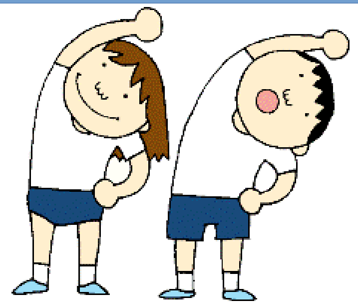
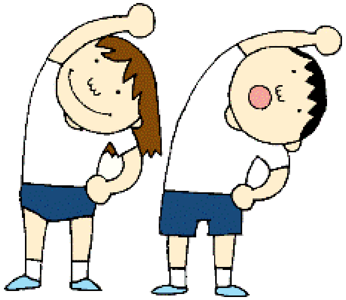


Warm-Up!

- You may have to use R interpreter from the terminal. Type "R" at the terminal and copy and paste in your code from your editor.
- General Question: What correlations exist in the BFI data concerning the factors of one's personality?
- Use **with()** and **corPlot()** to study correlations in the BFI dataset. Now Find:
  - The top thee most positively-correlated columns
  - The top three most negatively-correlated columns
  - The top three least-correlated columns.

Ideas? See File: warmUp_correlations.r



Warm-Up!

- Returning to the codebook, working with your group, can you offer a suggestions to explain the typess of correlations that you found?
- Use GGplot to graph some of your correlations. Can you tell from the graph what the correlation is?

Ideas? See File: warmUp_correlations.r

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?

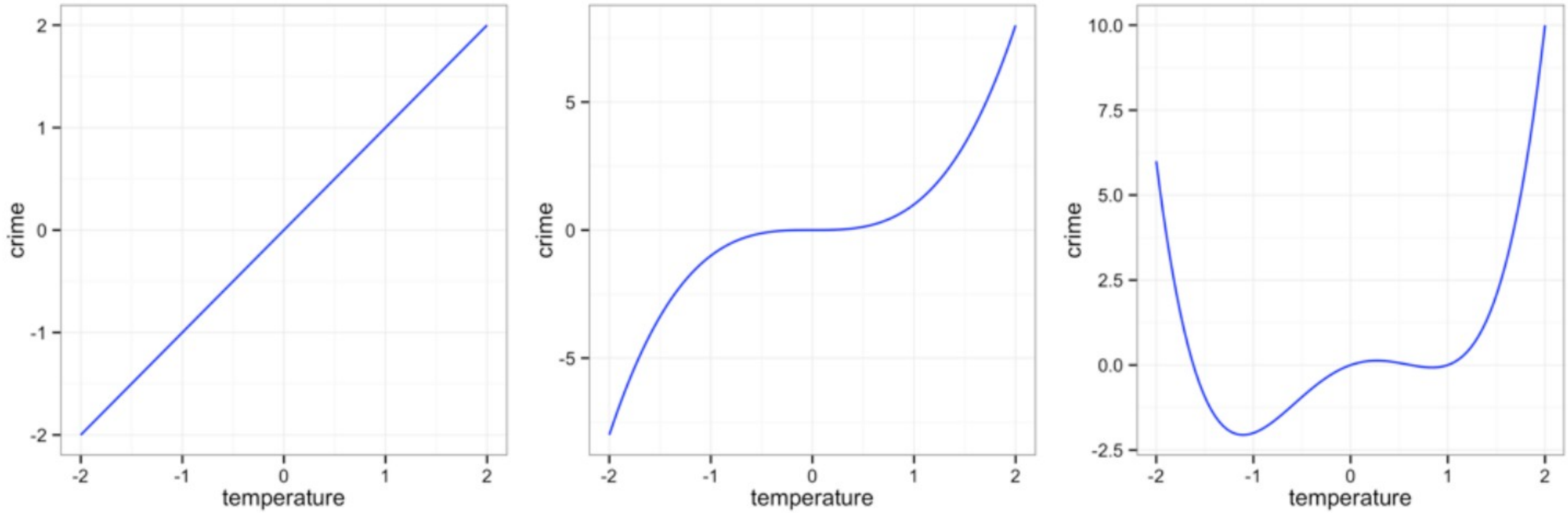
How Do we
Answer The Question?

