Data Analytics

CS390

Basic Statistics

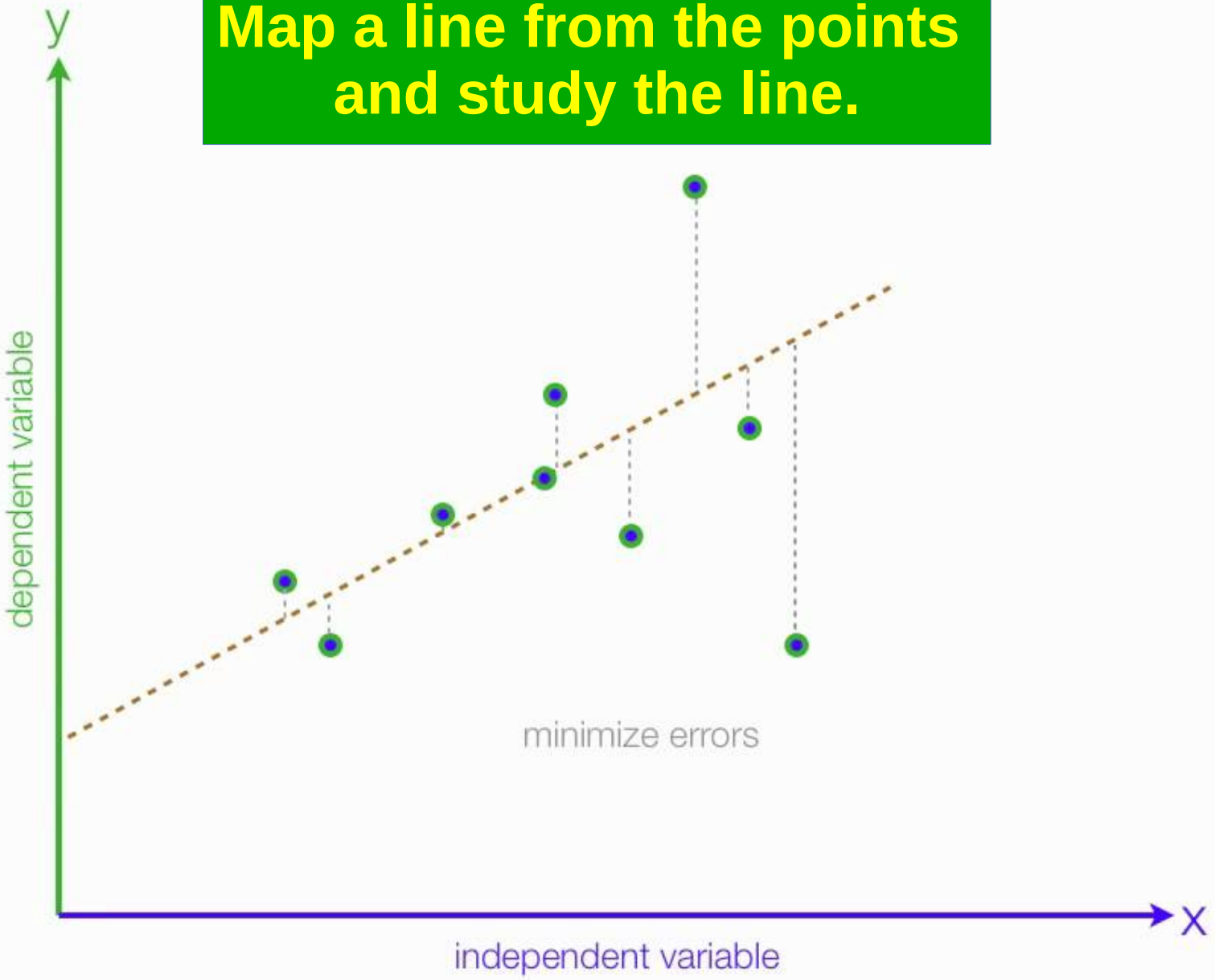
Fall 2017

Oliver Bonham-Carter

From last time...

Linear Regression

Map a line from the points and study the line.



From last time...

Linear Regression

- Is one thing able to influence another thing?
- A linear approach for modeling the relationship between a scalar dependent variable ***y*** and one or more explanatory variables, or independent variables, denoted by ***x***.
- *Simple linear regression*: Single explanatory variable; **models x and y**
- *Multiple linear regression*: More than one explanatory variable (***y's***); **models x and y1, y2**

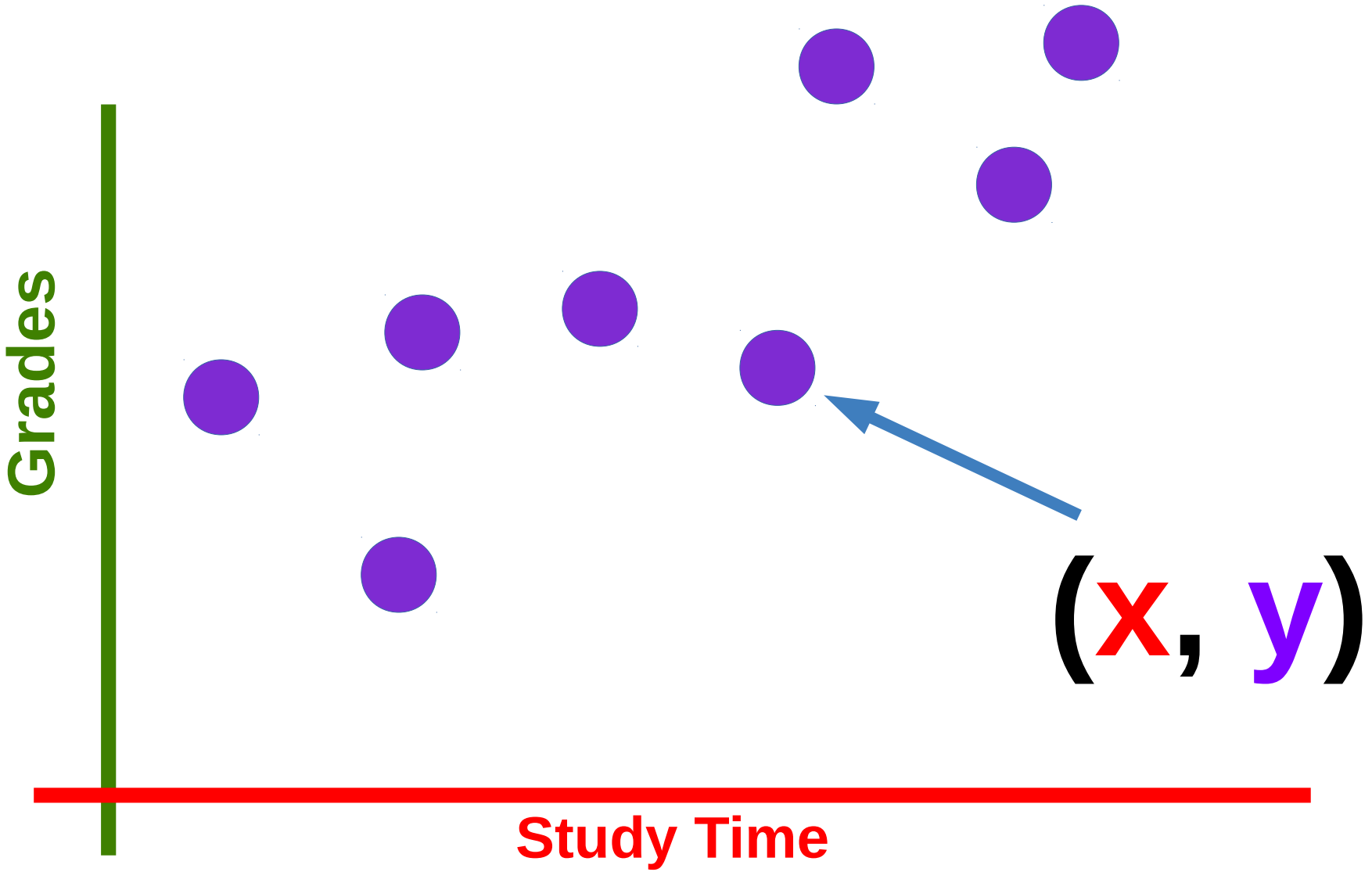
From last time...

Linear Regression

- A straight line is drawn through a dot cloud.
- As the independent variable progresses, what is the dependent variable doing? Is there a relationship?
- The line has a y-intercept and a slope and can be used to determine the positive or negative relationship

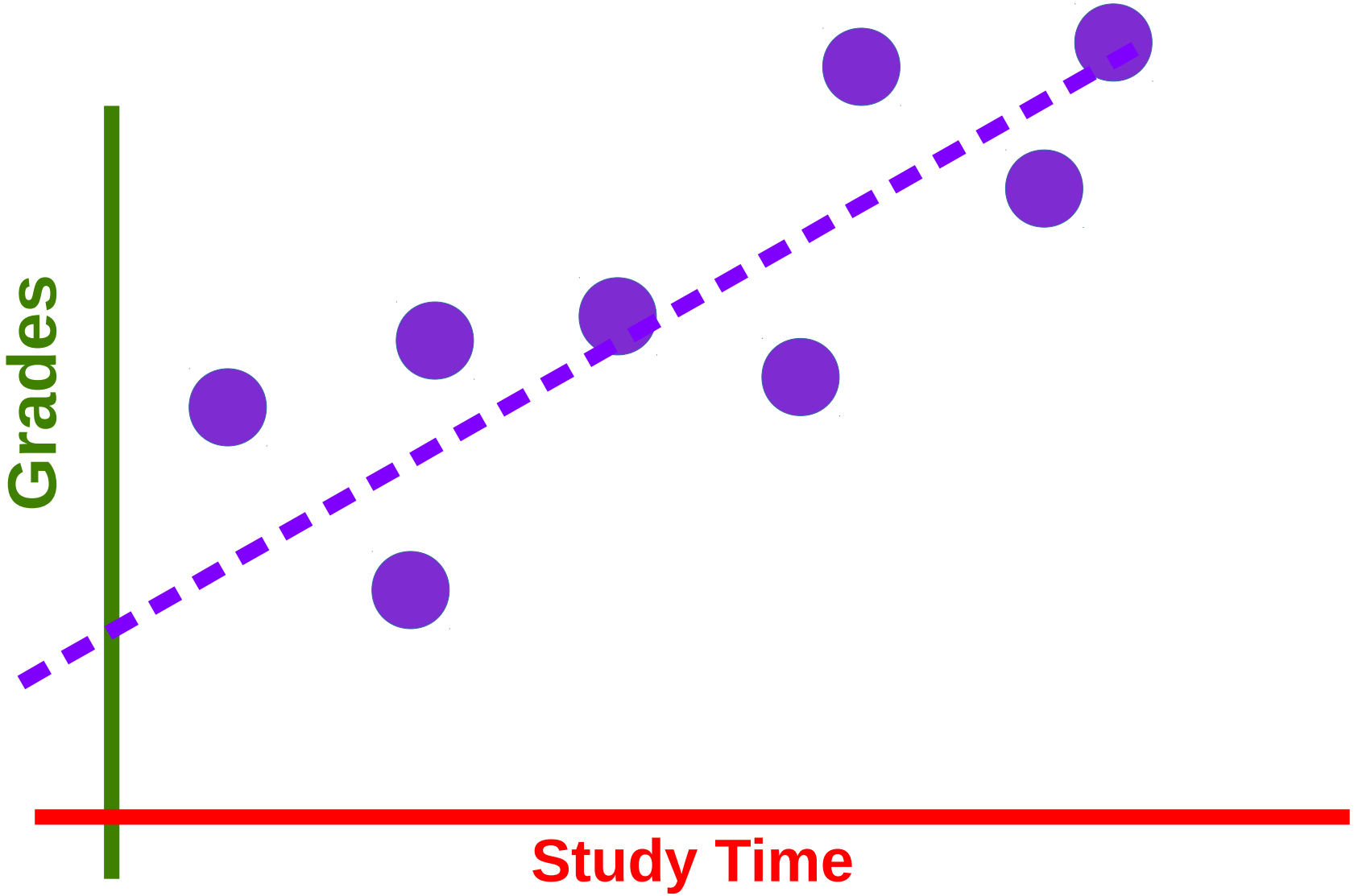
From last time...

Plot *Study Time* to *Grades* Points

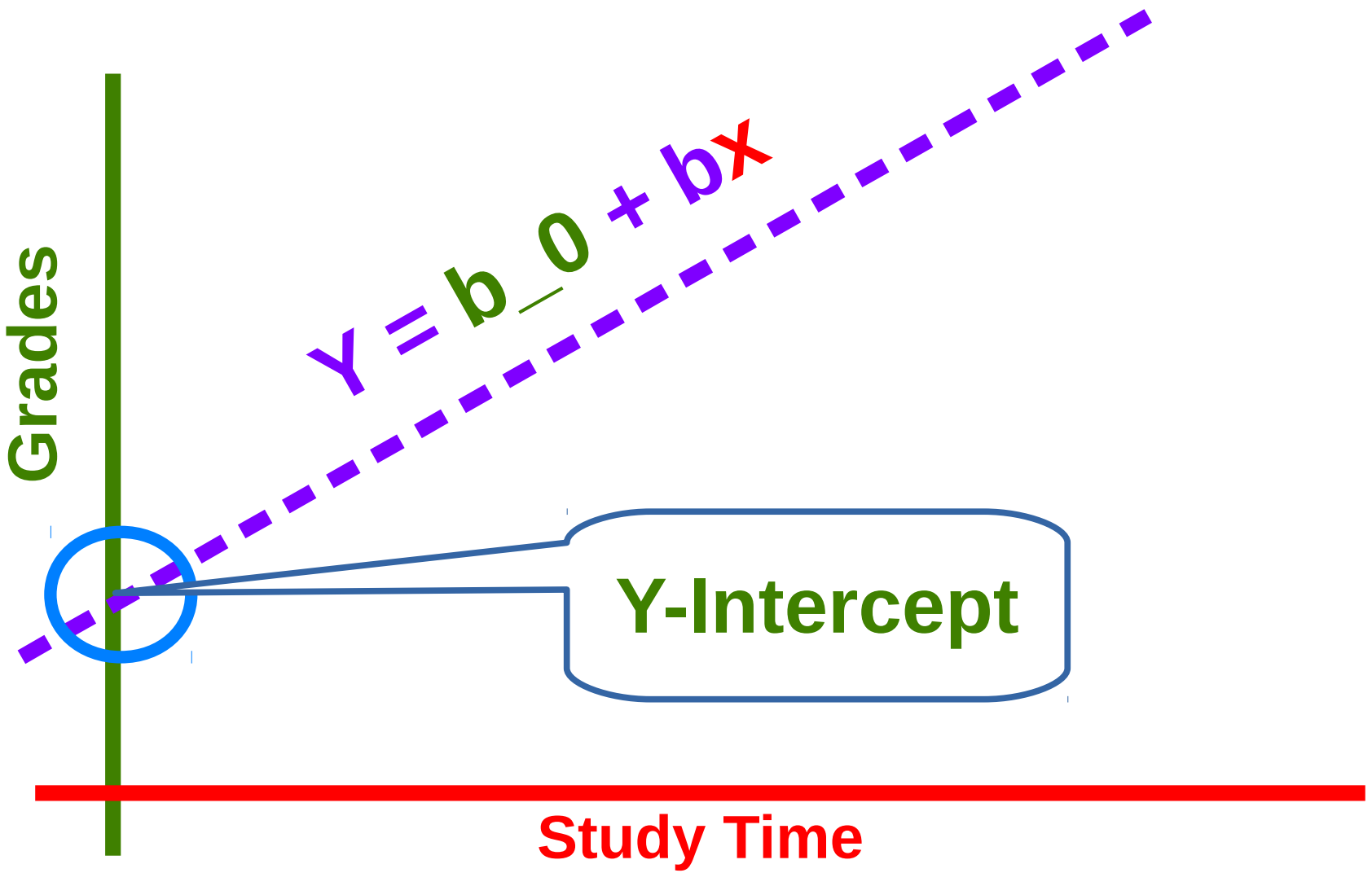


From last time...

Draw Line Through Points



Intercept and Slope: Positive Relationship



ALLEGHENY COLLEGE

Linear Regression

```
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept
summary(lm.D9)
```

- H0: there is no relationship between vars, m =0
- Ha: There is a relationship between vars, m!=0

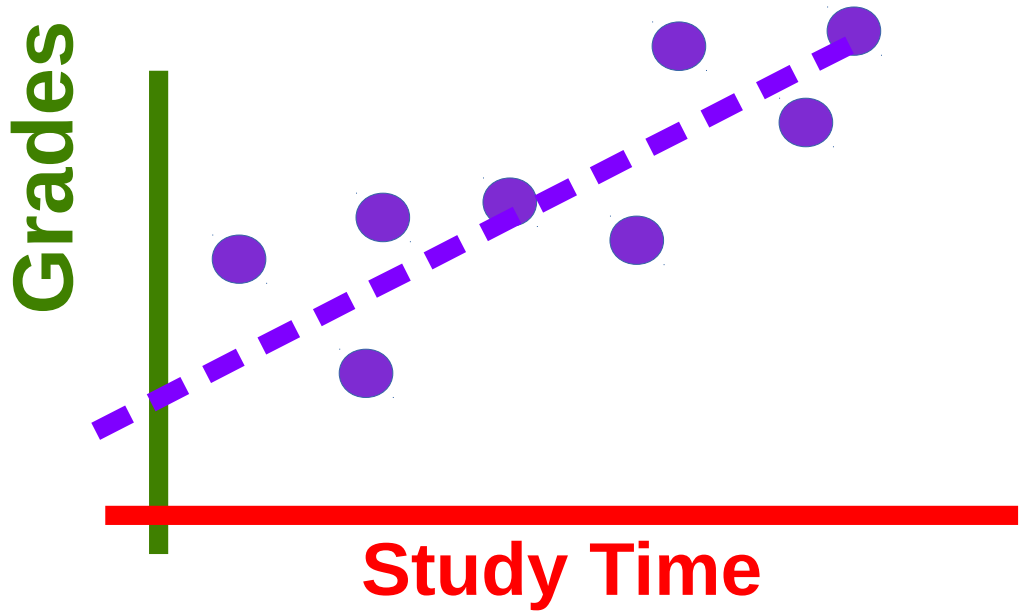
  # Check the p-value:
  - If p-val =< alpha = 0.05: reject H0.
  - If p-val > alpha = 0.05: do not reject H0.

