Cross validation

AKA most important lecture of your time here

with special thanks to Cary and Ryan

objectives

- figure out how to determine if a model is learning
- learn important vocab words
- think critically about model performance and how to score it

what are we doing here?

- lets talk about the process of data science

  A. define a business problem

     1. make tesla cars the most dependable cars around

  B. collect some relevant data

     2. car event logs, repair/service data, driver habits

  C. train a model

     3. features: event statistics, target: time to failure

  D. deploy model

     4. predict time to fail on parts, send notifications/technicians out to parts with low time

what does this mean?

what is relevant?

is there too much

(what model)

how do models work?

$$Y = f(\underline{X}) + \epsilon$$

$$\hat{Y} = \hat{f}(X)$$

$$\text{Error} \equiv F(Y, \hat{Y})$$

$$\text{MSE}: \tfrac{1}{2}(Y - \hat{Y})^2$$

how do models work?

coefficients (weights)

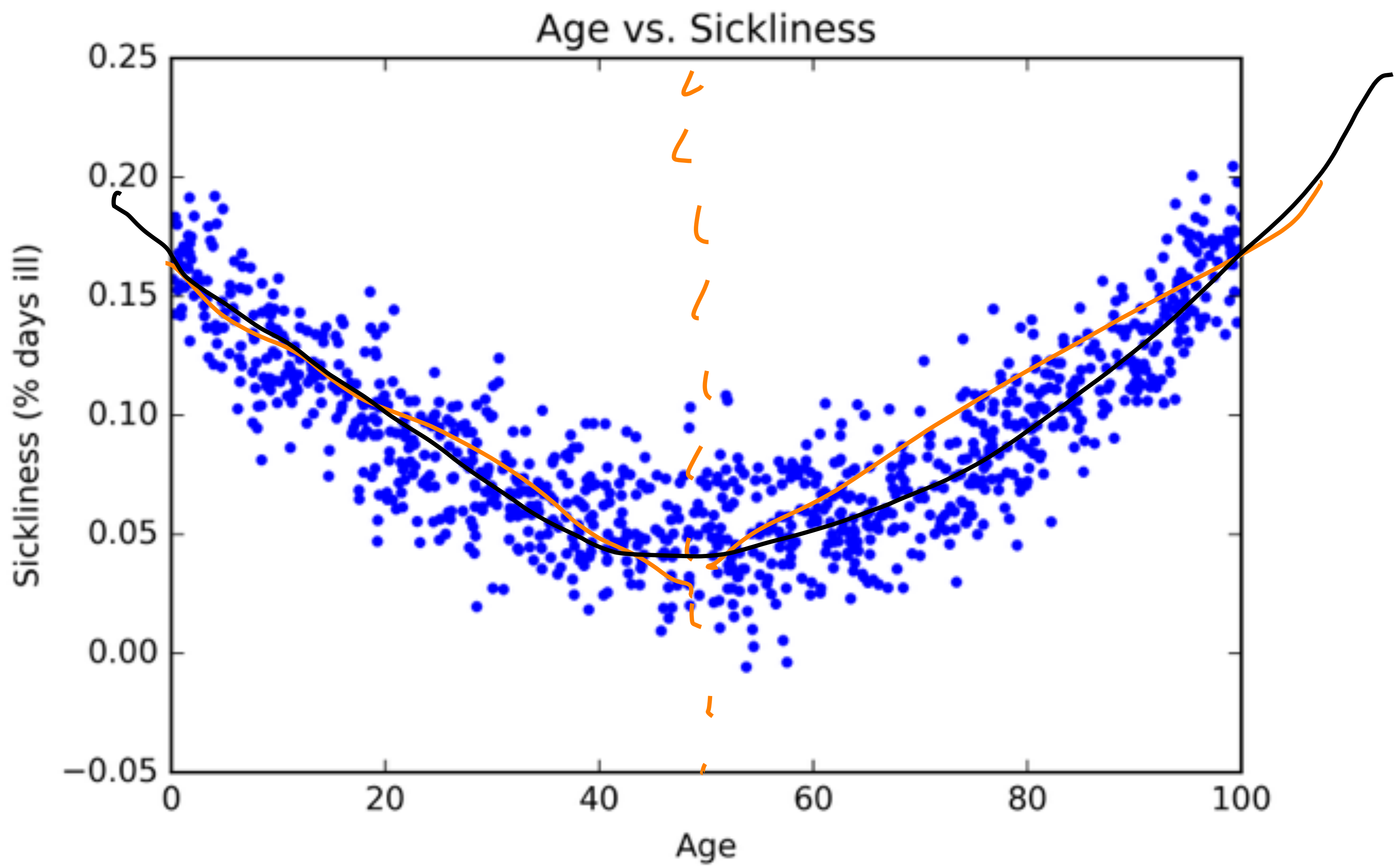
prediction

$$\boxed{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

intercept
(global average
TTF)

features

how do models work?