- 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...

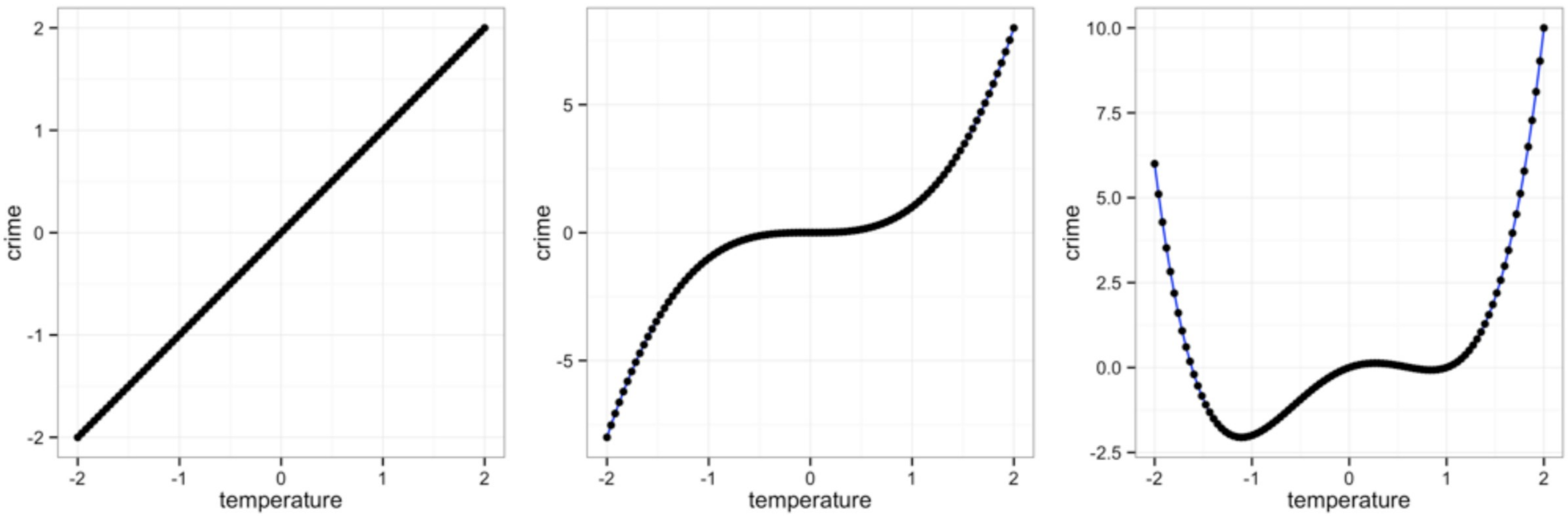
Functions:
the *stuff* behind the models

- A function is a mathematical description of a relationship.



