Multicollinearity and model selection

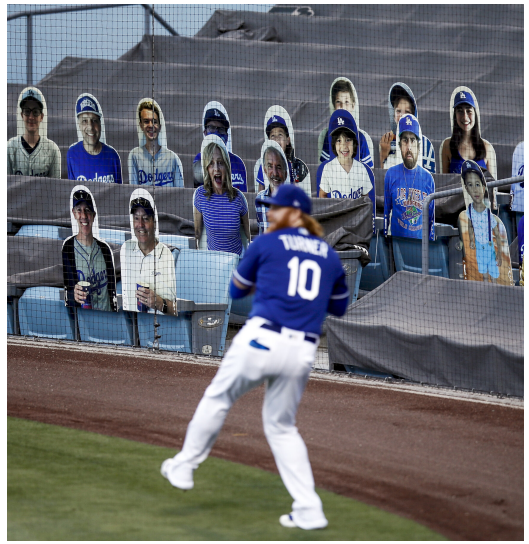
Overview

Multicollinearity

Model selection and overfitting

Review of baseball analysis and multicollinearity

Recall last class…



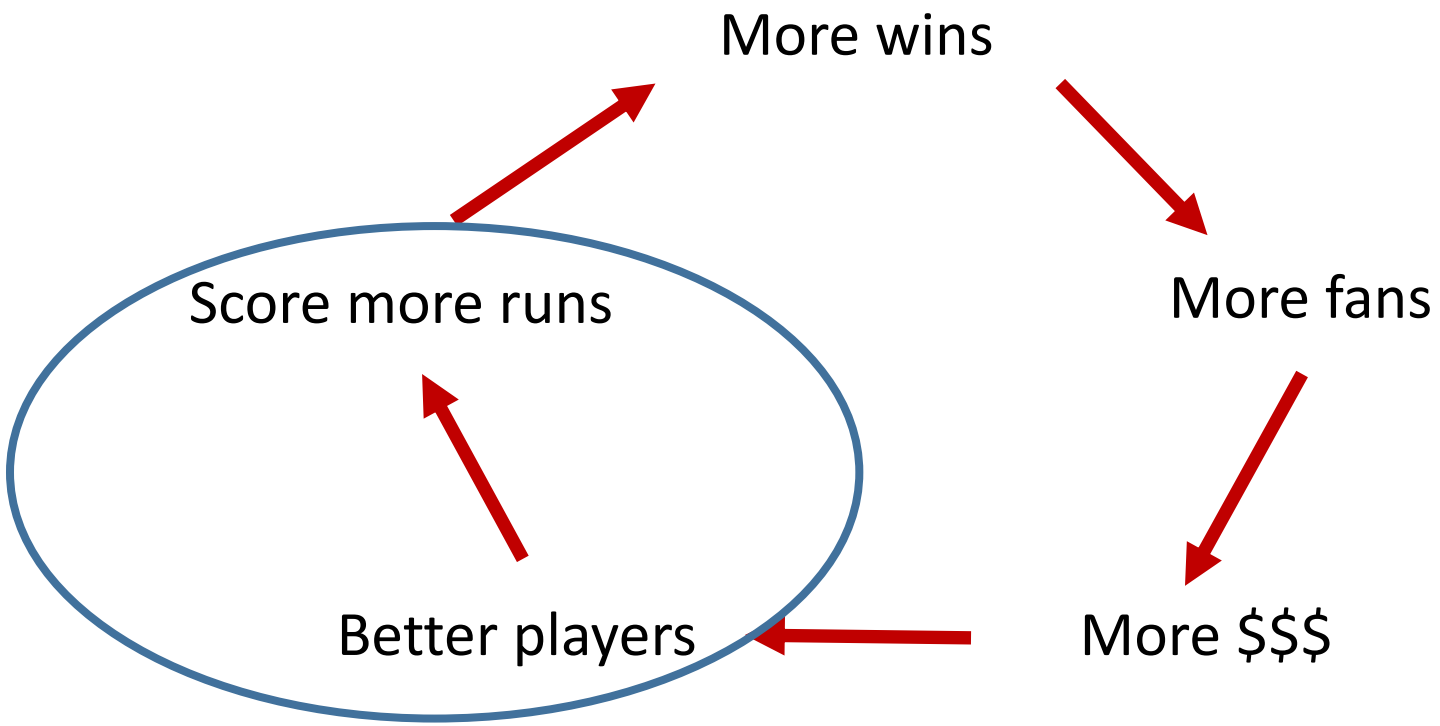
Motivation: Who is a better hitter- Derek Jeter or David Ortiz?

Derek Jeter

David Ortiz

If we are going to pay these players millions of dollars, how can we assess who is best?