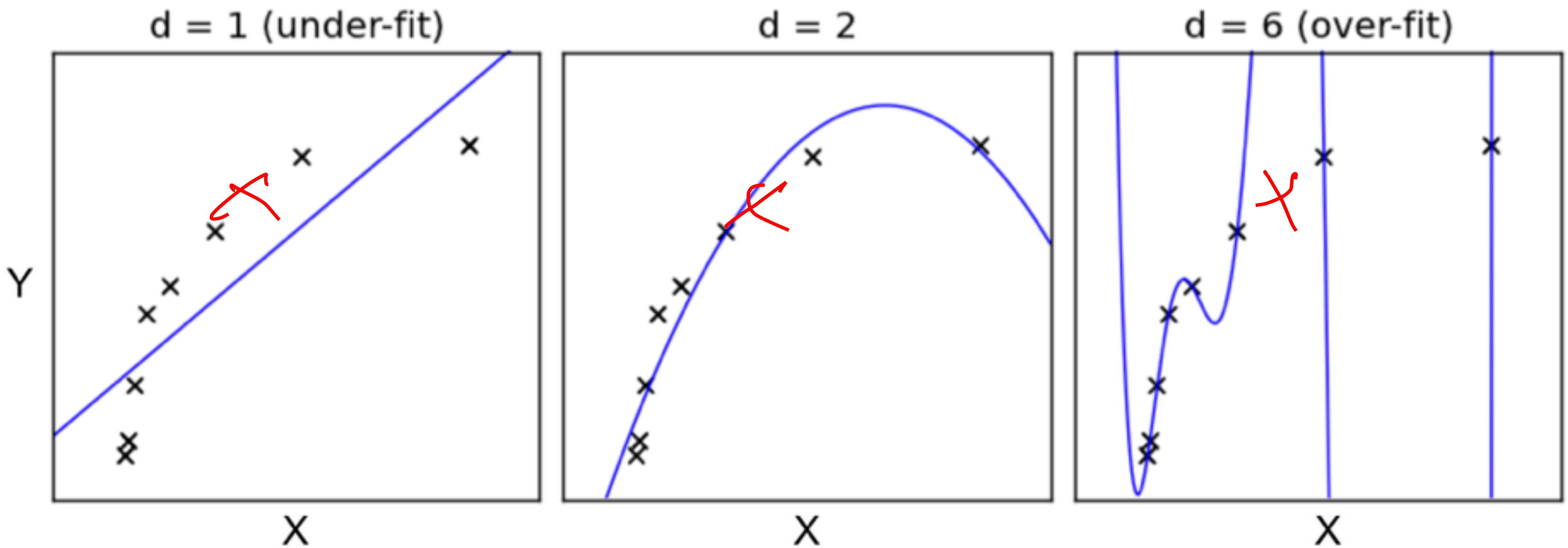
$$Y = \beta_0 + \beta_1 * \text{age}$$
$$Y = \beta_0 + \beta_1 * \text{age} + \beta_2 * \text{age}^2$$

solve all of data science

$y = \boxed{f(x)} + \epsilon$

```python
def super_awesome_model(X, y):
    model = LinearRegression()
    while True:
        model.fit(X, y)
        if calculate_r2(model, X, y) >= 0.999
            return model
        else:
            X = add_interaction_feature(X)
```

how you fit matters



underfitting and overfitting

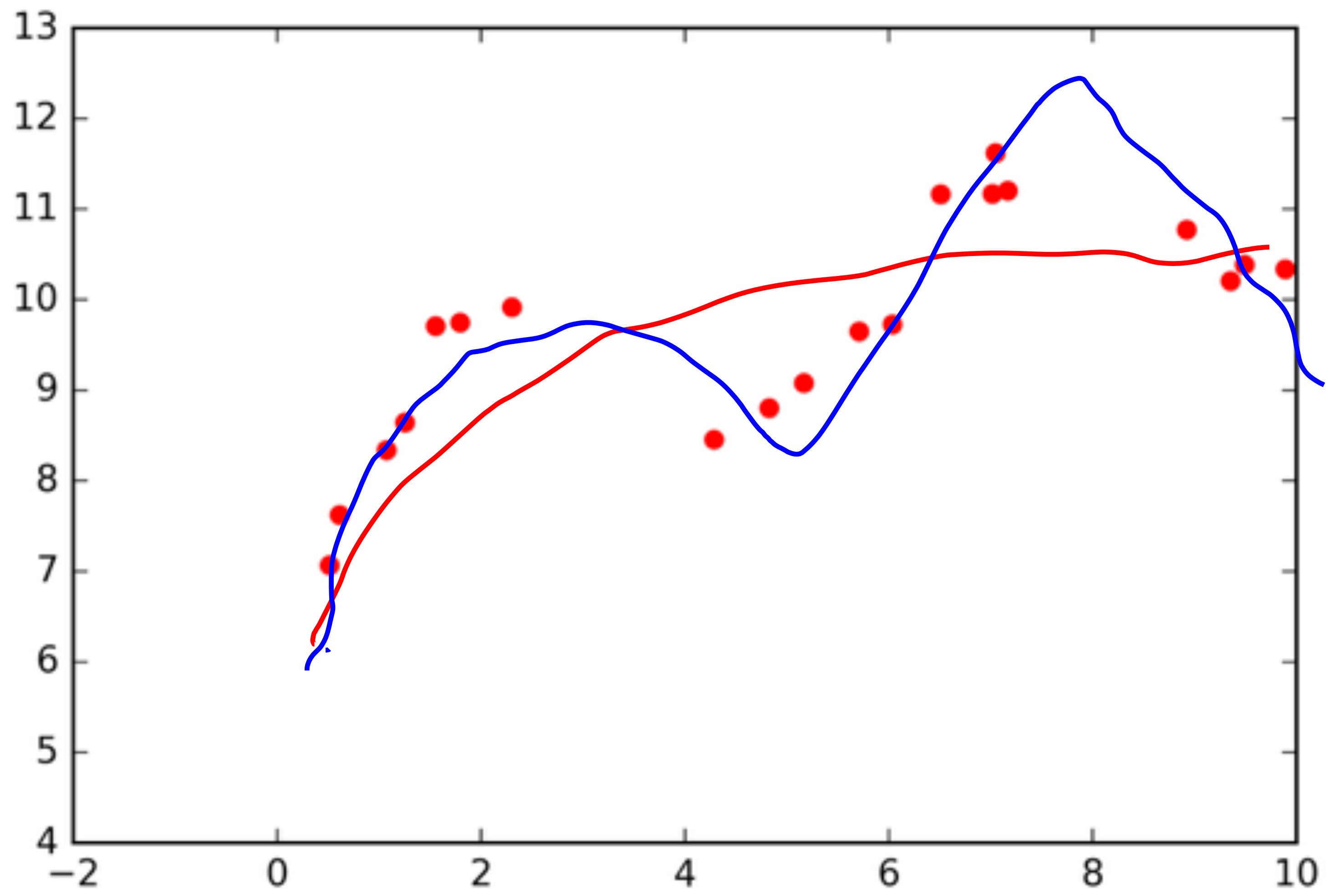
- underfitting is when we fail to properly learn the functional relationship in our data, we have not fully accounted for the signal
  - what can we do if we underfit our data?

