Introduction to linear regression

Batter up

The movie Moneyball focuses on the "quest for the secret of success in baseball". It follows a low-budget team, the Oakland Athletics, who believed that underused statistics, such as a player's ability to get on base, better predict the ability to score runs than typical statistics like home runs, RBIs (runs batted in), and batting average. Obtaining players who excelled in these underused statistics turned out to be much more affordable for the team.

In this lab we'll be looking at data from all 30 Major League Baseball teams and examining the linear relationship between runs scored in a season and a number of other player statistics. Our aim will be to summarize these relationships both graphically and numerically in order to find which variable, if any, helps us best predict a team's runs scored in a season.

The data

Let's load up the data for the 2011 season.

```
load("more/mlb11.RData")
```

In addition to runs scored, there are seven traditionally used variables in the data set: at-bats, hits, home runs, batting average, strikeouts, stolen bases, and wins. There are also three newer variables: on-base percentage, slugging percentage, and on-base plus slugging. For the first portion of the analysis we'll consider the seven traditional variables. At the end of the lab, you'll work with the newer variables on your own.

1. What type of plot would you use to display the relationship between `runs` and one of the other numerical variables? Plot this relationship using the variable `at_bats` as the predictor. Does the relationship look linear? If you knew a team's `at_bats`, would you be comfortable using a linear model to predict the number of runs?

The type of plot that I would use to display the relationship between runs and at_bats would be a scatterplot. See below for this plot:

```
plot(mlb11$at_bats, mlb11$runs, main="MLB",
   xlab="At Bats", ylab="Runs")
```

MLB
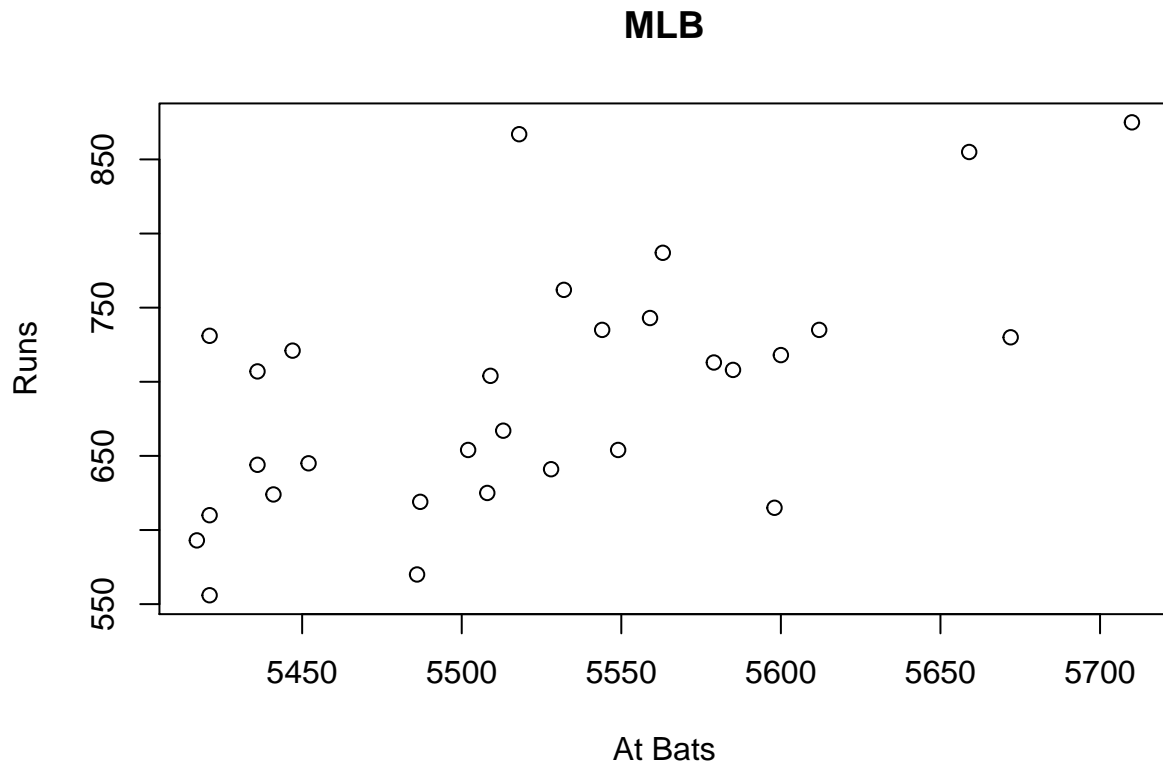
The relationship does appear to look linear. And yes, if I knew a team's at bats I would be comfortable predicting the number of runs.

If the relationship looks linear, we can quantify the strength of the relationship with the correlation coefficient.

```
cor(mlb11$runs, mlb11$at_bats)
```

```
## [1] 0.610627
```

Sum of squared residuals

Think back to the way that we described the distribution of a single variable. Recall that we discussed characteristics such as center, spread, and shape. It's also useful to be able to describe the relationship of two numerical variables, such as `runs` and `at_bats` above.

2. Looking at your plot from the previous exercise, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.

From the plot above, it looks like there is a positive linear relationship between runs and at_bats. As the number of at bats increases, the number of runs scored increases. The correlation coefficient of approximately 0.611 also supports this claim since it's positive. There are a few unusual observations, for instance, the New York Yankees had relatively few at-bats (5518), but scored the second most runs out of the 30 teams (867), which doesn't fit well with the majority of teams in the dataset. Additionally, the San Francisco Giants scored the second

fewest runs out of the 30 teams (570), but had more at bats than almost one third of the teams in the dataset.

Just as we used the mean and standard deviation to summarize a single variable, we can summarize the relationship between these two variables by finding the line that best follows their association. Use the following interactive function to select the line that you think does the best job of going through the cloud of points.
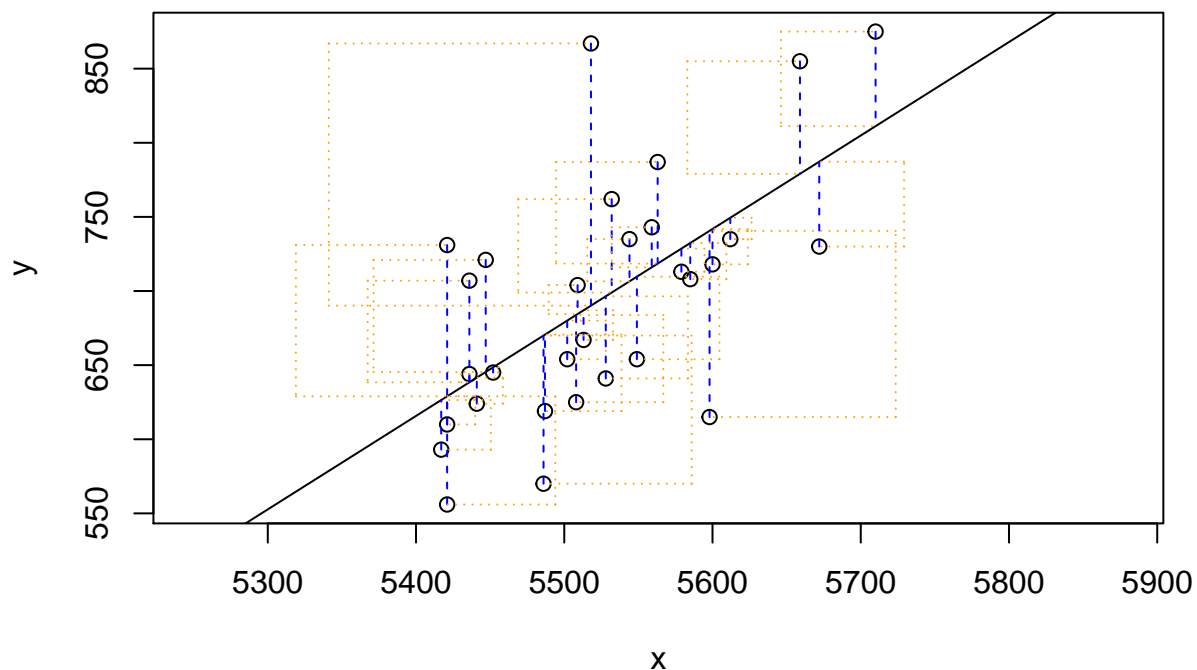
```r
plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)
```



```
## Click two points to make a line.

## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##  -2789.2429       0.6305
##
## Sum of Squares:  123721.9
```

