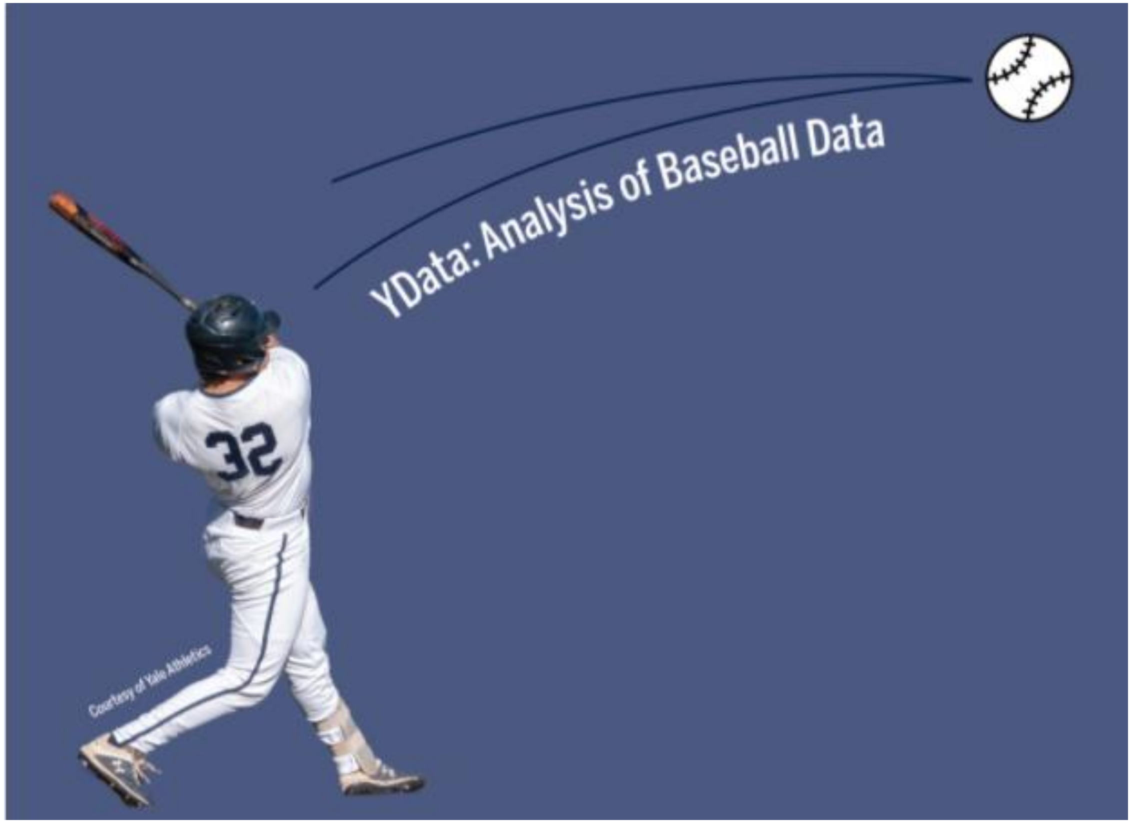
Additional topics in regression



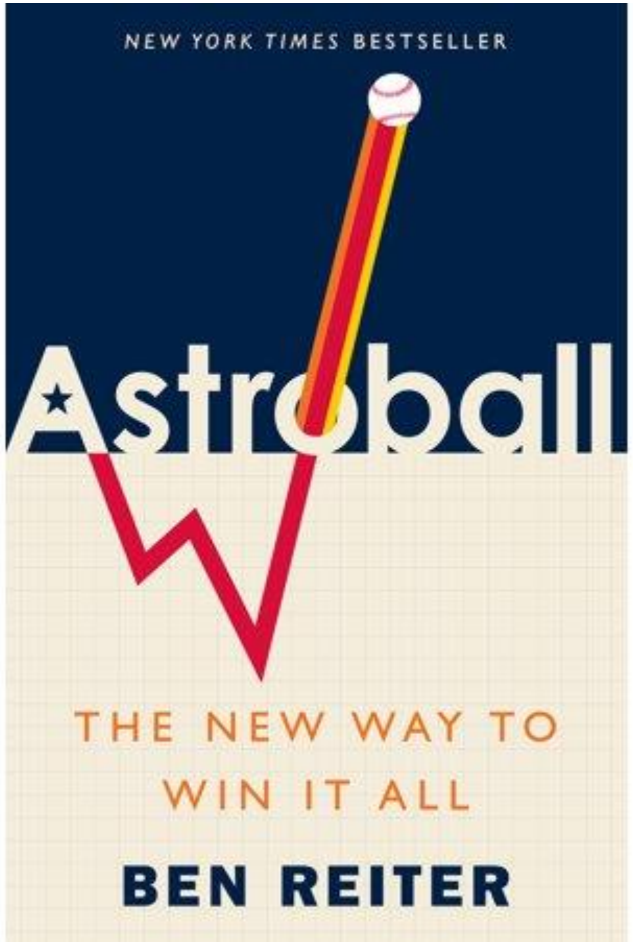
Overview

Discussion with Ben Reiter

Quick review and continuation of linear regression

- Regression to the mean, and the Sports Illustrated cover jinx
- Polynomial regression
- Overfitting
- Bill James' Pythagorean Expectation

Ben Reiter


NEW YORK TIMES BESTSELLER
Astroball
THE NEW WAY TO
WIN IT ALL
BEN REITER

Announcement: Final projects

Final project presentation will be live during next class

I would prefer prerecorded videos		0 %
I would prefer live presentations	7 respondents	78 %
I do not have a preference	2 respondents	22 %

~5 minute presentation with 2 minute Q&A

A final written reports are due at 11:30pm on May 13th (last day of reading period)

- Report should be 7-10 pages long

The MLB season is in week 3

Is this just a fluke that the Red Sox are still in first place or does this indicate that they might actually be good?

My final class project!