– add features

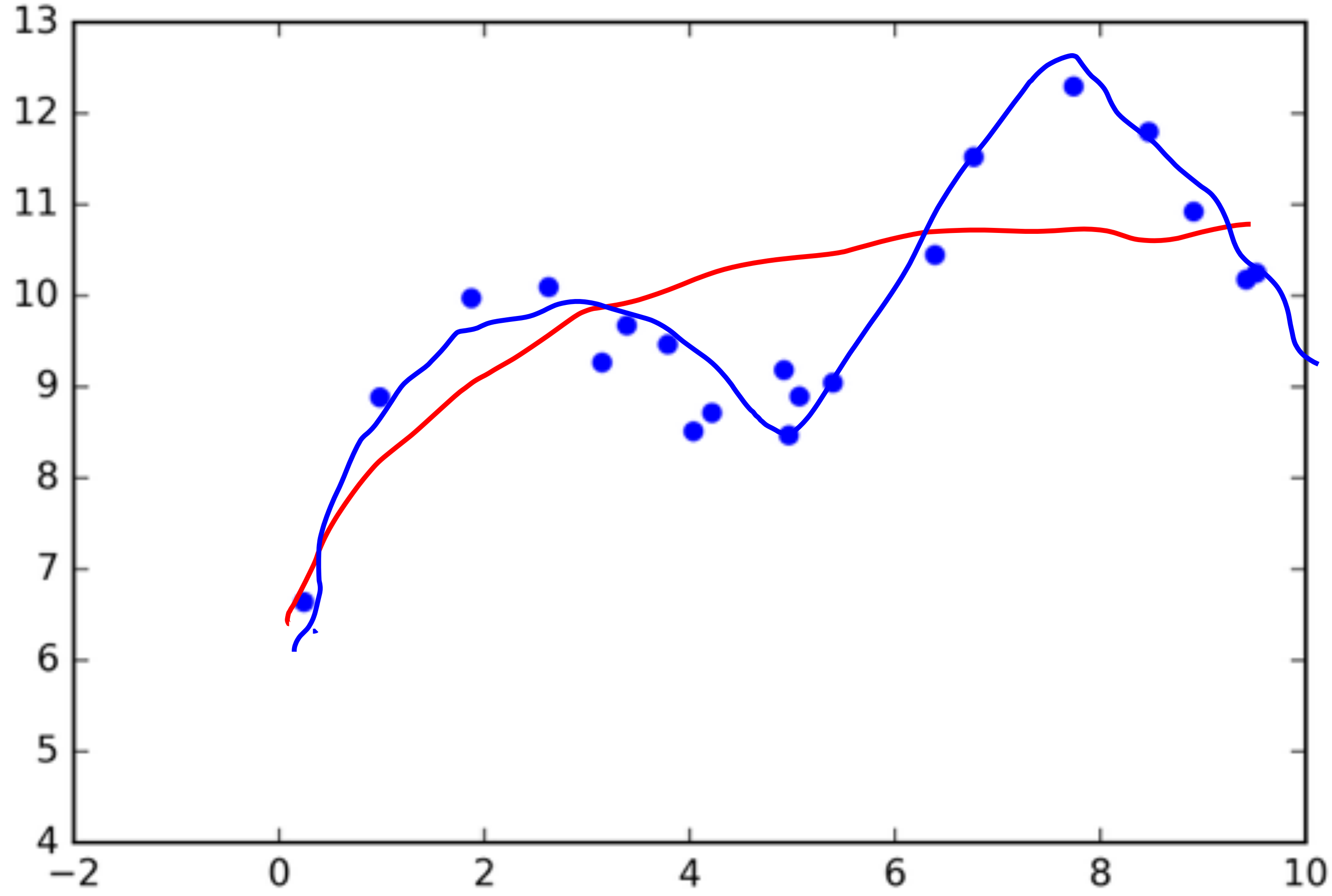
- overfitting is when we have learned the sampling error in our data, we have learned the signal and the noise
  - what can we do if we overfit our data?