After running this command, you'll be prompted to click two points on the plot to define a line. Once you've done that, the line you specified will be shown in black and the residuals in blue. Note that there are 30 residuals, one for each of the 30 observations. Recall that the residuals are the difference between the observed values and the values predicted by the line:

3

$$e_i = y_i - \hat{y}_i$$

The most common way to do linear regression is to select the line that minimizes the sum of squared residuals. To visualize the squared residuals, you can rerun the plot command and add the argument `showSquares = TRUE`.

```
plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)
```
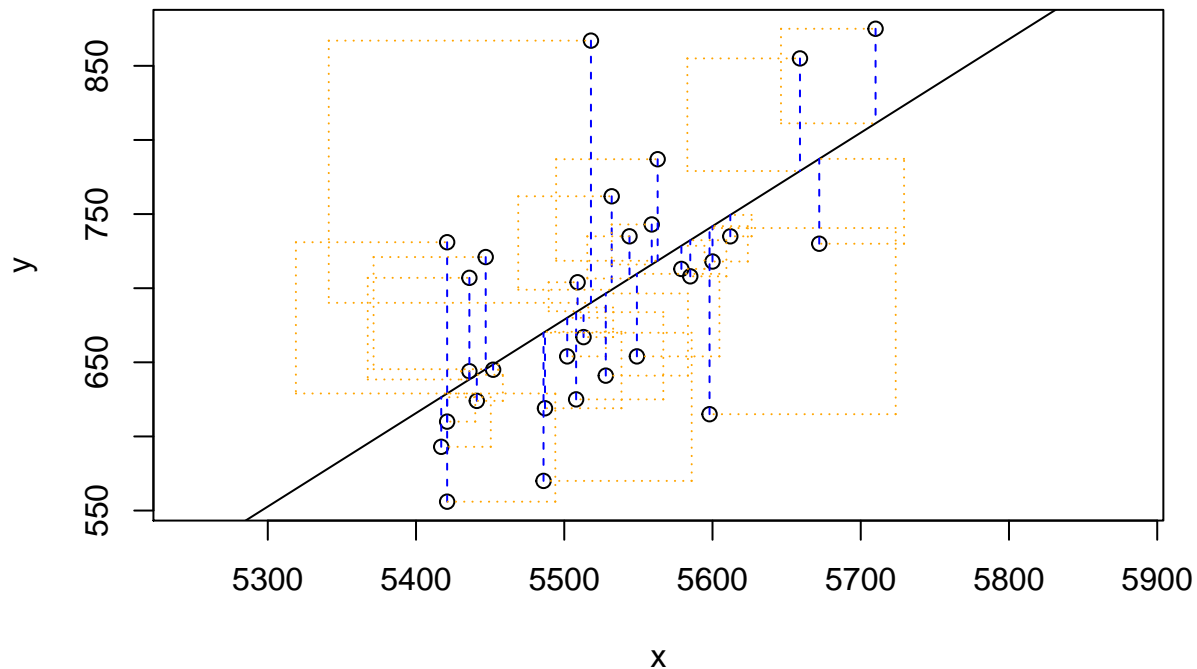


```
## Click two points to make a line.

## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##   -2789.2429       0.6305
##
## Sum of Squares:   123721.9
```

Note that the output from the `plot_ss` function provides you with the slope and intercept of your line as well as the sum of squares.

3. Using `plot_ss`, choose a line that does a good job of minimizing the sum of squares. Run the function several times. What was the smallest sum of squares that you got? How does it compare to your neighbors?

After running the function several times, the smallest sum of squares value I could come up with was **129877.5**. I don't have a neighbor sitting next to me, but I'm assuming that our sum of squares may not be the same, but would be close if we are both attempting to minimize the sum of squared residuals.

```
> plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)

Call:
lm(formula = y ~ x, data = pts)

Coefficients:
(Intercept)             x
  -3129.938         0.687

Sum of Squares:  149774.1
> plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)

Call:
lm(formula = y ~ x, data = pts)

Coefficients:
(Intercept)             x
 -2752.4661        0.6195

Sum of Squares:  141647.8
```

```
Call:
lm(formula = y ~ x, data = pts)

Coefficients:
(Intercept)            x
 -3007.7780      0.6667

Sum of Squares:  134844.2
> plot_ss(x = mlb11$at_bats, y = mlb11$runs, showSquares = TRUE)

Call:
lm(formula = y ~ x, data = pts)

Coefficients:
(Intercept)            x
 -2984.0380      0.6633

Sum of Squares:  129877.5
```

The linear model

It is rather cumbersome to try to get the correct least squares line, i.e. the line that minimizes the sum of squared residuals, through trial and error. Instead we can use the `lm` function in R to fit the linear model (a.k.a. regression line).

```
m1 <- lm(runs ~ at_bats, data = mlb11)
```

The first argument in the function `lm` is a formula that takes the form `y ~ x`. Here it can be read that we want to make a linear model of `runs` as a function of `at_bats`. The second argument specifies that R should look in the `mlb11` data frame to find the `runs` and `at_bats` variables.

The output of `lm` is an object that contains all of the information we need about the linear model that was just fit. We can access this information using the summary function.

```
summary(m1)
```

```
##
## Call:
## lm(formula = runs ~ at_bats, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125.58  -47.05  -16.59   54.40  176.87
##
## Coefficients:
```

6

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2789.2429   853.6957  -3.267 0.002871 **
## at_bats         0.6305     0.1545   4.080 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.47 on 28 degrees of freedom
## Multiple R-squared:  0.3729, Adjusted R-squared:  0.3505
## F-statistic: 16.65 on 1 and 28 DF,  p-value: 0.0003388
```

Let's consider this output piece by piece. First, the formula used to describe the model is shown at the top. After the formula you find the five-number summary of the residuals. The "Coefficients" table shown next is key; its first column displays the linear model's y-intercept and the coefficient of `at_bats`. With this table, we can write down the least squares regression line for the linear model:

$$\hat{y} = -2789.2429 + 0.6305 * atbats$$

One last piece of information we will discuss from the summary output is the Multiple R-squared, or more simply, R^2. The R^2 value represents the proportion of variability in the response variable that is explained by the explanatory variable. For this model, 37.3% of the variability in runs is explained by at-bats.

4. Fit a new model that uses `homeruns` to predict `runs`. Using the estimates from the R output, write the equation of the regression line. What does the slope tell us in the context of the relationship between success of a team and its home runs?

To fit a new model that uses homeruns to predict runs, we'll do the following:

```
m2 <- lm(runs ~ homeruns, data = mlb11)
summary(m2)
```

```
##
## Call:
## lm(formula = runs ~ homeruns, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -91.615 -33.410   3.231  24.292 104.631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 415.2389    41.6779   9.963 1.04e-10 ***
## homeruns      1.8345     0.2677   6.854 1.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.29 on 28 degrees of freedom
## Multiple R-squared:  0.6266, Adjusted R-squared:  0.6132
## F-statistic: 46.98 on 1 and 28 DF,  p-value: 1.9e-07
```

We can see from the new model that there is a positive correlation between homeruns and runs. The following equation can be made for this relationship:

$$\hat{y} = 415.24 + 1.835 * homeruns$$

This signifies that by adding the intercept of 415.24 to 1.835 times the number of homeruns a team hits, we can predict the number of runs this team will score. Since the slope here is a positive 1.835, we can see that the more homeruns a team hits, the more runs the team will score. Therefore, since more runs is connected to the success of a team, we can say that the number of homeruns hit will have a postive relationship with the success of the team.

Prediction and prediction errors

Let's create a scatterplot with the least squares line laid on top.