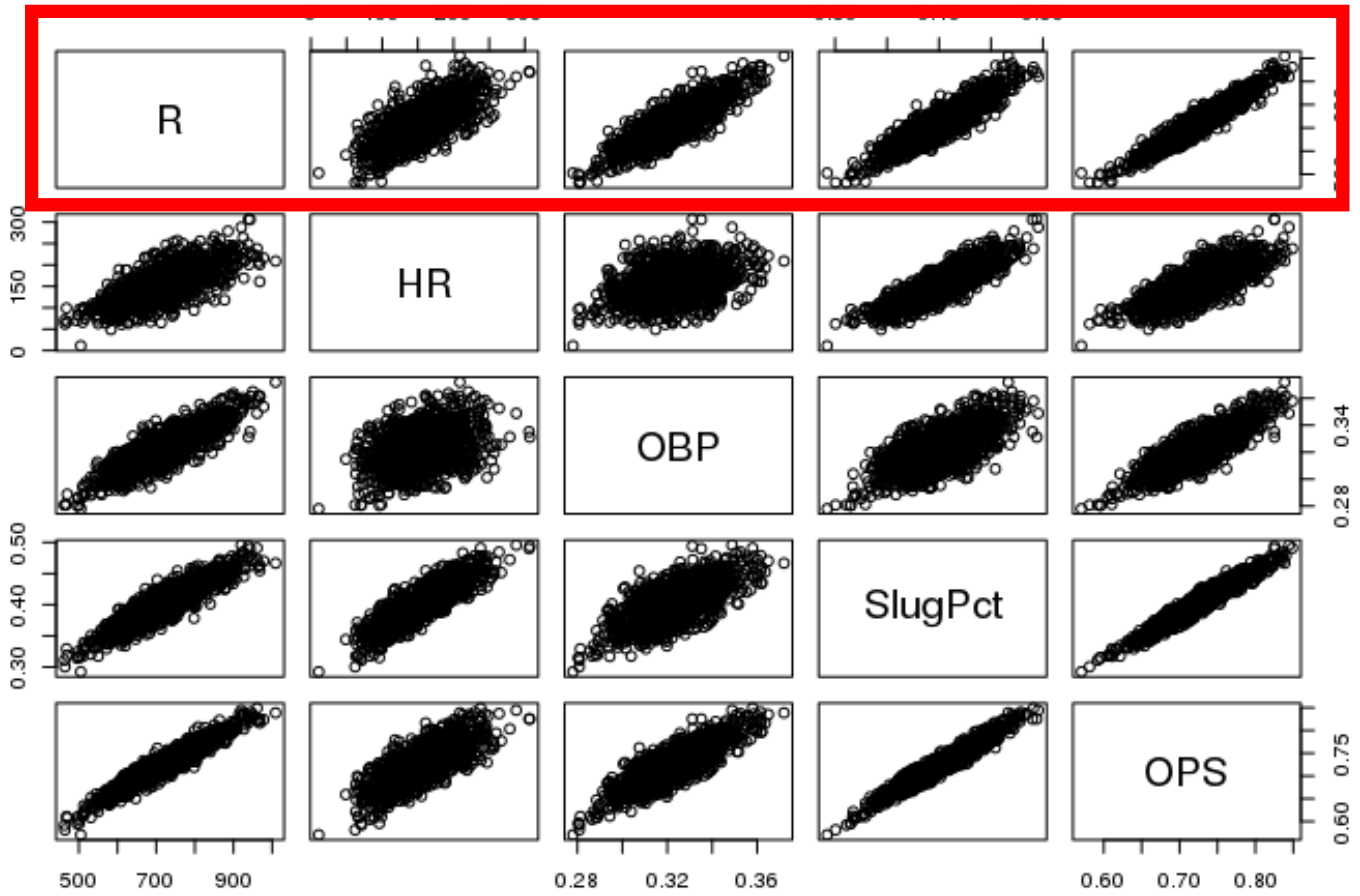
The great cycle of baseball



We can evaluate how ‘good’ a statistic is based on how well it correlates with the number of runs a team scores

Correlation between all variables

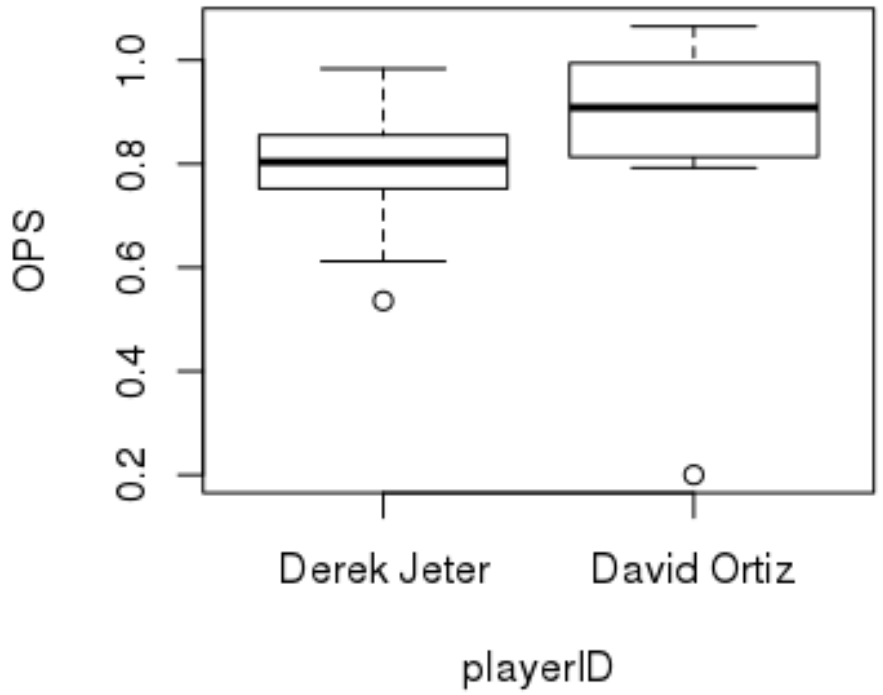
$R^2 = 0.50$ $R^2 = 0.73$ $R^2 = 0.82$ $R^2 = 0.91$



R: pairs()

What is the best statistic to use?

It seems like the winner is on-base plus slugging percentage!



Can we do better?

Fitting a multiple linear regression model to predict runs

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \ldots + \hat{\beta}_k \cdot x_k$$

