm(list = ls()) # remove old vars  
# open the wages.csv dataset from  
the data.  
  
w <- file.choose() # set the  
filename  
  
wages <- read.csv(w) # load and  
read the data.
```

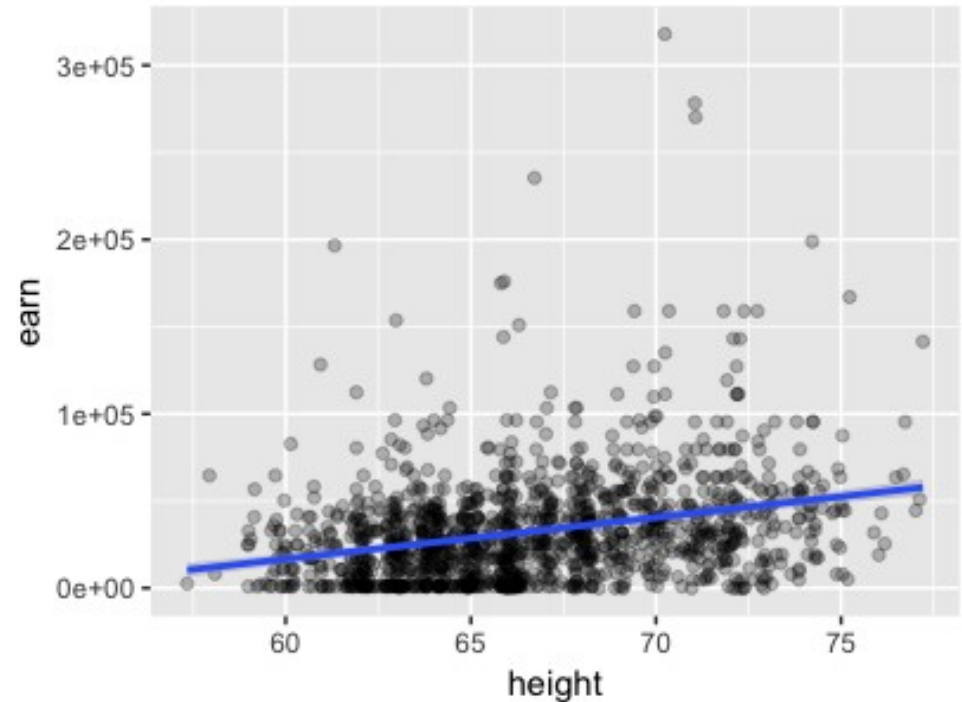
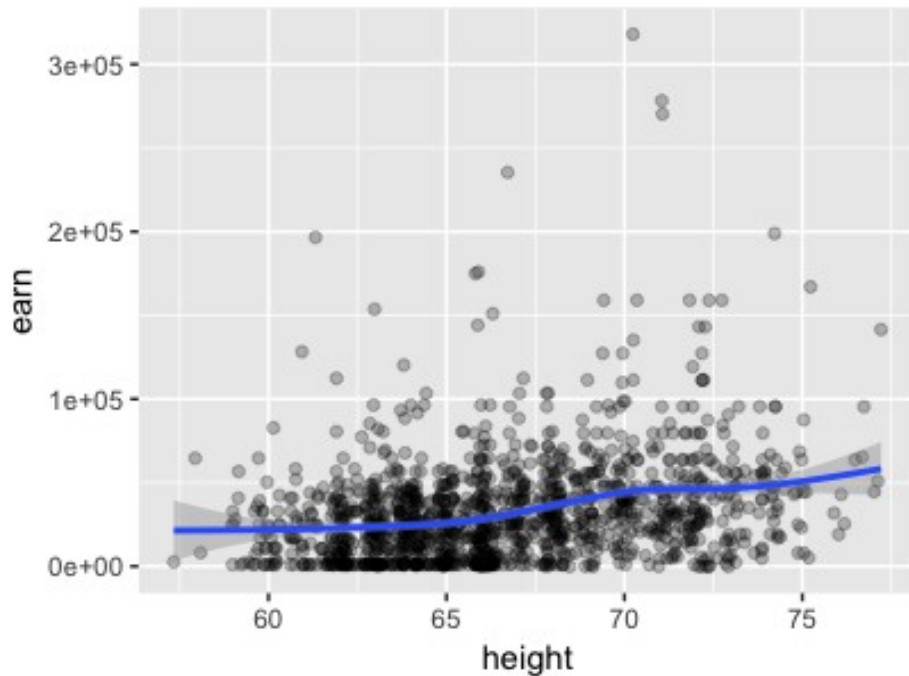


# Do *Tall* People *Earn* More?

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

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

**# Try switching the x's and y's for another view.**





# Correlations: Earn and Height

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

```
cor(wages$earn, wages$height, method =  
"pearson")
```

```
> # Find correlations using the "pearson" method  
> cor(wages$earn, wages$height, method = "pearson")  
[1] 0.2916002
```

Positive correlation

# Make a Model

```
hmod <- lm(earn ~ height, data = wages)  
summary(hmod)
```

Where **dependent** var is *earn*

And **independent** var is *height*

$$\textcircled{y} = \alpha + \beta \textcircled{x} + \epsilon$$



# Summary of Model

`summary(hmod)`

Build your  
model's  
line equation  
from these  
coefficients!