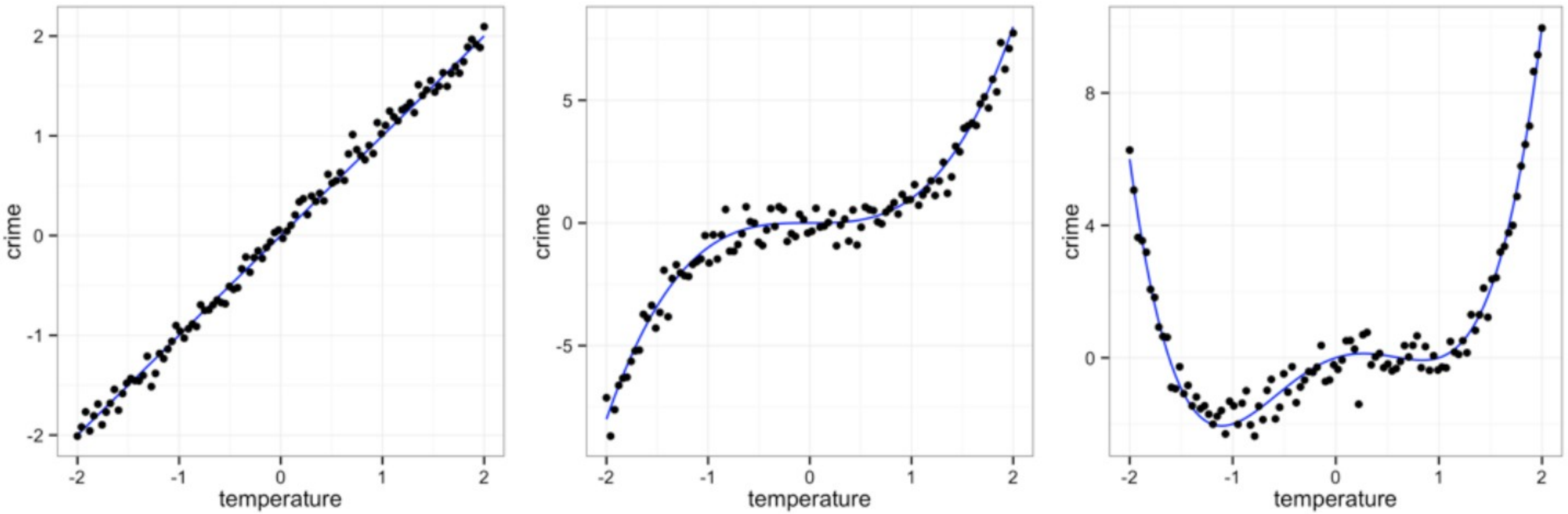
Functions:
the *stuff* behind the models

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



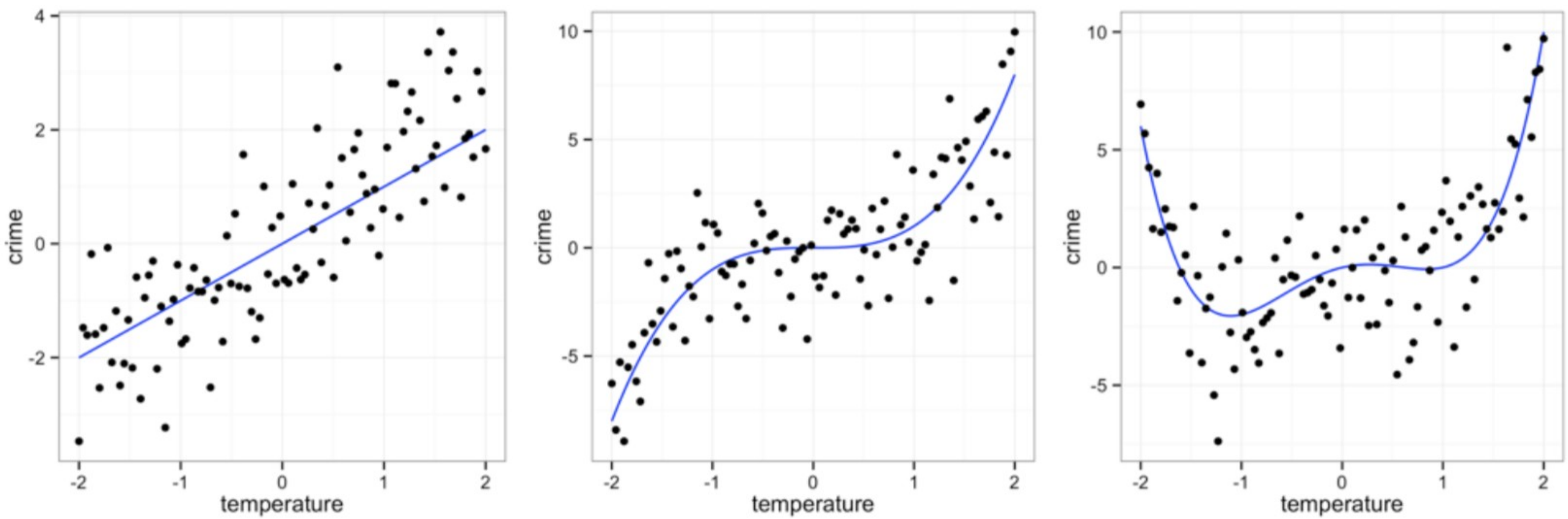
Relationships Between Variables Is Messy