Regression Assumptions

- The regression has five key assumptions:
  - Linear relationship
  - Multivariate normality
  - No or little multicollinearity
  - No auto-correlation
  - Homoscedasticity

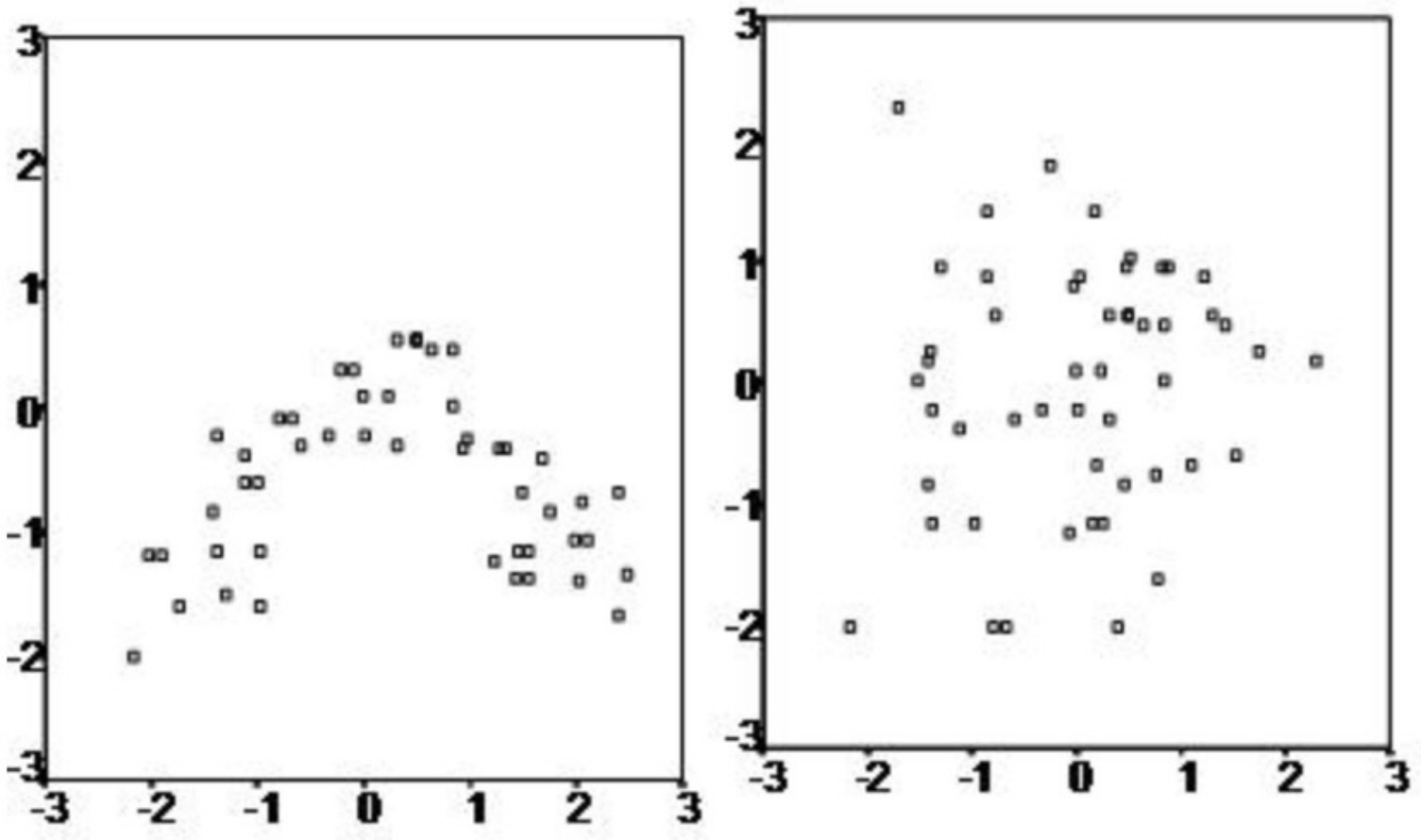
Linear Relationship

- Linear regression needs the relationship between the independent and dependent variables to be *linear*.
- Check for outliers linear regression is sensitive to outlier effects.



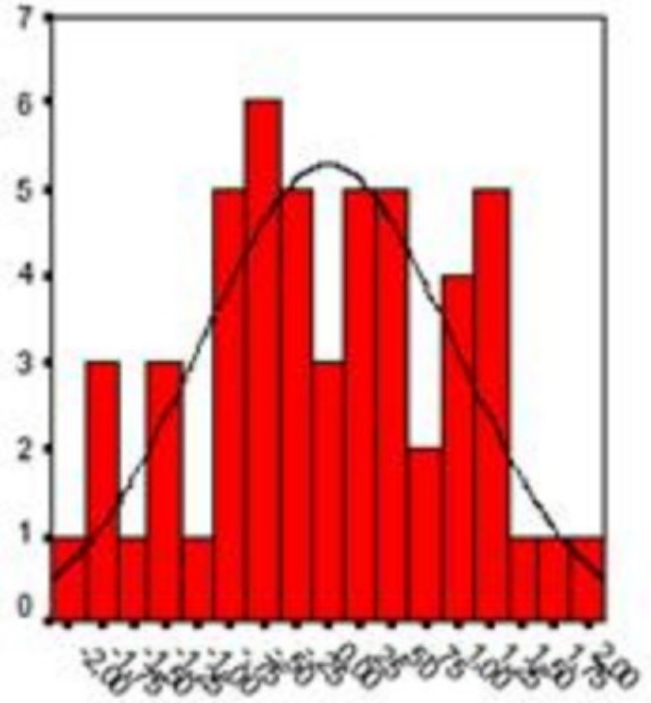
Linear Relationship

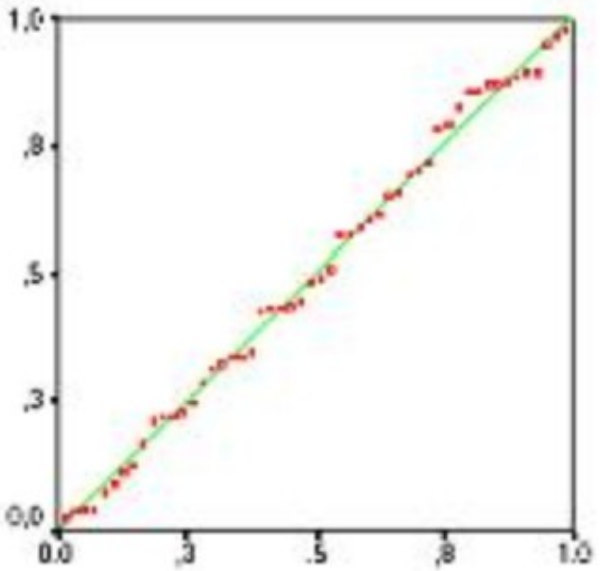
- Scatter plots: See where no and little linearity is present.



Multivariate Normality

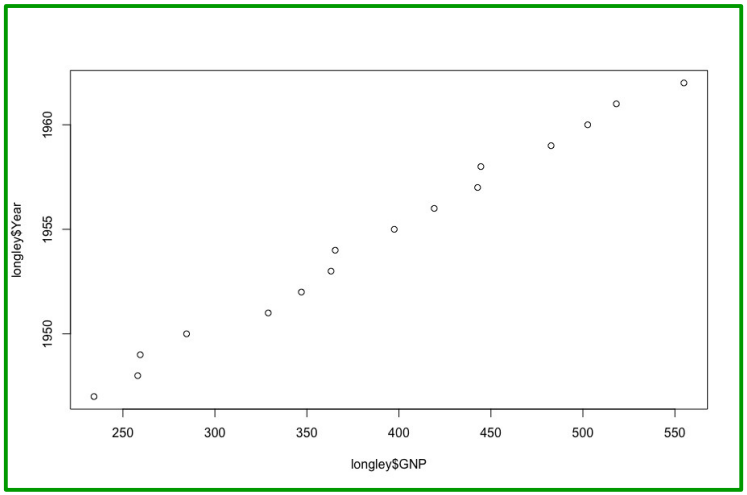
- The data must be of a *normal* distribution
- Check this with a QQ-plot

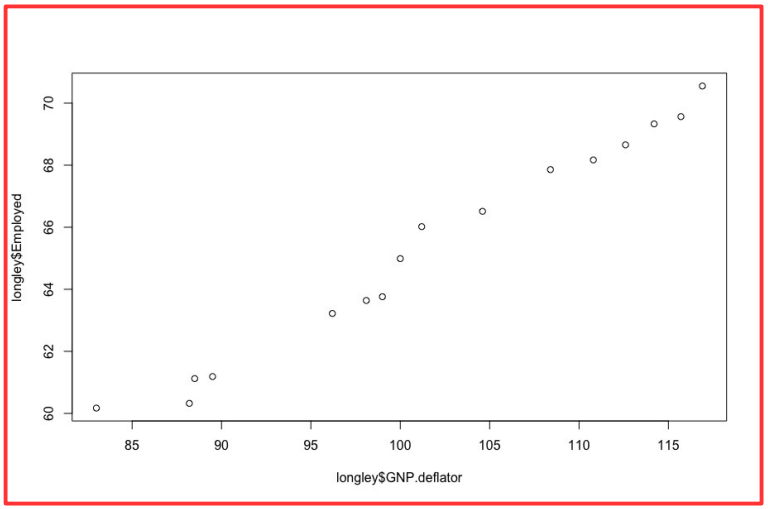




Multivariate Normality

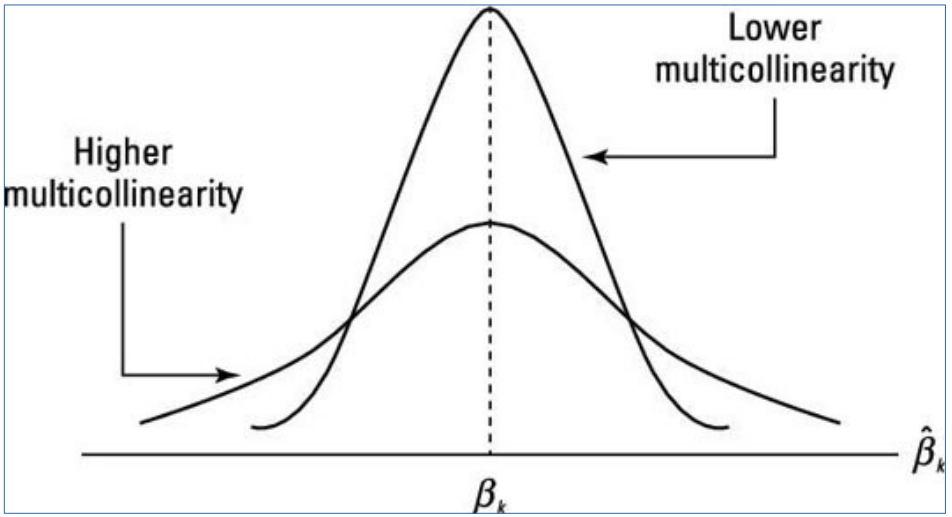
```
# Good
qqplot(x = longley$GNP, y = longley$Year)
# Not so good
qqplot(x = longley$GNP.deflator, y = longley$Employed)
```

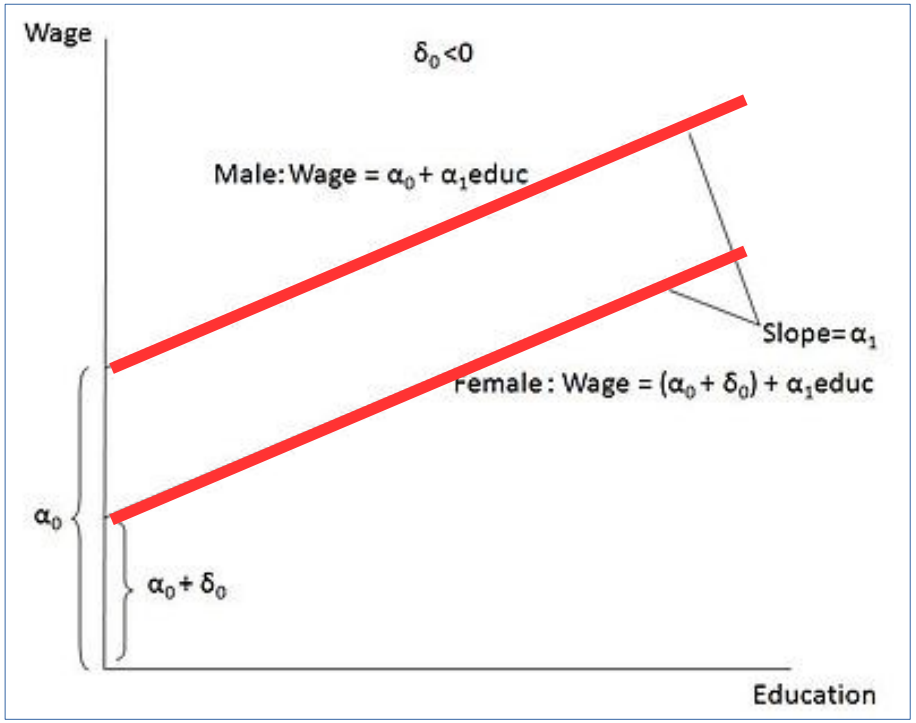




Multicollinearity

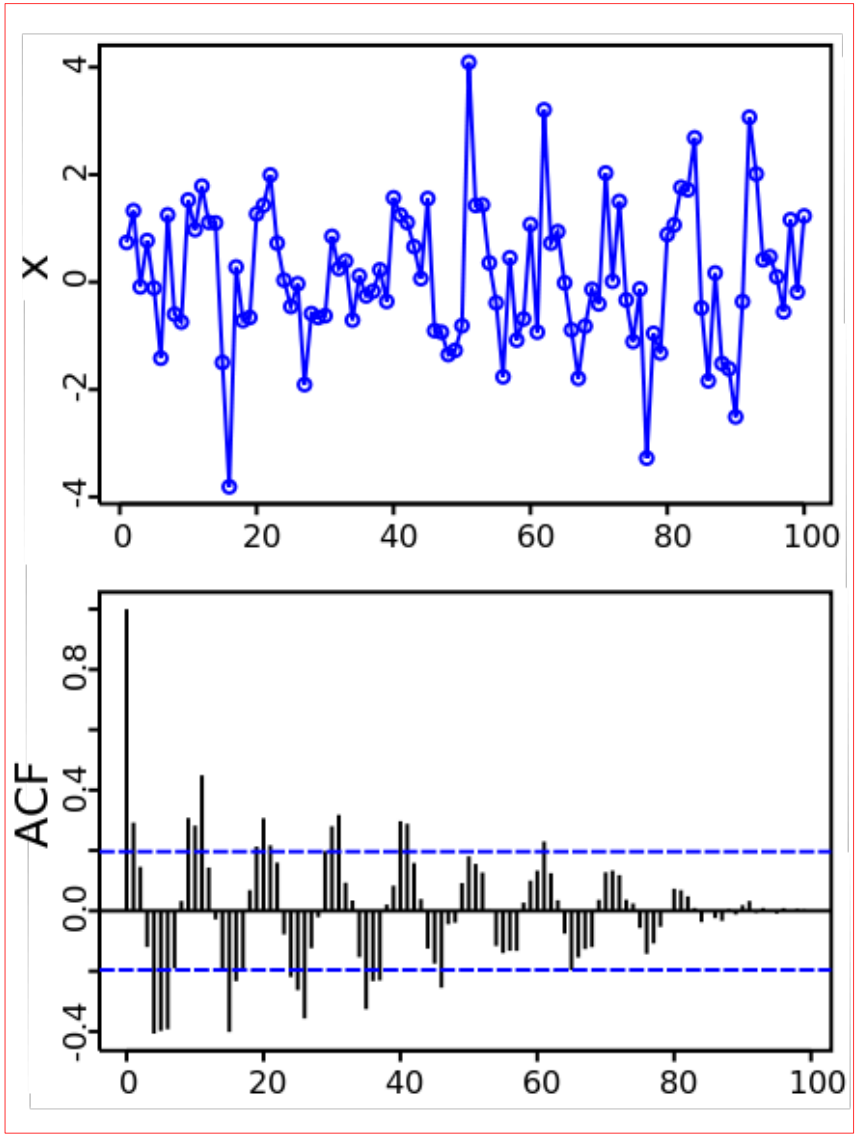
- A phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.
- Same slope; same line





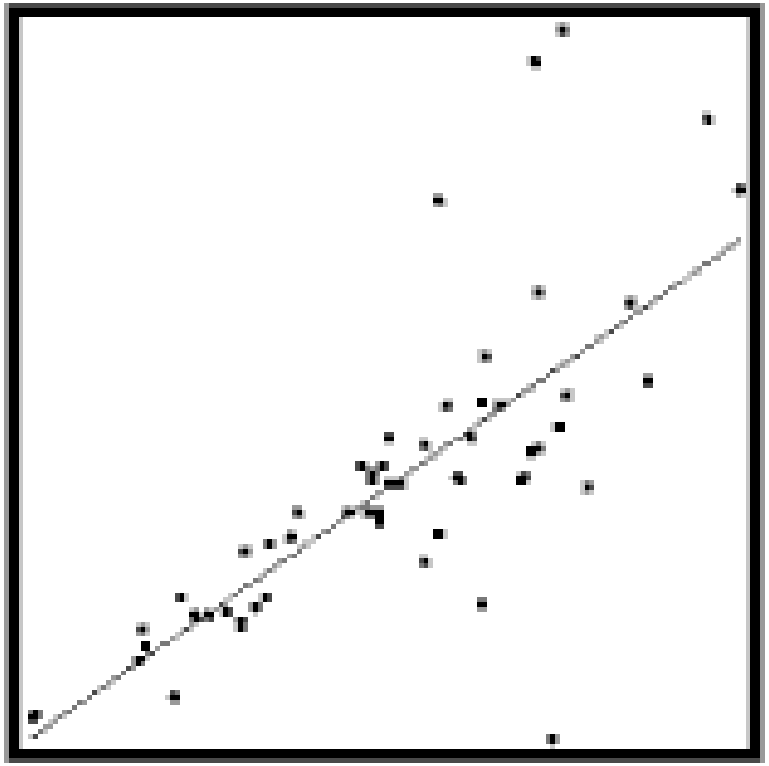
No Auto-correlation

- The correlation of a signal with a delayed copy of itself as a function of delay
- Ex: A plot of a series of 100 random numbers concealing a sine function. The sine function revealed in a correlogram produced by autocorrelation.
- Result: Non random output