	b_i
(Intercept)	-467
HBP	0.28
BB	0.36
X1B	0.54
X2B	0.68
X3B	1.35
HR	1.44

OPT model: $R^2 = 0.929$

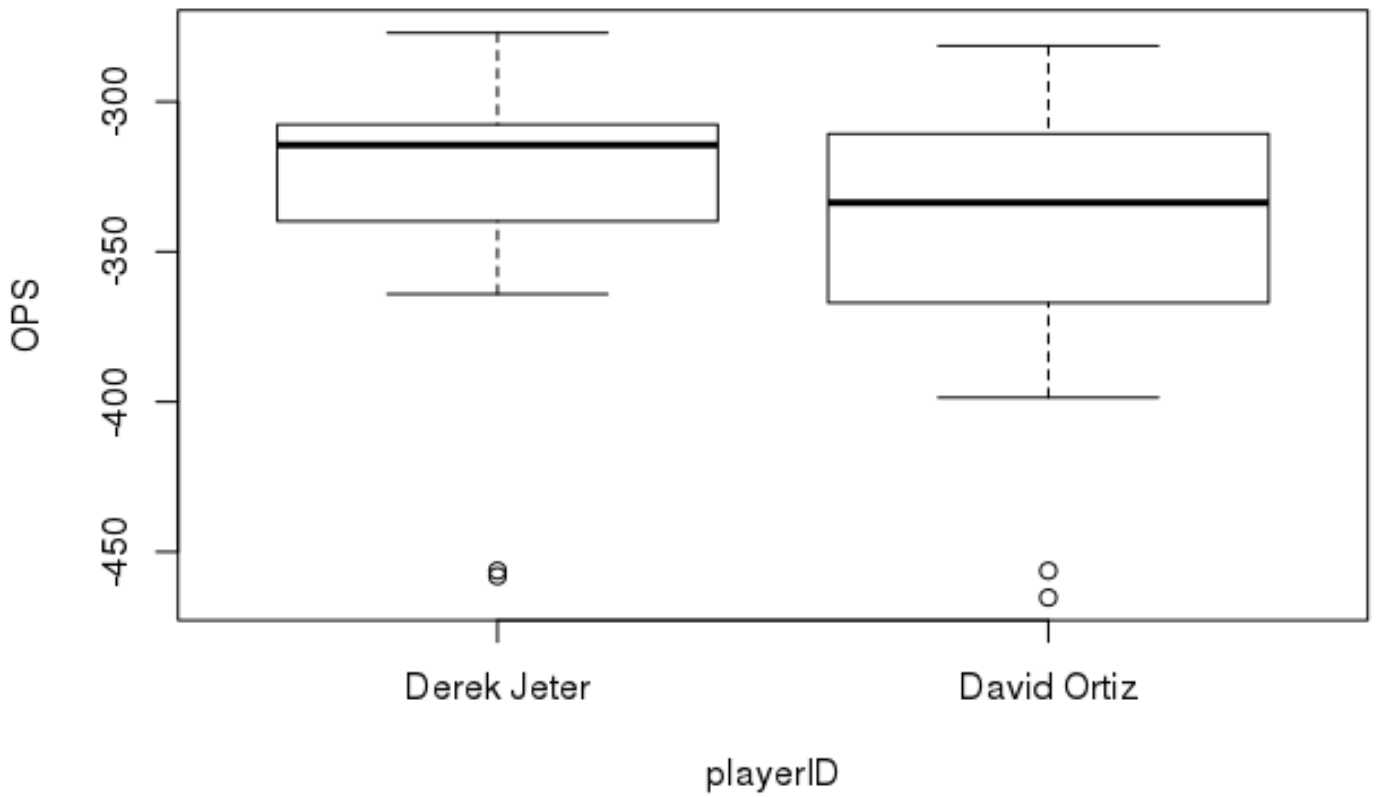
Previous best model

OPS: $R^2 = 0.921$

Our OPT statistic seems better!

$\hat{y}$ = *.36* · BB + *.28* · HBP + *.54* · 1B + *.68* · 2B + 1.35 · 3B + *1.44* · HR - 467

How do Derek Jeter or David Ortiz compare on our new statistic?



Can we do even better?

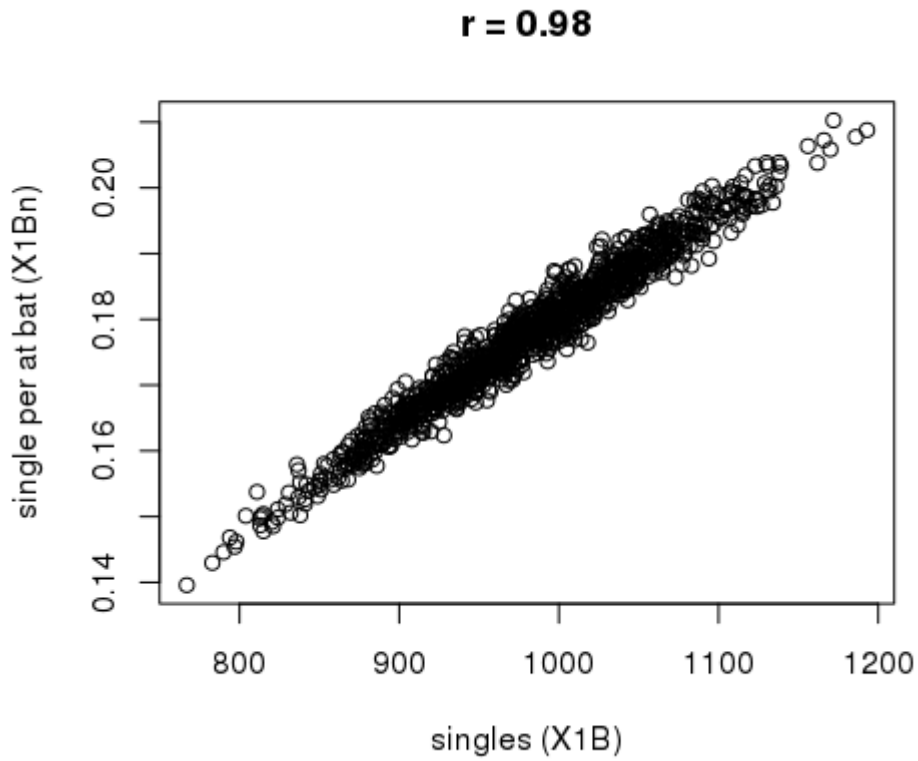
Can we add additional variables that will lead to a statistic with even higher R^2 ?

Let’s try it in R using the class 19 code!

Multicollinearity

Multicollinearity occurs when two or more variables are closely related to each other

- E.g., if they have a high correlation

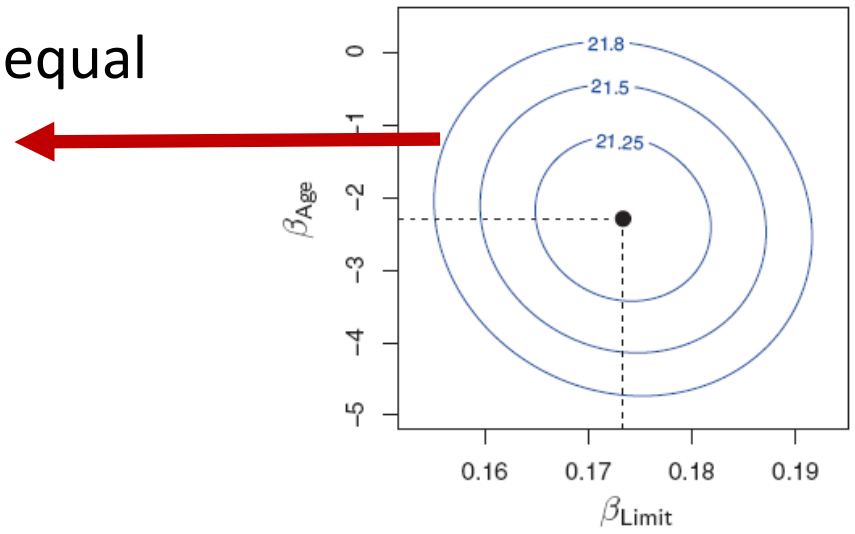


Multicollinearity

Multicollinearity can make our estimate of the regression coefficients unstable

- i.e., a large range of coefficient values give the same RSS and $\hat{\sigma}_\varepsilon$