- 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.**

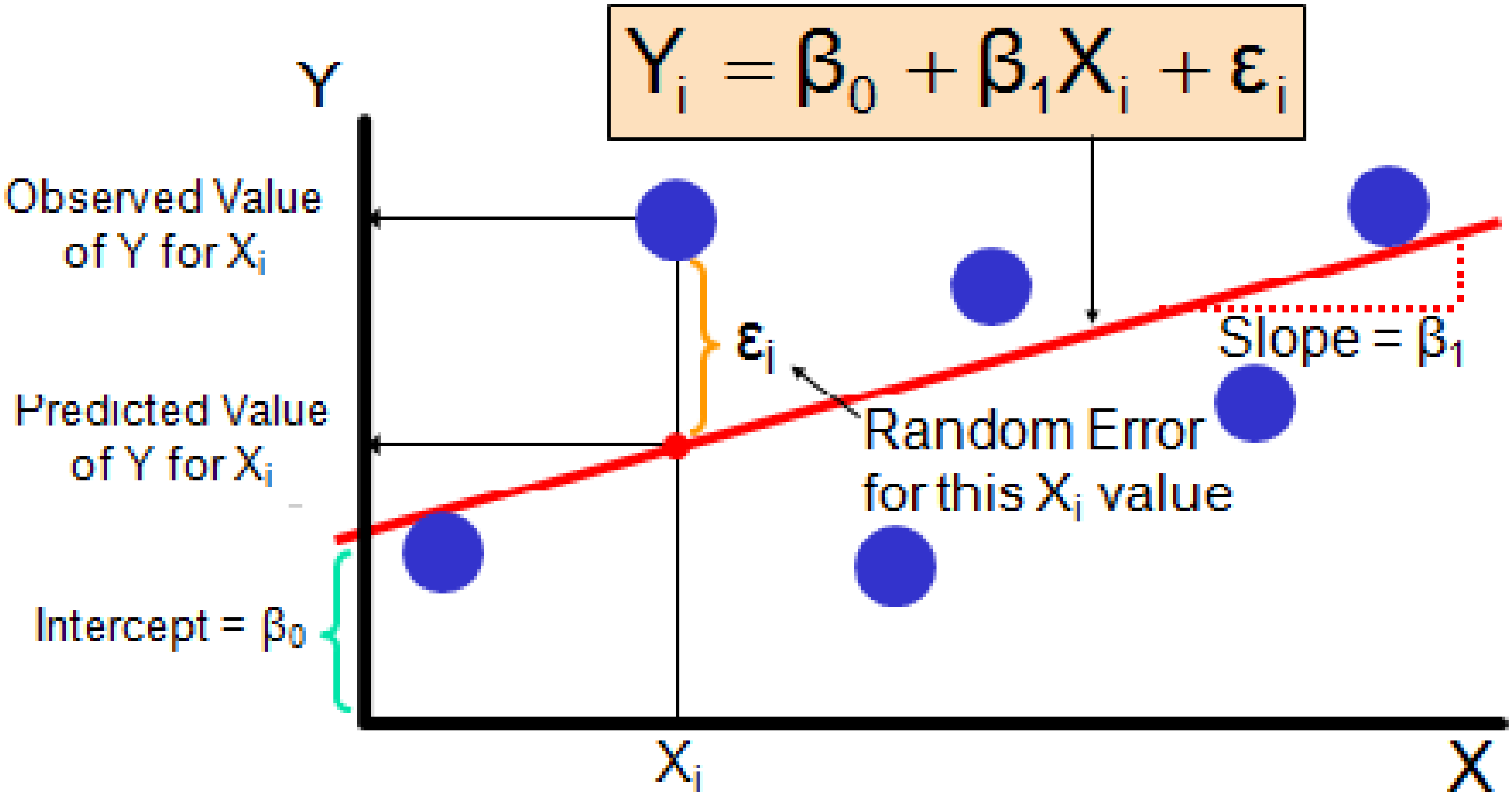