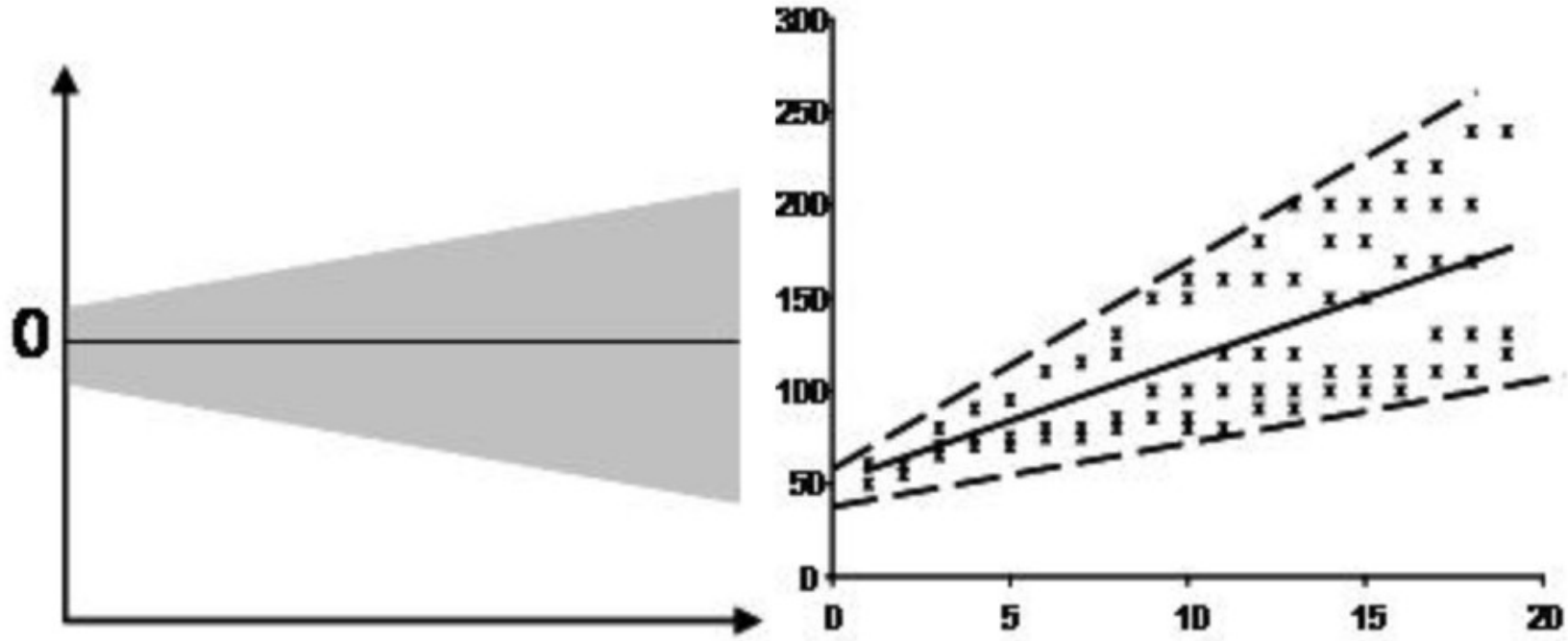
Must Have Homoscedasticity

- Data sets in the regression must have the same variance (same quality of being different or divergent)
- This assumption means that the variance around the regression line is the same for all values of the predictor variable (X).
- The plot shows a violation of this assumption. For the lower values on the X-axis, the points are all very near the regression line.



Must Have Homoscedasticity

- Heteroscedasticity examples below
- Differing variance is bad for regression models.



Data Analytics
CS390
Modeling: Formal Basics

Fall 2017

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

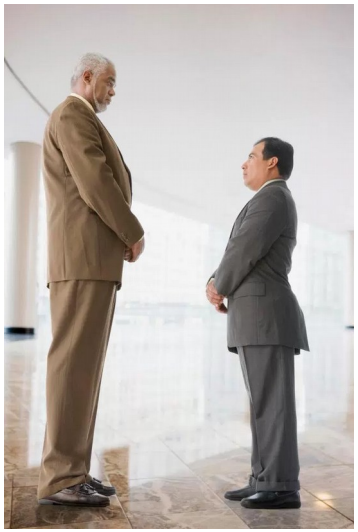
Let's Begin Our Discussion...

- 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?

There's Data For Each Question



Do you think that taller people make more money?

File: wages.csv

Do you think that hotter places have more crime?

File: crime.csv

Wages Data Set

- Earnings vs. height and demographic characteristics of 1379 individuals, collected in 1994. Earnings adjusted for inflation.
- Simulated data based on real data collected by Gelman and Hill. Data Analysis using Regression and Multilevel/ Hierarchical Models. Cambridge Press, 2007.

open the wages dataset from the data files.

```
options(stringsAsFactors = FALSE)
w <- file.choose() # set the filename
wages <- read.csv(w) # load and read the data.
```

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.
```

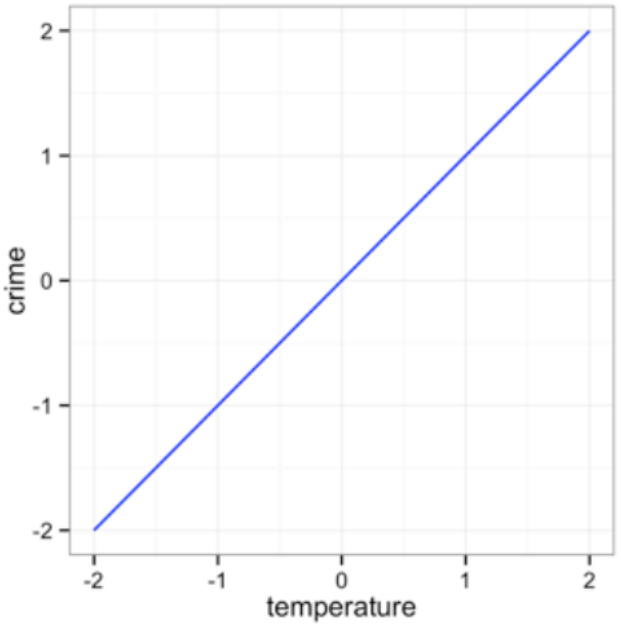
How Do we Answer The Question?

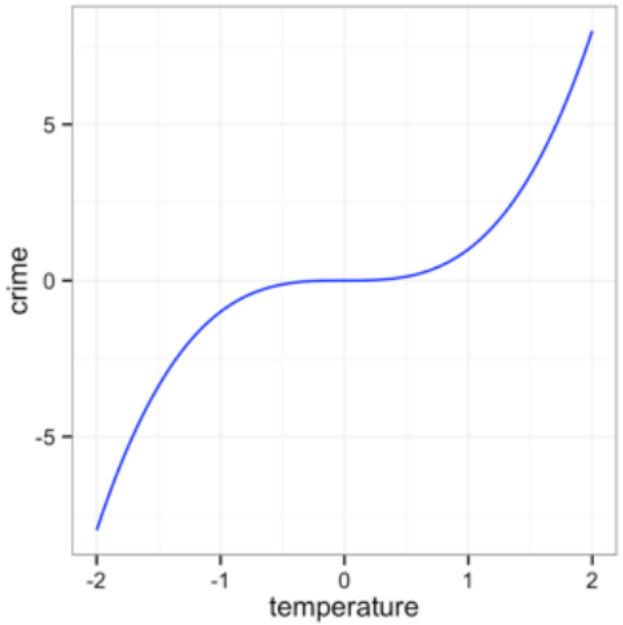
- Modeling: We employ a computational framework which we used data to build (for training).
- Play with the model to see what happens when we change a part of the data ...

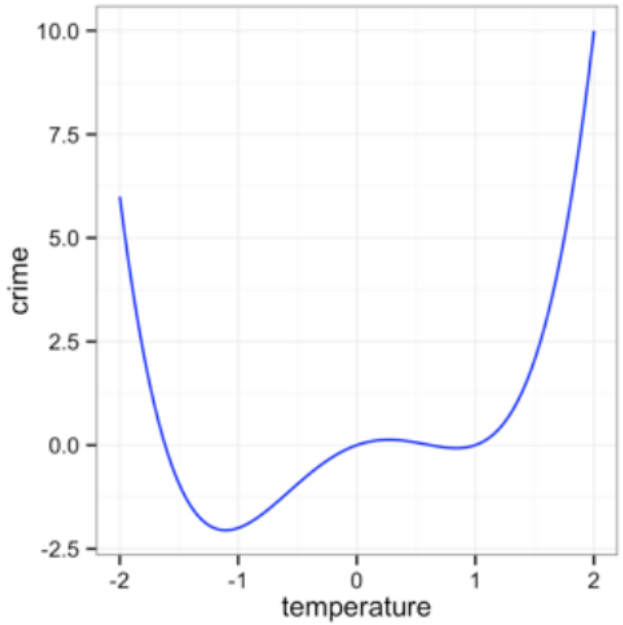
What if…

Functions: the *stuff* behind the models

- A function is a mathematical description of a relationship.
- If one variable completely determines another, every (x, y) data point will fall on the function line.

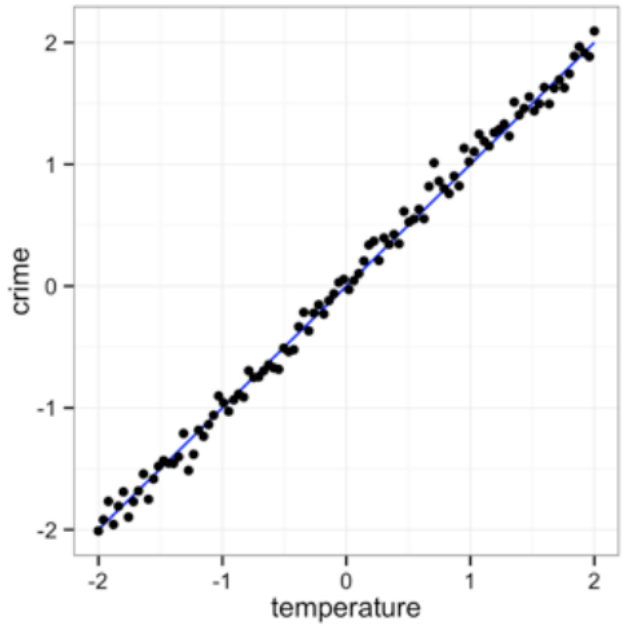


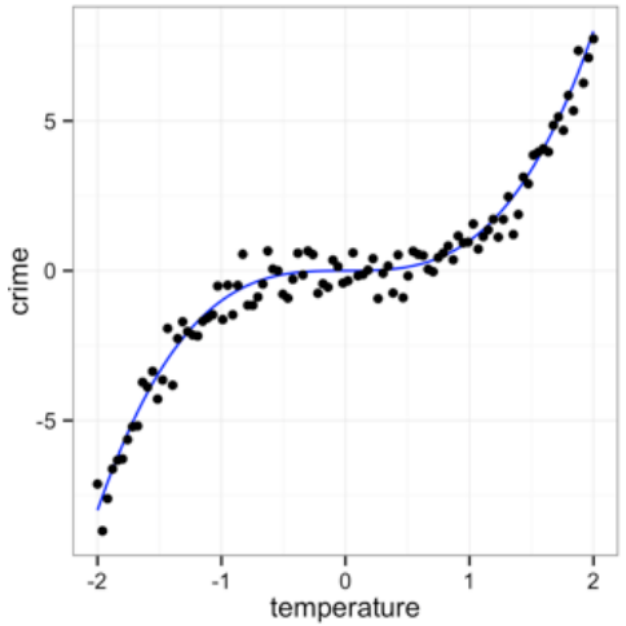


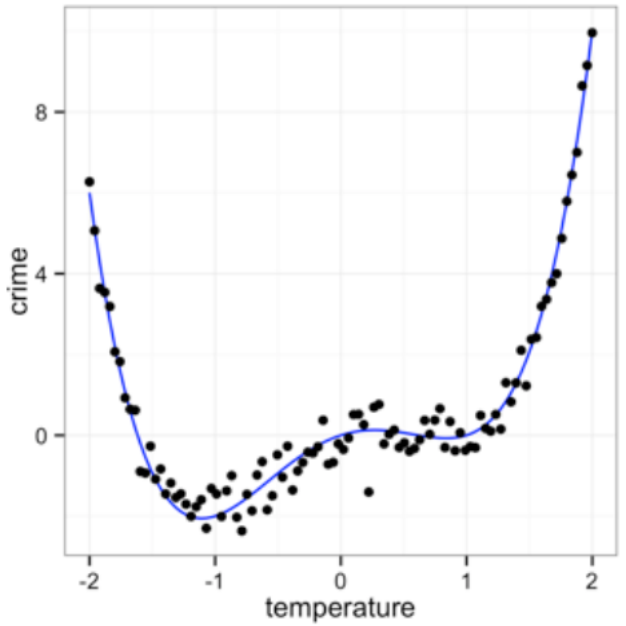


Relationships Between Variables

- If the relationship is also affected by other variables, data points may not fall directly on the function line.
- 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 correctly identify relationships in noisy data.**

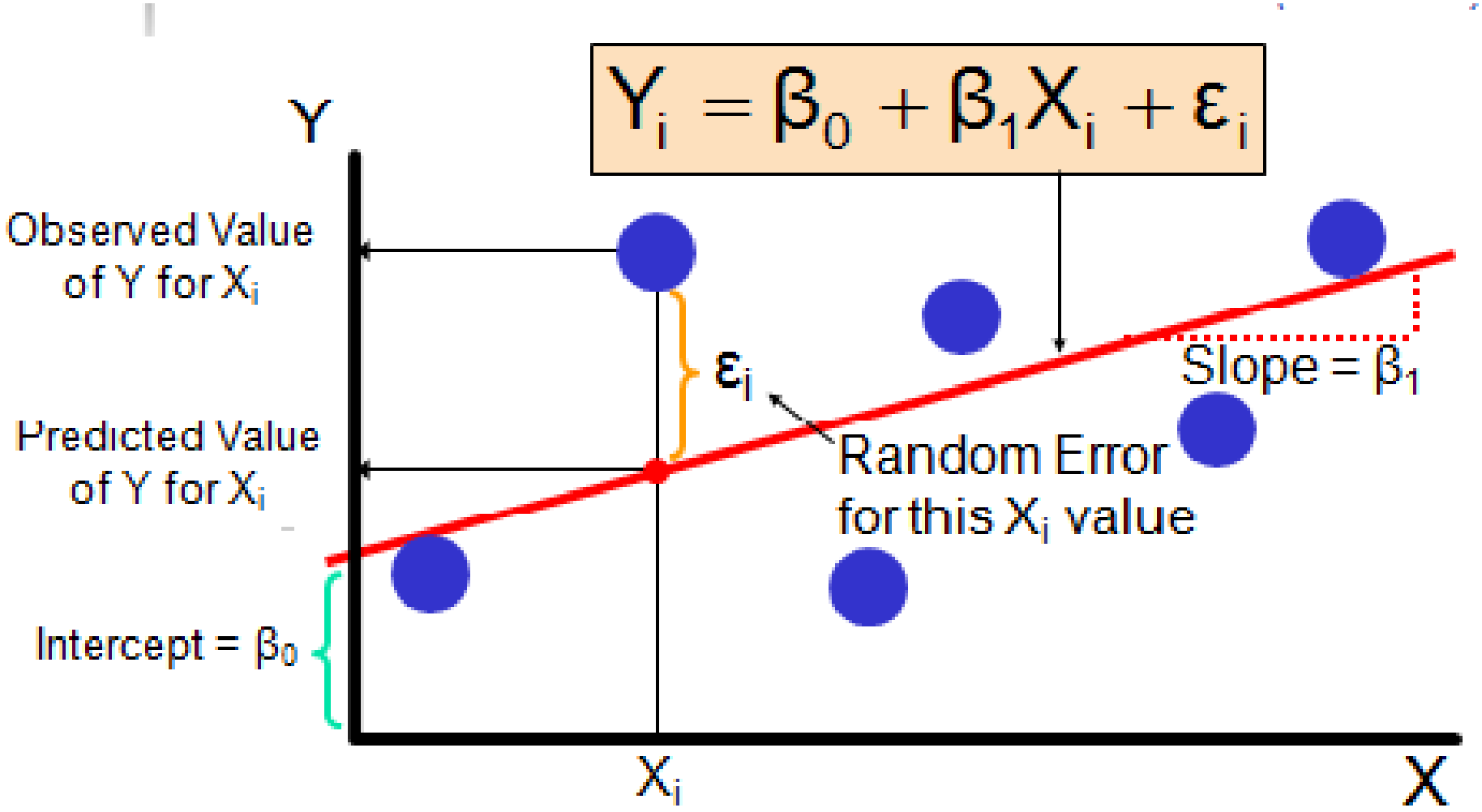
Types of Models

- **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
- **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)

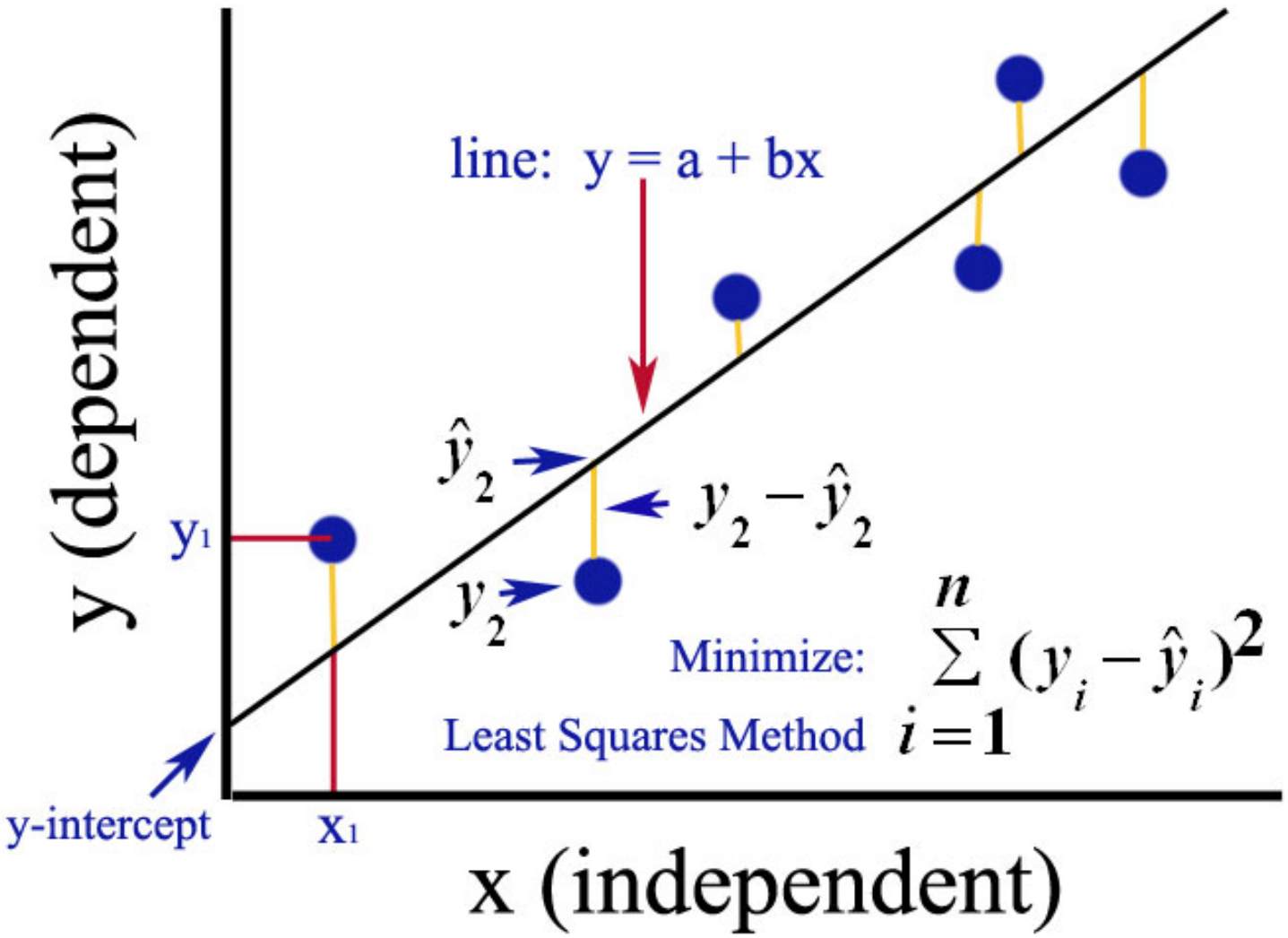
Let's Talk Linear Models

- Linear regression, formally is:
- The linear regression algorithm constrains *f(x)* to have the form:
- $f(x) = \alpha + \beta x + \epsilon$
  - Line formula alpha: intercept.
  - Beta: slope
  - Epsilon: account for the error
- *Note: f(x)* will be a straight line in *x*

Let's Talk Linear Models

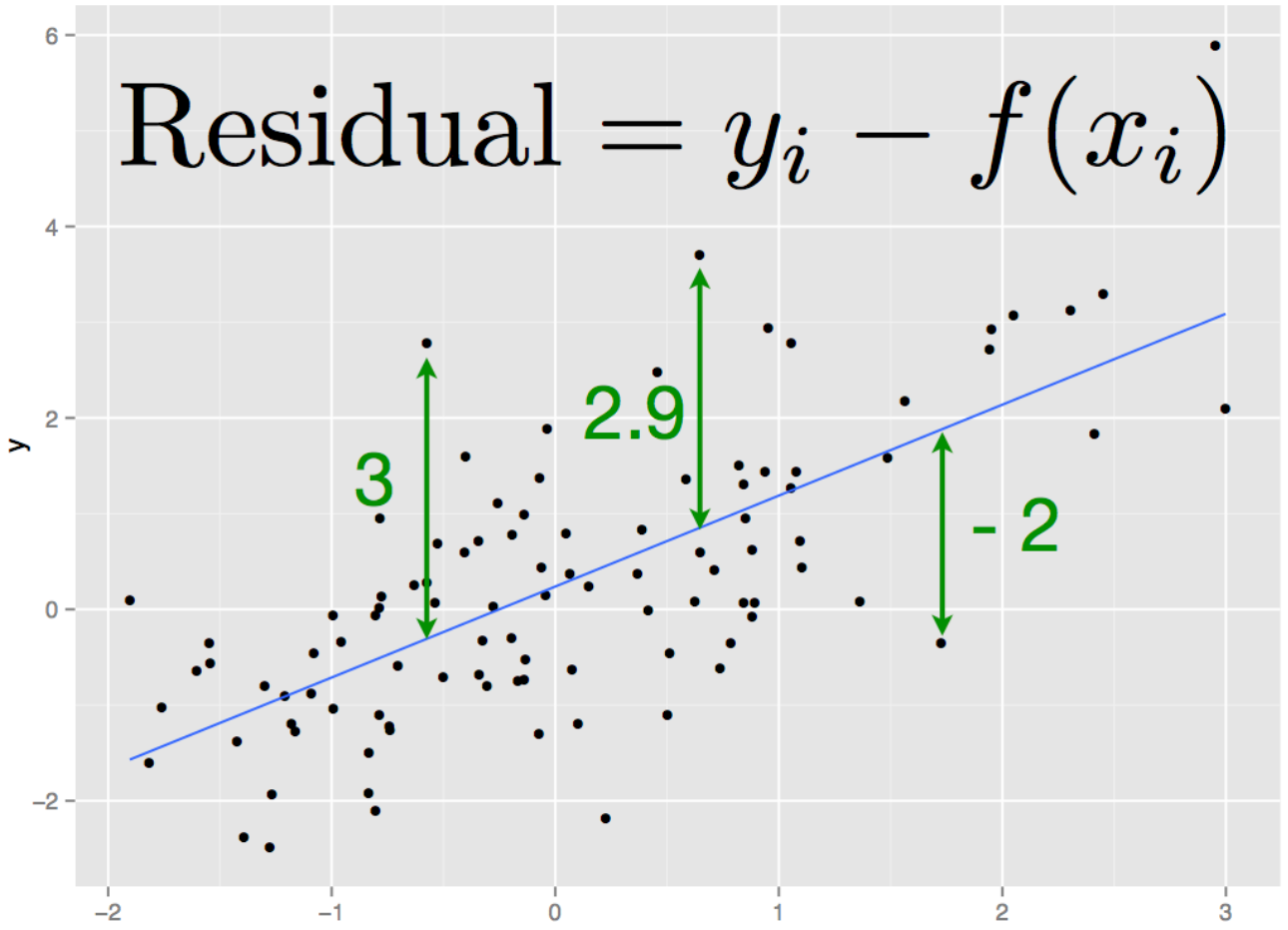


Another Linear Model



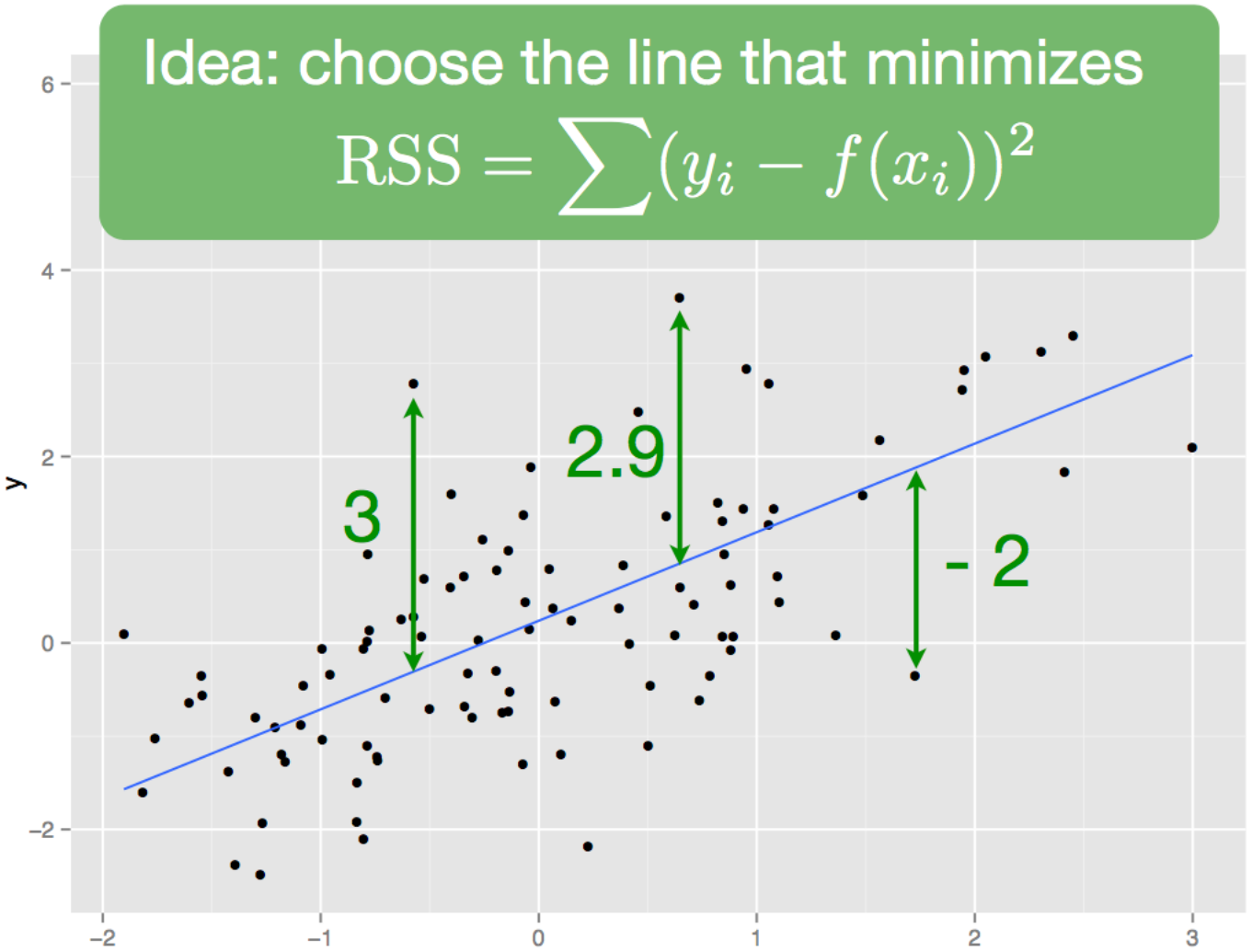
How To Best Draw a Line Through The Data?

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



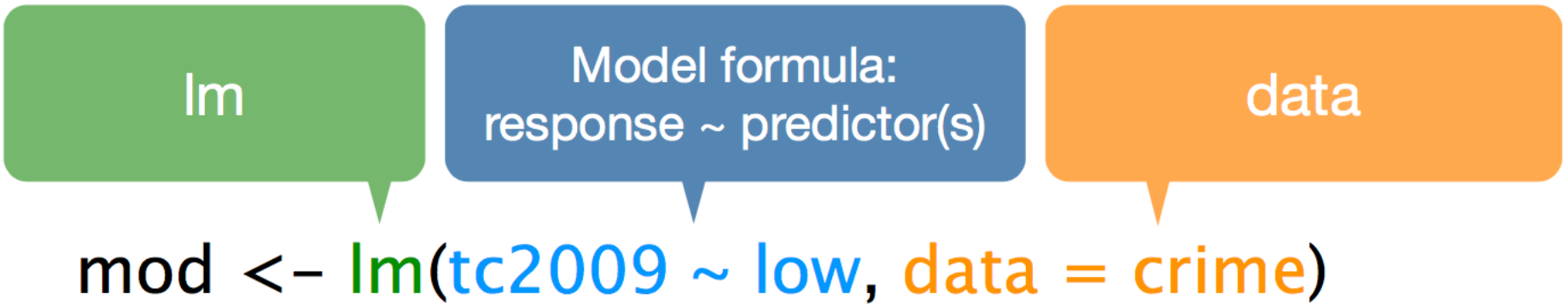
How To Best Draw a Line Through The Data?

- 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).



Let's Hit the Code

- Linear model syntax



Formulas

- R formulas are expressions built with ~ (tilda)

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


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

Formulas

- Formulas only need to include the response and predictor variables

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

#Syntax to Build the linear model:

$$y \sim x$$

Formulas