American League	National League

AL East

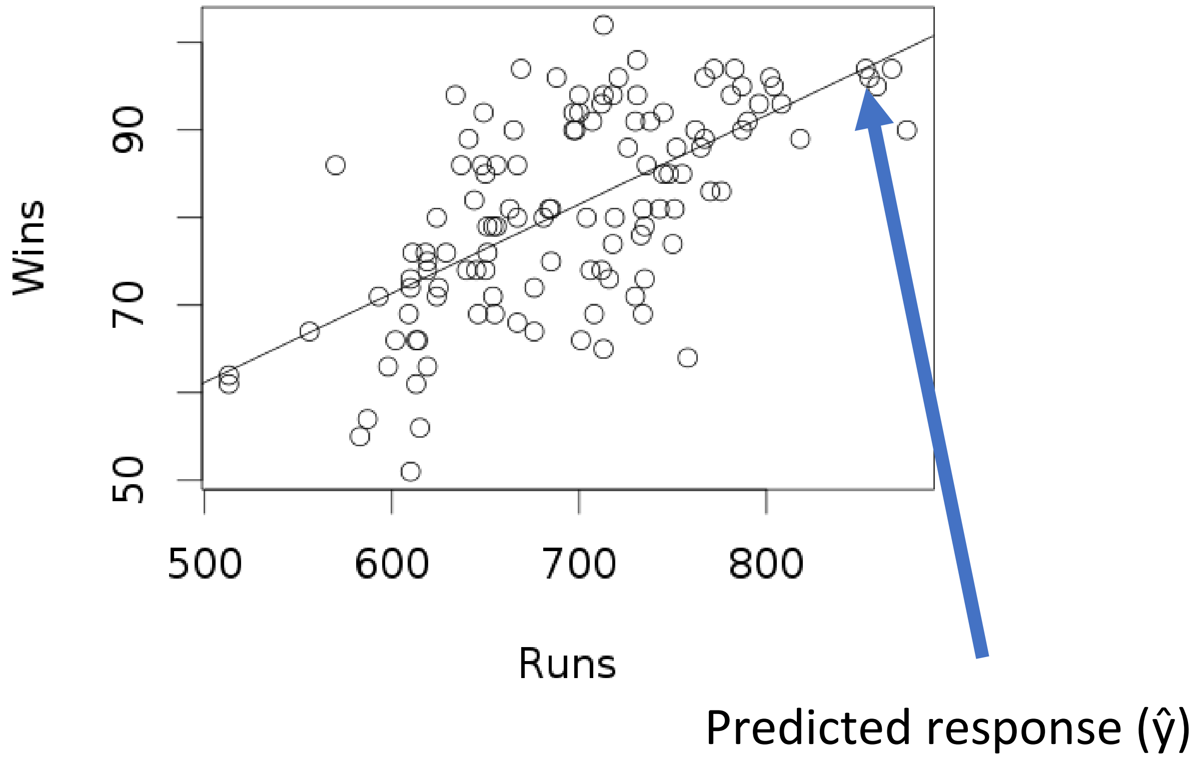
Team	W	L	Pct	GB	Home	Away	L10
Red Sox	15	9	.625	-	8-8	7-1	5-5
Blue Jays	11	11	.500	3.0	4-3	7-8	5-5
Rays	12	12	.500	3.0	5-7	7-5	6-4
Orioles	10	13	.435	4.5	3-9	7-4	5-5
Yankees	10	13	.435	4.5	4-7	6-6	5-5

Quick review and continuation of regression

Regression

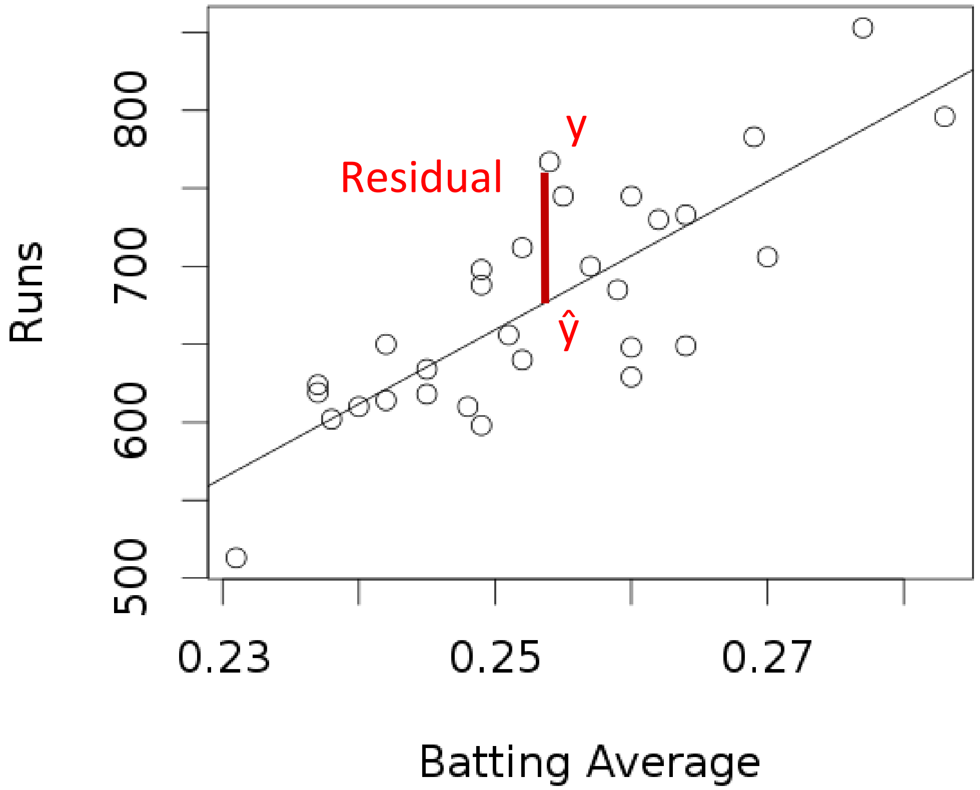
Regression is method of using one variable to predict the value of a second variable