– reduce features

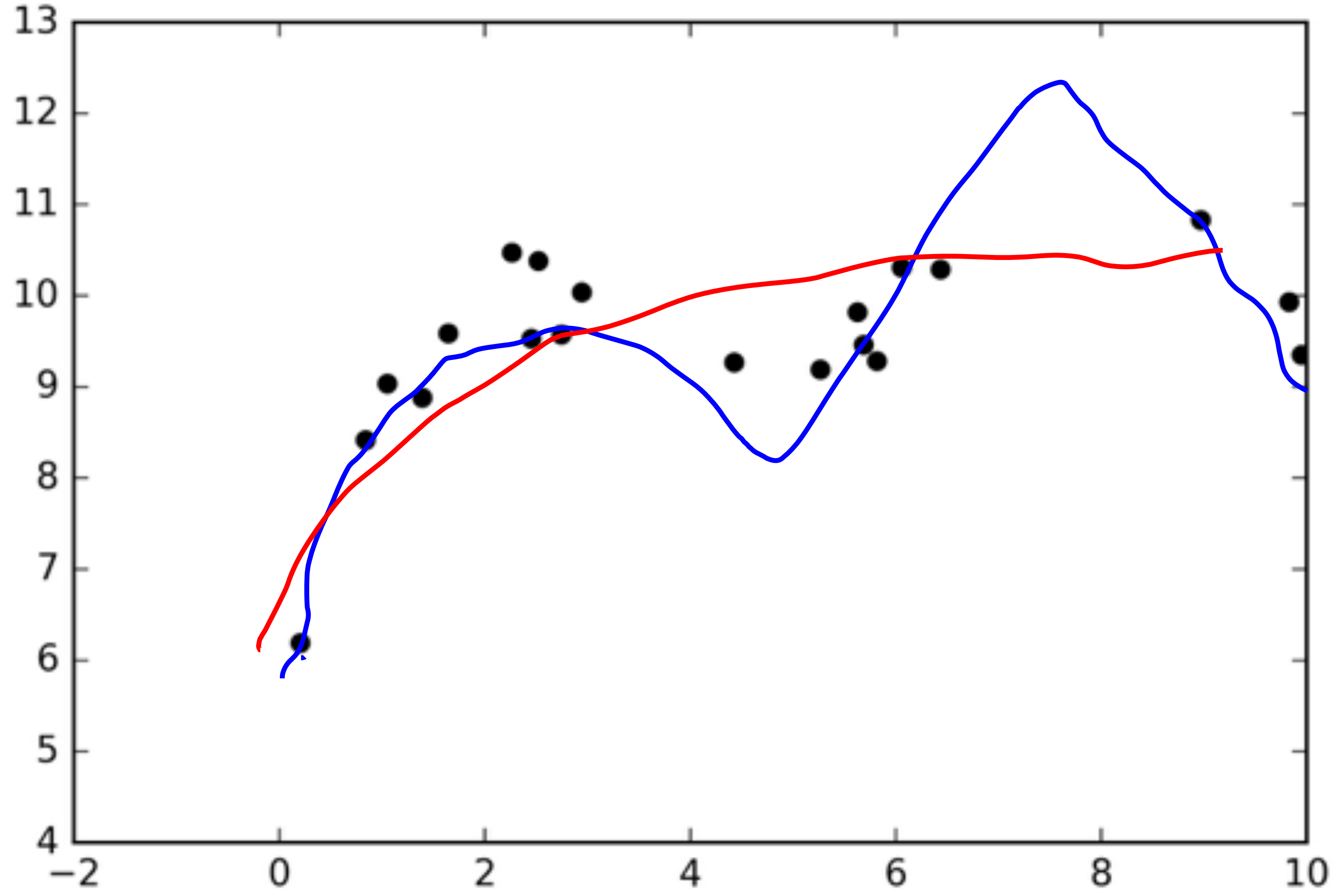
lets fit some data



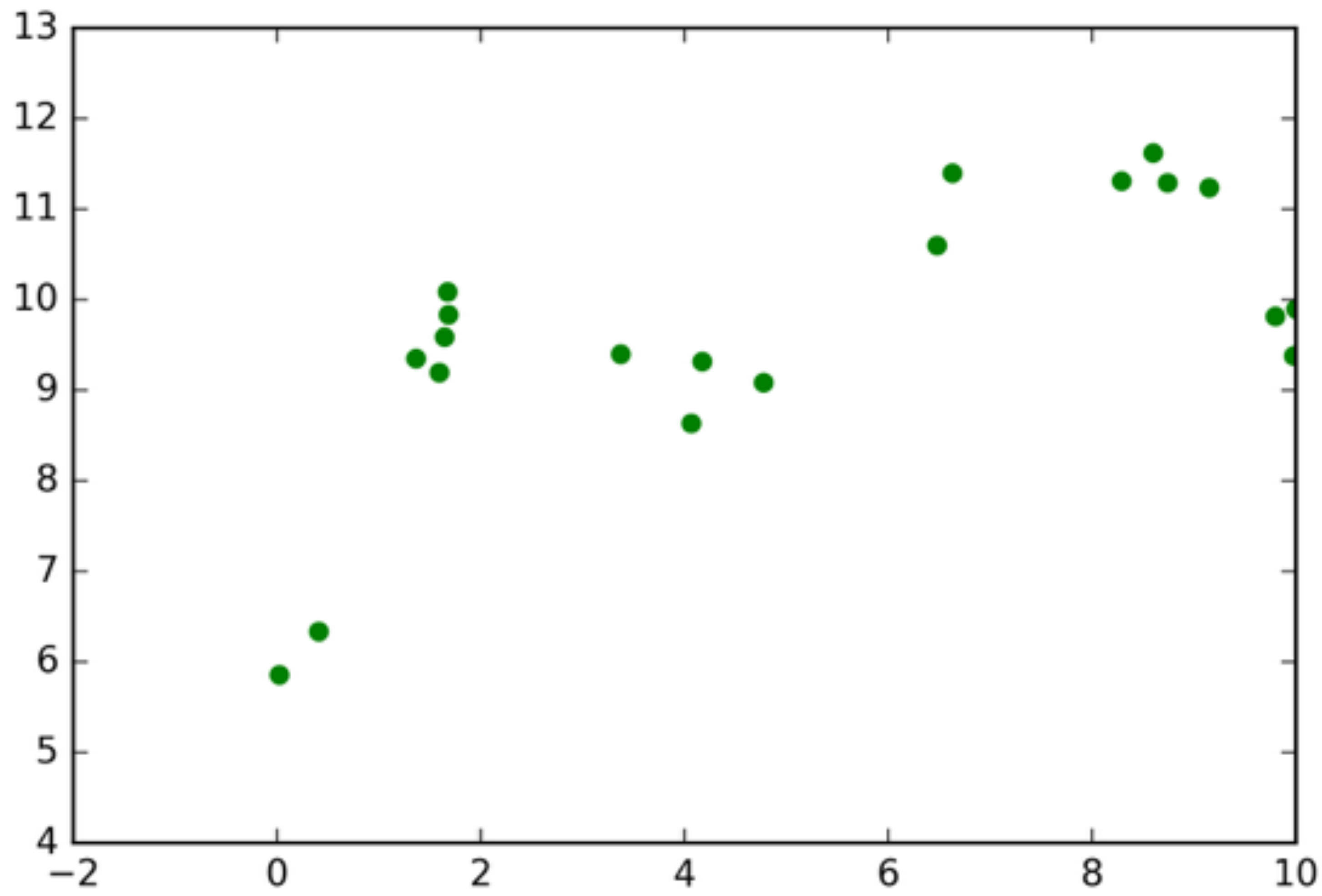
lets fit some data