Contours of equal $\hat{\sigma}_\varepsilon$ value



This increases our estimate of the variance of the coefficients we measure and hence can decrease the power to detect a statistically significant predictor

Multicollinearity

The **variance inflated factor** is a statistic that can be computed to test for multicollinearity

$$VIF_i = \frac{1}{1-R_i^2}$$

where R_i^2 is the coefficient of multiple determination for a model to predict x_i using the other predictors in the model

Rule of thumb: suspect multicollinearity for VIF > 5

car::vif(lm_fit)

Are any of the predictors x_i related to y?

We can set this up as a hypothesis test:

H_0: $\beta_1 = \beta_2 = \ldots = \beta_p = 0$

H_A: At least one $\beta_j \neq 0$

We can run a parametric hypothesis test based on an F statistic to test this hypothesis

Are any of the predictors x_i related to y?

```
(Intercept)         X1B         X2B         X3B          HR          BB          AB
106.4370957   0.6380842   0.8382352   1.3246222   1.5579544   0.3496647  -0.1267687

Call:
lm(formula = R ~ X1B + X2B + X3B + HR + BB + AB, data = team_batting)

Residuals:
    Min      1Q  Median      3Q     Max
-76.685 -15.609  -0.977  14.913  79.020

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 106.43710   73.32426   1.452    0.147
X1B           0.63808    0.01678  38.020  < 2e-16 ***
X2B           0.83824    0.02437  34.393  < 2e-16 ***
X3B           1.32462    0.07352  18.018  < 2e-16 ***
HR            1.55795    0.02812  55.406  < 2e-16 ***
BB            0.34966    0.01073  32.598  < 2e-16 ***
AB           -0.12677    0.01618  -7.835 1.09e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.45 on 1116 degrees of freedom
Multiple R-squared:  0.9294,    Adjusted R-squared:  0.929
F-statistic:  2449 on 6 and 1116 DF,  p-value: < 2.2e-16
```

summary(lm_fit)

```
Call:
lm(formula = R ~ X1B + X2B + X3B + HR + BB + X1Bn + X2Bn + X3Bn +
    XHRn + XBBn, data = team_batting2)

Residuals:
    Min      1Q  Median      3Q     Max
-78.695 -15.457  -0.798  15.480  76.092

Coefficients:
              Estimate  Std. Error t value            Pr(>|t|)
(Intercept) -574.88241    20.89696 -27.510 <0.0000000000000002 ***
X1B           -0.08976     0.43995  -0.204               0.838
X2B            1.70203     1.36050   1.251               0.211
X3B           -0.20163     4.71591  -0.043               0.966
HR             1.19258     1.47183   0.810               0.418
BB             0.24157     0.65658   0.368               0.713
X1Bn        3930.66847  2443.75215   1.608               0.108
X2Bn       -4839.59898  7517.51009  -0.644               0.520
X3Bn        8493.67060 26119.44048   0.325               0.745
XHRn        2061.44301  8146.72963   0.253               0.800
XBBn         588.32226  3628.53349   0.162               0.871
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.61 on 1140 degrees of freedom
Multiple R-squared:  0.9297,    Adjusted R-squared:  0.929
F-statistic:  1507 on 10 and 1140 DF,  p-value: < 0.00000000000000022
```

None of the coefficients are significant at the $\alpha = 0.05$ level

Overall H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$ is highly significant

This can happen when there is multicolinearity

Model selection

Model selection

Model selection is the process of selecting a statistical model from a set of candidate models

Model selection depends on our goal, which often are:

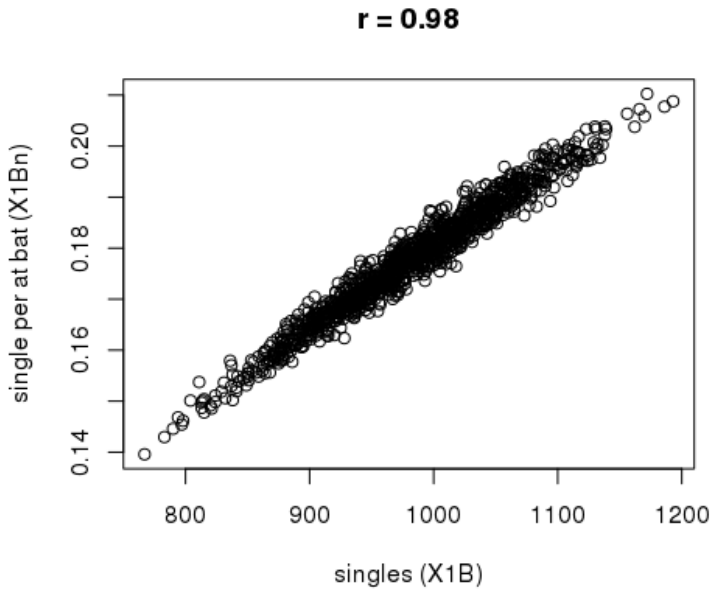
- Making accurate predictions
- Understanding relationships in our data

“All models are wrong but some are useful”

- Model selection is a little bit of an art
  - And there is definitely some bad art out there

Baseball data example

```
team_batting2 <-
  mutate(team_batting,
         X1Bn = X1B/AB,
         X2Bn = X2B/AB,
         X3Bn = X3B/AB,
         XHRn = HR/AB,
         XBBn = BB/AB)
```