```
plot(mlb11$runs ~ mlb11$at_bats)
abline(m1)
```
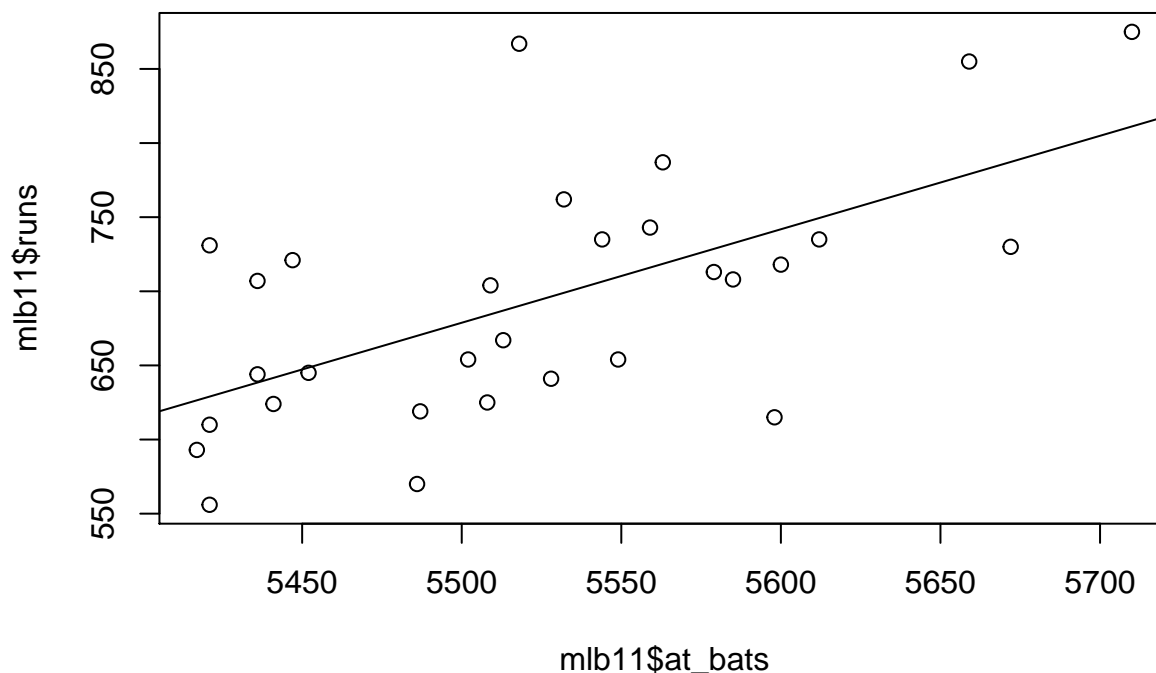

The function `abline` plots a line based on its slope and intercept. Here, we used a shortcut by providing the model `m1`, which contains both parameter estimates. This line can be used to predict y at any value of x. When predictions are made for values of x that are beyond the range of the observed data, it is referred to as *extrapolation* and is not usually recommended. However, predictions made within the range of the data are more reliable. They're also used to compute the residuals.

5. If a team manager saw the least squares regression line and not the actual data, how many runs would he or she predict for a team with 5,578 at-bats? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction?

We can use this equation to predict the number of runs based on the number of at-bats for a team:

$$\hat{y} = -2789.2429 + 0.6305 * atbats$$

$$\hat{y} = -2789.2429 + 0.6305 * (5578)$$

$$\hat{y} = 727.6861 runs$$

We can see here that the model would predict **727.69** runs scored based on **5578** at bats. To check to see whether or not this is an overestimate or an underestimate, we can check our dataset:

```
library(tidyverse)
library(DescTools)
x <- mlb11$at_bats
mlb11 %>% filter(at_bats == Closest(x, 5578)) %>%
  select(team, runs, at_bats)
```

```
##                      team runs at_bats
## 1 Philadelphia Phillies  713    5579
```

It looks like the closest at_bats count in the dataset is **5579** by the Philadelphia Phillies. With one extra at-bat, they scored **713** runs. Therefore, we would expect **0.6305** runs for one less at-bat, according to our slope, so it would be **712.3695** runs for 5578 at-bats. To calculate the residual of our previous prediction, we have the following calculation:

$$\hat{e} = y_i - \hat{y}$$

$$\hat{e} = 712.3695 - 727.6861$$

$$\hat{e} = -15.3166 runs$$

Our model overestimated the number of runs scored by about 15 runs.

Model diagnostics

To assess whether the linear model is reliable, we need to check for (1) linearity, (2) nearly normal residuals, and (3) constant variability.

Linearity: You already checked if the relationship between runs and at-bats is linear using a scatterplot. We should also verify this condition with a plot of the residuals vs. at-bats. Recall that any code following a # is intended to be a comment that helps understand the code but is ignored by R.