```
response ~ explanatory

dependent ~ independent

outcome ~ predictors
```

```
# Make a model called, mod
mod <- lm(tc2009 ~ low, data = crime)
```

Consider This!

- Fit a linear model to the crime data set.
- Predict **tc2009** (dep) with **low** (ind). What function describes the best fit line?

$$Y = \underline{?} + \underline{?} * x + \epsilon$$

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
```

Let's Hit the Code

- We run the code
- Next time, we interpret these results.

Data Analytics
CS390
Modeling: Formal Basics

Fall 2017

Oliver Bonham-Carter

What does it mean to regress *Y* on *X*?

- A function defines one variable in terms of another.
- The statement "y is a function of x" (denoted y = y(x)) means that y varies according to whatever value x takes on.
- A causal relationship is often implied (i.e. "x causes y"), but does not *necessarily* exist.

Extracting Info

- Create model object to look for “What If?” patterns.
- Run functions on model object to get details

  Try these commands

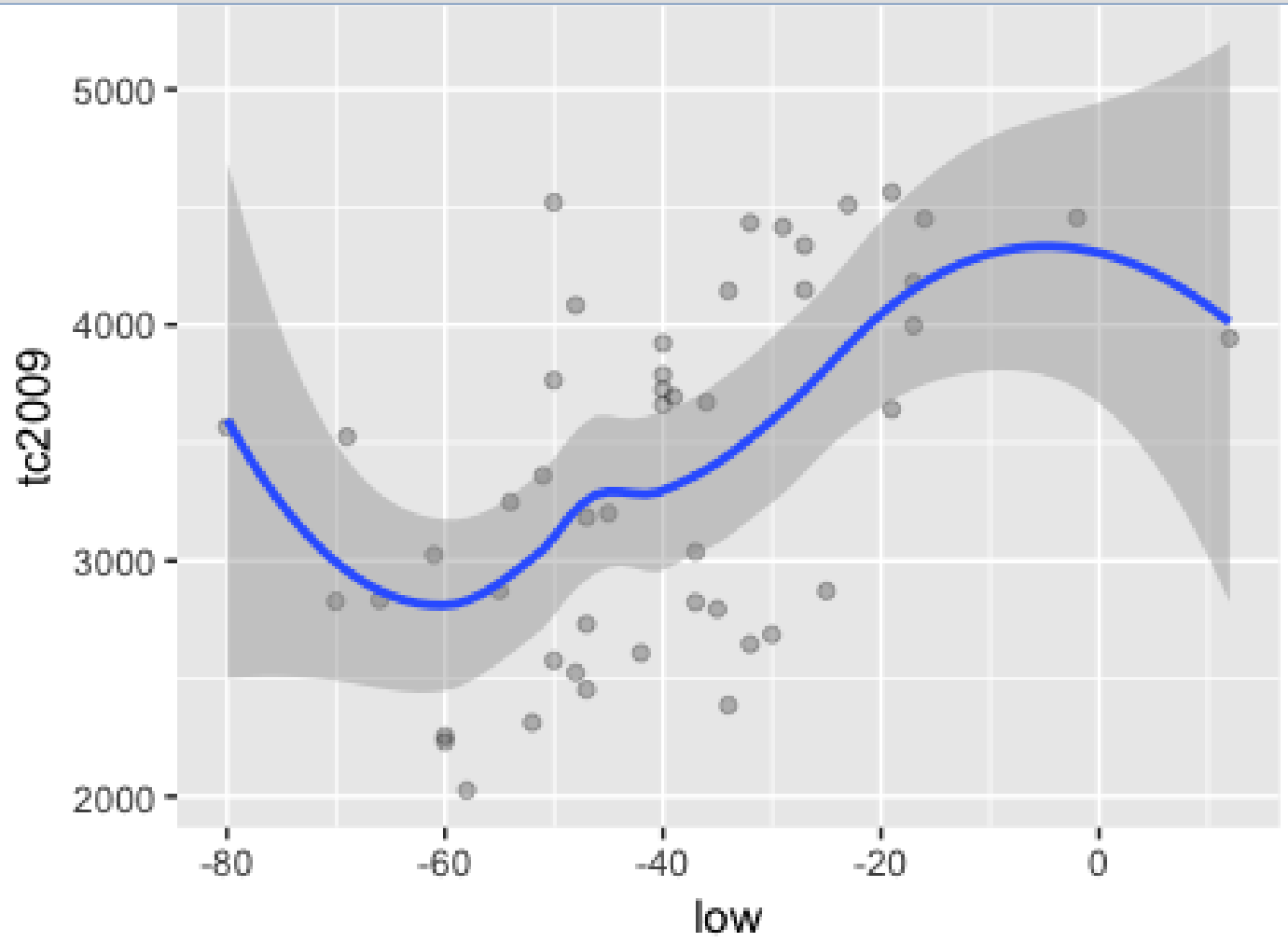
```
summary(mod)
predict(mod) # predictions at original vals
resid(mod) # residuals
```

Let's Hit the Code

```
#plot the data

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

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

#Build the model

mod1 <- lm(tc2009 ~ low, data = crime)
```