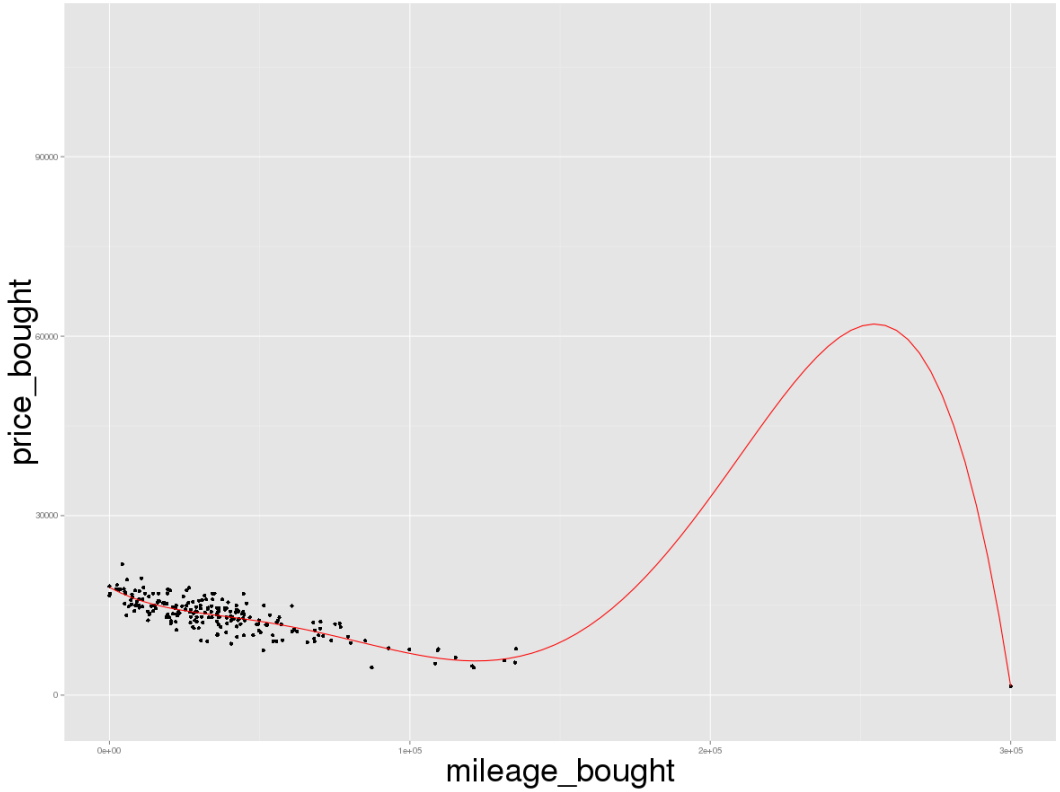

```
fit1 <- lm(R ~ X1B + X2B + X3B + HR + BB + AB, data = team_batting2)

fit2 <- lm(R ~ X1B + X2B + X3B + HR + BB +
             X1Bn + X2Bn + X3Bn + XHRn + XBBn, data = team_batting2)
```

Used Toyota Corollas example

On homework 8 you fit 1, 3, and 5 degree polynomials to the Edmunds data

Is the 5th degree polynomial fit a "good model" for predicting car price from miles drive?



Degree	1	2	3	4	5
R^2	0.596197	0.642271	0.6425	0.643618	0.654087

Fitting complex models

Last year I had students use the Edmunds data to create a multiple regression models that could explain as much of the variability in a car's sale price as possible.

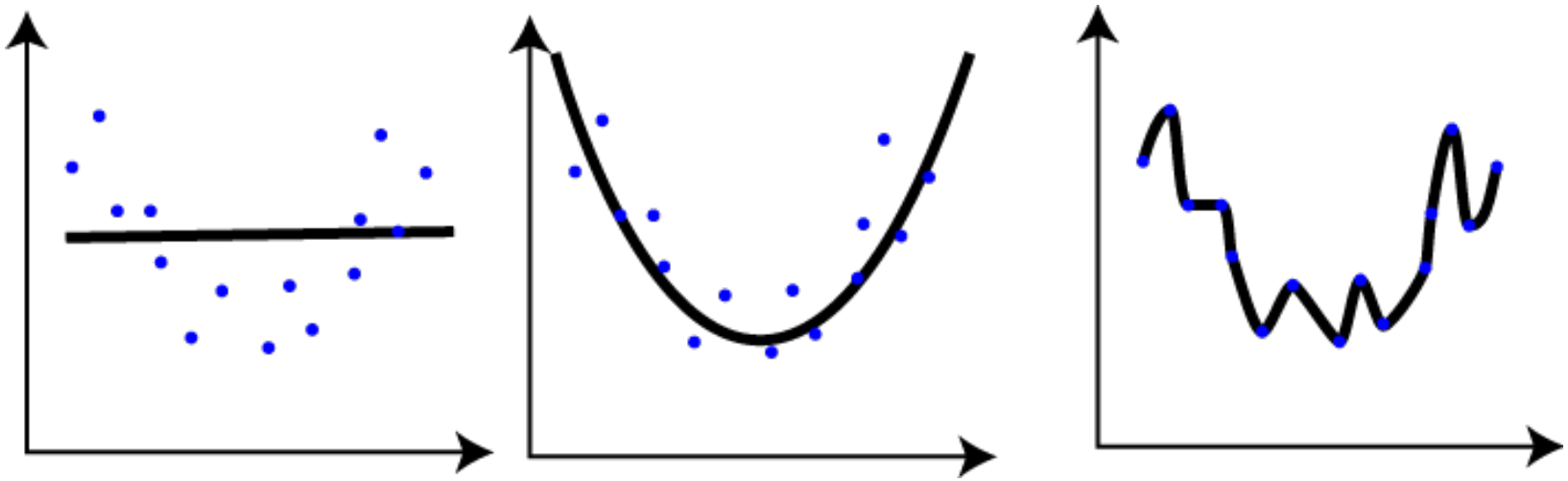
One student came up with a model had $R^2 = 0.9434$