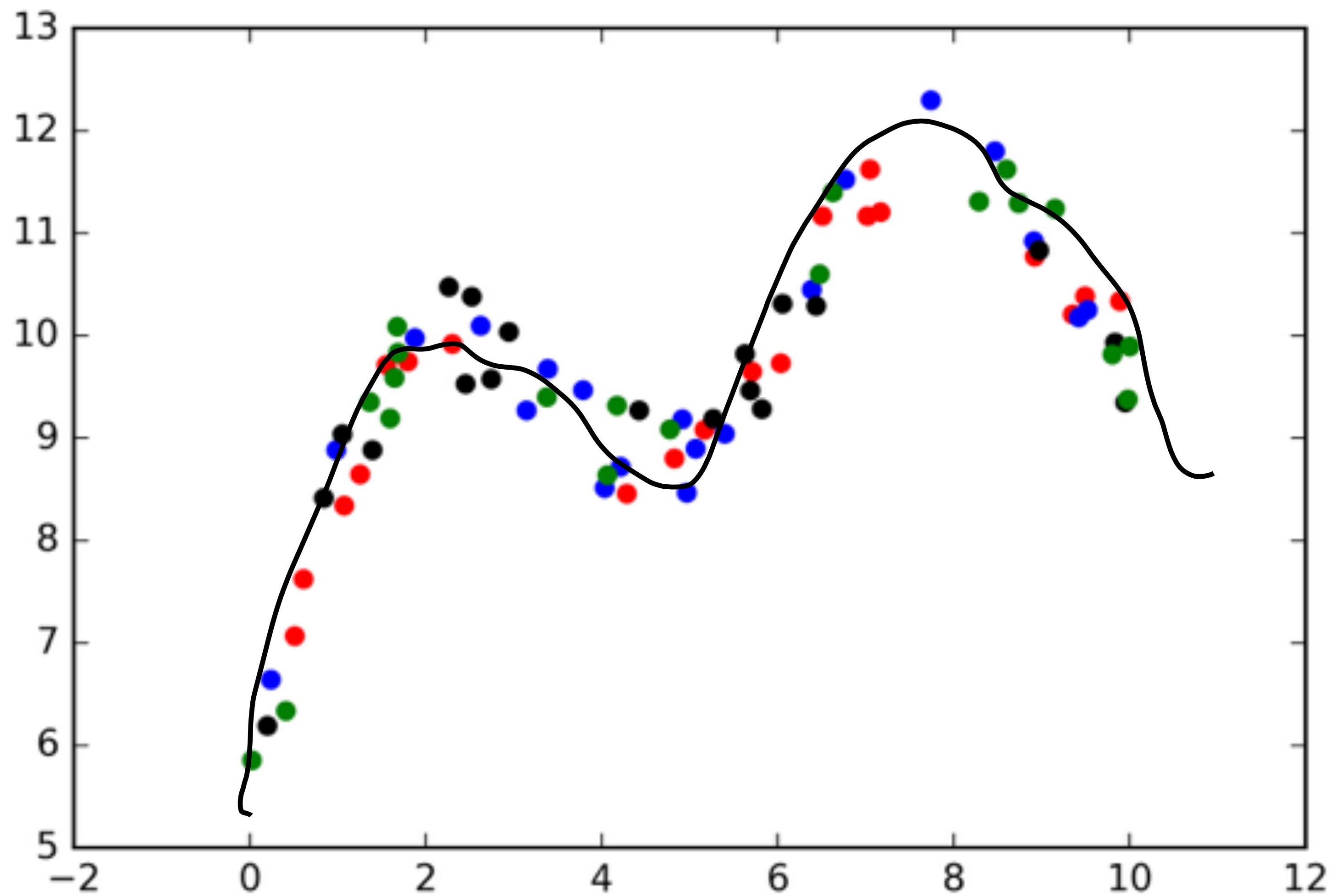
lets fit some data



lets fit some data



but what is going on behind the sampling?



what does this lead us to conclude?