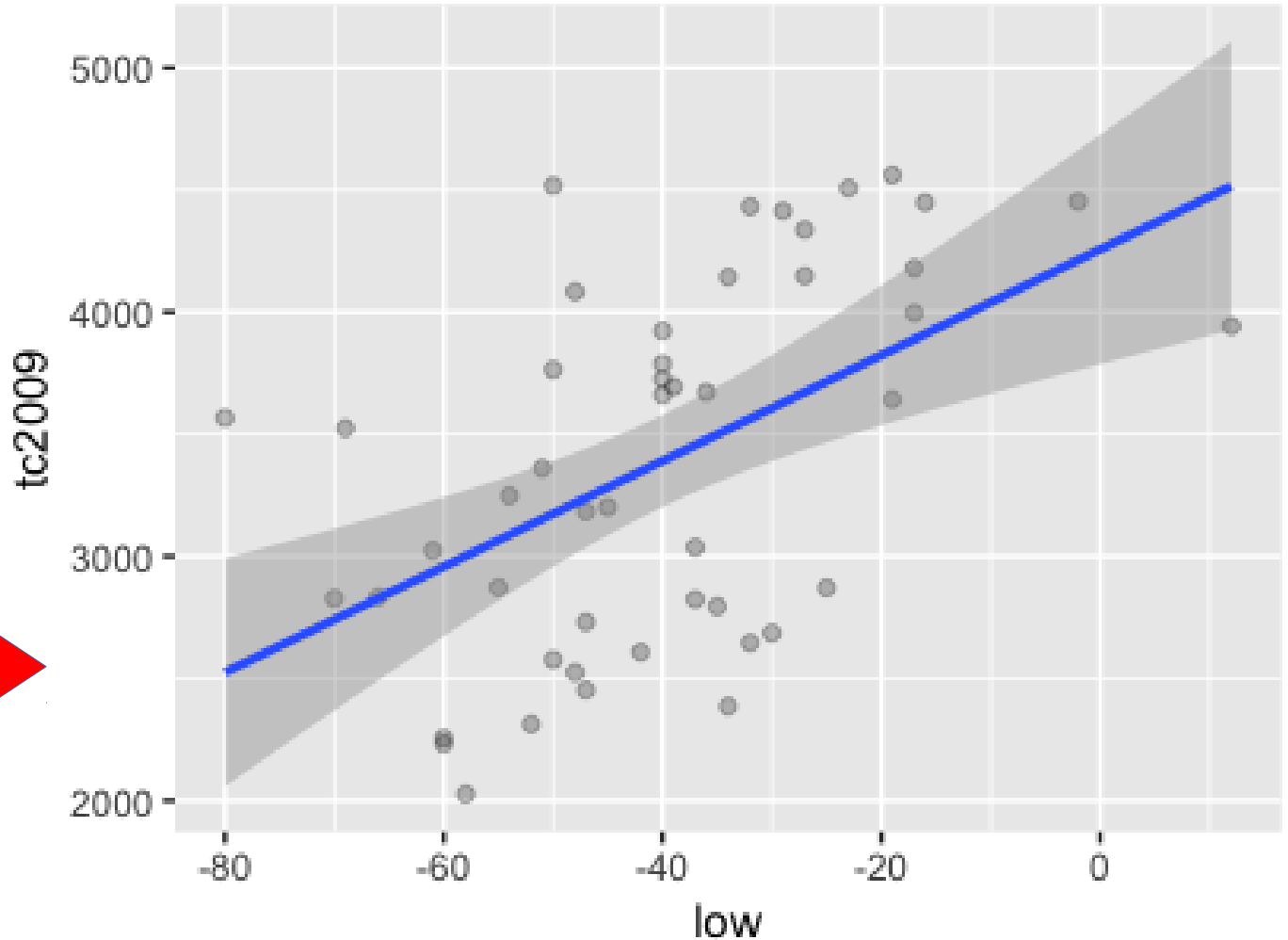
Plots

```
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 Model
To Play With

```
mod1 <- lm(tc2009 ~ low, data = crime)
```

```
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

Coef

```
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)

Coef Calculator

This function is now my data!!

```
# 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 :",mod1$coefficients[1] )
  cat("\n  slope     :",mod1$coefficients[2] )
  y = mod1$coefficients[1] + x_int * mod1$coefficients[2]
  cat("\n  y = ",y)
}

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

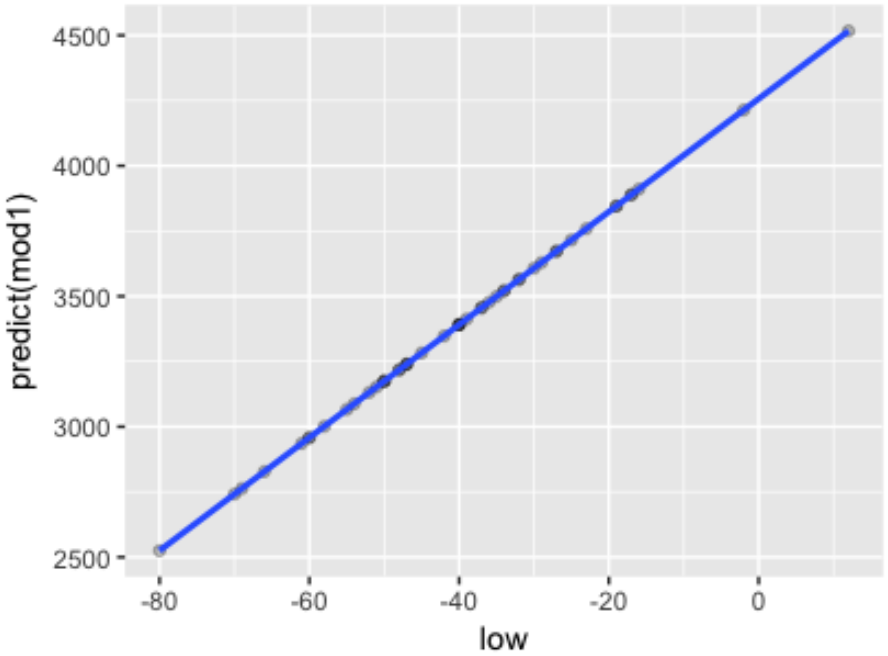
The best estimate of
tc2009 for a state with low = -10 is
4256.86 + 21.6 * (-10) = 4040.86

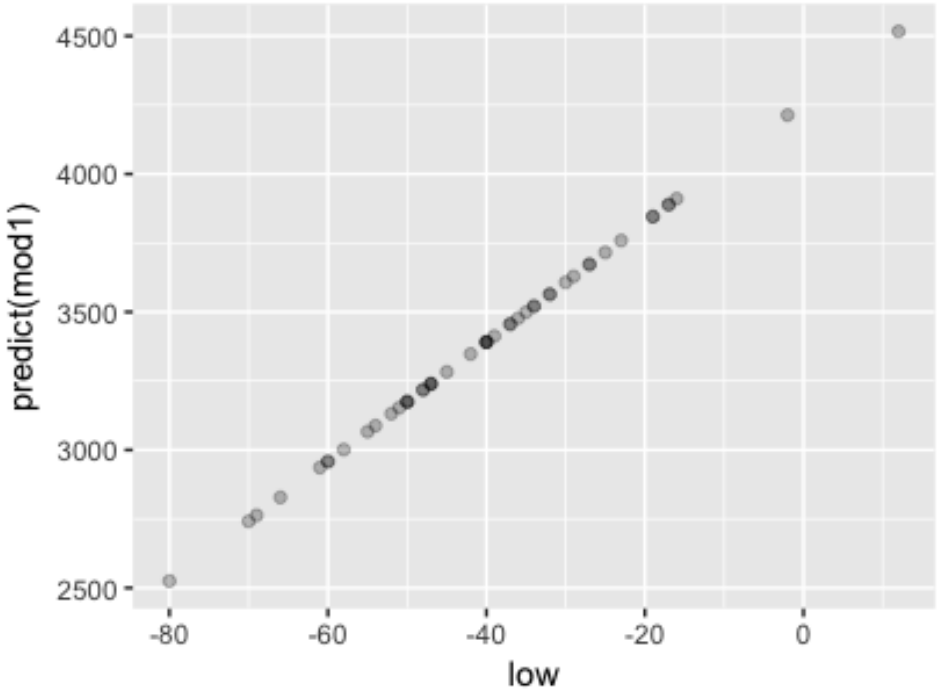
Due to error, there is a slight difference between this and our value.

The Model's Line

```
crime %>% ggplot(aes(x = low, y = predict(mod1))) +
geom_point(alpha = I(1/4))
```

```
crime %>% ggplot(aes(x = low, y = predict(mod1))) +
geom_point(alpha = I(1/4)) + geom_smooth()
```





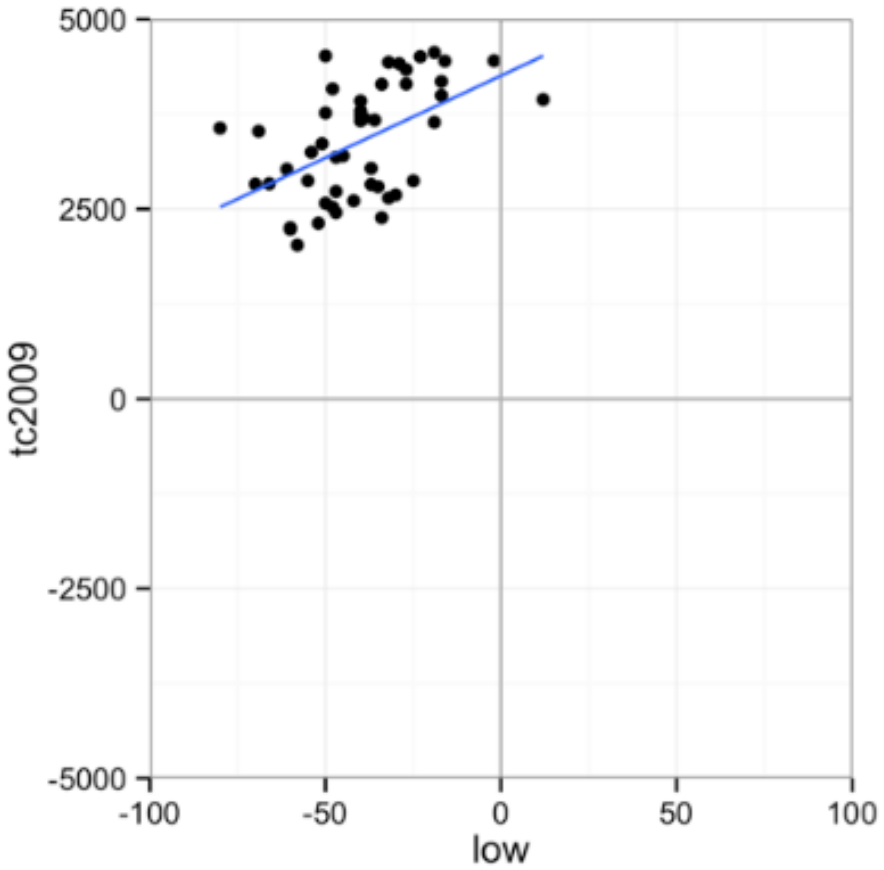
Aside: intercept terms

R includes an intercept term in each model by default

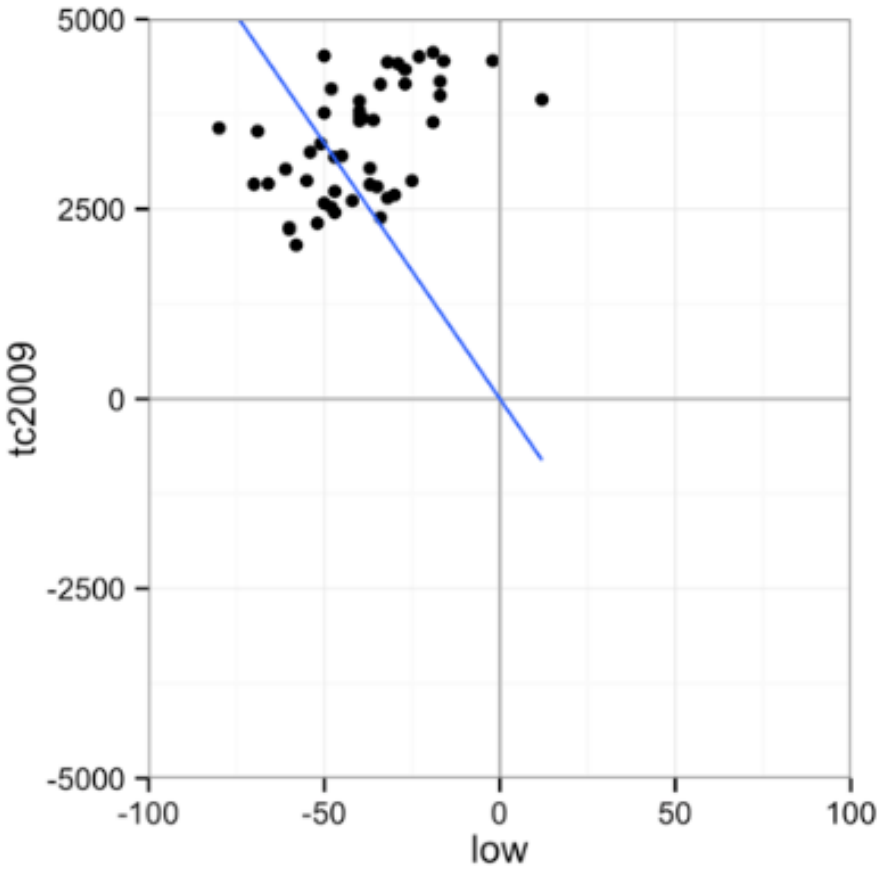
$$y = \alpha + \beta x + \epsilon$$

```
y ~ x
```

Want the Zeros or Not?



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).

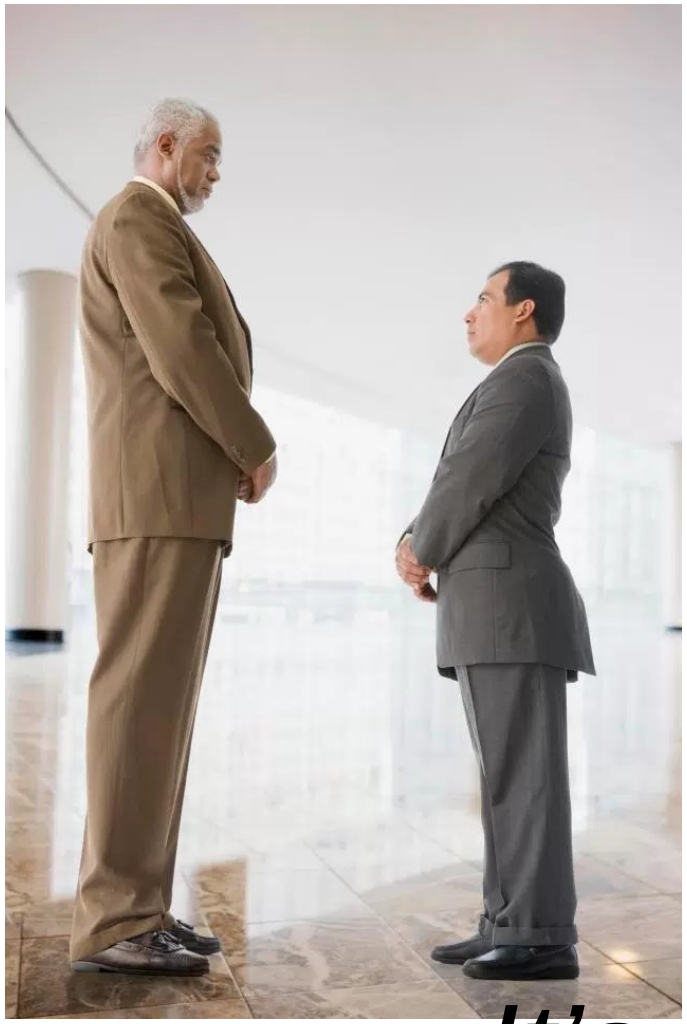
An Intercept Term: To Use or Not?

ALLEGHENY COLLEGE

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)
```

Now, Back to Our Question...



Do you think that taller people make more money?

File: wages.csv

Remember:

It's not you, it's your data.

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")
```

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
```

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

$$earn = -126523.36 + 2387.20 \times height + \epsilon$$

Earn Regressed Over *height*

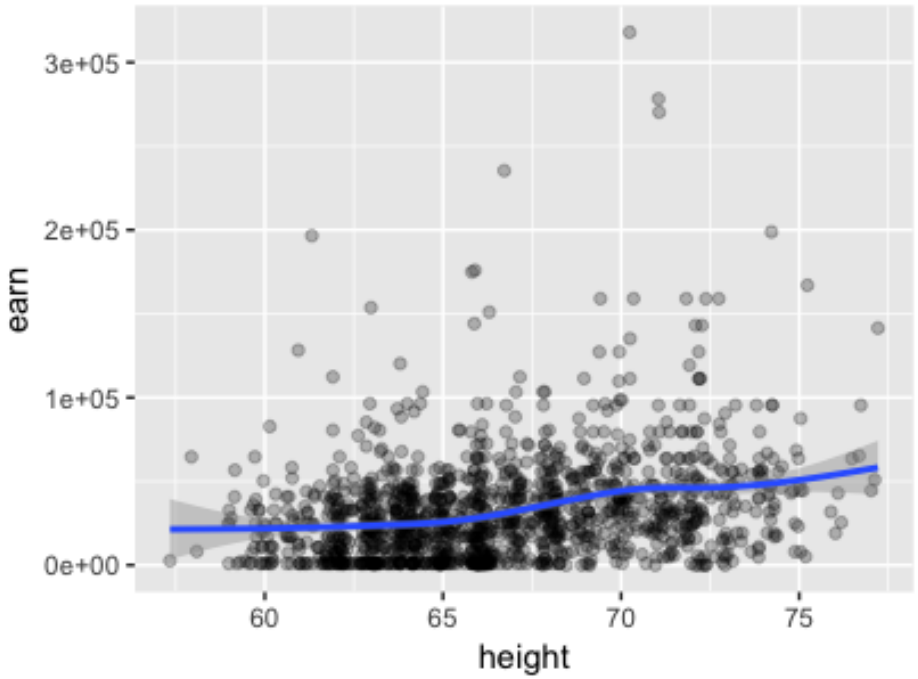
The best estimate of earn for someone 68 inches tall is

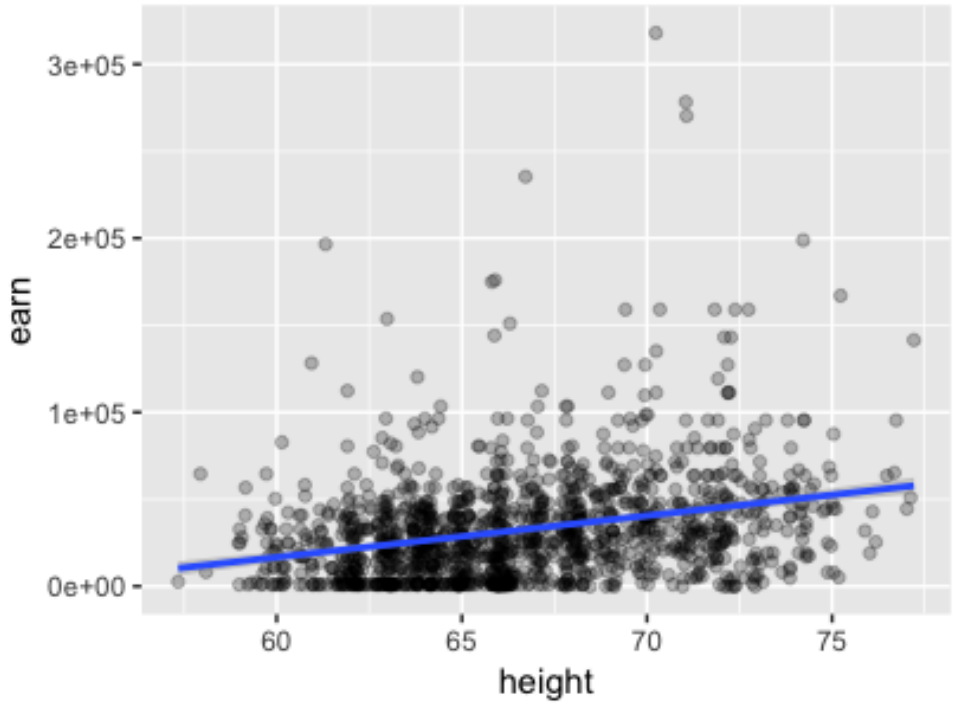
$$earn = -126523.36 + 2387.20 \times 68 + \epsilon$$

$$earn = 35806.24$$

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
```