$$\hat{y} = 24919.66 + 101976420913.07MS - 8801176308.15MS^2 + 1241642151.11MS^3 + -294845652.56MS^4 + 81081110.14MS^5 - 19486206.21MS^6 + 4627457.40MS^7 - 962171.87MS^8 + 159411.19MS^9 + 4758.12MS^10 - 215919.33YO + 61629.11YO^2 - 46883.91YO^3 + 27616.00YO^4 + -41.24YO^5 + 986604.79MB - 35074977.23MB^2 - 65070922.16MB^3 - 80975553.98MB^4 - 39704725.17MB^5 - 13029640.26MB^6 - 4181962.33MB^7 - 1234301.18MB^8 - 444859.17MB^9 - 130775.72MB^10 + 10404.48MPY - 3884.70MPY^2 + 2328.80MPY^3 + 13563.89MPY^4 - 5277.89MPY^5 - 15338.18MPY^6 + 12851.91MPY^7 + 4899.51MPY^8 + 383.03MPY^9 - 68.52MPY^10 - 82637866000.10(log(MS)) - 41355241329.94(log(MS))^2 - 32425545929.48(log(MS))^3 - 24481236288.29(log(MS))^4 - 14974293674.01(log(MS))^5 - 7624035557.33(log(MS))^6 - 2264871890.73log(MS)^7 - 411050984.04log(MS)^8 - 54557306.69log(MS)^9 - 2606070.61log(MS)^10 + 172377.75(YO * MS) - 85365.06(YO * MS)^2 + 20333.05(YO * MS)^3 + 38388.69(YO * MS)^4 - 11361.46(YO * MS)^5 + 37428.48(YO * MS)^6 + 5120.32(YO * MS)^7 + 15285.89(YO * MS)^8 + 9300.88(YO * MS)^9 + 11552.40(YO * MS)^10 + 35723937.44w^2 + 66215674.21w^3 + 80347987.00w^4 + 38445702.35w^5 + 12868623.19w^6 + 4105787.54w^7 + 1161535.74w^8 + 426310.43w^9 + 133553.84w^10$$

Do you think this model would do well making predictions on new data?

Overfitting

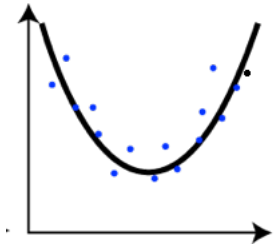
Overfitting occurs when we generate a function that too closely matches random sample we have, but does not generalize to the full probability distribution

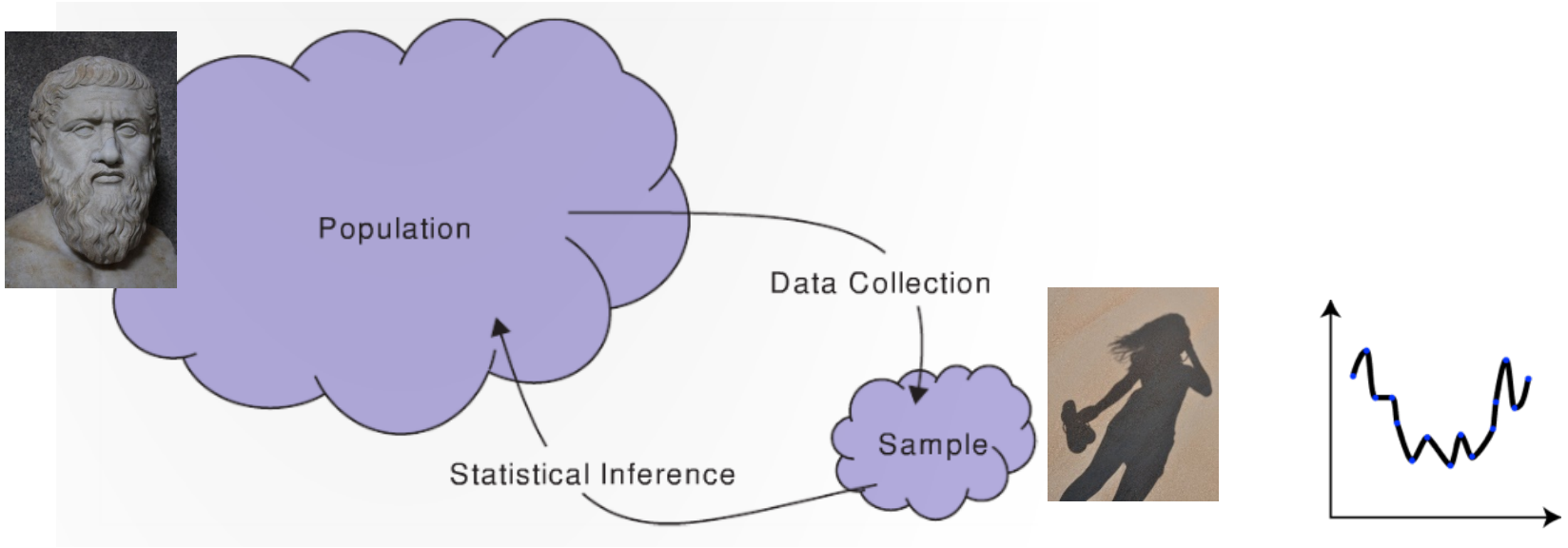


Overfitting

Overfitting occurs when we generate a function that too closely matches random sample we have, but does not generalize to the full probability distribution

- The model is fit to closely to the shadows and not getting at the Truth





Overfitting song

https://www.youtube.com/watch?v=DQWI1kvmwRg

Selecting models methods

There are a number of different methods for selecting models. Three we will briefly discuss are:

1. Creating measures of fit (statistics) that penalize models with more predictors

2. Creating simpler models by removing predictors

3. Evaluating models using cross-validation

Model method selection 1: Selected models using statistics that penalize larger models

R^2 as a measure of model fit

We have used the coefficient of multiple determination (R^2) to determine how well our model is fitting the data:

$$R^2 = \frac{SSModel}{SSTotal} = 1 - \frac{RSS}{SSTotal} = 1 - \frac{\Sigma_{i=1}^{n}(y_i - \hat{y})^2}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2}$$

R^2 always increases with more predictors x_i because $\hat{y}$ can always fit better with more predictors