In **linear regression** we fit a line to the data, called the **regression line**



$$\hat{y} = a + b \cdot x$$

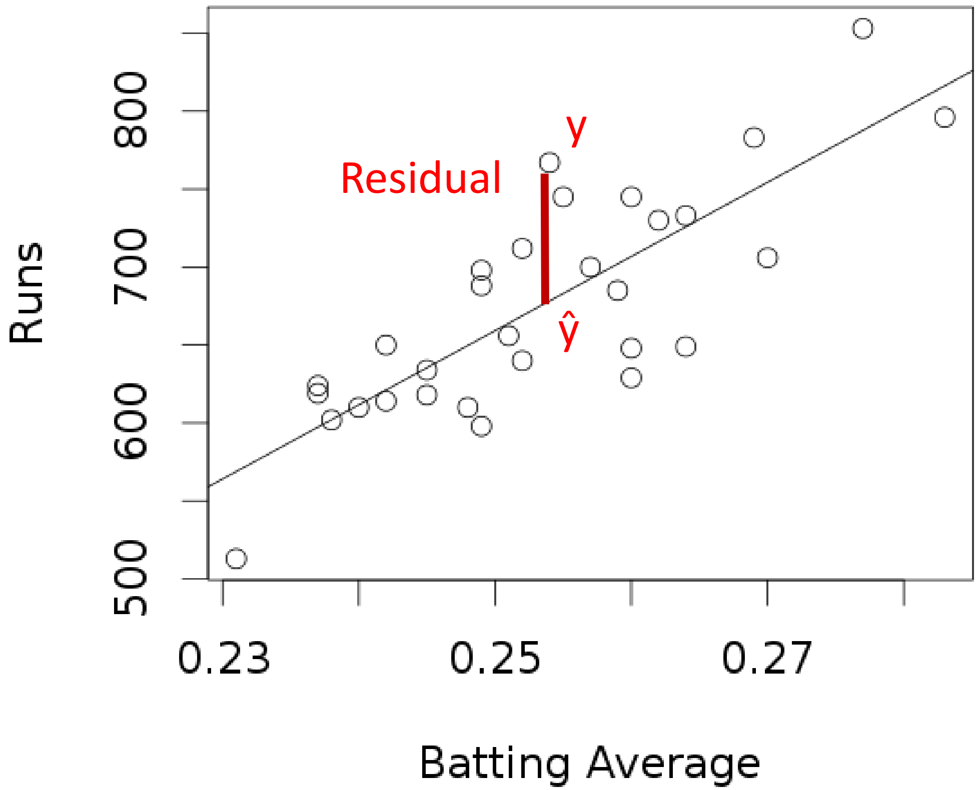
Residuals



The **residual** at a data value is the difference between the observed (y) and predicted value ($\hat{y}$) of the response variable

Residual = Observed – Predicted

Measuring goodness of fit



$r^2 = 1 - \text{MSE/var}(y) \quad \cdot \quad [(n-1)/n]$

We can measure how well the line fits the data using the equation:

$$MSE = \frac{1}{n}\sum_i^n (y_i - \hat{y}_i)^2$$

Least squares line

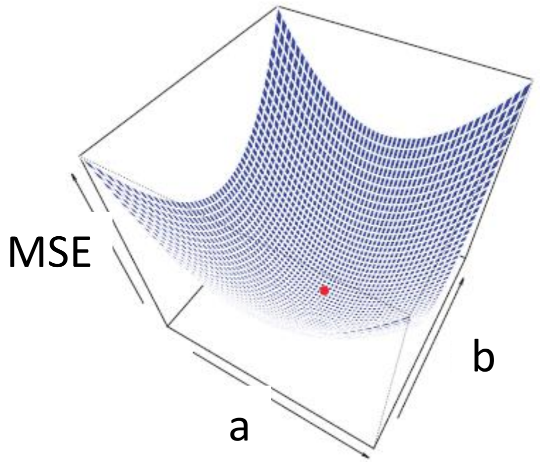
The **least squares line**, also called "**the line of best fit"**, is the line which minimizes the sum of squared residuals

- i.e., the least squares line are the coefficients *a*, and *b* that minimize the Mean Squared Error (MSE)

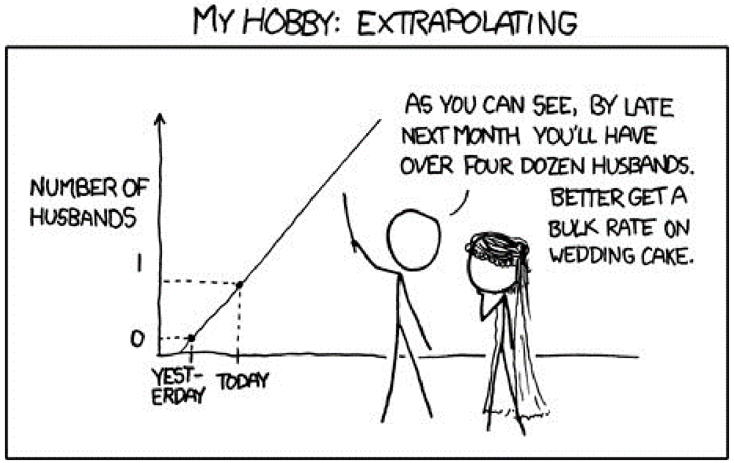
$$MSE \ = \ \tfrac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2 \ =$$

The regression coefficients can be found using calculus:

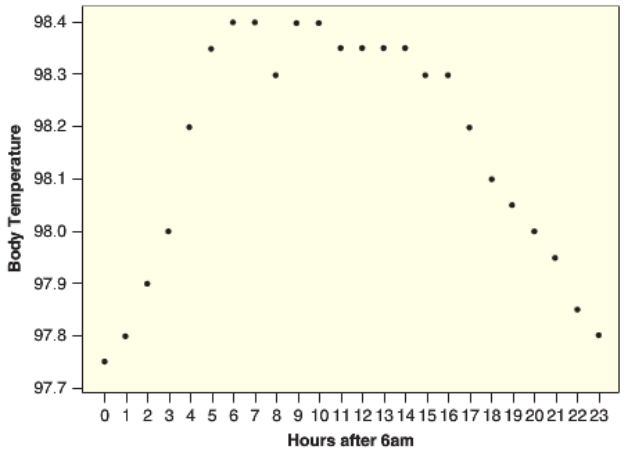
- This can be done by setting the partial derivative of the MSE with respect for a and b to 0 and solving for a and b