```
plot(m1$residuals ~ mlb11$at_bats)
abline(h = 0, lty = 3)  # adds a horizontal dashed line at y = 0
```

9
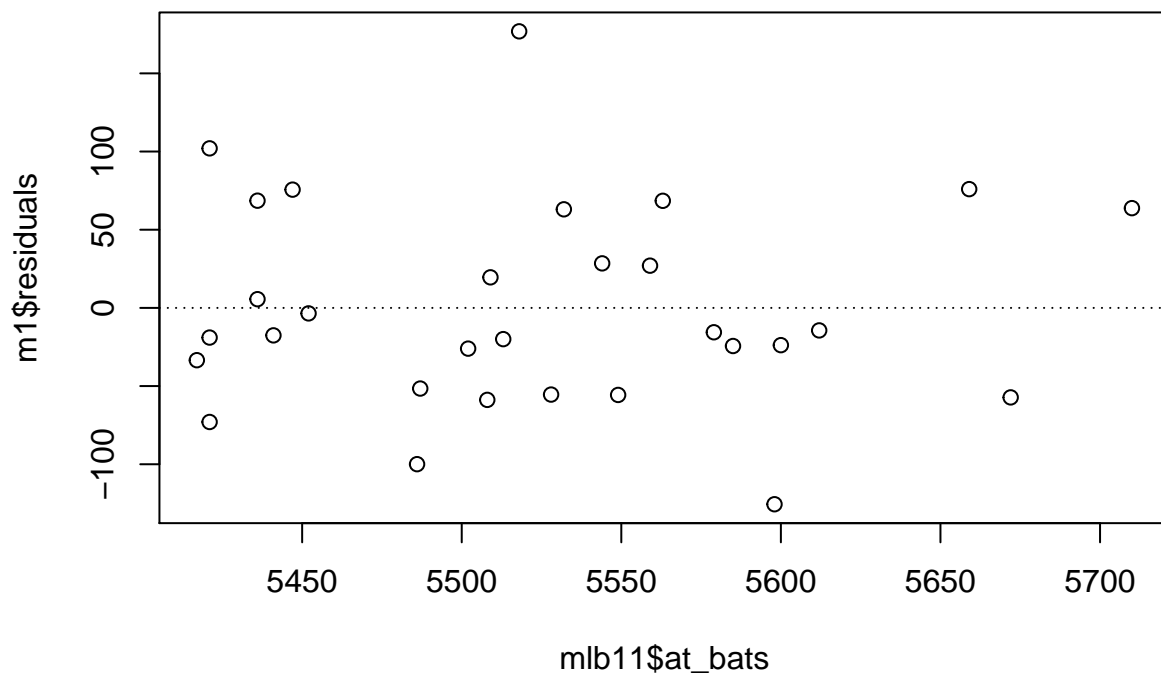
6. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between runs and at-bats?

There does not appear to be an apparent pattern in the residuals plot. This indicates that it is reasonable to try to fit a linear model to this data, however, it is unclear whether there is statistically significant evidence that the slope parameter is different from zero.
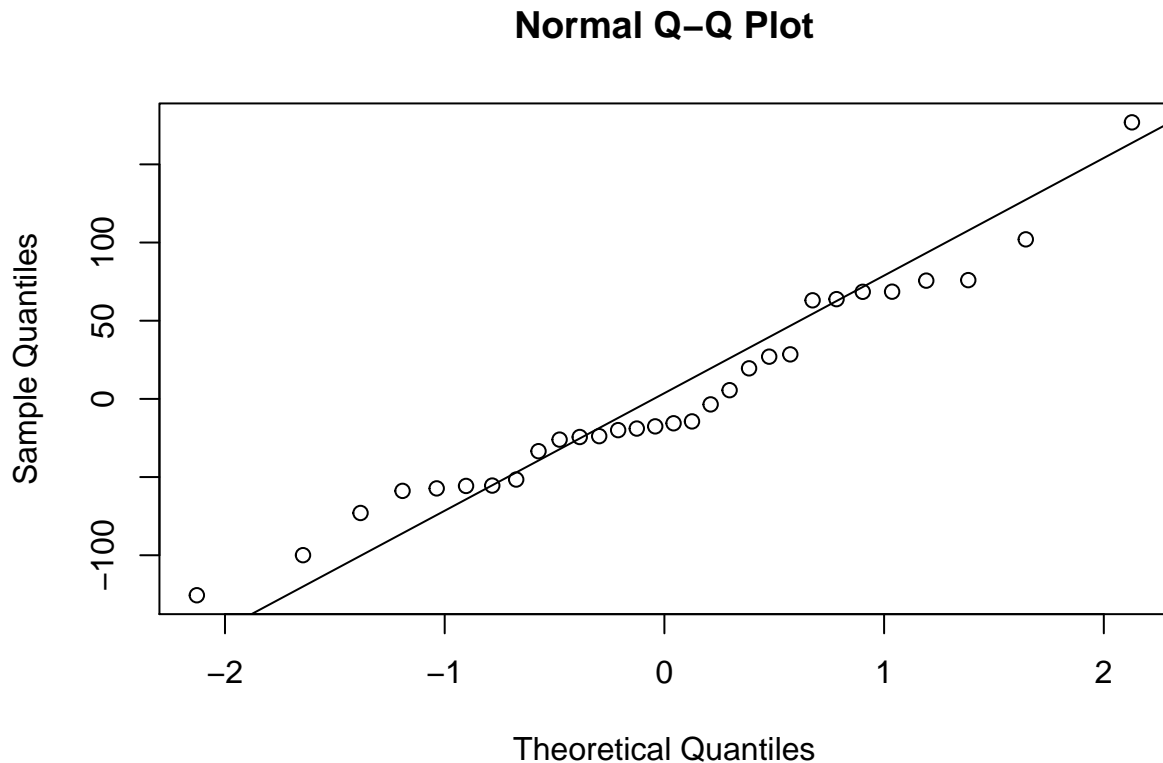
Nearly normal residuals: To check this condition, we can look at a histogram

```
hist(m1$residuals)
```

Histogram of m1$residuals
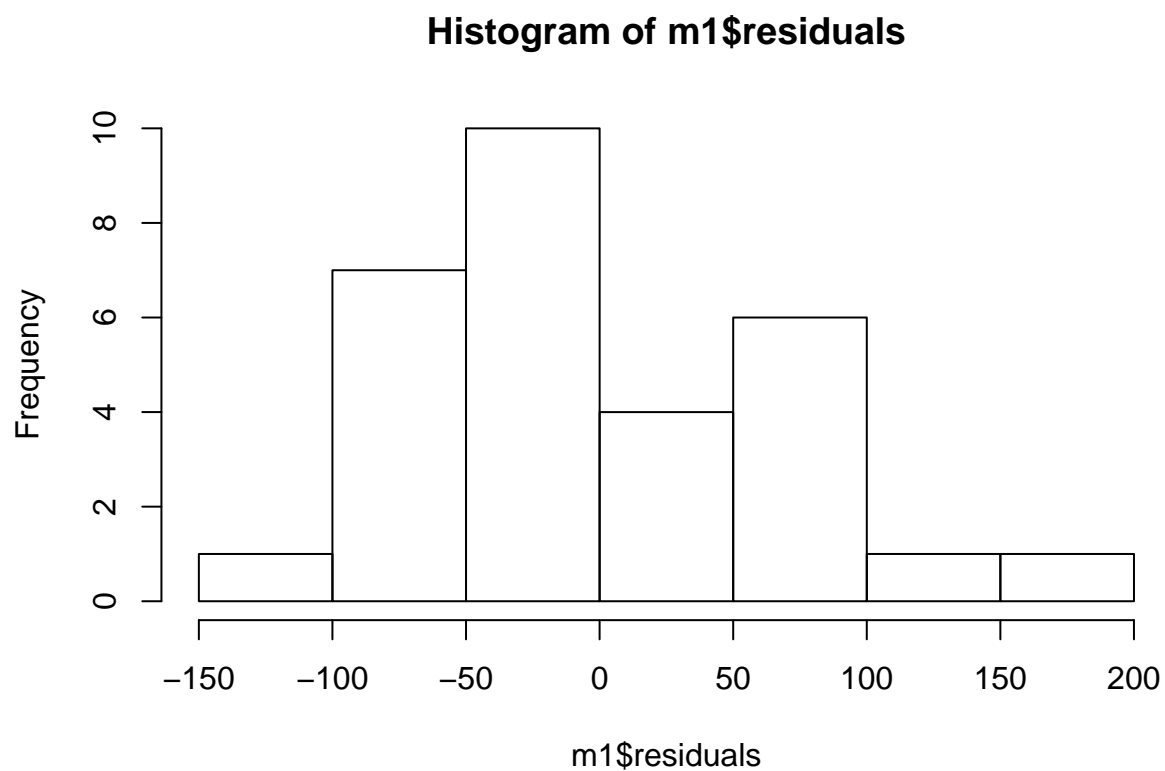
or a normal probability plot of the residuals.

```r
qqnorm(m1$residuals)
qqline(m1$residuals)   # adds diagonal line to the normal prob plot
```

Normal Q–Q Plot

7. Based on the histogram and the normal probability plot, does the nearly normal residuals condition appear to be met?

Looking at the histogram and the normal probability plot, it does appear that the nearly normal residuals condition is met – although it appears to be slightly right skewed, it is approaching a normal distribution. Additionally, the normal probability plot also appears to show most values close to the qqline.

Constant variability:

8. Based on the plot in (1), does the constant variability condition appear to be met?

The constant variability condition does appear to be met, since the first plot in this lab shows roughly constant variability of points around the least squares line. There seems to be a pretty good balance between the variability of y and the variability of x.

On Your Own

- Choose another traditional variable from `mlb11` that you think might be a good predictor of `runs`. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship?

I'll choose the 'wins' variable to see if it is a good predictor of 'runs', since it would make sense that there would be a relationship between the number of runs scored and the number of wins. See below for my scatterplot:

```
plot(mlb11$runs ~ mlb11$wins)
m3 <- lm(runs ~ wins, data = mlb11)
abline(m3)
```
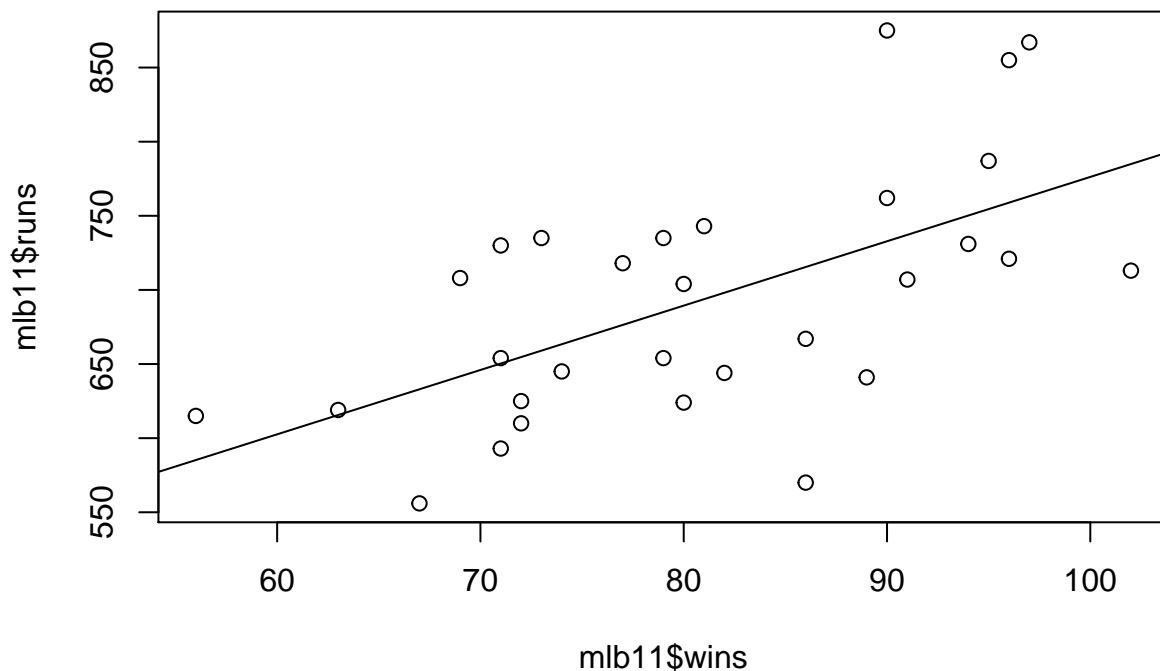

At a glance, there does seem to be a linear relationship between the number of runs and wins. It also appears to be a positive relationship, where the more runs that are scored by a team reflects more wins by a team.

- How does this relationship compare to the relationship between `runs` and `at_bats`? Use the R^2 values from the two model summaries to compare. Does your variable seem to predict `runs` better than `at_bats`? How can you tell?

```
summary(m2)
```

```
##
## Call:
## lm(formula = runs ~ homeruns, data = mlb11)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -91.615  -33.410    3.231   24.292  104.631
```

13

```
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 415.2389    41.6779   9.963 1.04e-10 ***
## homeruns      1.8345     0.2677   6.854 1.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 51.29 on 28 degrees of freedom
## Multiple R-squared:  0.6266, Adjusted R-squared:  0.6132
## F-statistic: 46.98 on 1 and 28 DF,  p-value: 1.9e-07
```

summary(m3)

```
## 
## Call:
## lm(formula = runs ~ wins, data = mlb11)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -145.450  -47.506   -7.482   47.346  142.186
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  342.121     89.223   3.834 0.000654 ***
## wins           4.341      1.092   3.977 0.000447 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 67.1 on 28 degrees of freedom
## Multiple R-squared:  0.361,  Adjusted R-squared:  0.3381
## F-statistic: 15.82 on 1 and 28 DF,  p-value: 0.0004469
```

It appears that 'wins' does not predict 'runs' as well as 'at-bats' when comparing the adjusted r-squared values, 0.3381 and 0.6132 respectively. Since the wins variable would only account for about 34% of the variability in runs, and at-bats would account for about 61% of the variability in runs, this can be concluded.

- Now that you can summarize the linear relationship between two variables, investigate the relationships between **runs** and each of the other five traditional variables. Which variable best predicts **runs**? Support your conclusion using the graphical and numerical methods we've discussed (for the sake of conciseness, only include output for the best variable, not all five).

We will now look at the remaining traditional variables of 'hits', 'homeruns', bat_avg', 'strike-outs', and 'stolen_bases'.

summary(m1)

```
## 
## Call:
## lm(formula = runs ~ at_bats, data = mlb11)
## 
```

```
## Residuals:
##     Min     1Q  Median     3Q     Max
## -125.58  -47.05  -16.59   54.40  176.87
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2789.2429   853.6957  -3.267 0.002871 **
## at_bats         0.6305     0.1545   4.080 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.47 on 28 degrees of freedom
## Multiple R-squared:  0.3729, Adjusted R-squared:  0.3505
## F-statistic: 16.65 on 1 and 28 DF,  p-value: 0.0003388
```

```r
m4 <- lm(runs ~ hits, data = mlb11) # runs and hits
summary(m4)
```

```
##
## Call:
## lm(formula = runs ~ hits, data = mlb11)
##
## Residuals:
##      Min      1Q  Median      3Q     Max
## -103.718  -27.179   -5.233   19.322  140.693
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -375.5600   151.1806  -2.484   0.0192 *
## hits           0.7589     0.1071   7.085 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.23 on 28 degrees of freedom
## Multiple R-squared:  0.6419, Adjusted R-squared:  0.6292
## F-statistic:  50.2 on 1 and 28 DF,  p-value: 1.043e-07
```

```r
m5 <- lm(runs ~ homeruns, data = mlb11) # runs and homeruns
summary(m5)
```

```
##
## Call:
## lm(formula = runs ~ homeruns, data = mlb11)
##
## Residuals:
##     Min     1Q  Median     3Q     Max
## -91.615 -33.410   3.231  24.292 104.631
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 415.2389    41.6779   9.963 1.04e-10 ***
## homeruns      1.8345     0.2677   6.854 1.90e-07 ***
## ---
```

```
## Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.29 on 28 degrees of freedom
## Multiple R-squared:  0.6266, Adjusted R-squared:  0.6132
## F-statistic: 46.98 on 1 and 28 DF,  p-value: 1.9e-07
```

```r
m6 <- lm(runs ~ bat_avg, data = mlb11) # runs and bat_avg
summary(m6)
```

```
##
## Call:
## lm(formula = runs ~ bat_avg, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -94.676 -26.303  -5.496  28.482 131.113
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -642.8      183.1  -3.511  0.00153 **
## bat_avg       5242.2      717.3   7.308 5.88e-08 ***
## ---
## Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.23 on 28 degrees of freedom
## Multiple R-squared:  0.6561, Adjusted R-squared:  0.6438
## F-statistic: 53.41 on 1 and 28 DF,  p-value: 5.877e-08
```

```r
m7 <- lm(runs ~ strikeouts, data = mlb11) # runs and strikeouts
summary(m7)
```

```
##
## Call:
## lm(formula = runs ~ strikeouts, data = mlb11)
##
## Residuals:
##      Min      1Q  Median      3Q     Max
## -132.27  -46.95  -11.92   55.14  169.76
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1054.7342   151.7890   6.949 1.49e-07 ***
## strikeouts    -0.3141     0.1315  -2.389   0.0239 *
## ---
## Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.5 on 28 degrees of freedom
## Multiple R-squared:  0.1694, Adjusted R-squared:  0.1397
## F-statistic: 5.709 on 1 and 28 DF,  p-value: 0.02386
```

```r
m8 <- lm(runs ~ stolen_bases, data = mlb11) # runs and stolen_bases
summary(m8)
```