R^2 > R^2

Recall: the standard deviation of the errors: σ_ε

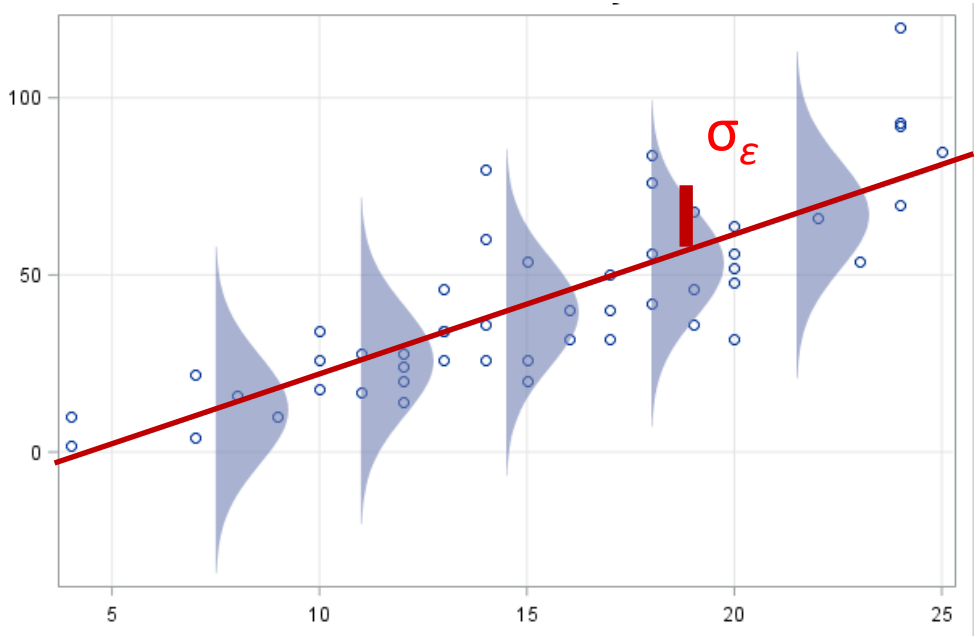
Recall the standard deviation of the errors is denoted $\boldsymbol{\sigma_\varepsilon}$ is the standard deviation of how the points fall off the true regression line.

We can use the **standard deviation of residuals** as an estimate for σ_ε
This is known as the **regression standard error ($\hat{\sigma}_\varepsilon$)**

For simple linear regression we had:

$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{n-2}RSS}$$

$$= \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

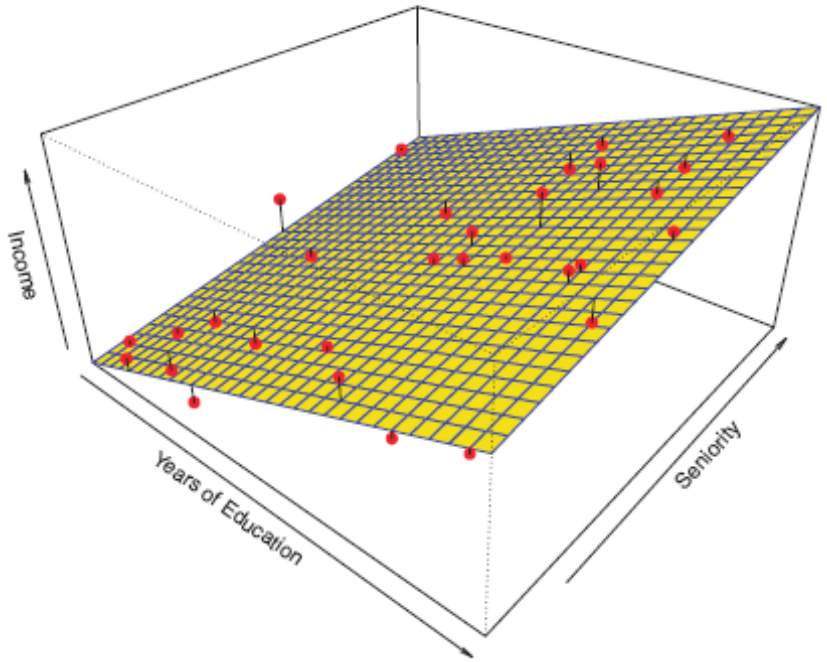


Recall: the standard deviation of the errors: σ_ε

For multiple regression, we with k parameters (i.e., k - 1 predictors) an (almost) unbiased estimate of $\hat{\sigma}_\varepsilon$ is:

$$\hat{\sigma}_\epsilon = \sqrt{\frac{1}{n-k}RSS}$$

$$= \sqrt{\frac{1}{n-k}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$



The residual standard error $\hat{\sigma}_\varepsilon$ is corrects for bias by dividing the RSS by 1/(n - k)

This estimate does not always decrease with more predictors x

Adjusted R^2

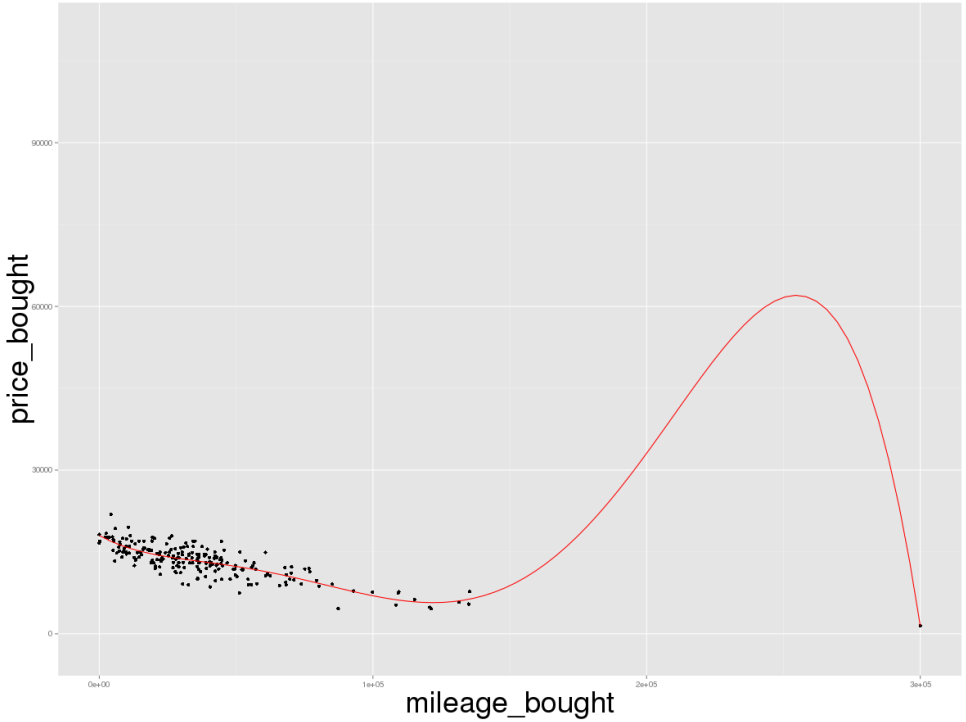
The **adjusted R^2** helps account for the number of predictors in the model by using $\hat{\sigma}_\varepsilon^2$

$$R^2_{adj} = 1 - \frac{\frac{RSS}{n-k}}{\frac{SST}{n-1}} = 1 - \frac{\hat{\sigma}^2_\epsilon}{s^2_y}$$