Regression cautions

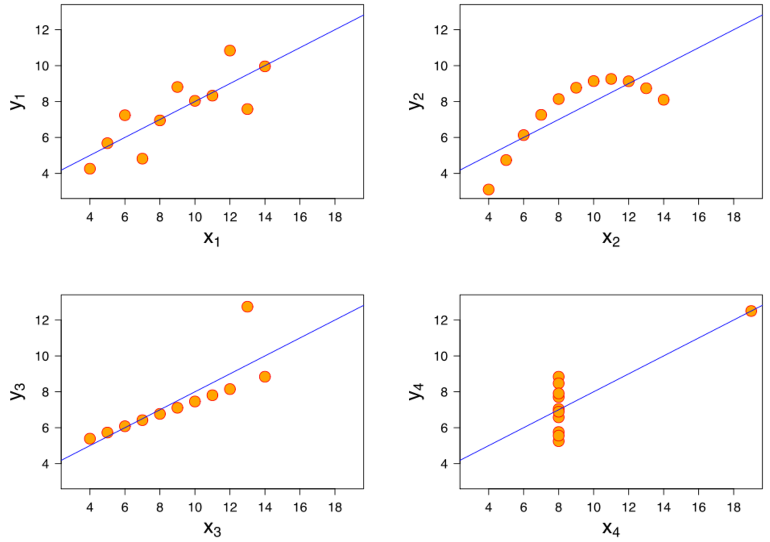


Do not extrapolate too far



Plot the data! Regression lines are only appropriate when there is a linear trend in the data

Be aware of outliers – they can have an huge effect on the regression line

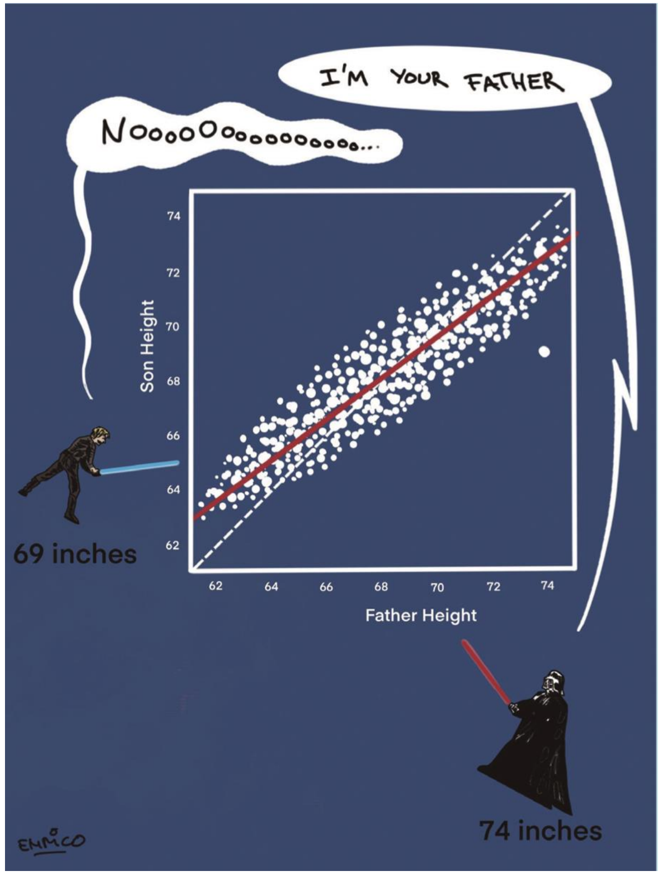


Linear regression in Python

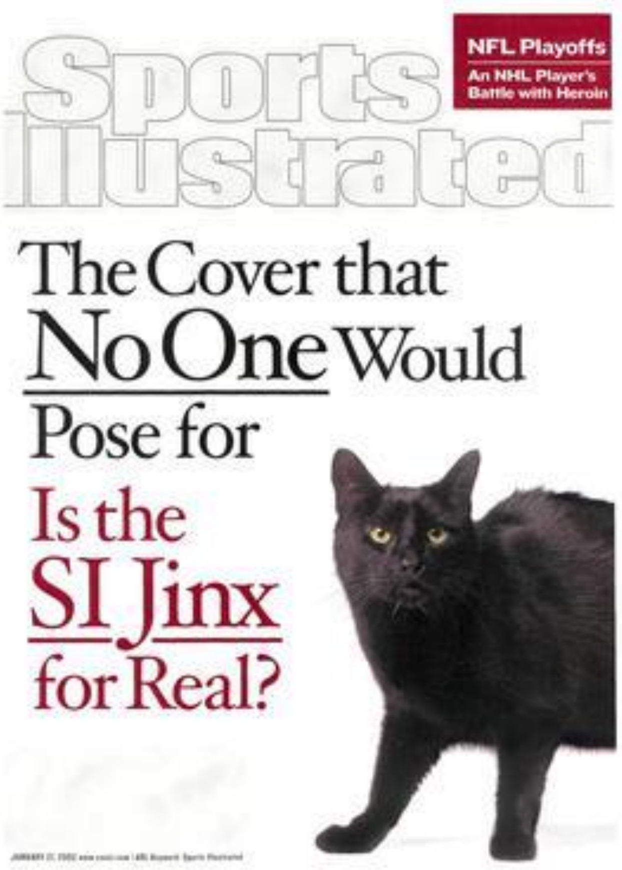
```
import statsmodels.formula.api as smf

tb.scatter('x', 'y', fit_line = True)

lm = smf.ols('y ~ x', data = my_df).fit()

params = lm.params

sm_predictions = lm.predict(the_data)
```

```
Intercept    -526.921684
BA           4744.561329
dtype: float64
```

Regression to the mean



Original data from Galton, 1886



- Sports Illustrated Cover Jinx
- Rookie of the year curse

Regression to the mean

Does anyone know what is causing this phenomenon?

Lab 10 you will briefly explore this in Python

Multiple regression

In multiple regression we try to predict a quantitative response variable *y* using several predictor variables $x_1, x_2, \ldots, x_k$

For multiple linear regression our equation as the form of:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \ldots \beta_k \cdot x_k + \epsilon$$

We estimate coefficients using a data set to make predictions ŷ

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

What are the optimal weights?

$$\textbf{OPT} = \boldsymbol{b_1} \cdot \textbf{BB} + \boldsymbol{b_2} \cdot \textbf{HBP} + \boldsymbol{b_3} \cdot \textbf{1B} + \boldsymbol{b_4} \cdot \textbf{2B} + \boldsymbol{b_5} \cdot \textbf{3B} + \boldsymbol{b_6} \cdot \textbf{HR} + \boldsymbol{b_0}$$

Let's use multiple regression to find the b_i's that minimize sum of $(R - OPT)^2$

```
lm = smf.ols('R ~ BB + HBP + H + X2B + X3B + HR', data = teams_2013).fit()

the_params = lm.params
```

What are the optimal weights?

	b_i
(Intercept)	-497.44
HBP	0.42
BB	0.34
X1B	0.56
X2B	0.75
X3B	1.40
HR	1.44

lm.params

Do these coefficients make sense?

$$\hat{r} = .34 \cdot \text{BB} + .42 \cdot \text{HBP} + .56 \cdot \text{1B} + .75 \cdot \text{2B} + 1.40 \cdot \text{3B} + 1.44 \cdot \text{HR} - 497.44$$

How low can you go?