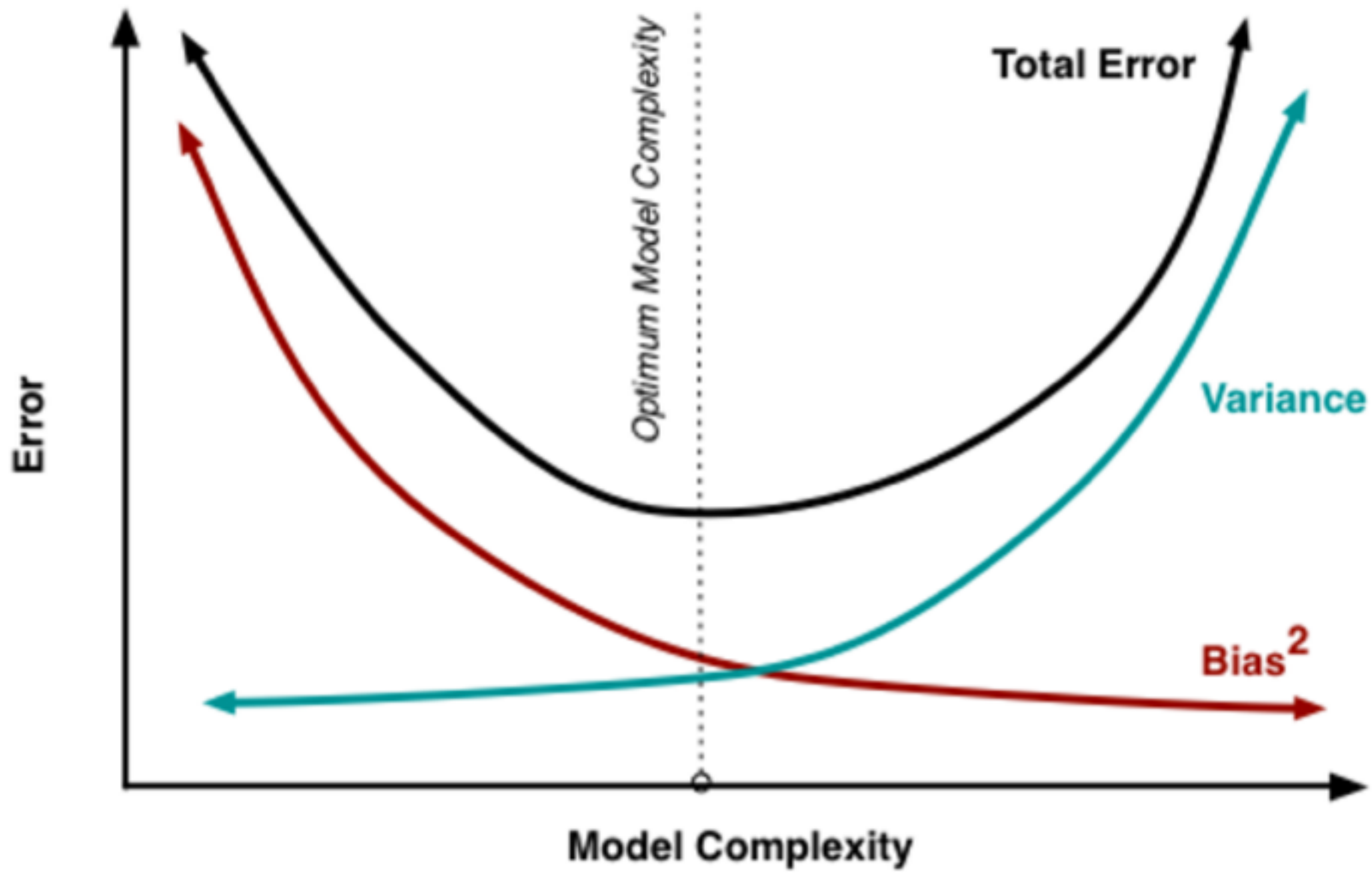
bias/variance tradeoff

MSE $(y-\hat{y})^2$

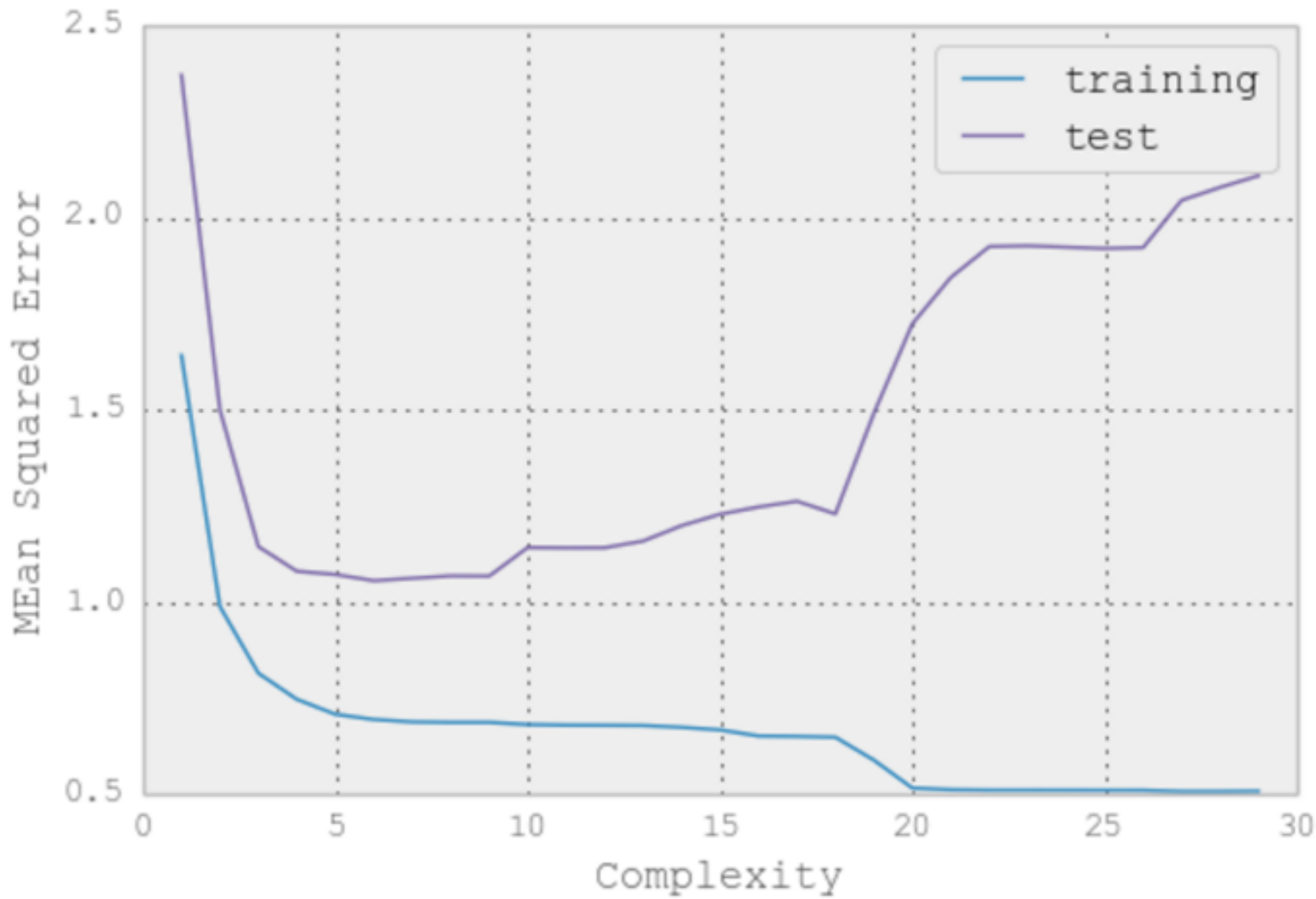
$= \text{bias}^2 + \text{variance} + \epsilon$