```
##
## Call:
## lm(formula = runs ~ stolen_bases, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -139.94  -62.87   10.01   38.54  182.49
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  677.3074    58.9751  11.485 4.17e-12 ***
## stolen_bases   0.1491     0.5211   0.286    0.777
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.82 on 28 degrees of freedom
## Multiple R-squared:  0.002914,   Adjusted R-squared:  -0.0327
## F-statistic: 0.08183 on 1 and 28 DF,  p-value: 0.7769
```

It appears that batting average seems to be the best predictor of runs with an adjusted R-squared value of 0.6438. This was the highest adjusted R-square value out of the the traditional variables, and indicates that about 64% of the variability in runs can be explained by batting average. Here's the plot:
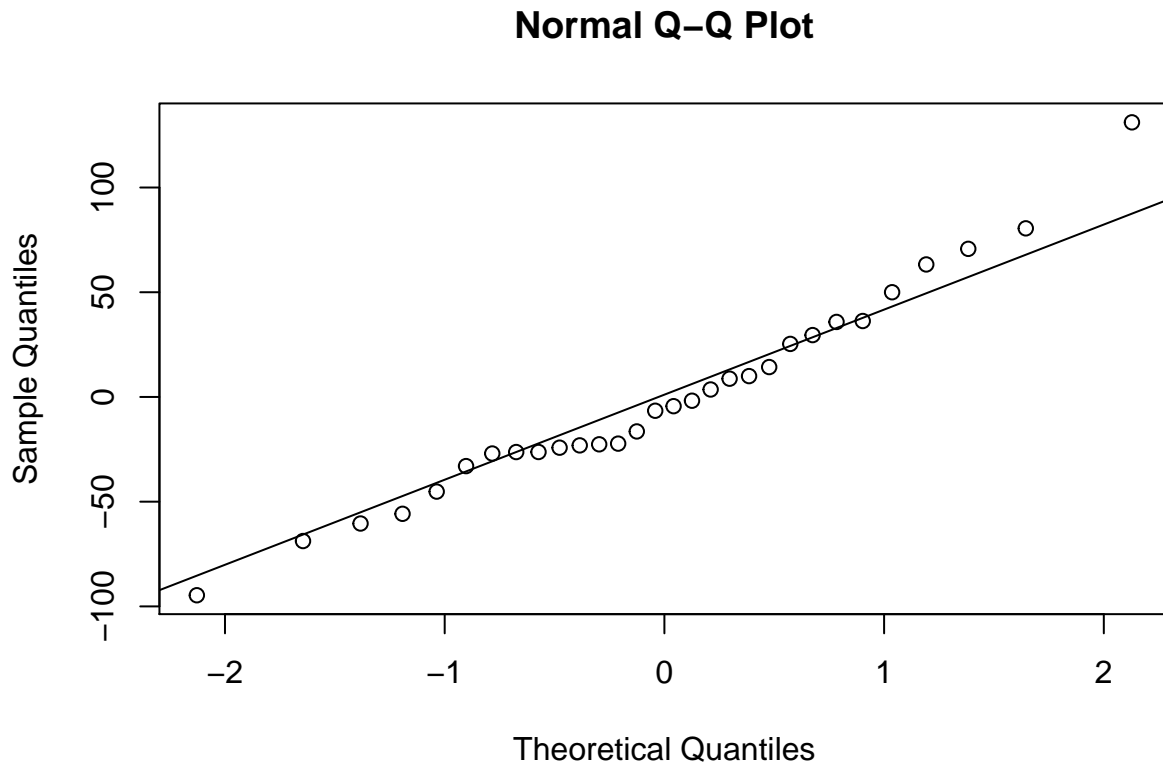
```
plot(mlb11$runs ~ mlb11$bat_avg)
abline(m6)
```
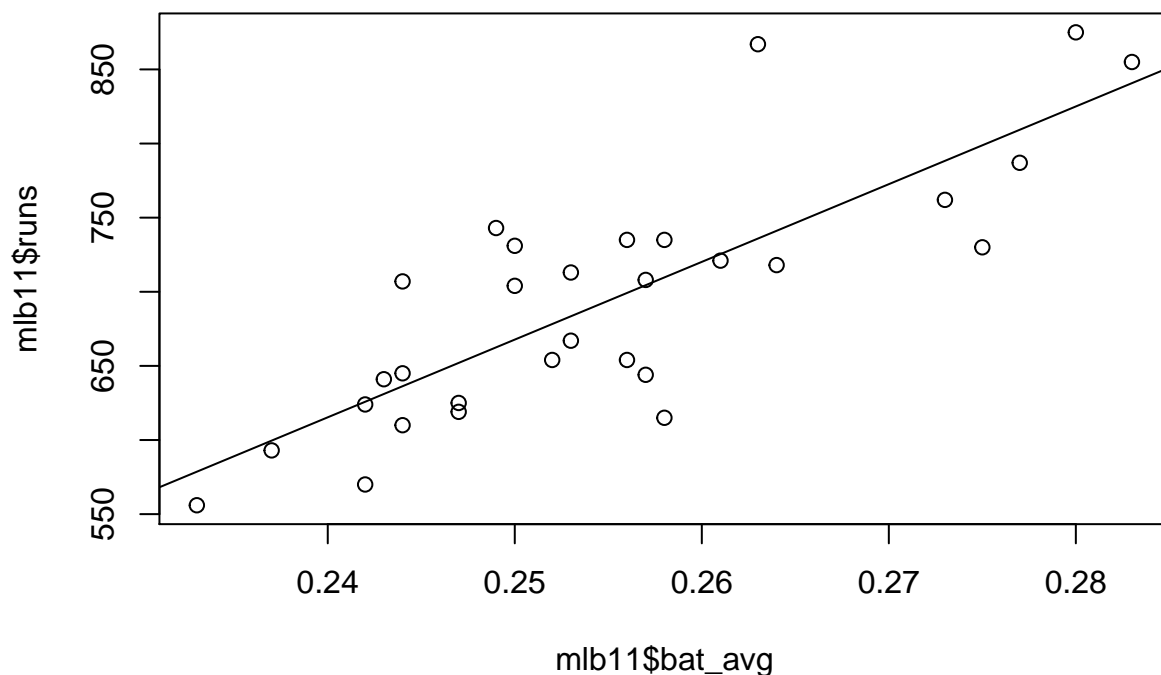
Also, by plotting the residuals we can see the following:

```
plot(m6$residuals ~ mlb11$bat_avg)
abline(h = 0, lty = 3)  # adds a horizontal dashed line at y = 0
```
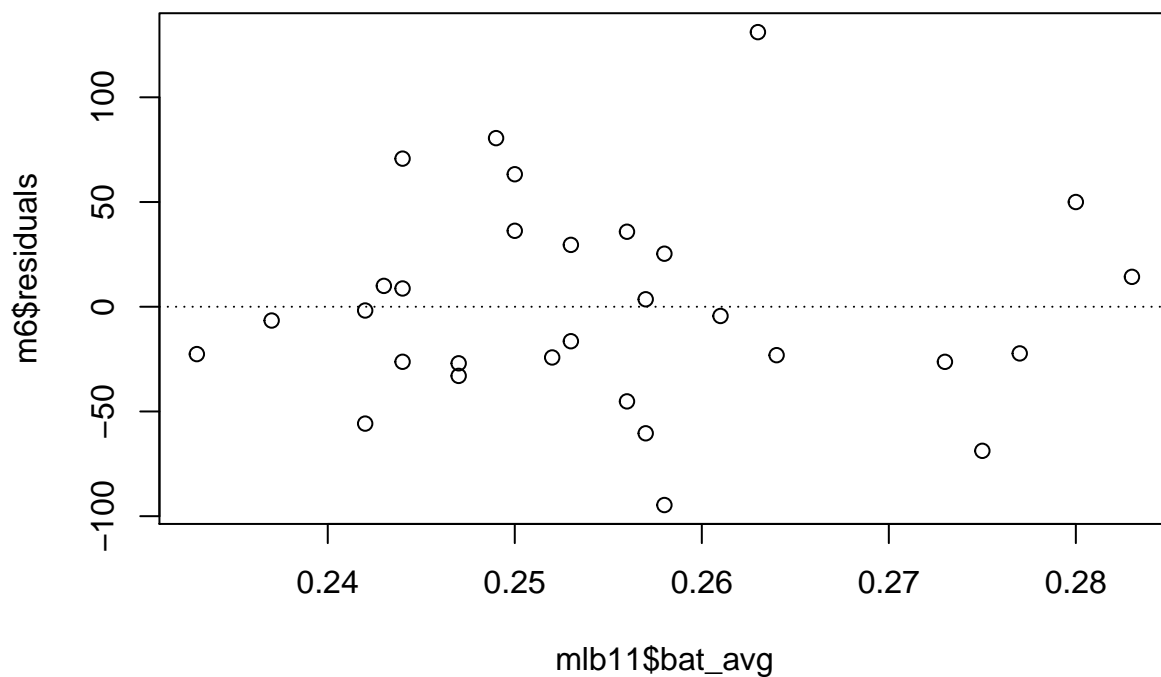