Data

Model

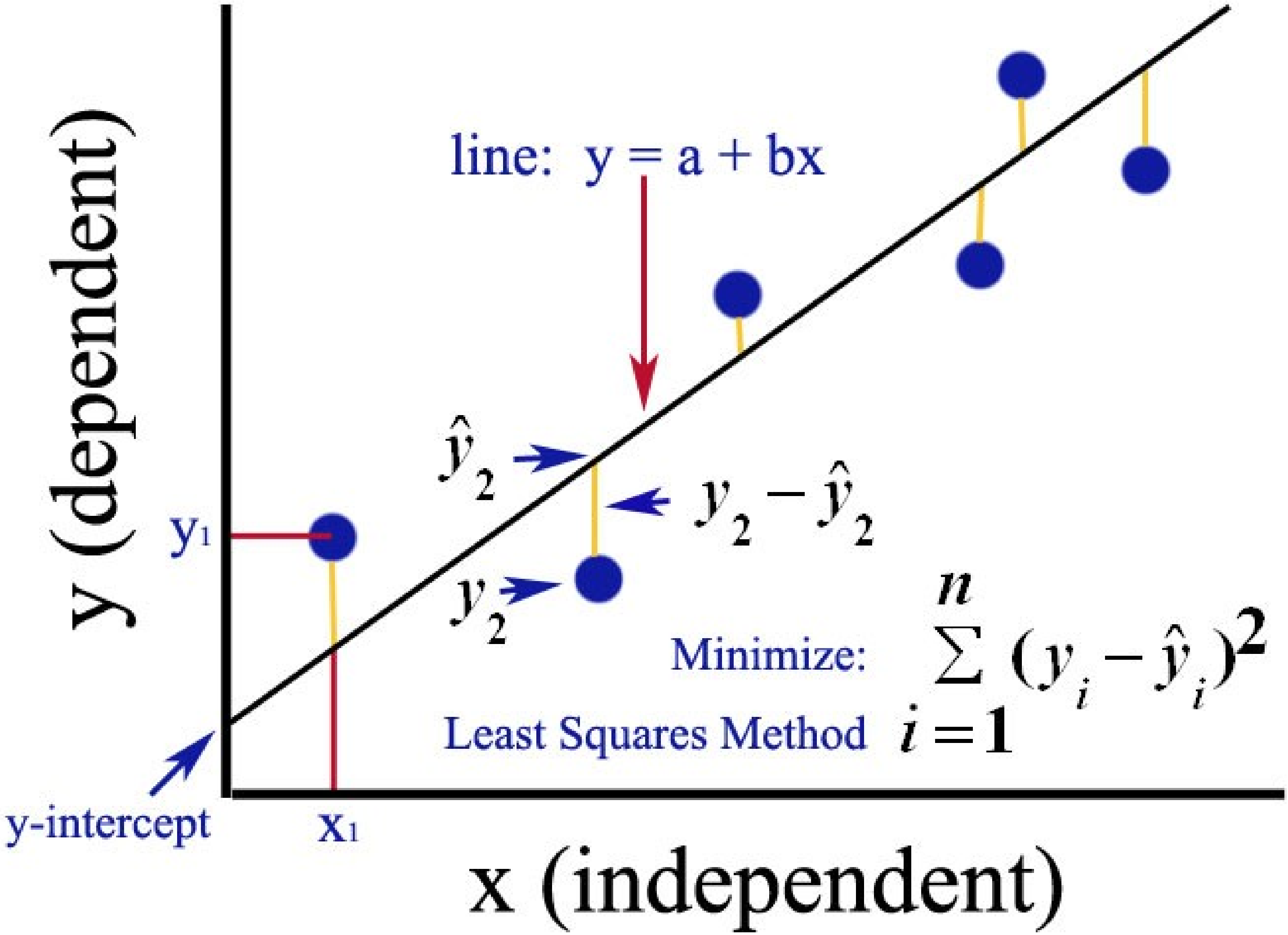
Let's Talk Linear Models

