Data manipulation

Outline for today

Better know a sport: Roberto Clemente

Review of multiple linear regression

Manipulating data with dplyr

Worksheet 5

Better know a player

Roberto Clemente

Announcements

Extra office hours this Friday. There is a sign up link on Moodle, or email me

Start thinking about your final project for the class. The guidelines for the final project are on Moodle.

A project proposal is due on Wednesday March 29th

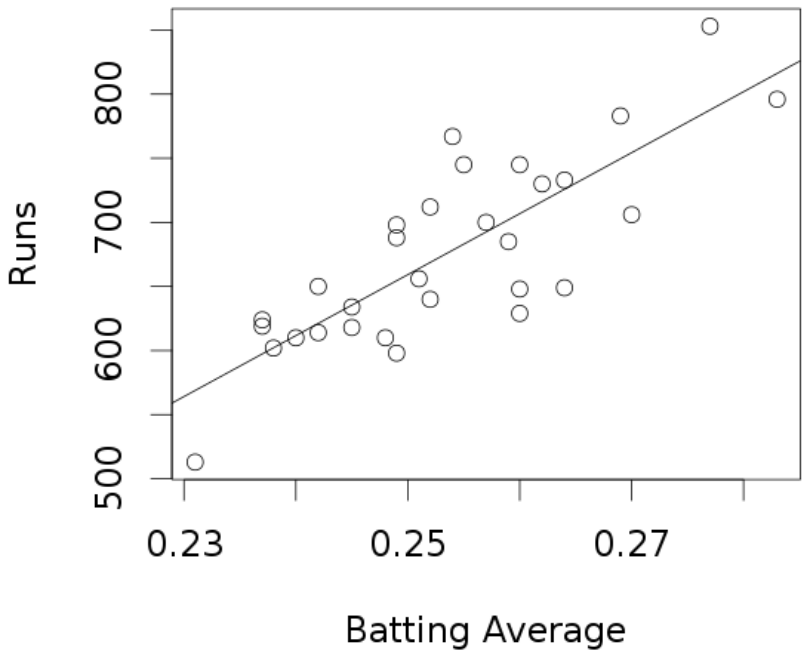
Review

Regression

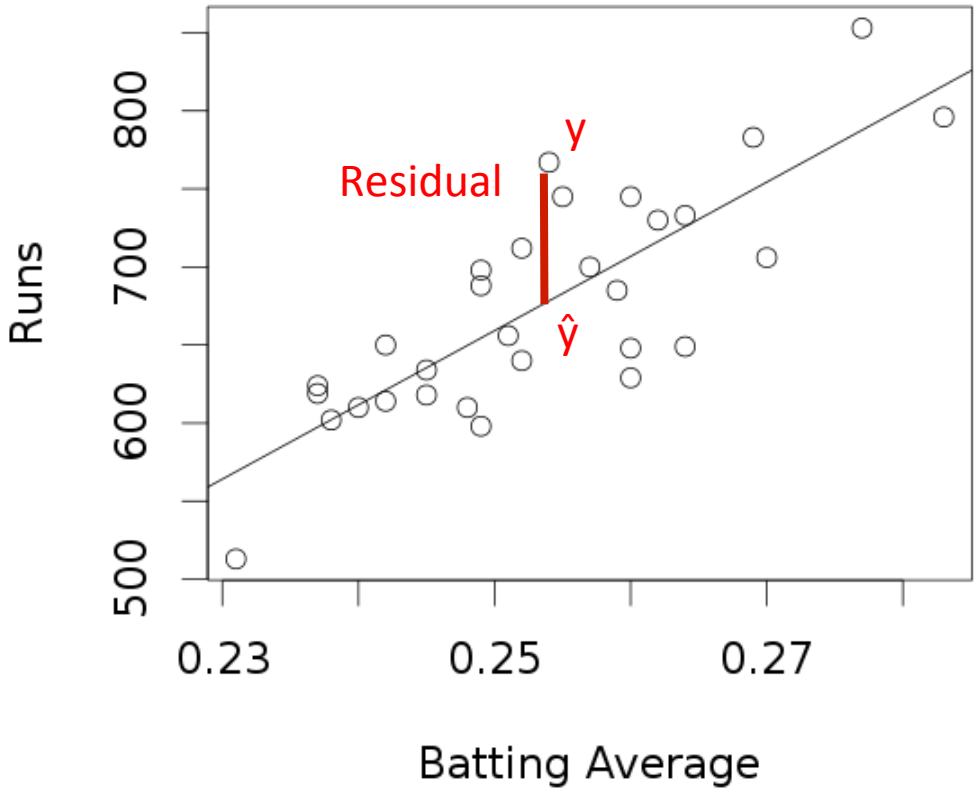
Regression is method of using one variable to predict the value of a second variable.