On lab 9 problem 3.3 you added additional variables in the team_batting to get the lower RMSE

- Whoever can come up with the lowest RMSE value wins bragging rights

The winner is… Raphael!

$$\hat{R} = -264 + 3.14W + 0.83H - 0.01X2B + 0.39X3B + 0.69HR + 0.50BB + 0.078SB + 0.19CS + 0.48HB \\ + 251.17ERA - 0.24CG - 0.58SHO - 0.88SV + 0.10HRA - 0.016SOA - 0.24X1B + 691BA + 445SLG$$

RMSE: 18.31

Do we believe Raphael's model is the best?

Non-linear relationships

You can get even lower RMSEs by including non-linear terms

- E.g., $1B^2$, HR^5 etc.

Polynomial regression extends linear regression to non-linear relationships by including nonlinear transformations of predictors

$$BA = \beta_0 + \beta_1 \cdot year + \beta_2 \cdot (year)^2 + + \beta_3 \cdot (year)^3 + \varepsilon$$

Still a linear equation but non-linear in original predictors

Non-linear relationships

We can add non-linear predictors by simply adding new columns to our table that are non-linear functions of the original columns

```
tb = tb.with_column('x2', tb['x']**2)

lm = smf.ols('y ~ x + x2’, tb).fit()
```

You will also try this on lab 10

Overfitting

Do these optimal weights yield the best model?

As we just discussed, we can use least squares to find the optimal weights:

$$\mathbf{OPT} = \boldsymbol{w_1} \cdot \mathbf{BB} + \boldsymbol{w_2} \cdot \mathbf{HBP} + \boldsymbol{w_3} \cdot \mathbf{1B} + \boldsymbol{w_4} \cdot \mathbf{2B} + \boldsymbol{w_5} \cdot \mathbf{3B} + \boldsymbol{w_6} \cdot \mathbf{HR} + \boldsymbol{w_0}$$

	w_i
(Intercept)	-478.22
HBP	0.52
BB	0.28
X1B	0.52
X2B	0.96
X3B	0.84
HR	1.38

$$\hat{r} = .28 \cdot \text{BB} + .52 \cdot \text{HBP} + .52 \cdot \text{1B} + .96 \cdot \text{2B} + .84 \cdot \text{3B} + 1.38 \cdot \text{HR} - 478.22$$

How good is our new optimal statistic based as measured through RMSE (R^2)?

Is our RSMSE using least squares better than using OPS to predict runs?

OPT* also includes PA
(this is included in OPS too)

Question: do we really believe that OPT* is better at predicting runs than OPS?

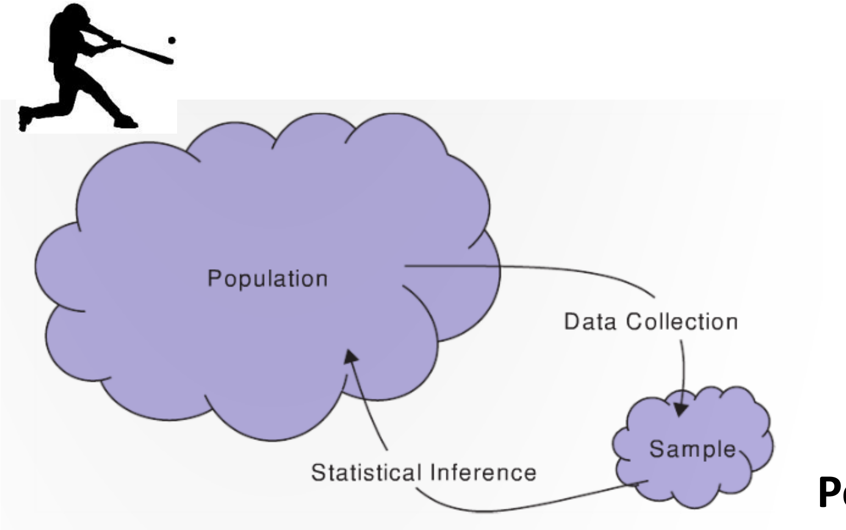
	RMSE
HR	60.42
BA	42.17
OBP	31.83
SlugPct	31.55
OPS	23.46
OPT	24.53
OPT*	21.62

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 observed performance and not getting at the players' ability

Ability



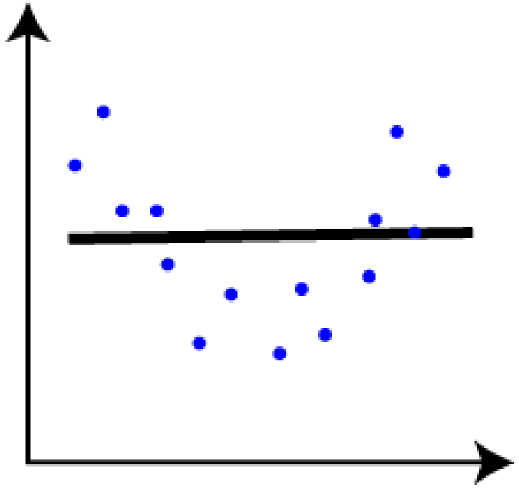
Performance: Hit, Out, Hit, Out, …

Fitting on the 2012 season, measuring the fit on the 2013 season

	RMSE
HR	62.85
BA	49.29
OBP	38.40
Slug	34.50
OPS	26.61
OPT*	30.39

“Optimal” fit no longer that optimal

Overfitting



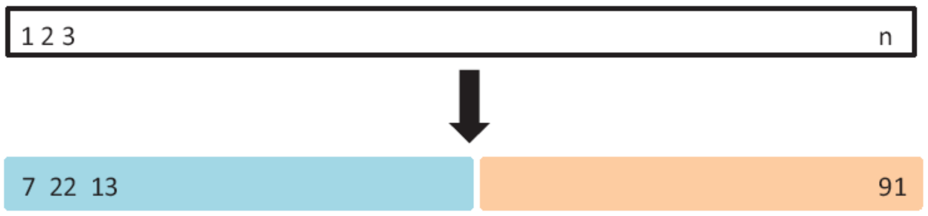
Cross-validation

To realistically assess how well our classifier can make accurate predictions on new data (i.e. to estimate the generalization error) we use cross-validation