```
qqnorm(m6$residuals)
qqline(m6$residuals)  # adds diagonal line to the normal prob plot
```

Normal Q–Q Plot

There seems to be a strong relationship between the number of runs scored and batting average. The plots confirm that there is linearity, nearly normal residuals, and constant variability.

- Now examine the three newer variables. These are the statistics used by the author of *Moneyball* to predict a teams success. In general, are they more or less effective at predicting runs that the old variables? Explain using appropriate graphical and numerical evidence. Of all ten variables we've analyzed, which seems to be the best predictor of **runs**? Using the limited (or not so limited) information you know about these baseball statistics, does your result make sense?

```
m_new1 <- lm(runs ~ new_onbase, data = mlb11) # runs and on-base percentage
summary(m_new1)
```

```
##
## Call:
## lm(formula = runs ~ new_onbase, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.270 -18.335   3.249  19.520  69.002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1118.4      144.5  -7.741 1.97e-08 ***
## new_onbase     5654.3      450.5  12.552 5.12e-13 ***
```

19

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.61 on 28 degrees of freedom
## Multiple R-squared:  0.8491, Adjusted R-squared:  0.8437
## F-statistic: 157.6 on 1 and 28 DF,  p-value: 5.116e-13
```

```r
m_new2 <- lm(runs ~ new_slug, data = mlb11) # runs and slugging percentage
summary(m_new2)
```

```
##
## Call:
## lm(formula = runs ~ new_slug, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.41  -18.66   -0.91   16.29   52.29
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -375.80      68.71   -5.47 7.70e-06 ***
## new_slug      2681.33     171.83   15.61 2.42e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.96 on 28 degrees of freedom
## Multiple R-squared:  0.8969, Adjusted R-squared:  0.8932
## F-statistic: 243.5 on 1 and 28 DF,  p-value: 2.42e-15
```

```r
m_new3 <- lm(runs ~ new_obs, data = mlb11) # runs and on-base plus slugging
summary(m_new3)
```

```
##
## Call:
## lm(formula = runs ~ new_obs, data = mlb11)
##
## Residuals:
##      Min      1Q  Median      3Q     Max
## -43.456 -13.690   1.165  13.935  41.156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -686.61      68.93  -9.962 1.05e-10 ***
## new_obs       1919.36      95.70  20.057  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.41 on 28 degrees of freedom
## Multiple R-squared:  0.9349, Adjusted R-squared:  0.9326
## F-statistic: 402.3 on 1 and 28 DF,  p-value: < 2.2e-16
```

After looking at these new variables, it appears that they are more effective at predicting
runs than the old variables. I believe this is the case because the adjusted R-square values

are much higher than the old variables, and can account for more of the variability in runs. For instance, the on-base plus slugging number can account for about **93%** of the variability in runs, whereas the highest adjusted R-square value for the old variables was around **64%**. This can also be confirmed by the plots of batting average (best predictor of old variables), and on-base plus slugging (best predictor of new variables). See below for the comparison:

```
plot(mlb11$runs ~ mlb11$bat_avg) # best predictor of old variables
abline(m6)
```
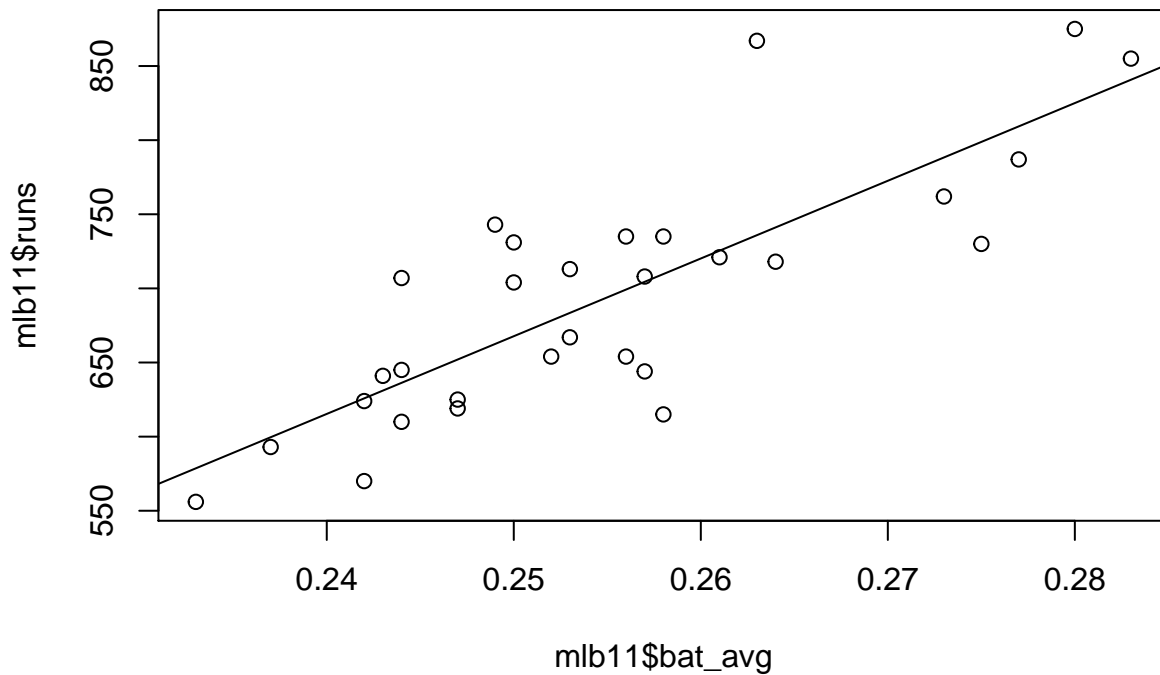


```
plot(mlb11$runs ~ mlb11$new_obs) # best predictor of new variables
abline(m_new3)
```
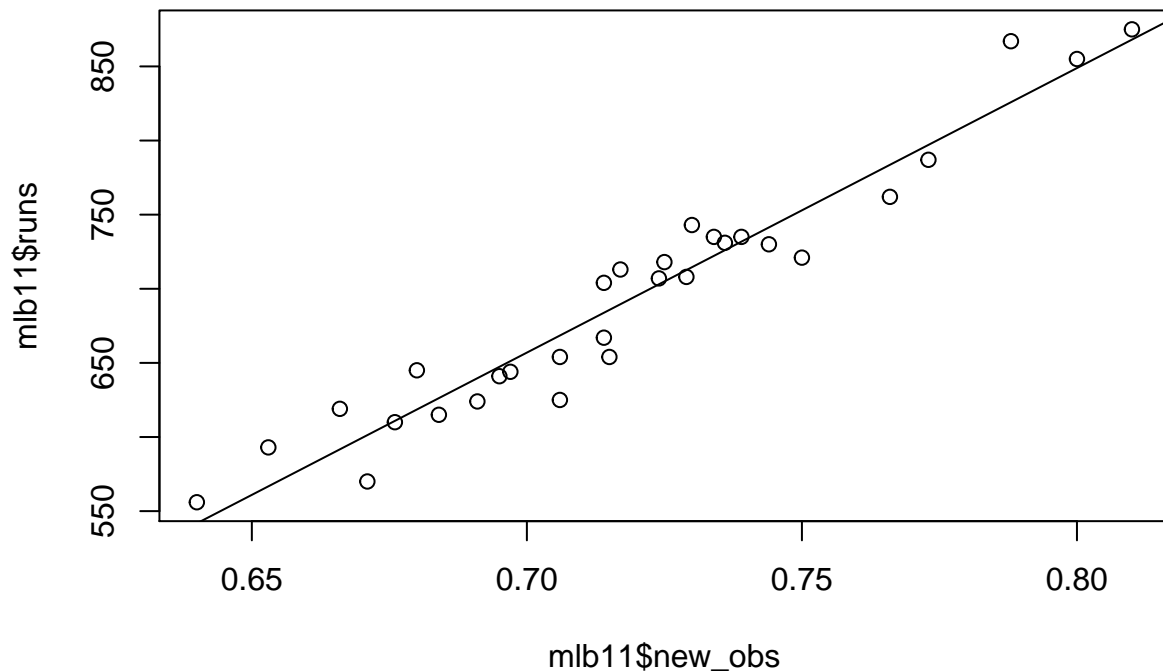
We can visually see that the residuals are much smaller for 'new_obs' than they are for 'bat_avg'.