- 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

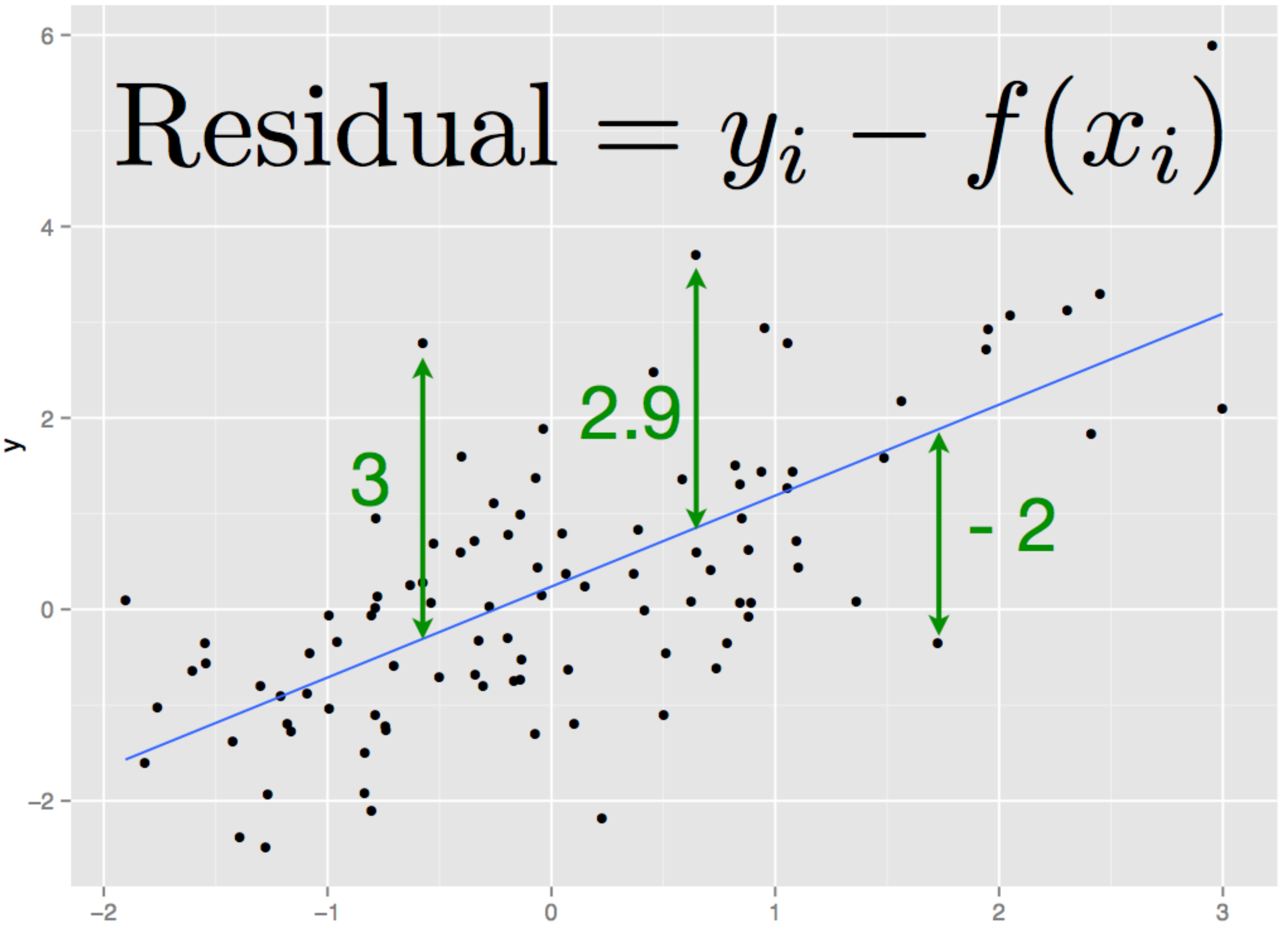