Cross-validation consists of splitting your 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



Cross-validation

Training error rate: model predictions are made on using the same data that the model was fit with

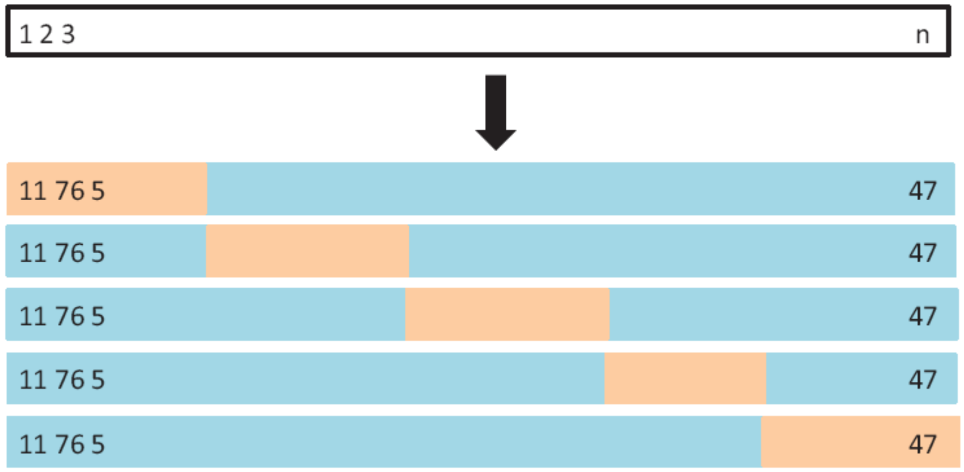
Test error rate: model predictions are made on a separate set of data

The test error rate is an estimate of how accurate your predictions will be on new (future) data

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

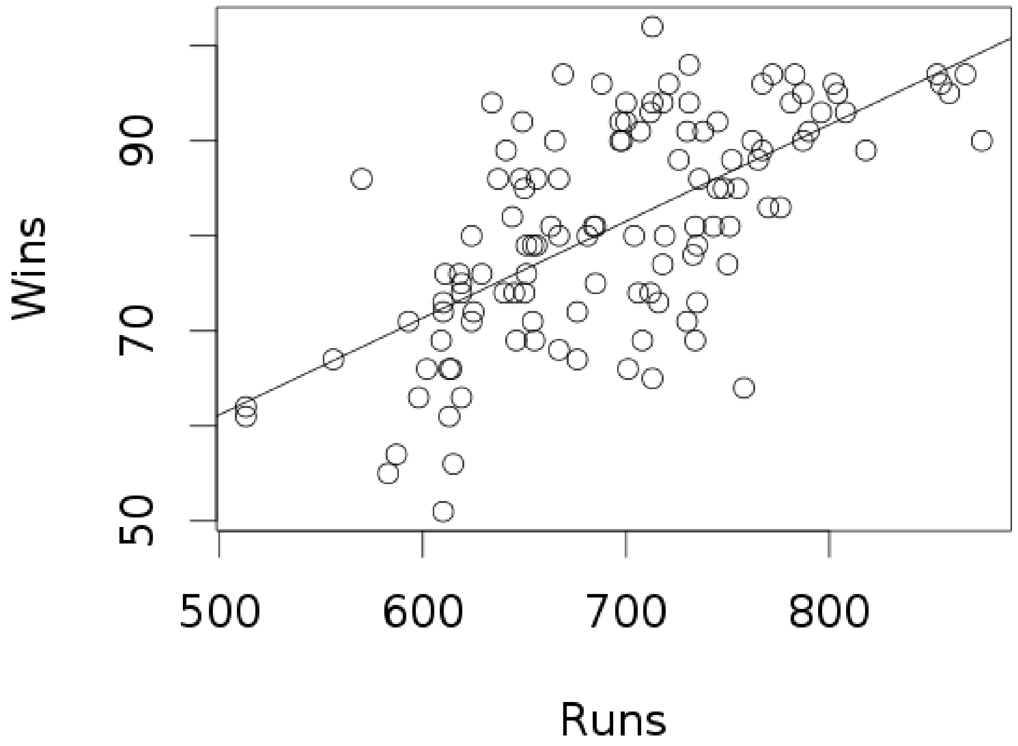
Fitting on the 2012 season, measuring the fit on the 2013 season

	RMSE
HR	62.85
BA	49.29
OBP	38.40
Slug	34.50
OPS	26.61
OPT*	30.39

This is a form of cross-validation!   (out of sample predictions)

Bill James’ “Pythagorean Method”

Recall that our equation for predicting the number of **wins** a team would score as a **function of the number of runs** they produced had some issues…



$\hat{w} = 14.47 + .088 \cdot Runs$

What happens when 0 runs are scored all season?

Bill James' "Pythagorean Method"

Bill James came up with a formula that he called the "Pythagorean Method" that relates:

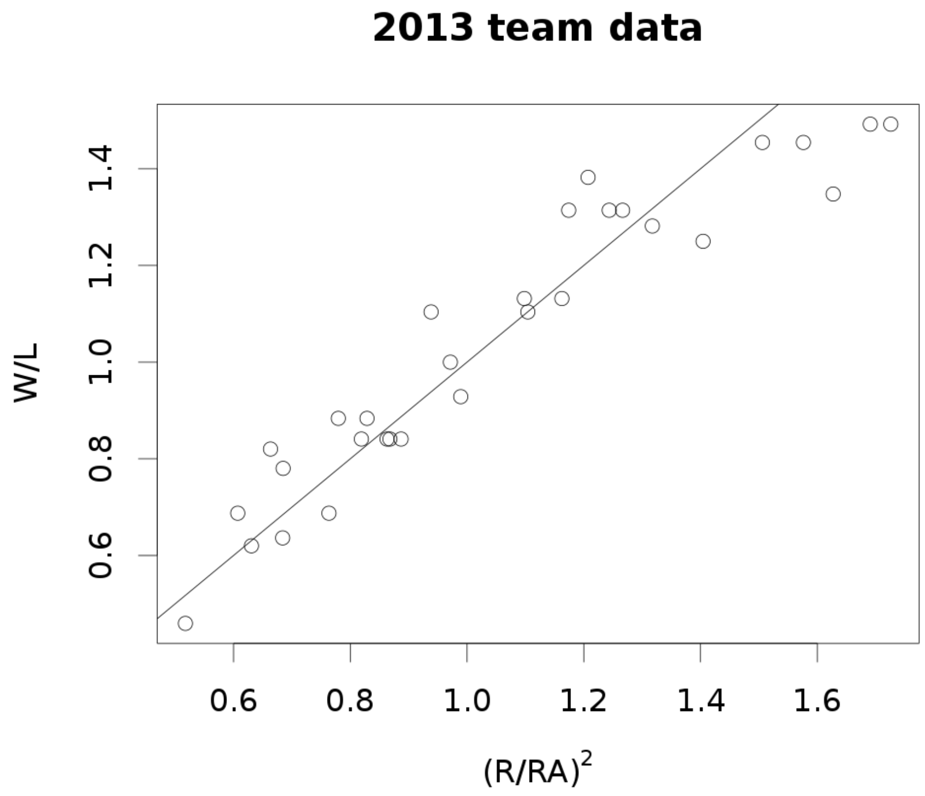
- wins (W) and losses (L) to
- runs scored (R) and runs allowed (RA)