```
> summary(hmod)
```

```
Call:
```

```
lm(formula = wages$earn ~ wages$height)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-47903	-19744	-5184	11642	276796

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-126523	14076	-8.989	<2e-16 ***
wages\$height	2387	211	11.312	<2e-16 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 29910 on 1377 degrees of freedom
```

```
Multiple R-squared:  0.08503,    Adjusted R-squared:  0.08437
```

```
F-statistic:   128 on 1 and 1377 DF,  p-value: < 2.2e-16
```



# *Earn* Regressed Over *height*

```
hmod <- lm(earn ~ height, data = wages)
```

```
coef(hmod)
```

```
## (Intercept)          height
```

```
## -126523.359      2387.196
```

$$\textit{earn} = \alpha + \beta \times \textit{height} + \epsilon$$



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



# An Estimation

The best estimate of earn for someone 68 inches tall is

$$\text{earn} = -126523.36 + 2387.20 \times 68 + \epsilon$$

$$\text{earn} = 35806.24$$



# Conclusions

## *Earn* Regressed Over *height*

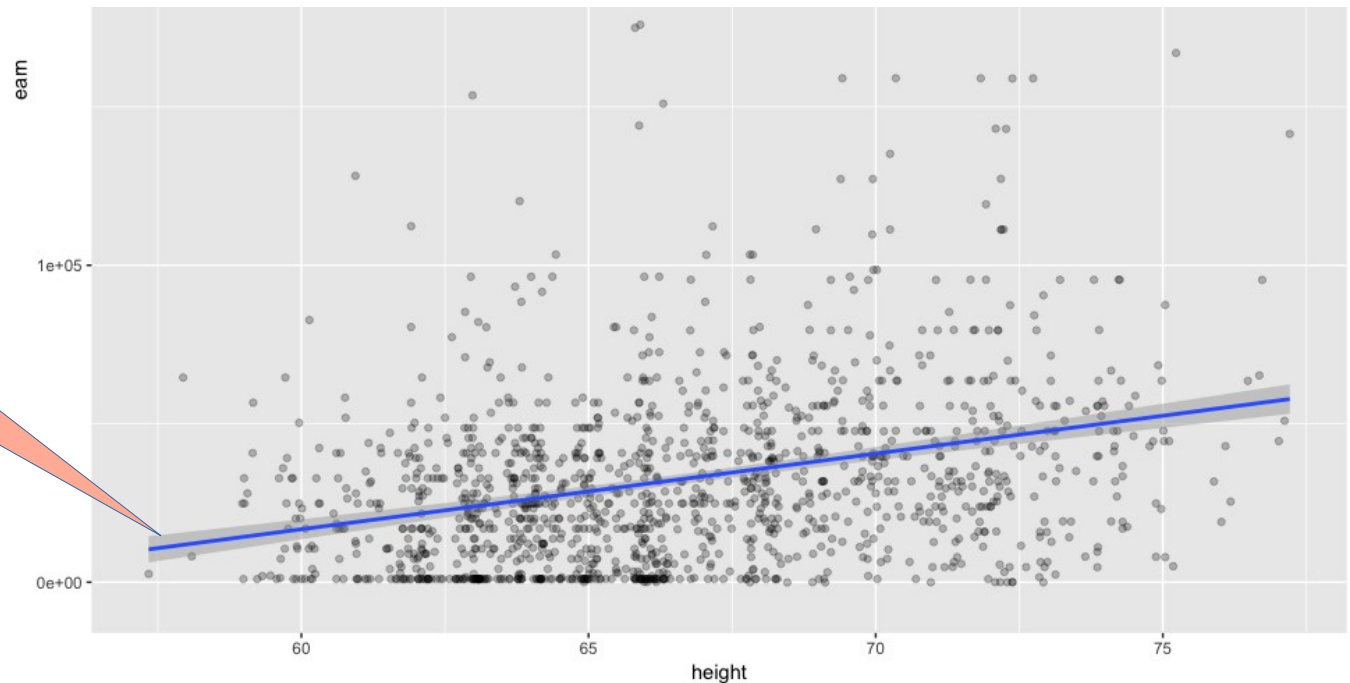
```
cor(wages$earn, wages$height, method = "pearson")
```

- Slight, but positive