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

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



Measuring goodness of fit



Residual =

Observed – Predicted

= $y - \hat{y}$

We can measure how well the line fits the data using the mean squared error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad = \frac{1}{n}\sum_{i=1}^{n}(y_i - (a + b \cdot x_i))^2$$

Least square regression line minimizes the MSE

Multiple regression

We have observed that some statistics combine multiple types of measurements

Batting average:

$$BA = [(1) \cdot 1B + (1) \cdot 2B + (1) \cdot 3B + (1) \cdot HR]/AB$$

Slugging percentage:

$$Slug = [(1) \cdot 1B + (2) \cdot 2B + (3) \cdot 3B + (4) \cdot HR]/AB$$

Generic linear statistic:

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

Multiple regression

Multiple regression makes predictions ($\hat{y}$) using multiple variables

We can find the optimal weights (w_i's) for a combination of basic statistics by minimizing the MSE on

$$\hat{y} = w_1 \cdot \mathrm{BB} + w_2 \cdot \mathrm{HBP} + w_3 \cdot \mathrm{1B} + w_4 \cdot \mathrm{2B} + w_5 \cdot \mathrm{3B} + w_6 \cdot \mathrm{HR} + w_0$$

```
fit <- lm(R ~ BB + HBP + H + X2B + X3B + HR, data = team.batting.162)
```

The R model fit contains the weights w_i's for making predictions $\hat{y}$

```
coef(fit)
```

What is the best statistic we can create?

Suppose we fit:

```
fit <- lm(R~BB+H+X2B+X3B+HR+BRA+OBP+ AB, data = team.batting.162)
```

RMSE:

```
sqrt(mean(linear_model$residuals^2))
```

23.49

How accurate would this RMSE be if we applied it to new data?

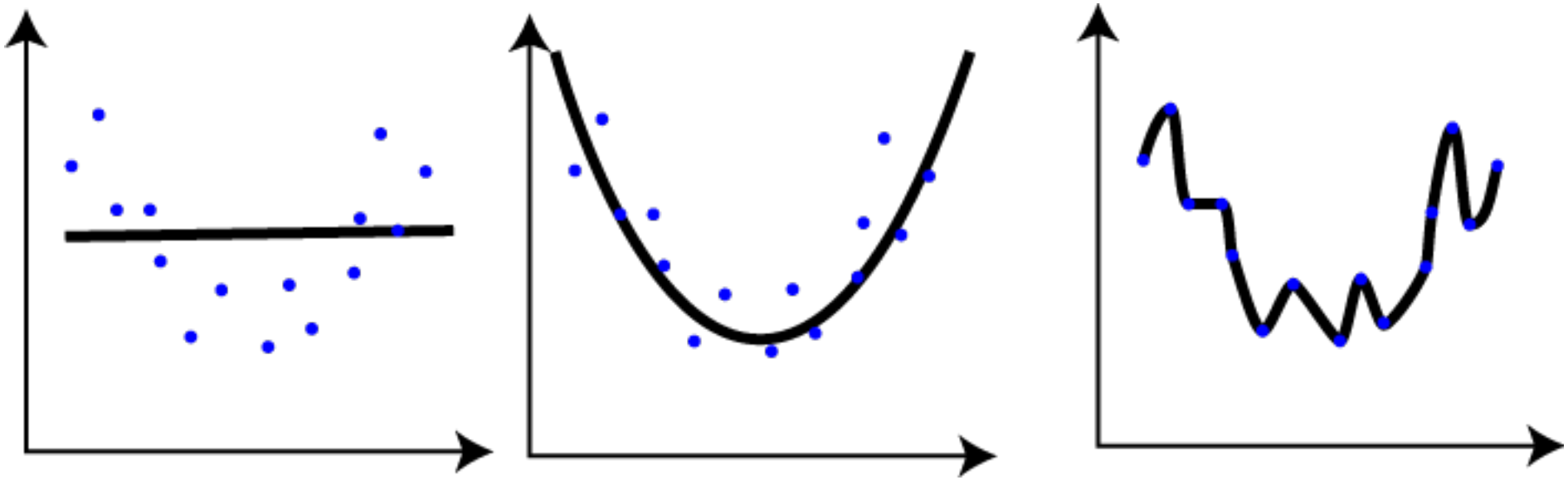
Overfitting

Fitting a model to precisely to the data at hand in such a way that it does not generalize to new data is called **overfitting**

If we used the same data to fit our model (find the w_i's) as we did to evaluate whether it was a good fit our estimate of RMSE might be too optimistic

- i.e., our estimate of the RMSE might be too small

Overfitting



One should always you different data when fitting and evaluating a model!