$$\frac{W}{L} = \left(\frac{R}{RA}\right)^2$$

What happens when a team scores 0 runs with this formula?

How can we tell how good this formula is?

An answer: look at a scatter plot of W/L ratio predicted by $(R/RA)^2$ and the actual W/L ratio



How can we tell how good this formula is?

<u>An answer</u>: compare the number of wins predicted by the R and RA values, to the number of wins actually scored by each team

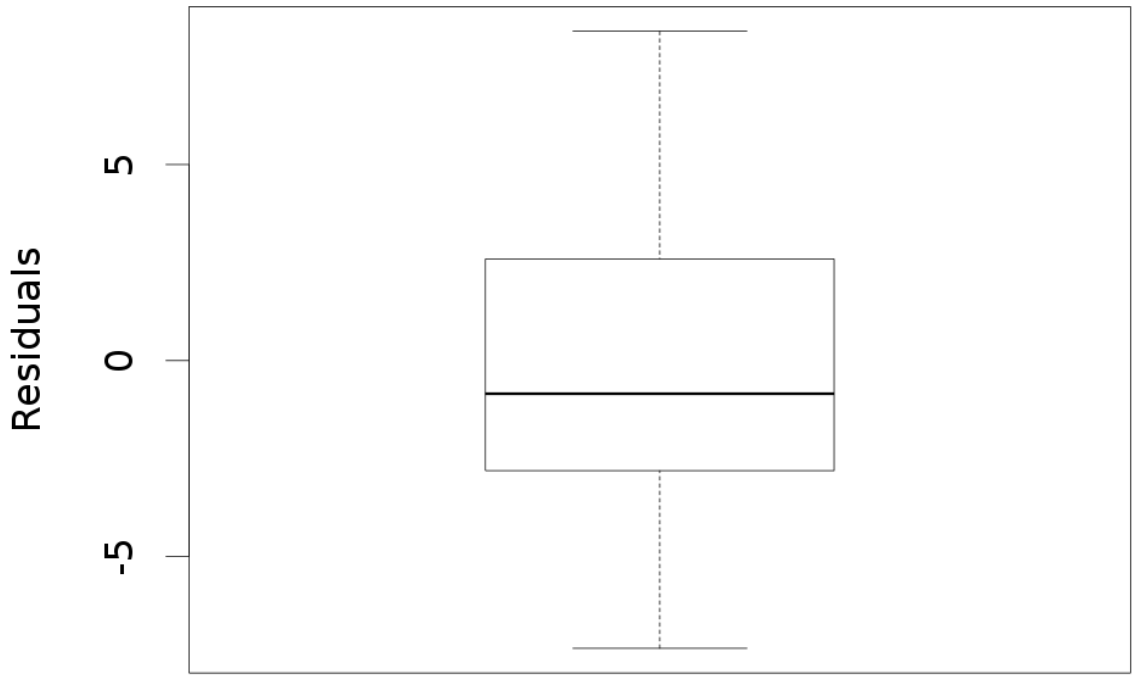
- i.e., look at the residuals of $W - \hat{W}_{Pythag}$

$(W/L)_{pred} = (R/RA)^2$

$(W_{pred} /(162 - W_{pred}) = (R/RA)^2$ ... some algebra ...

$W_{pred} = (162 \cdot (R/RA)^2)/(1 + (R/RA)^2)$

How can we tell how good this formula is?



RMSE = 3.9

95% of the time off by < 8 wins

- Assuming the residuals are normal

Five number summary of the residuals:
(-7.34, -2.81, -0.85, 2.59, 8.30)

Can we do better the James’ formula?

Any ideas how we could modify James' formula to do better?

One idea: try to find a better exponent on R/RA rather than just assuming it is 2

$$\frac{W}{L} = \left(\frac{R}{RA}\right)^2 \qquad\qquad \frac{W}{L} = \left(\frac{R}{RA}\right)^k$$

How can we do this?

Can we do better the James' formula?

If we take the logarithm of James' formula, it becomes a linear equation

$$\log\left(\frac{W}{L}\right) = 2 \cdot \log\left(\frac{R}{RA}\right)$$

$$\log\left(\frac{W}{L}\right) = k \cdot \log\left(\frac{R}{RA}\right)$$

Since this equation is linear we can find k with linear regression!

Can we do better the James' formula?