$\left(E(\hat{y})-f(x)\right)^2, \; E\left(\hat{y}-E(\hat{y})\right)^2$

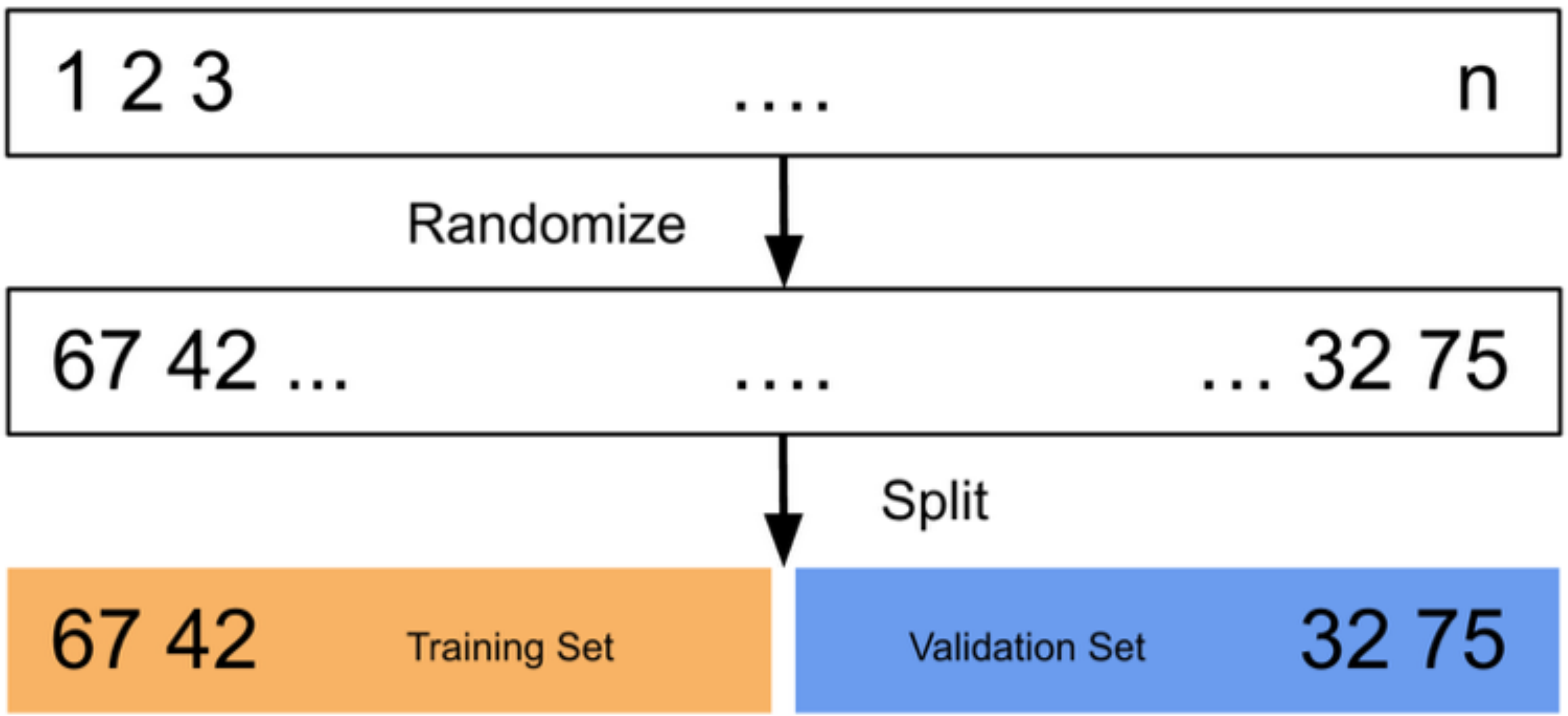
bias/variance tradeoff



train and test error



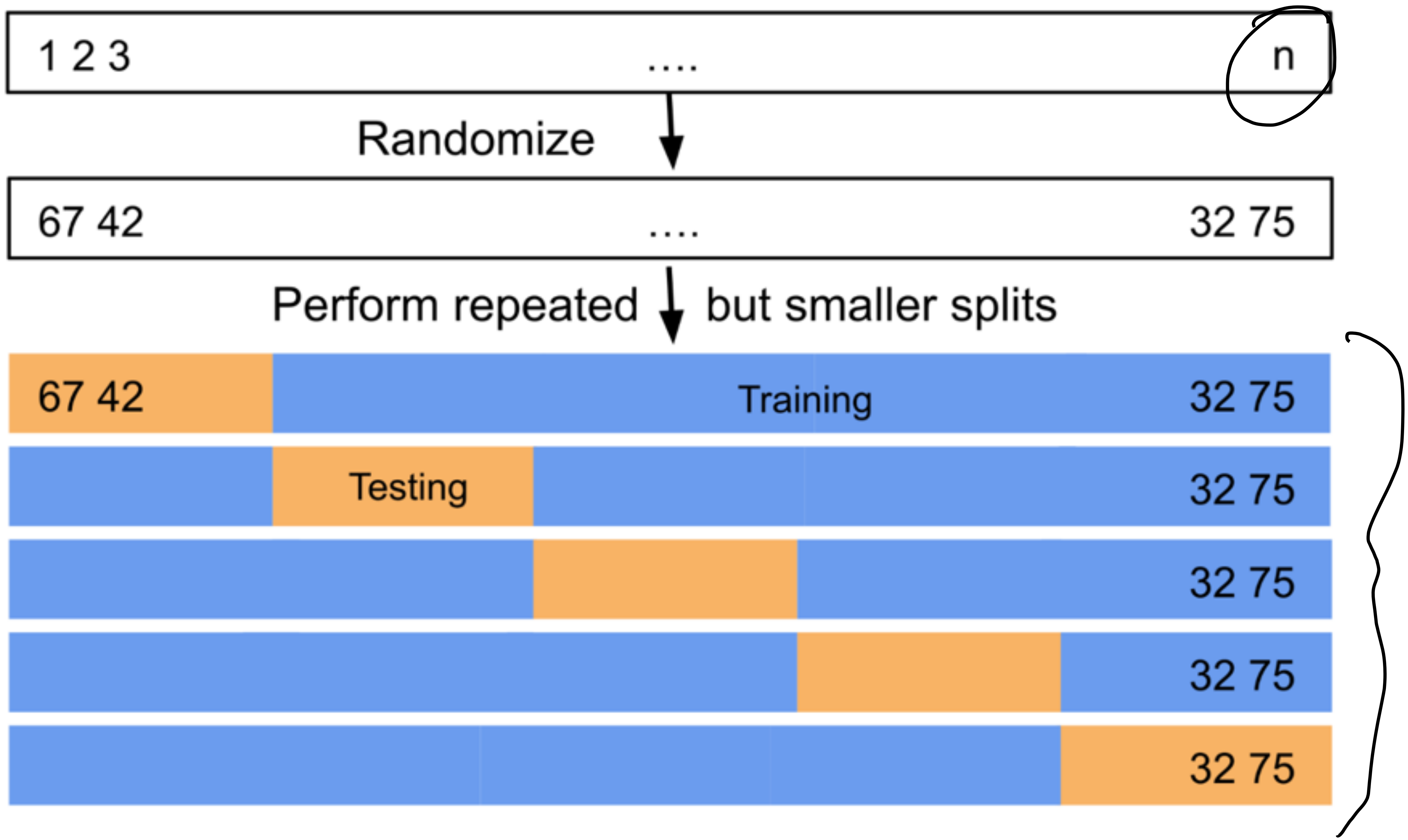
lets split off some of our data



what do we do now?

train	valid	test

k fold cross validation



what do we do now?

what if its overfitting?

- get more data
- reduce the dimensionality
- add a regularization term to the cost function

subset selection

- ~~figure out the best subset of features to use~~ 2^p $p!$

or

- iterate through features, pick the best model you find

forward stepwise selection

$p + p - 1$

- M_0 -> M_1 -> M_2 -> … -> M_p

- how many models does this generate?

- how do we pick the best one?

backward stepwise selection

- M_p -> M_{p-1} -> M_{p-2} -> … -> M_0

- how many models does this generate?

- how do we pick the best one?

error metrics

$$C_p = \frac{1}{n}(RSS + 2\underline{p}\hat{\sigma}^2)$$

Mallow's Cp
p is the total # of parameters
$\hat{\sigma}^2$ is an estimate of the variance of the error, ε