Data Analytics
CS390
Modeling: Formal Basics

Fall 2017

Oliver Bonham-Carter

Deducing a Relationship

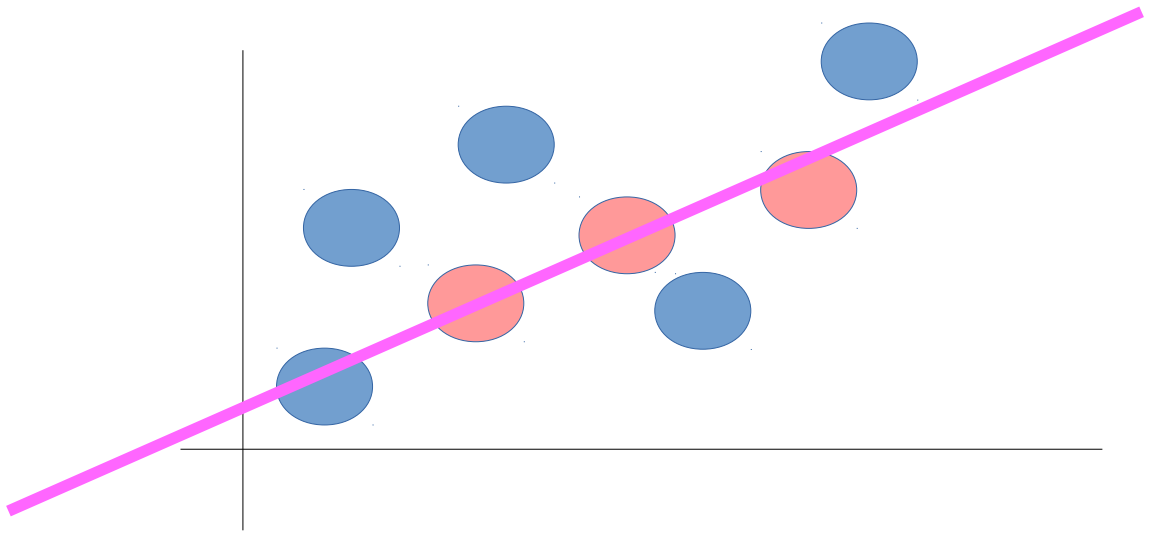
- Entities, Variables in data
- Correlation?
- Are new patterns in keeping with our model?
- How to build Models?
  - Linear Regression: one entity regresses over another.

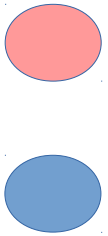
We Made a Model From Our Data

- How do models turn out when several are made from the same source data?
- Reliable?
- Are some models just lucky?

Building a Model From a Subset of the Data

- Sometimes, we get too much data. Not all the data has be used.
- We could build a model using a random selection of points (partial grouping of points)
- These points should describe the same (basic) model if all points where used.





Partial set of data

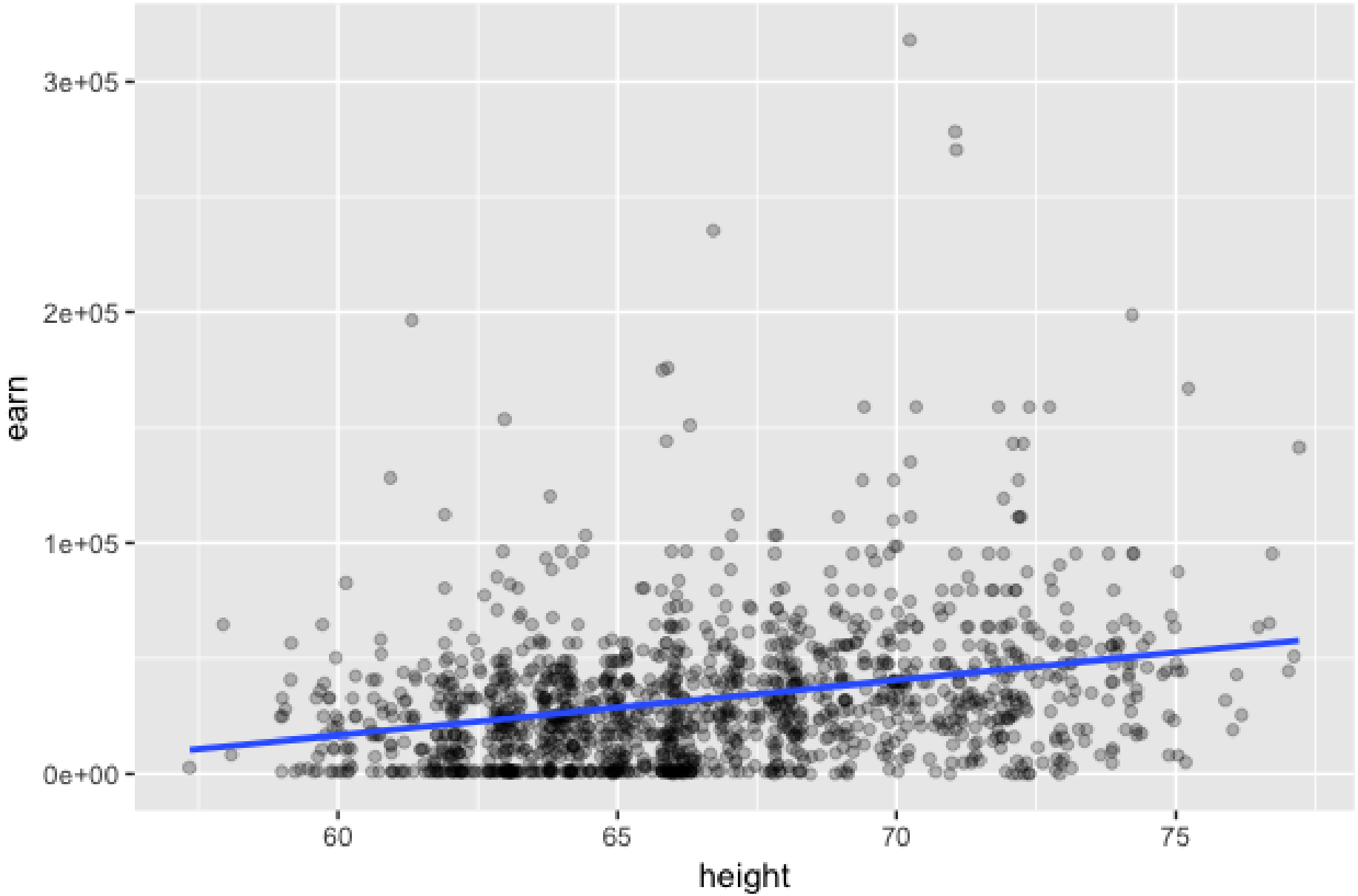
Full set of data

Wages: Plot **Full Set**

```
#full wages data
w <-file.choose()
wages <- read.csv("wages.csv")
# plot full data set
qplot(height, earn, data = wages, alpha = I(1/4))
+ geom_smooth(method = lm, se = F)
```

```
lm(earn ~ height, data = wages)
```

Wages: Plot **Full Set**

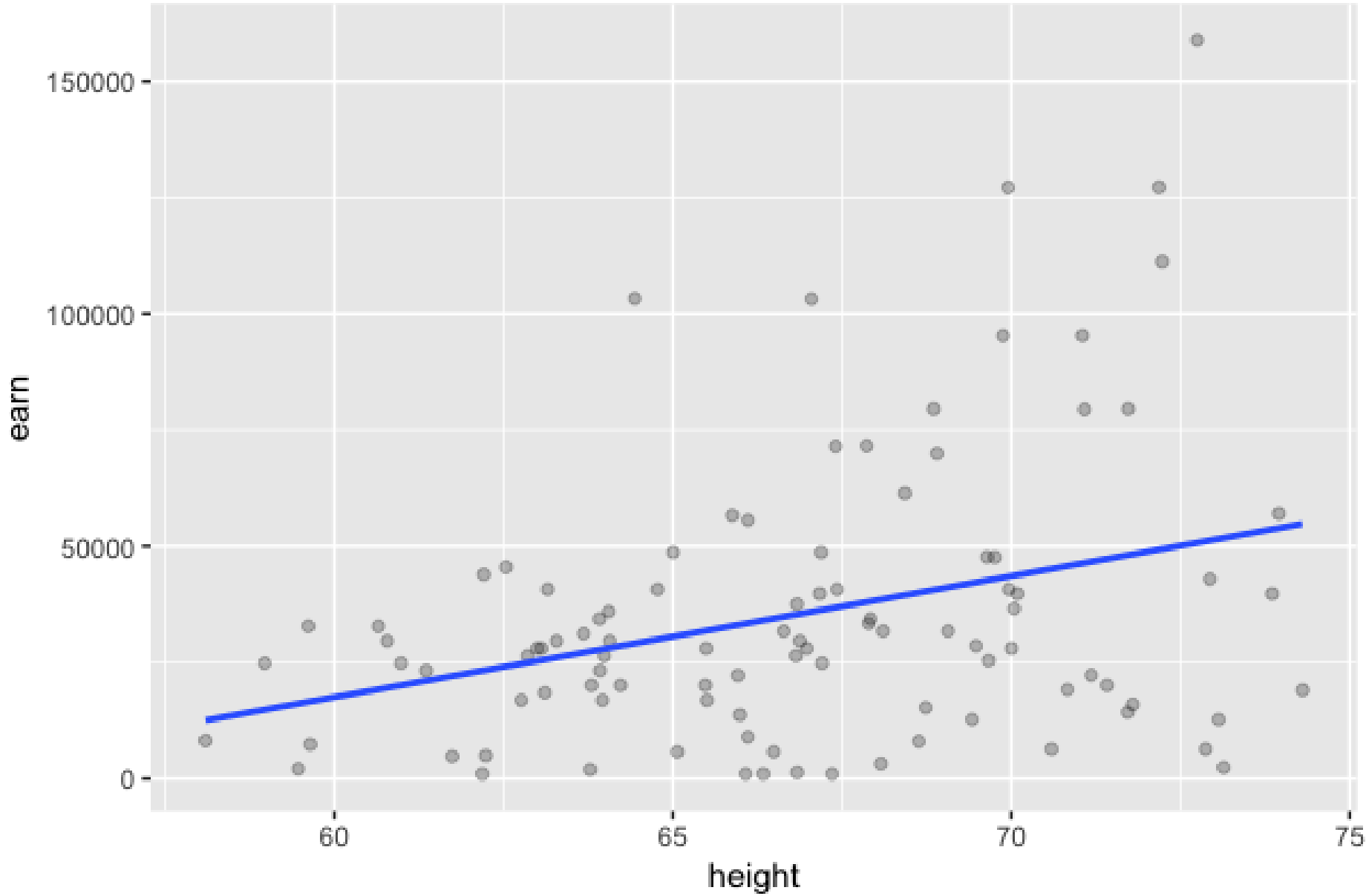


Wages: Plot **Partial** Points

```
# partial wages data

w1 = file.choose()

p\Wages <- read.csv(w1)

# plot partial data set

qplot(height, earn, data = pWages, alpha =
I(1/4)) + geom_smooth(method = lm, se =F)
```

```
lm(earn ~ height, data = pWages)
```

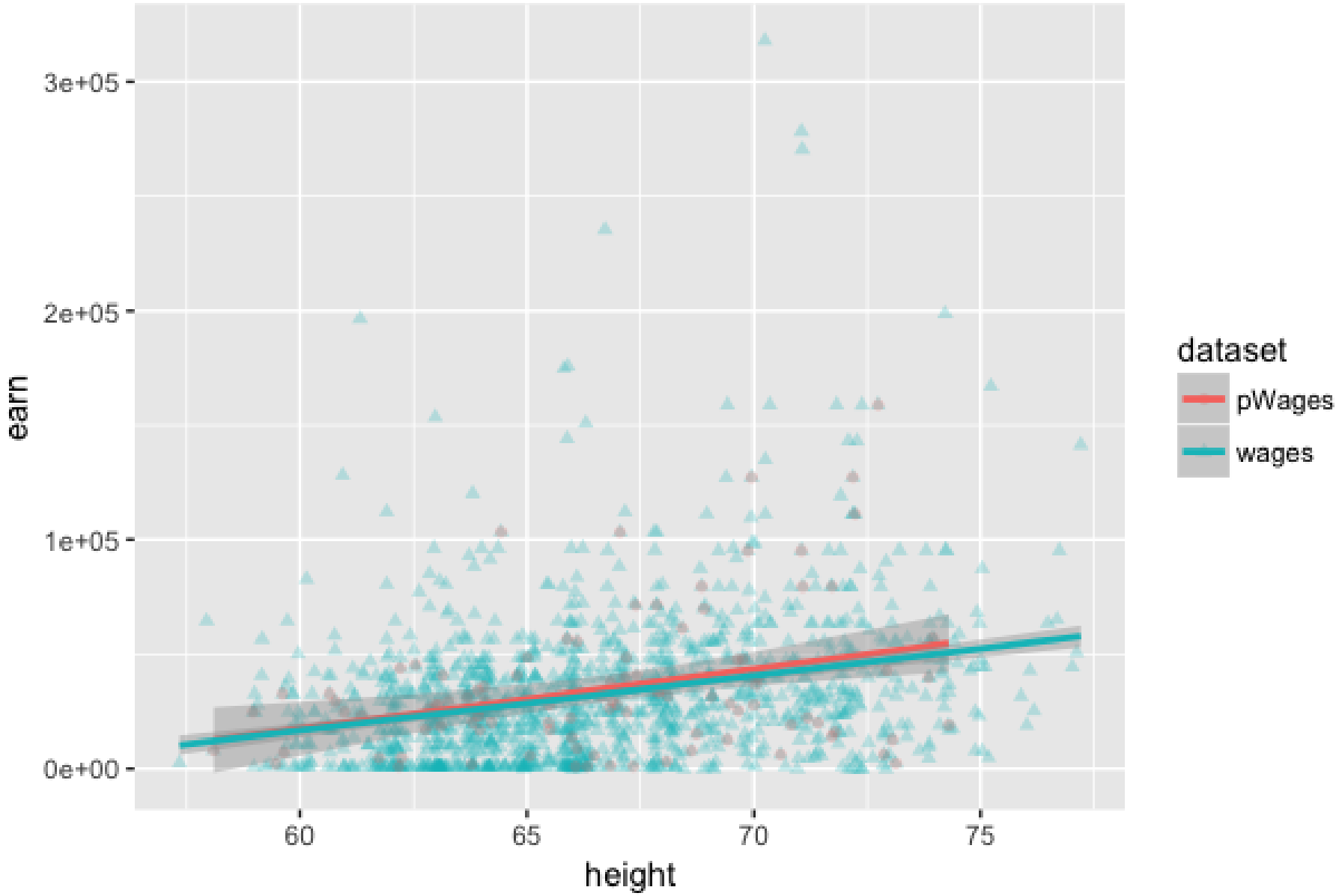
Wages: Plot **Partial** Points



Wages: Plot **Partial Set** with the **Full Set**

```
#both plots together

dat <- rbind(wages, pWages)

dat$dataset <- factor(c(rep("wages",
dim(wages)[1]), rep("pWages",
dim(pWages)[1])))

ggplot(dat, aes(x=height, y=earn, col =
dataset, shape = dataset)) +
geom_point(alpha = I(1/4)) +
geom_smooth( method = lm)
```

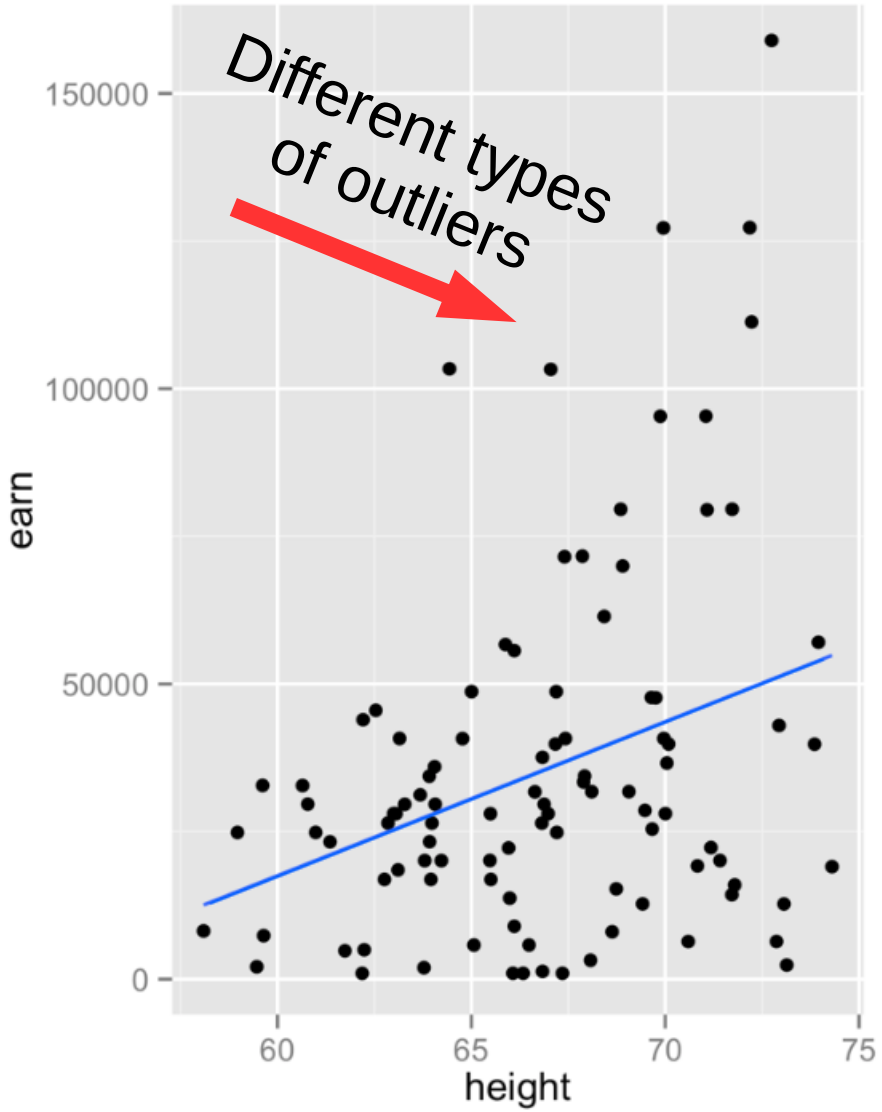
Wages: Plot **Partial Set** with the **Full Set**