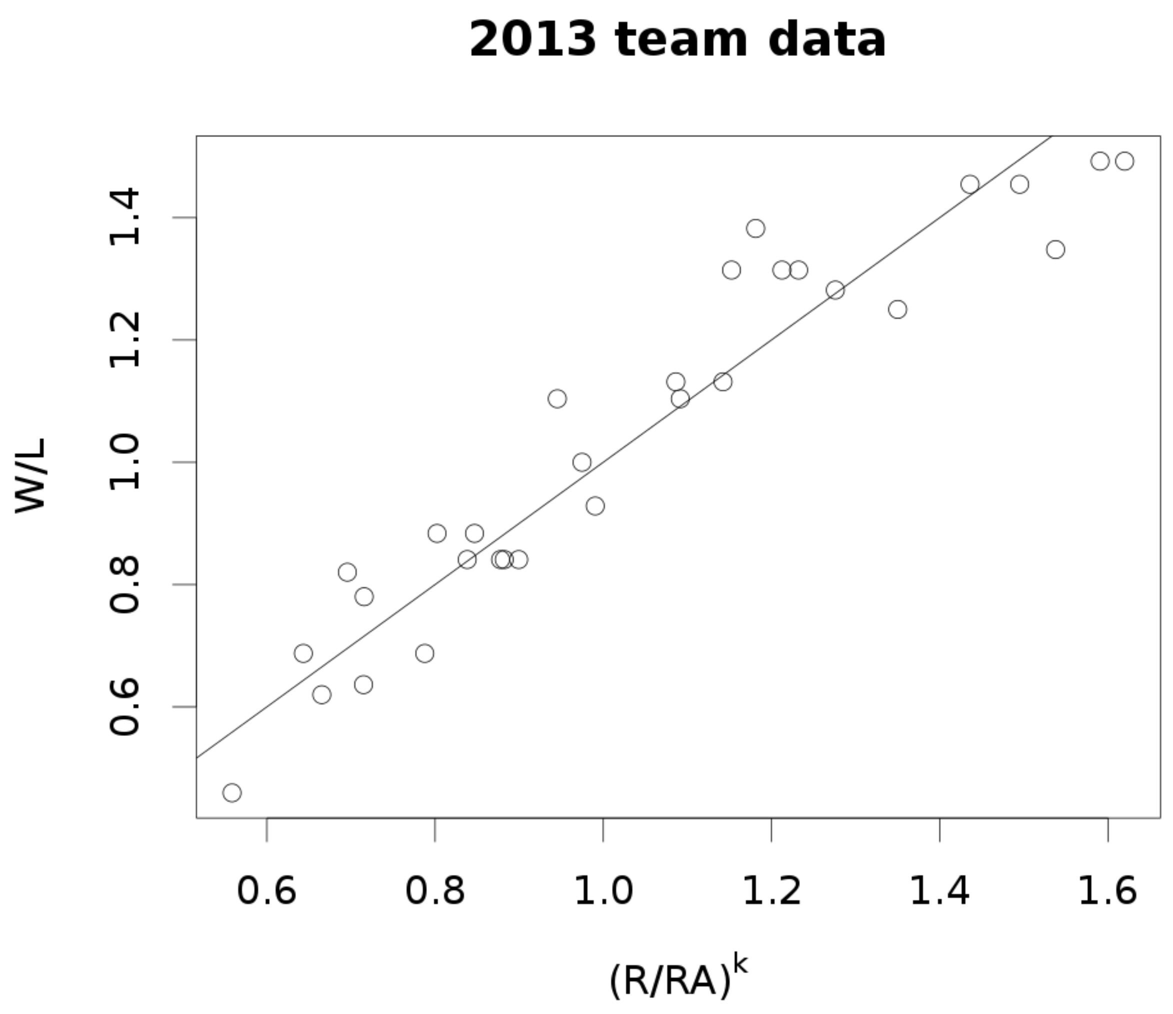
In R:

```
lm(formula = log(W.L.ratio) ~ log(R.RA.ratio))
    Coefficients:
    (Intercept)  log(R.RA.ratio)
    0.0003601        1.7675268
```

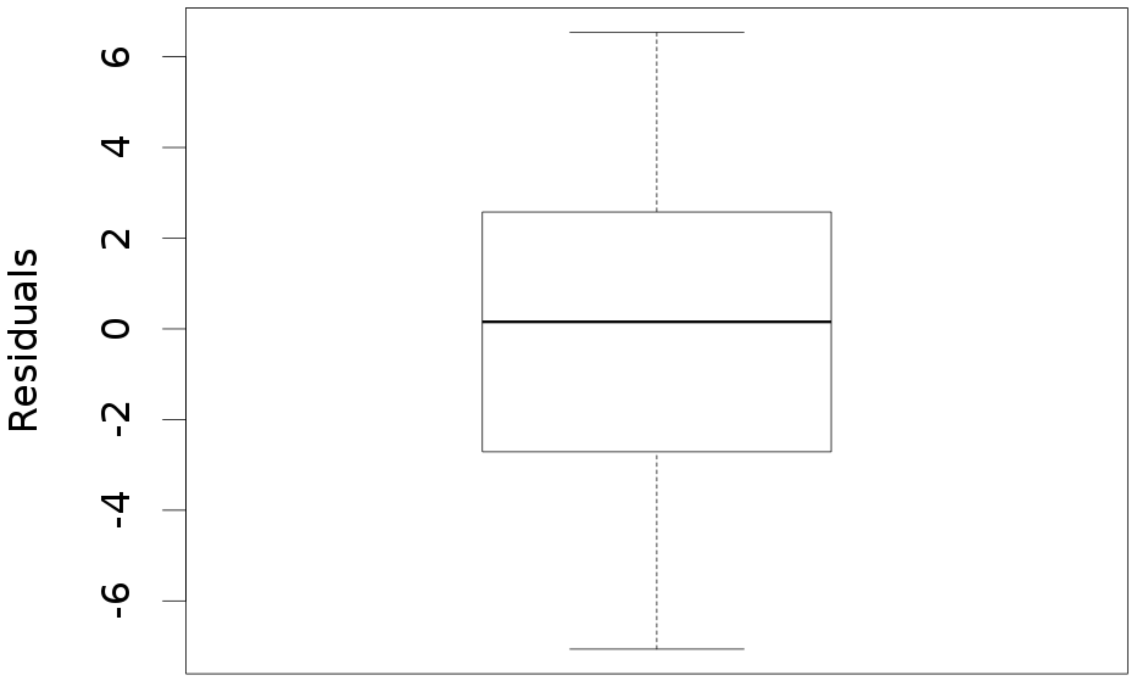
$$\frac{W}{L} = \left(\frac{R}{RA}\right)^{1.77}$$

You will try this in Python on a slightly different data set for homework 10!

Can we do better the James' formula?



Can we do better the James' formula?



Five number summary of the residuals:

Old: (-7.34, -2.81, -0.85, 2.59, 8.30)

New: (-7.06, -2.71, 0.15, 2.57, 6.54)

Old: RMSE = 3.9

New: RMSE = 3.6

Is this an improvement?

How can we better assess if this is a real improvement?

- Cross-validation!

Lab 10

If there's time, we can start on lab 10 now

Please be prepared with your 5 minute presentation for next class