Another Linear Model



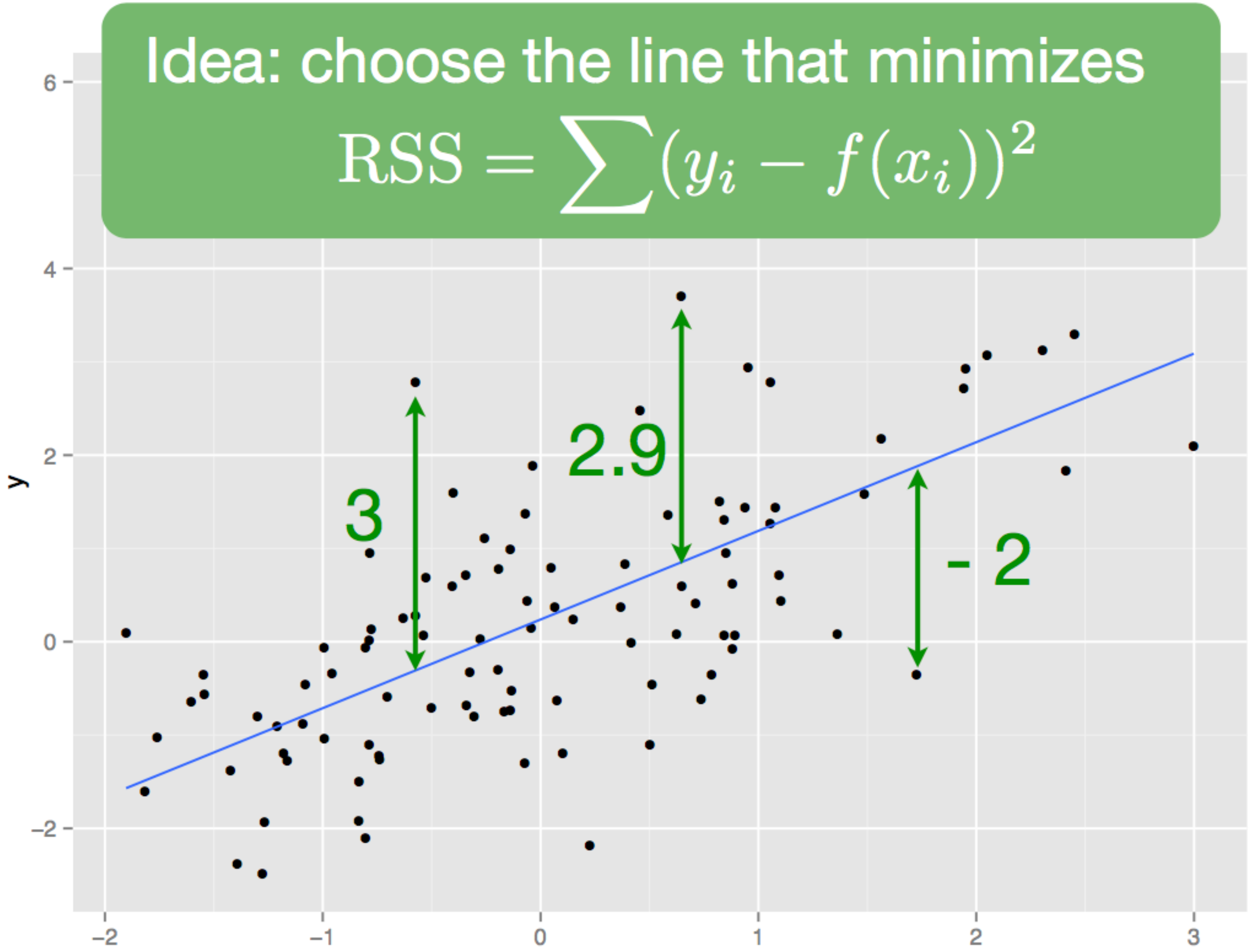
How To Best Draw a Line Through The Data?