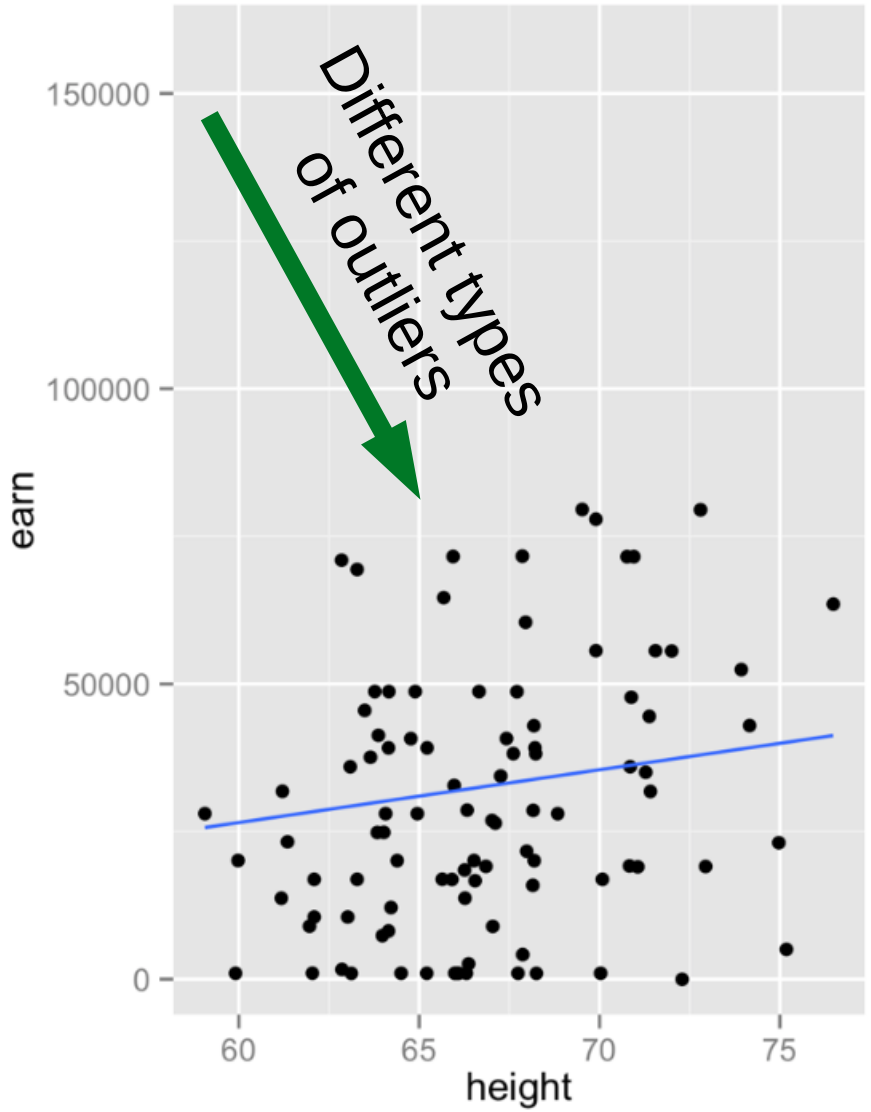
Deduce Real Relationships

- Using the *wages* data.
- What would happen if we used other randomly selected data to create a model?
- Would the another model be different, if we had chosen other data points from the *wages* set?

Wages: Two partial Models Created From Same Set of Data

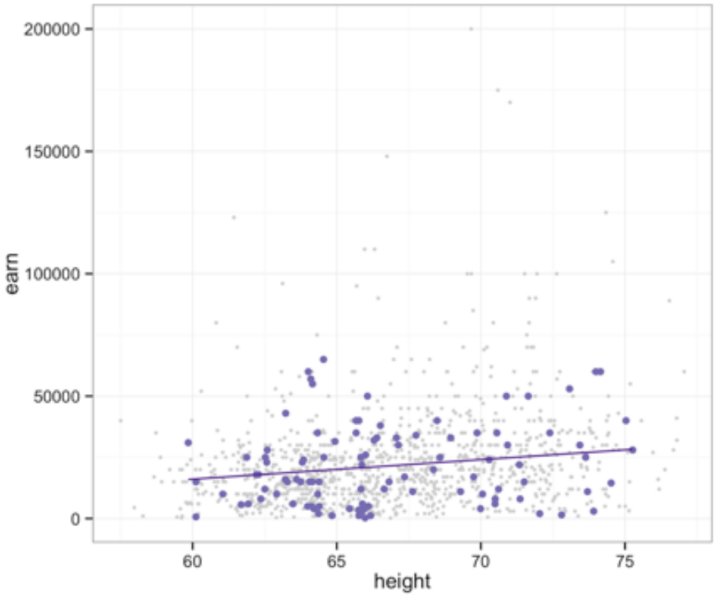


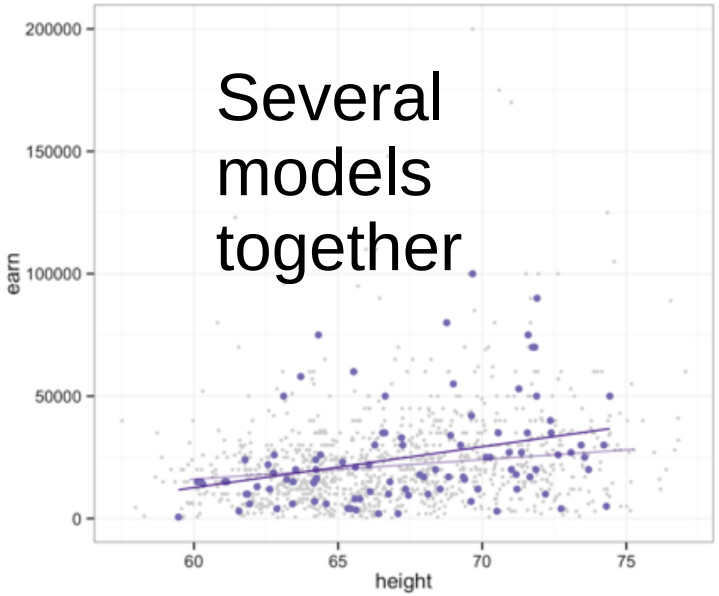
w1 sample



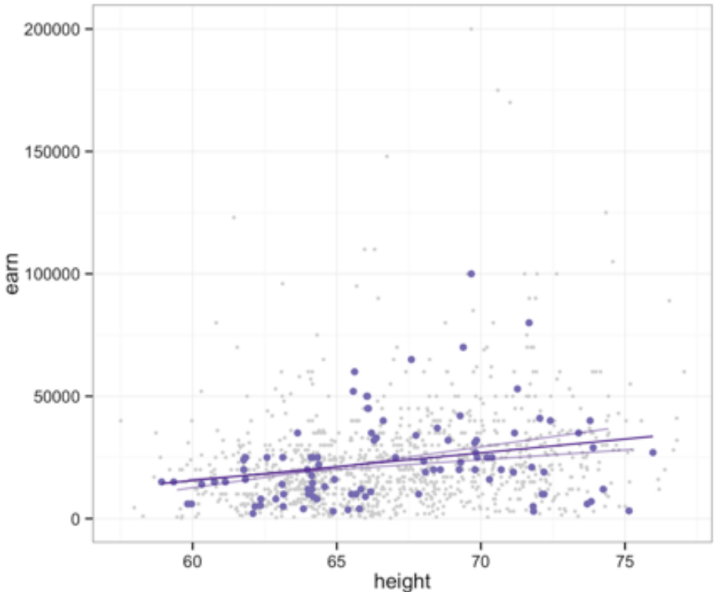
w2 sample

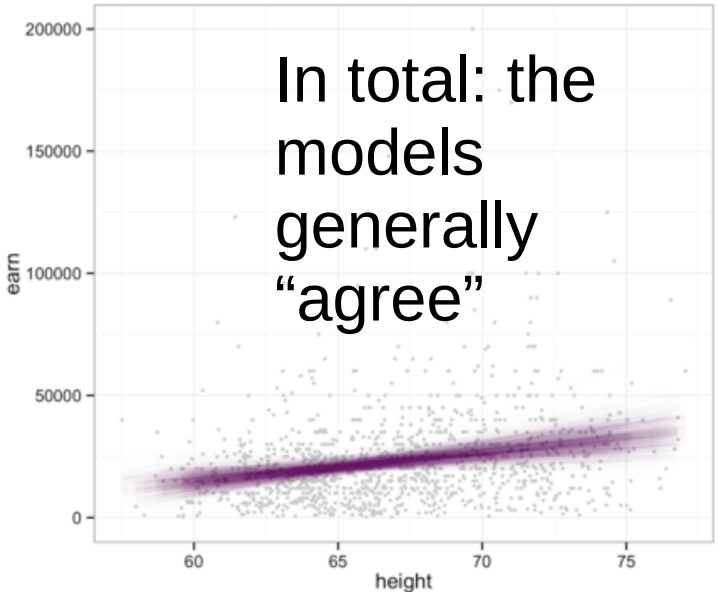
Repeated Model Making From Same Source





Several models together





In total: the models generally “agree”

Remember!

You still need to check your models no matter how they were made.

So, Are My Models Made From Sampling Full Data Set Any Good?

- Use *Parametric statistics* to check your model before you use it!

```
> summary(mod)

Call:
lm(formula = earn ~ height, data = pWages)

Residuals:
   Min     1Q Median     3Q    Max
-49392 -17589  -4448  10236 108209

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -138901.1    50897.3  -2.729 0.007530 **
height         2607.4      760.6   3.428 0.000891 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29100 on 98 degrees of freedom
Multiple R-squared:  0.1071,    Adjusted R-squared:  0.09795
F-statistic: 11.75 on 1 and 98 DF,  p-value: 0.0008909
```

mod = lm(earn ~ height, data = pWages)

So, Are My Models Made From Sampling Full Data Set Any Good?

Probability that the **βheight** value greater than 2607.4, if the true value is actually zero.

```
Residuals:
   Min     1Q Median     3Q    Max
-49392 -17589  -4448  10236 108209

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -138901.1    50897.3  -2.729 0.007530 **
height         2607.4      760.6   3.428 0.000891 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29100 on 98 degrees of freedom
Multiple R-squared:  0.1071,    Adjusted R-squared:  0.09795
F-statistic: 11.75 on 1 and 98 DF,  p-value: 0.0008909
```

Alpha Level

- If a *p*-value is very low (less than 0.05 and “close to zero”), it suggests that either:
  - 1. you have an unusual sample
  - 2. the true β does not equal 0
  - 3. your model assumptions are wrong

Confidence Intervals

Knowing probabilities also lets us calculate confidence intervals for β

```
confint(mod, level = 0.95)
```

```
> confint(mod, level = 0.95)
                    2.5 %     97.5 %
(Intercept) -239905.188 -37896.930
height         1097.902   4116.846
```

If we are wrong, then we are still 95% confident that the true coefficients are found in these intervals.

Data Analytics
CS390
Modeling: Multiple Linear Regression

Fall 2017

Oliver Bonham-Carter

Up To Now in Regression

- We have discussed how one entity influences another.
- What about having two entities (independent) which may have some kind of influence on a dependent variable.
- Especially if a dependent variable has a high correlation with more multiple independent variable.

Main Idea

- GPA could be dependent on studying
- Student performance may be based on more than just one entity.
- GPA could be dependent on studying AND getting enough rest
- OR maybe even more variables are involved?
- GPA could be dependent on studying AND rest AND eating good food AND …

So, Multiple Linear Regression Is What … ?

- Multiple linear regression is the most common form of linear regression analysis.
- A predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables.
- The independent variables can be continuous or categorical (dummy coded as appropriate).

Types of Questions Answered

- Do age and IQ scores effectively predict GPA?
- Do weight, height, and age explain the variance in cholesterol levels?
- Are video game sales explained by their exciting graphics and inexpensive costs?
- Is road safety a combination of relaxed and defensive driving?
- Are there more independent variables to be used to answer to these above dependents?

Equation of Multiple Independent Variables

- The model is now a multi-independent variable equation.

Dependent Variable

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \epsilon_i$$

Independent Variables

Equation and Assumptions

- A population model for a multiple regression model that relates a *y*-variable to $p - 1$ predictor variables is written as the following.

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{p-1} x_{i,p-1} + \varepsilon_i$$

• We assume that the ε_i have a normal distribution with mean 0 and constant variance σ^2. These are the same assumptions that we used in simple regression with one *x*-variable.

• The subscript *i* refers to the ith individual or unit in the population. In the notation for the *x*-variables, the subscript following *i* simply denotes which *x*-variable it is.

Hypothesis

As an example, to determine whether variable X_1 is a useful predictor variable in a model, we use the following hypothsis:

$$H_0 : \beta_1 = 0$$

$$H_A : \beta_1 \; != \; 0$$

(think slope values)

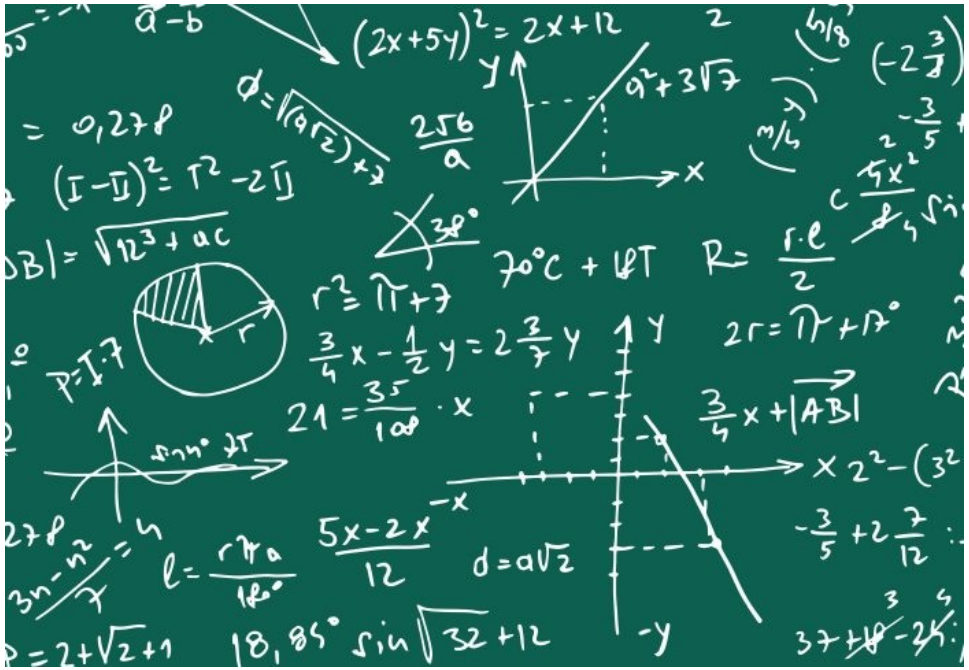
If the null hypothesis above were the case, then a change in the value of X_1 would not change Y, so Y and X_1 are not related.

We would still be left with variables X_2 and X_3 being present in the model and so we could not reject the null hypothesis above. Instead we should say that we do not need variable X_1 in the model given that variables X_2 and X_3 will remain.

In general, the interpretation of a slope in multiple regression can be tricky. Correlations among the predictors can change the slope values dramatically from what they would be in separate simple regressions.

Analysis Question

- We ask: Do *Age* and *Height* influence the capacity of lungs (*LungCap*)?
- Asking actually, can we make a model that takes the following form?
  - Age =**?** Age*b_1 + Height*b_2 + b_3



Lung Capacity Data

```
library(tidyverse)
library(psych)


#open lung capacity data
lc <-file.choose()
dataLungCap <- read.csv(lc)
View(dataLungCap)
```

Create the Multiple-Variable Regression Model

```
#model creation
mod <- lm(LungCap ~ Age + Height)
#get a report of the model
summary(mod)
```

Summary

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

R-squared Value:

"How do the independents explain the dependent?"

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t valu     (>|t|)
(Intercept) -11.747065   0.476899 -24      < 2e-16 ***
Age           0.126368   0.017851      079 3.45e-12 ***
Height        0.278432   0.009926   28.051  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

Approximately 84% of the variation in *Lung Capacity* can be explained by our model (*Age* and *Height*)

F-Statistic of Test:

"What value do I look up in a table to check on significance?"

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t valu     (>|t|)
(Intercept) -11.747065   0.476899 -24       < 2e-16 ***
Age           0.126368   0.017851        079 3.45e-12 ***
Height        0.278432   0.009926   28.051  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

Degrees of Freedom:
There are 725 rows in the data and three groups.
722 = 725 – 3

The p-Value:
"Is this model statistically meaningful?"

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>
(Intercept) -11.747065   0.476899 -24.632  < 2e     ***
Age           0.126368   0.017851   7.079 3.45e     ***
Height        0.278432   0.009926  28.051  < 2e-    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05    ' 0.1 ' ' 1

Residual standard error: 1.056 on 722 degrees of f eedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

The p-value is very close to zero and so we reject Ho (i.e., all the model coefficients are zero (slope = 0).