```
hmod <- lm(earn ~ height, data = wages)
```

- Significant

Our model  
with the  
data points





# 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*?

Or, How does ***race*** influence ***earn***?

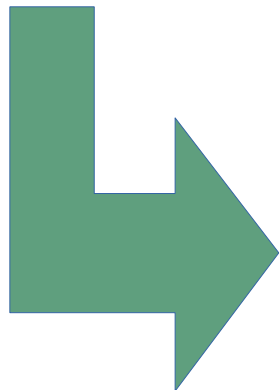


# Summary

```
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  
 $28372.09 = 28372.09$

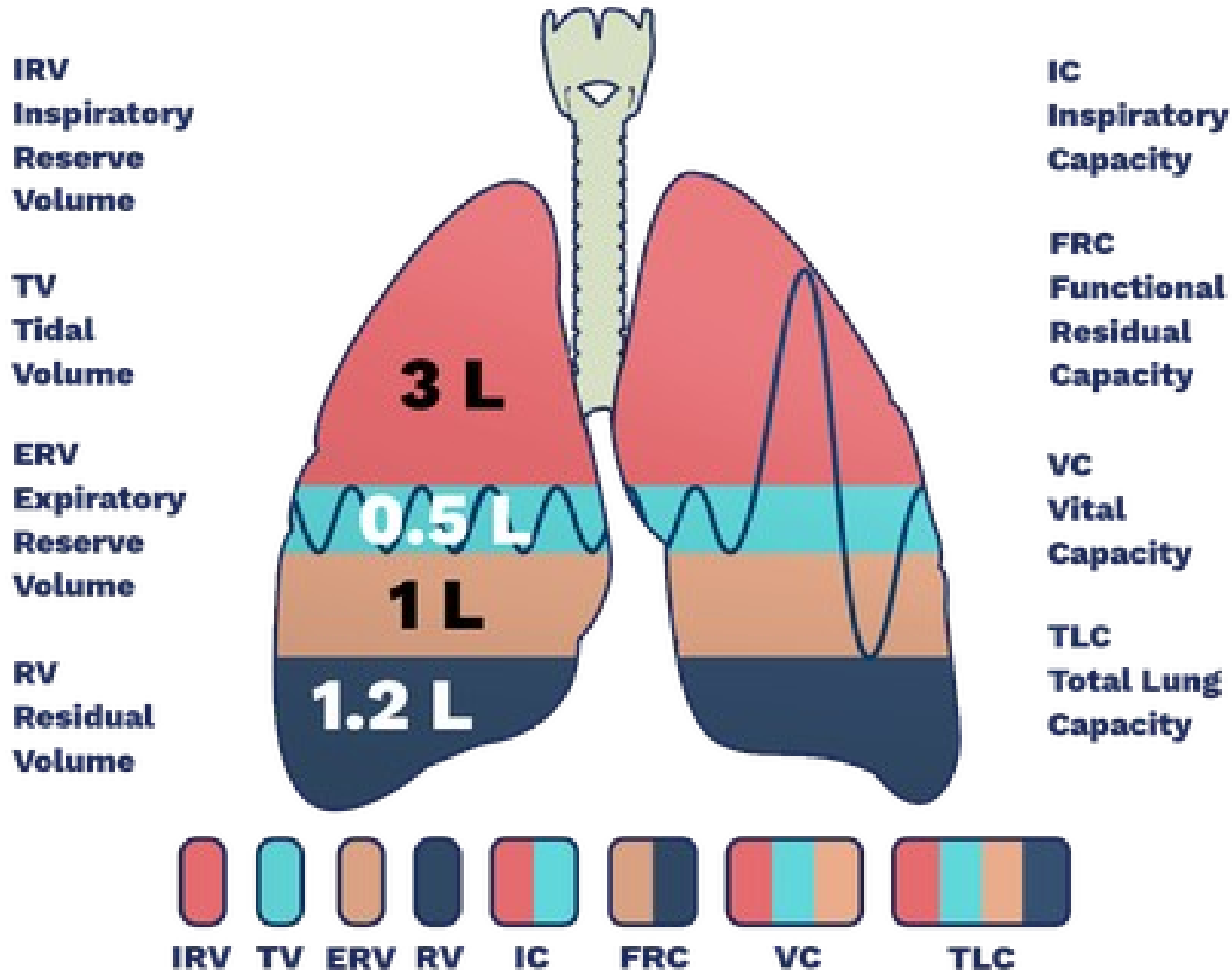


# What Do Plots Also Indicate??



```
ggplot(data = wages) +  
  geom_point(mapping = aes(y = earn, x = height, color = race )) +  
  geom_smooth(mapping = aes(y = earn, x = height )) + facet_wrap(~race)
```

# Lungs Capacity



File: lungCapDemo.R