$$AIC = -2logL + 2 \cdot \underline{p}$$

L is the maximized value of the likelihood function for the model estimated

$$BIC = \frac{1}{n}(RSS + log(n)\underline{p}\hat{\sigma}^2)$$

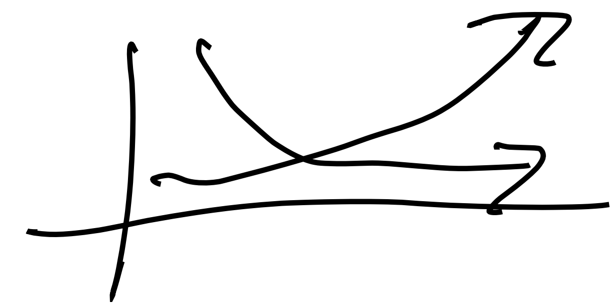
This is Cp, except 2 is replaced by log(n). log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adjusted\ R^2 = 1 - \frac{RSS/(n - \underline{p} - 1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

what you just learned



- figuring out if your model is working is hard
- cross validation is a tool for estimating how well your model does on unseen data
    - because of this you can use it to set hyperparameters (we will see our first of those this afternoon)
- bias-variance trade off is really important
    - similar to overfitting and underfitting, but instead of relating to a single dataset, is a feature of the modeling process used
    - you will see it all the time, remember what it means, it will make people think you know what you are talking about