Lesson 13. The Multiple Linear Regression Model – Part 1

Note. In Part 2 of this lesson, you can run the R code that generates the plots and outputs in here Part 1.

1 Overview

- We still want to study or predict the behavior of a response variable Y...
- But now, we will use multiple explanatory variables $X_1, X_2, \dots, X_k$

2 Choosing a multiple linear regression model

- We need:
  1. One quantitative response variable
  2. Multiple explanatory variables (quantitative or categorical)
- Suppose we have n observations of k explanatory variables $(X_1, \dots, X_k)$ and a response variable Y
- The **multiple linear regression model** is:

  - β_j describes the relationship between Y and X_j when all the other explanatory variables are held constant

3 Fitting a multiple linear regression model

- We use **least squares** to estimate the best fit
- The **fitted model** (or prediction equation) is:

- The **residual** of observation i is still defined as:

- The **estimated standard error of the multiple regression model** with k predictors is:

  - This is still interpreted as the size of a "typical" prediction error

4 Assessing a multiple linear regression model

- The conditions and assumptions are analogous to those in simple linear regression

Condition	Where to check	What we want
Linearity	Residuals vs. fitted values plot	Points evenly distributed above and below residual = 0 line, moving from left to right
Constant variance	Residuals vs. fitted values plot	Points span constant vertical width, moving from left to right
Normality	Normal Q-Q plot of residuals	Points in approximately straight line
Zero mean	Automatically met!	
Independence (of errors)	Description of data collection	No indication that errors influence each other
Randomness	Description of data collection	Data obtained using a random process

Example 1. How is an NFL team's winning percentage related to its offensive and defensive performance? The dataset `NFLStandings2016` from `Stat2Data` contains the records for all NFL teams during the 2016 regular season. *WinPct* is the winning percentage, *PointsFor* is the total number of points scored, and *PointsAgainst* is the total number of points allowed.

a. What is the response variable? What are the explanatory variables?

b. Write the model we will fit. Include the distribution of the error term.

c. We can fit the multiple regression using R with the following code:

```
fit <- lm(WinPct ~ PointsFor + PointsAgainst, data = NFLStandings2016)
summary(fit)
```

We get the following output:

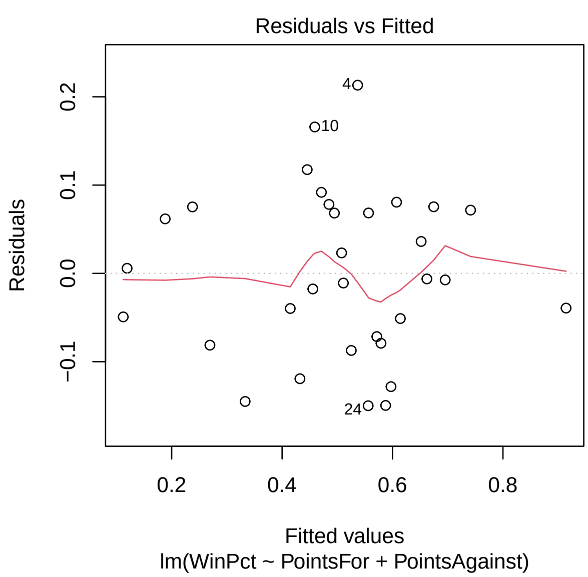
```
Call:
lm(formula = WinPct ~ PointsFor + PointsAgainst, data = NFLStandings2016)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.149898 -0.073482 -0.006821  0.072569  0.213189 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.7853698  0.1537422   5.108 1.88e-05 ***
PointsFor      0.0016992  0.0002628   6.466 4.48e-07 ***
PointsAgainst -0.0024816  0.0003204  -7.744 1.54e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09653 on 29 degrees of freedom
Multiple R-squared:  0.7824,	Adjusted R-squared:  0.7674 
F-statistic: 52.13 on 2 and 29 DF,  p-value: 2.495e-10
```

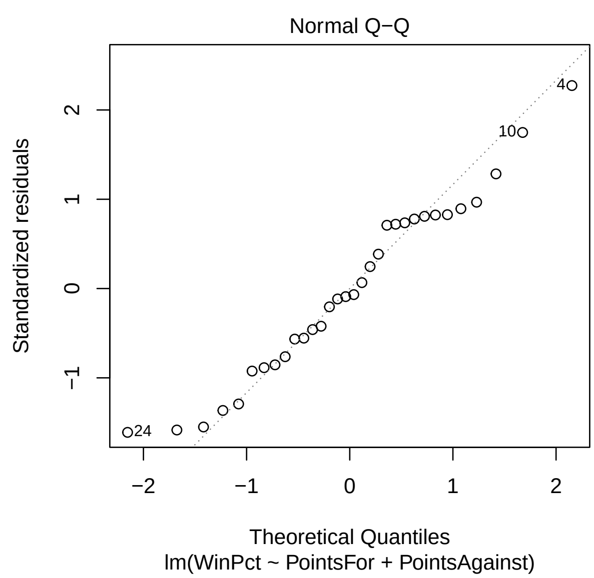
Write the prediction equation.

d. Assess whether the conditions for multiple regression appear to be met. The code below should look familiar to you – it creates a residuals vs. fitted values plot and a Normal Q-Q plot of the residuals:

```
plot(fit, which=1)
plot(fit, which=2)
```

The output is below:





Linearity	
Constant variance	
Normality	
Independence (of errors)	
Randomness	

e. Consider the Baltimore Ravens who scored 343 points while allowing 321 points during the 2016 season.

   i. What is their predicted winning percentage?

   ii. Their winning percentage was actually 0.500. What is the corresponding residual?

f. What is the estimated regression standard error?

g. Interpret the estimated coefficient of *PointsFor*.

h. What is the predicted increase in *WinPct* associated with a 7 point increase in *PointsFor* (holding *PointsAgainst* fixed)?