- 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?

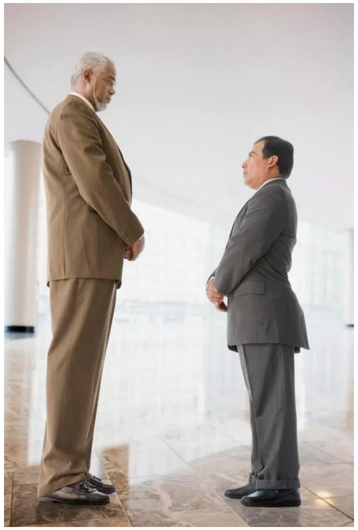
- Residual sum of squares (RSS), also known as the sum of squared residuals (SSR) or the sum of squared errors of prediction (SSE)
- The sum of the squares of residuals (deviations predicted from actual empirical values of data).



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
...					

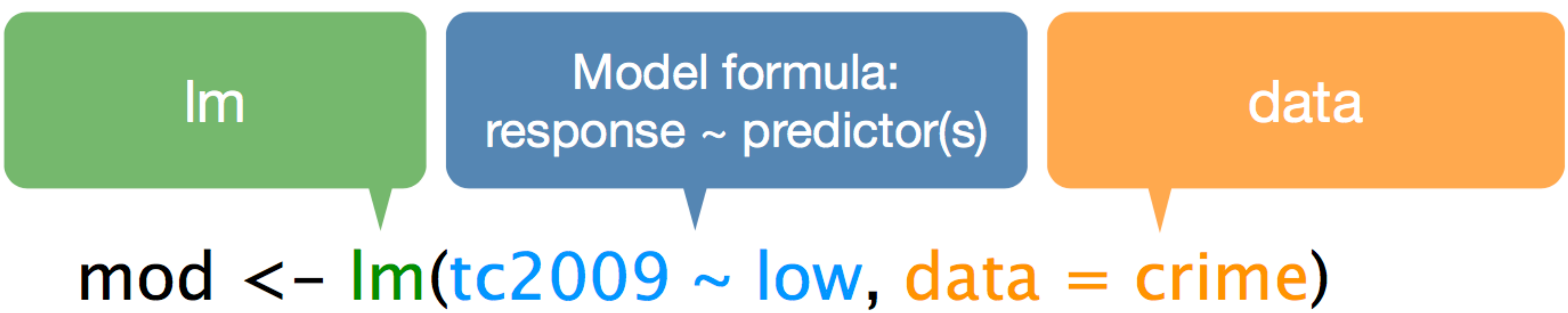
Yearly low temp

Murder rate

Training data

Let's Hit the Code

- How much ***low (indep)*** influence ***tc2009 (dep)***
- 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 ~ **x**

Formulas

```
response ~ explanatory

dependent ~ independent

outcome ~ predictors
```

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

Results: summary(mod)

mod

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

Coefficients:
(Intercept)          low
    4256.86        21.65
```

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
```

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 lm()

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

Let's Hit the Code

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