Thus the adjusted R^2 will not always give a higher values to the model with the most predictors

- i.e., using this statistic we will not always say that a model with the most predictors is a “better” fit to the data

Adjusted R^2 on used Toyota Corollas



According to the adjusted R^2 statistic, the degree 5 model is “best”, which doesn’t seem right

Degree	1	2	3	4	5
R^2	0.596	0.642	0.642	0.644	0.654

Other statistics that penalize larger models

There are several other statistics that also ***penalize models that have more predictors***

- These statistics are only meaningful for within data set comparisons

Akaike information criterion: $AIC = 2k + n\ln(RSS)$ R: AIC(lm_fit)

Bayesian information criterion: $BIC = k\ln(n) + n\ln(RSS/n)$ R: BIC(lm_fit)

One should select the model with the <u>lowest value</u> on these statistics

Model method selection 2: Using algorithms to select a subset of variables

Brief mention: Variable selection

Variable selection refers to finding models that rely on a small subset of predictors

- This can help make the regression model more interpretable as well

We could use individual feature p-values to determine which predictors to use, however…

- Some of these will be spuriously significant
  - i.e., if H_0 is true for all predictors, ~5 will be significant at $\alpha = .05$ level
- The p-values change as predictors are added and removed
  - Due to multicollinearity

```
team_batting2 <-
    mutate(team_batting,
           X1Bn = X1B/AB,
           X2Bn = X2B/AB,
           X3Bn = X3B/AB,
           XHRn = HR/AB,
           XBBn = BB/AB)
```

Feature selection: deciding which variables to use

Ideally we would like to try all combinations of predictors, however, if there are k features, there are 2^k possible models which can be intractable

A few heuristic methods exist for selecting smaller models

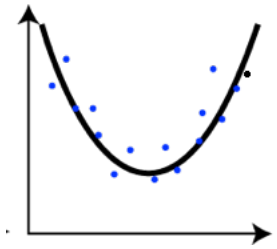
- Forward selection: start with a model with no predictors and add predictors (until you have enough)
- Backward selection: Start with the full model and delete predictors
- Mixed selection: Use a combination of forward and backward selection

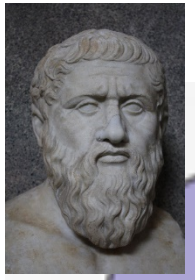
```
Coefficients:
       Estimate Std. Error t value Pr(>|t|)
X1B   1.355e+00  5.700e-01   2.378   0.0176 *
X2B   3.170e-01  1.786e+00   0.177   0.8592
X3B   4.598e+00  6.128e+00   0.750   0.4532
HR    4.143e+00  2.041e+00   2.030   0.0426 *
BB   -2.925e-01  8.564e-01  -0.342   0.7328
X1Bn -6.359e+03  3.150e+03  -2.019   0.0437 *
X2Bn  9.474e+02  9.871e+03   0.096   0.9236
X3Bn -2.001e+04  3.393e+04  -0.590   0.5555
XHRn -1.683e+04  1.128e+04  -1.492   0.1359
XBBn  3.600e+03  4.732e+03   0.761   0.4469
```

In R: leaps::regsubsets()

Model method selection 3: Choosing a model through cross-validation

Cross-validation

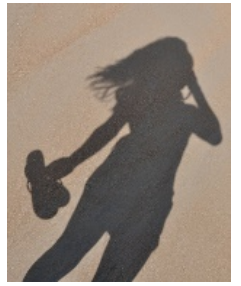


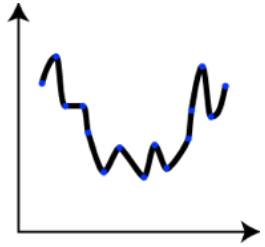


Population

Training set

Sample





Test set

Sample

Good predictions on the test set give an estimate of how accurate the model will be on new data from the population

Cross-validation

We run cross-validation by splitting data into two sets:

A training set in which the parameters of classification/regression model are fit

A test set in which the prediction accuracy of our model is assessed

Mean squared prediction error

To evaluate how effective a model is, we can use the mean squared prediction error (MSPE) using the following steps:

1. Fit a model using the training data
2. Make predictions on the test data
3. We can use the MSPE to assess how accurate the predictions are:

$$MSPE = \frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \hat{y}_i)^2$$

Actual y values in the test set

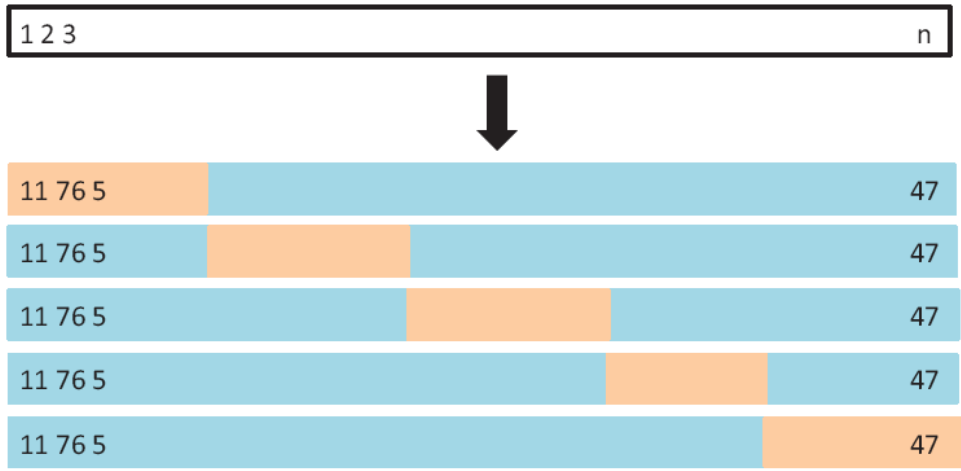
Predicted y values in the test set

n_t is the number of points in the test set

K-fold cross-validation

K-fold cross-validation

- Split the data into k parts
- Train on k-1 of these parts and test on the left out part
- Repeat this process for all k parts
- Average the prediction accuracies to get a final estimate of the generalization error



Leave-one-out (LOO) cross-validation: k = n

Let’s try it in R…