Cross-validation

Cross-validation is a method for assessing the goodness of a model in a way that can avoid overfitting

What we do is build the model on one set of data, called the training set

- i.e., find the coefficients on one set of data

Then we evaluate whether the model fits well on a second set of data, called the test set

- i.e., predict the $\hat{y}$'s based on x's from a new data set

If the model is truly good, we should get good predictions on the test set

- i.e., a small RMSE on the test set

Matthew's Batting Statistic (MBS)

Randomly split the data:

- ½ if the data is in the training set
- ½ of the data is in the test set

Fit the model using the training data for the MBS:

```
fit <- lm(R ~ BB+H+X2B+  …, data = training.data)
```

Make predictions on the test data

```
predicted.yhats <- predict(fit, newdata = test.data)
cross.validated.RMSE <- sqrt(mean((predicted.yhats - test.data$R)^2))
```

MSE for predictions made using the same training data x's and y's: 23.18

MSE for predictions made on the test data x's and y's: 24.63

Manipulating data with dplyr

R packages add additional functions to R

```
library('package.name')
```

dplyr is a very useful package for manipulating data frames

```
library('dplyr')
```

There are several very useful functions in the dplyr package including:

- filter()
- select()
- mutate()
- group_by()
- summarize()

All these functions take a data frame as input and return a data frame as output

filter()

The filter() function returns a subset of *rows* in a data frame

Example:

```
all.data <- get.Lahman.batting.data()
red.sox.data <- filter(all.data, teamID == "BOS")
```

Question:

How could we get all players who have less than 300 PA?

```
max.300PA <- filter(all.data, PA < 300)
```

The pipe operator %>%

The pipe operator %>% allows us to chain commands together

```
> all.data <- get.Lahman.batting.data()
```

```
> red.sox.2015 <- all.data   %>%
        filter(teamID == "BOS") %>%
        filter(yearID == 2015)
```

select()

The select() function returns a subset of the variables

- i.e., subset of the columns of a data frame

Example:

```
all.data <- get.Lahman.batting.data()
data.hits.and.walks <- select(all.data, H, BB)
```

Question:

How could we only home runs and doubles?

```
data.homeruns.and.doubles <- select(all.data, HR, X2B)
```

mutate()

The mutate() function adds new variables to a data frame from variables that are already in the data frame

- i.e., creates new columns from old columns

Example:

```
data.with.1B <- mutate(all.data, X1B = H – X2B – X3B - HR)
```

Question:

- How can we add BRA (which is OBP * SlugPct) to our data frame?

```
data.with.BRA <- mutate(all.data, BRA = OBP * SlugPct)
```

group_by()

The group_by() function assigns categorical variables to groups

- by itself it does nothing, but it is useful in conjunction with the summarize() function as described on the next slide

Example:

```
data.team.grouped <- group_by(all.data, teamID)
```

Question:

- How can group data by year?

```
data.year.grouped <- group_by(all.data, yearID)
```

summarize()

The summarize() function reduces the data based on the grouping assigned by the group_by() function

- i.e., it takes many cases and create summary statistics from these cases separately for each grouping.

Example:

```
data.team.grouped <- all.data  %>%
    group_by(teamID) %>%
    summarize(sum(H, na.rm = TRUE))
```

Question: How can we get the total number of hits as a function of the year?

```
data.year.grouped <- all.data   %>%
    group_by(yearID)   %>%
    summarize(sum(H, na.rm = TRUE))
```

Summary of some of what we have learned about descriptive statistics

Descriptive statistics: median, mean, standard deviation, percentiles, five number summary, range, interquartile range, z-scores, correlation

Plots: bar plots, pie charts, histograms, box plots, scatter plots

Regression: linear regression, multiple regression, residuals, RMSE, overfitting

A lot about baseball and analyzing data in R

Worksheet 5

> get.worksheet(5)

Please get started on this worksheet early, some of the questions on the worksheet might be challenging!

After the break: probability and inferential Statistics