Null Hypothesis

- Ho: $\beta_1 = \beta_2 = \ldots = \beta_k$
- In our case,
  - Ho: $\beta_{age} = \beta_{height}$

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \epsilon_i$$

Residual Errors:

"What is the estimation of the difference between observed and predicted values?"

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

This error gives an idea about how far the observed *Lung Capacity* (dependent) values are from the predicted or fitted *Lung Capacity* (the "y-hats")

Slope of Age:

"How is my *Age* variable related to *Height*?"

```
Call:
lm(formula = LungCap ~ Age

Residuals:
    Min      1Q  Median
-3.4080 -0.7097 -0.0078

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

The effect of *Age* on *Lung Capacity* adjusting or controlling for *Height*. There is an increase of 1 year in *Age* with an increase of 0.126 in *Lung Capacity* adjusting or controlling for *Height*

Test Statistic:

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

The test statistic that we use to perform the hypothesis test that the slope for *Age* = 0.

Test Statistic:

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

The estimated effect of *Height* on *Lung Capacity* adjusted for *Age*.

Test Statistic:

```
Call:
lm(formula = LungCap ~ Age + Height)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4080 -0.7097 -0.0078  0.7167  3.1679

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.747065   0.476899 -24.632  < 2e-16 ***
Age           0.126368   0.017851   7.079 3.45e-12 ***
Height        0.278432   0.009926  28.051  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.056 on 722 degrees of freedom
Multiple R-squared:  0.843,     Adjusted R-squared:  0.8425
F-statistic:  1938 on 2 and 722 DF,  p-value: < 2.2e-16
```

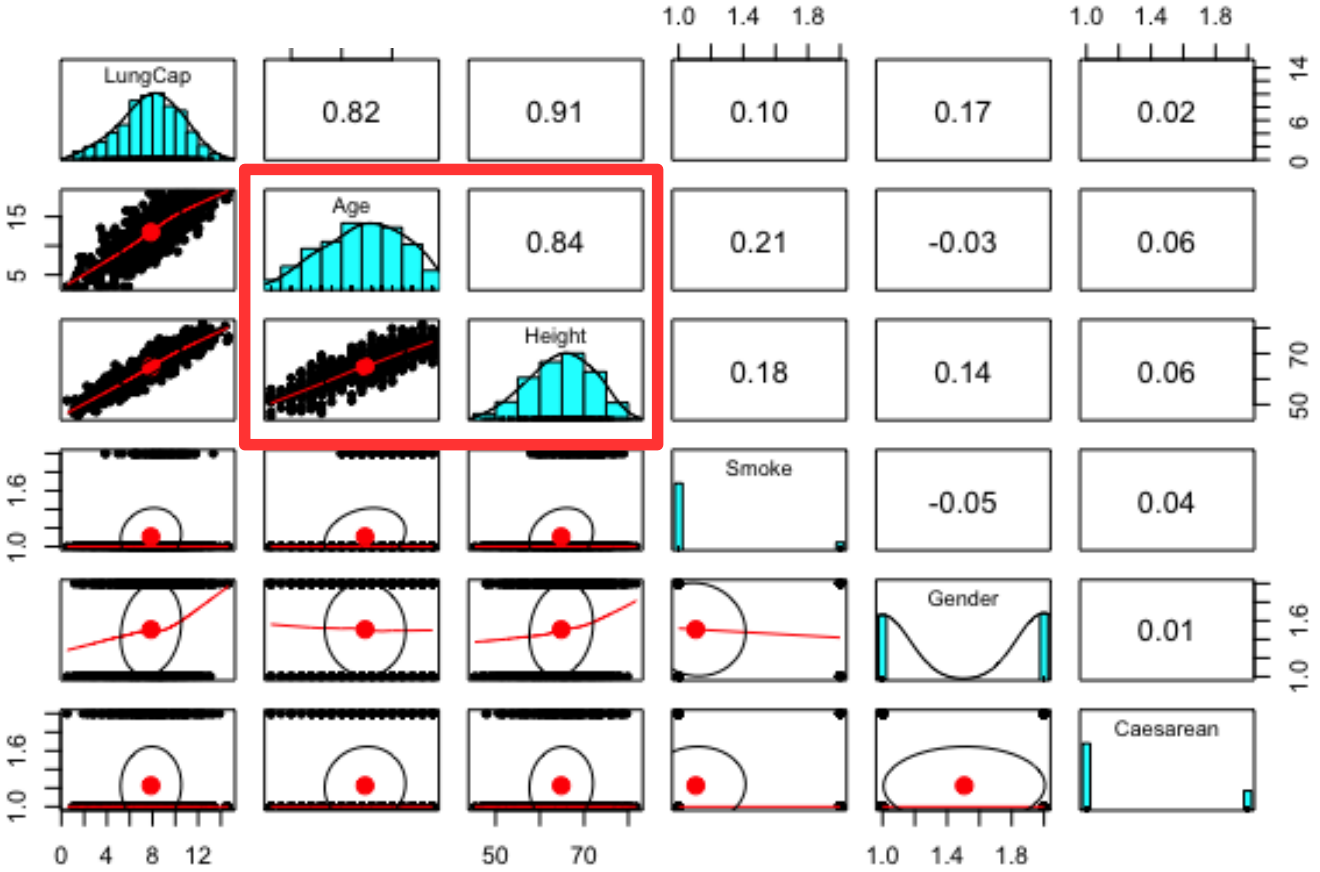
The test statistic that we use to perform the hypothesis test that the slope for *Height* = 0.

Correlation between *Age* and *Height*

- Pearson correlation between Age and Height = 0.84

```
> cor(Age, Height, method = "pearson")
[1] 0.8357368
```

- The high correlation between *Age* and *Height* suggests that these two effects are related.



Correlation and Confidence

```
#Pearson correlation test
cor(Age, Height, method = "pearson")
#output: 0.8357368
#Examine the 95 per cent confidence level
confint(mod, conf.level = 0.95)
```

The **estimated slope** for Age is 0.126 and we are 95 per cent sure that the **true slope** of Age is between 0.09 and 0.16.

```
> confint(mod, conf.level = 0.95)
                    2.5 %      97.5 %
(Intercept) -12.68333877 -10.8107918
Age           0.09132215   0.1614142
Height        0.25894454   0.2979192
```

Create Bigger Model!!

- Use this data set to make a bigger model.
  - Run Pearson's Correlation analysis over the elements.
  - Describe the outcome of the model.
- Are any the independent variables significant?
- What is the result of the correlations?
- Describe the significance of model.

Data Analytics
CS390
Text Analysis:
Sentiment Determination

Fall 2017

Oliver Bonham-Carter

Text Analysis: Sentiment of Content

- The determination of the text's "message" or "mood" based on the actual individual *words*.
- How good, how bad is the writer feeling about some topic?
- Is a body of text describing some idea where many of the words are emotionally charged with some type of feeling?
- Sentiment analysis is able to determine what the general feeling is behind some written work.

Packages and Libraries

```
install.packages(“janeaustenr")
install.packages(“stringr”)

library(janeaustenr)
library(dplyr)
library(stringr)
```

Data: Jane Austen's Text

- Jane Austen's 6 completed, published novels from the *janeaustenr* package.
  - Sense & Sensibility
  - Pride & Prejudice
  - Mansfield Park
  - Emma
  - Northanger Abbey
  - Persuasion

Research Question

- Jane Austen's written work:

  How many *Bad* words did she use?

  How many *Good* words did she use?

The *Sentiments* dataset

```
install.packages("tidytext")
library(tidytext)
sentiments
```

```
## # A tibble: 27,314 × 4
##           word sentiment lexicon score
##          <chr>     <chr>   <chr> <int>
## 1       abacus     trust     nrc    NA
## 2      abandon      fear     nrc    NA
## 3      abandon  negative     nrc    NA
## 4      abandon   sadness     nrc    NA
## 5    abandoned     anger     nrc    NA
## 6    abandoned      fear     nrc    NA
## 7    abandoned  negative     nrc    NA
```

Three general-purpose lexicons

- *AFINN* from Finn Årup Nielsen,
  - assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment
- *bing* from Bing Liu and collaborators,
  - categorizes words in a binary fashion into positive and negative categories
- *nrc* from Saif Mohammad and Peter Turney
  - categorizes words in a binary fashion ("yes"/"no") into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

- Used to determine the general mood of words.
- Lexicons are based on unigrams, (i.e., single words).
- Words are assigned scores for positive/negative sentiment,
- Emotions: joy, anger, sadness and etc.

Sentiments: afinn

- get_sentiments("afinn")

```
> get_sentiments("afinn")
# A tibble: 2,476 x 2
          word score
         <chr> <int>
 1     abandon    -2
 2   abandoned    -2
 3    abandons    -2
 4    abducted    -2
 5   abduction    -2
 6  abductions    -2
 7       abhor    -3
 8    abhorred    -3
 9   abhorrent    -3
10       abhors    -3
# ... with 2,466 more rows
```

Returns a score for each word [-5, 5] (Bad to Good)

Sentiments: nrc

- get_sentiments("nrc")

```
> get_sentiments("nrc")
# A tibble: 13,901 x 2
          word sentiment
         <chr>     <chr>
 1       abacus     trust
 2      abandon      fear
 3      abandon  negative
 4      abandon   sadness
 5    abandoned     anger
 6    abandoned      fear
 7    abandoned  negative
 8    abandoned   sadness
 9 abandonment     anger
10 abandonment      fear
# ... with 13,891 more rows
```

Returns a *synonym* for each word

Sentiments: nrc

```
get_sentiments("bing")
```

```
> get_sentiments("bing")
# A tibble: 6,788 x 2
          word sentiment
         <chr>     <chr>
 1     2-faced  negative
 2     2-faces  negative
 3          a+  positive
 4    abnormal  negative
 5     abolish  negative
 6  abominable  negative
 7  abominably  negative
 8   abominate  negative
 9 abomination  negative
10       abort  negative
# ... with 6,778 more rows
```

Returns
A
Positive
or
Negative
measurement
for each word

Setup

```
original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
  chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]", ignore_case = TRUE)))) %>%
  ungroup()


original_books
```

Chapter Words

- The words in the order that they appear in the text.
- Note the first line is the title of the book.

```
## # A tibble: 73,422 x 4
##    text                  book                linenumber chapter
##    <chr>                 <fctr>                   <int>   <int>
##  1 SENSE AND SENSIBILITY Sense & Sensibility          1       0
##  2 ""                    Sense & Sensibility          2       0
##  3 by Jane Austen        Sense & Sensibility          3       0
##  4 ""                    Sense & Sensibility          4       0
##  5 (1811)                Sense & Sensibility          5       0
##  6 ""                    Sense & Sensibility          6       0
##  7 ""                    Sense & Sensibility          7       0
##  8 ""                    Sense & Sensibility          8       0
##  9 ""                    Sense & Sensibility          9       0
## 10 CHAPTER 1             Sense & Sensibility         10       1
## # ... with 73,412 more rows
```

Unnesting Book Words

We need the words in list (un-nested) to work with them.

```
tidy_books <- original_books %>%
  unnest_tokens(word, text) #make a list of words from the paragraphs


tidy_books
```

Unnested Words

```
## # A tibble: 725,055 x 4
##    book                linenumber chapter word
##    <fctr>                   <int>   <int> <chr>
##  1 Sense & Sensibility          1       0 sense
##  2 Sense & Sensibility          1       0 and
##  3 Sense & Sensibility          1       0 sensibility
##  4 Sense & Sensibility          3       0 by
##  5 Sense & Sensibility          3       0 jane
##  6 Sense & Sensibility          3       0 austen
##  7 Sense & Sensibility          5       0 1811
##  8 Sense & Sensibility         10       1 chapter
##  9 Sense & Sensibility         10       1 1
## 10 Sense & Sensibility         13       1 the
## # ... with 725,045 more rows
```

When words are in one-word-per-row format, manipulation with tidy tools like *dplyr* is possible

Stop Words

- Stop words are words which do not add any distinguishing information to a body of text.
  - Contractions: hasn’t, didn’t won’t
  - In-betweens: been, is, had, having

```
data("stop_words")

View(stop_words)

cleaned_books <- tidy_books %>%
anti_join(stop_words)

# anti_join() returns all rows from x where there are not matching values in y, keeping just columns from x.
```

Counting Common Words Across All Books

```
cleaned_books %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 13,914 x 2
##    word       n
##    <chr>  <int>
##  1 miss    1855
##  2 time    1337
##  3 fanny    862
##  4 dear     822
##  5 lady     817
##  6 sir      806
##  7 day      797
##  8 emma     787
##  9 sister   727
## 10 house    699
## # ... with 13,904 more rows
```

Joy in Emma

- Consider common words having *joy* scores using the nrc lexicon in Emma

```
nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
tidy_books %>%
  filter(book == "Emma") %>%
  semi_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

Oh Joy ...

```
tidy_books %>%
  filter(book == "Emma") %>%
  semi_join(nrcjoy) %>%
 count(word, sort = TRUE)
```

```
## # A tibble: 303 x 2
##    word          n
##    <chr>     <int>
##  1 good        359
##  2 young       192
##  3 friend      166
##  4 hope        143
##  5 happy       125
##  6 love        117
##  7 deal         92
##  8 found        92
##  9 present      89
## 10 kind         82
## # ... with 293 more rows
```

How Does Sentiment Change? (In each novel?)

```
library(tidyr)

bing <- get_sentiments("bing")


janeaustensentiment <- tidy_books %>%

  inner_join(bing) %>%

  count(book, index = linenumber %/% 80, sentiment)
        %>% spread(sentiment, n, fill = 0) %>%
        mutate(sentiment = positive - negative)
```

What Are The Most Common Good and Bad Words?

- Count the common positive words across the books.

```
bing_word_counts <- tidy_books %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

bing_word_counts
```

Such Positivity ...

bing_word_counts

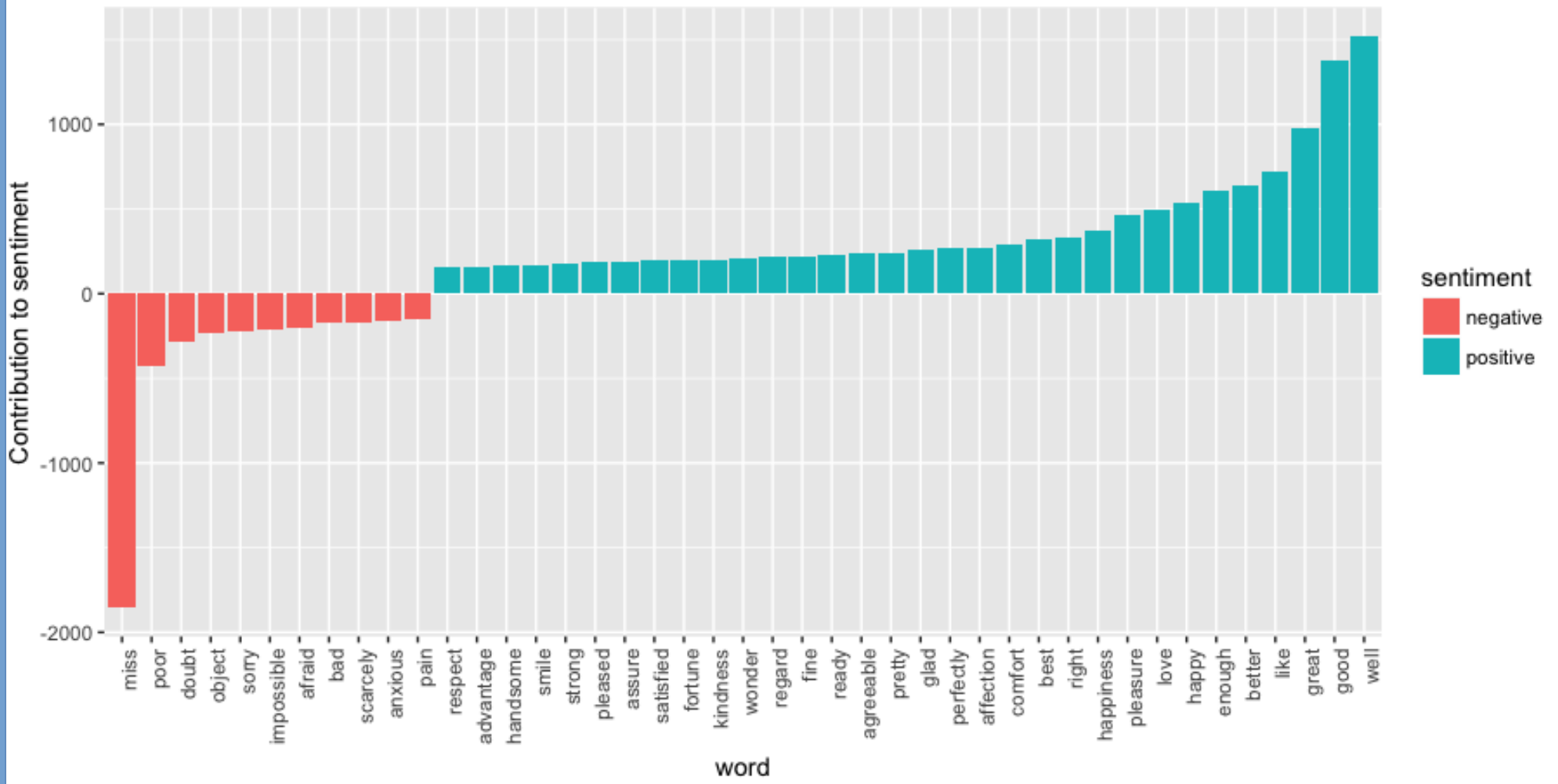
```
## # A tibble: 2,585 x 3
##    word      sentiment      n
##    <chr>     <chr>      <int>
##  1 miss      negative    1855
##  2 well      positive    1523
##  3 good      positive    1380
##  4 great     positive     981
##  5 like      positive     725
##  6 better    positive     639
##  7 enough    positive     613
##  8 happy     positive     534
##  9 love      positive     495
## 10 pleasure  positive     462
## # ... with 2,575 more rows
```

Plot of The Common Good and Bad words

- Plot the common positive words across the books.

```
bing_word_counts %>%
  filter(n > 150) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
 ylab("Contribution to sentiment")
```

Plot of Positive and Negative Sentiment Words



Header

- This

Header

- This

Header

- This