Slugging percentage is calculated as the total number of bases divided by the number of at-bats, whereas batting average is the number of hits divided by the total number of at-bats. Given that we see on-base percentage account for roughly 84% of the variability in runs, and this metric added to the slugging percentage that accounts for even more variability of runs, it does make sense that the 'new_obs' variable which adds these two together would account for the most variability in runs. In terms of thinking through it from a baseball perspective, it makes sense that if a team has more people on base, there would intuitively be a better chance that they would score more runs, since there would be more of an opportunity to advance those runners to score runs.

- Check the model diagnostics for the regression model with the variable you decided was the best predictor for runs.
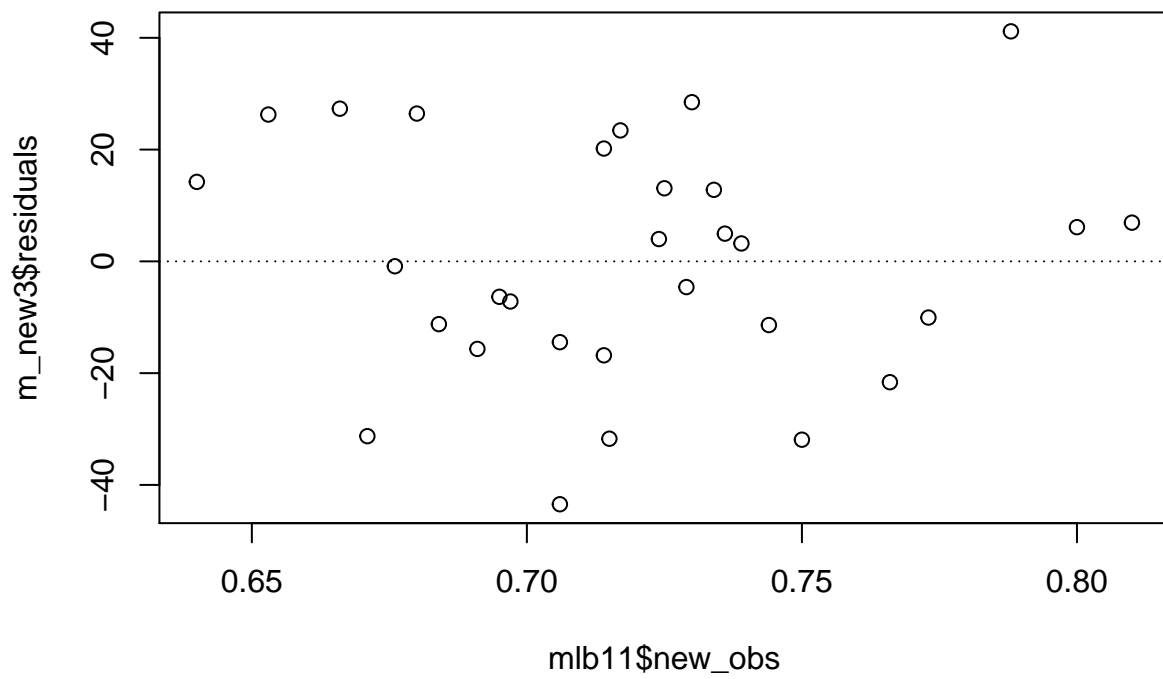
Here's a plot of the histogram for 'new_obs':

```
hist(m_new3$residuals)
```

Histogram of m_new3$residuals
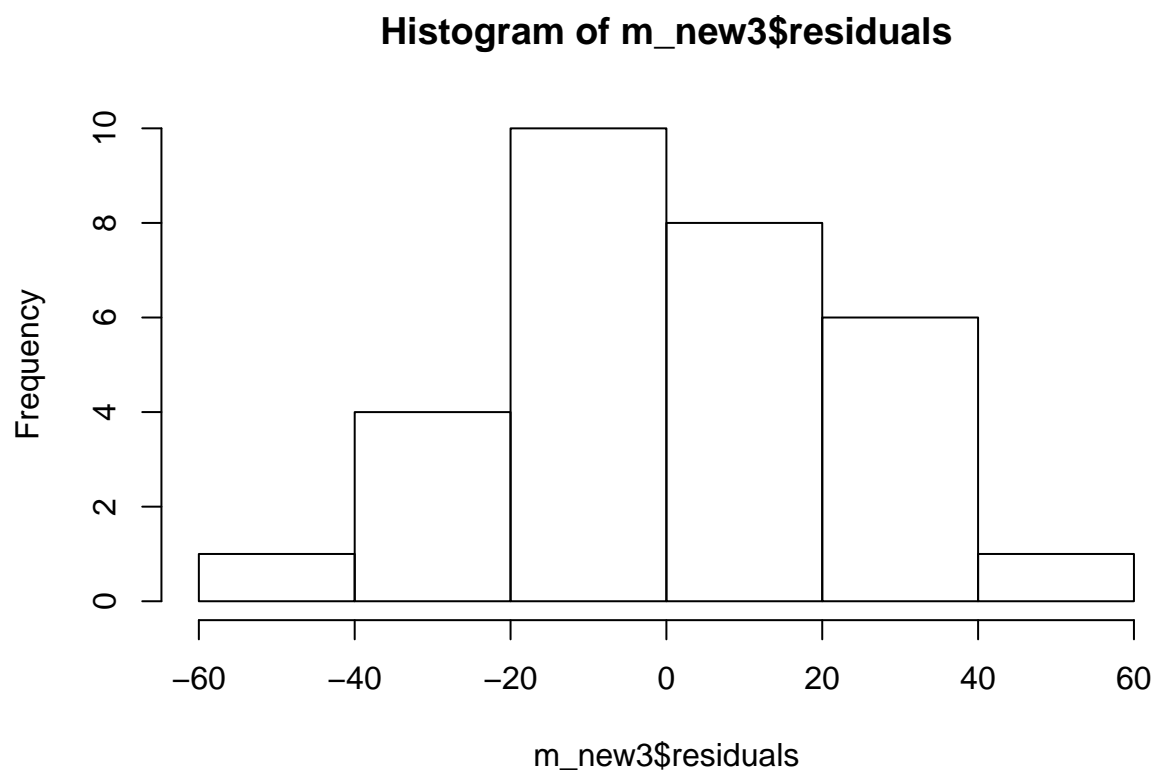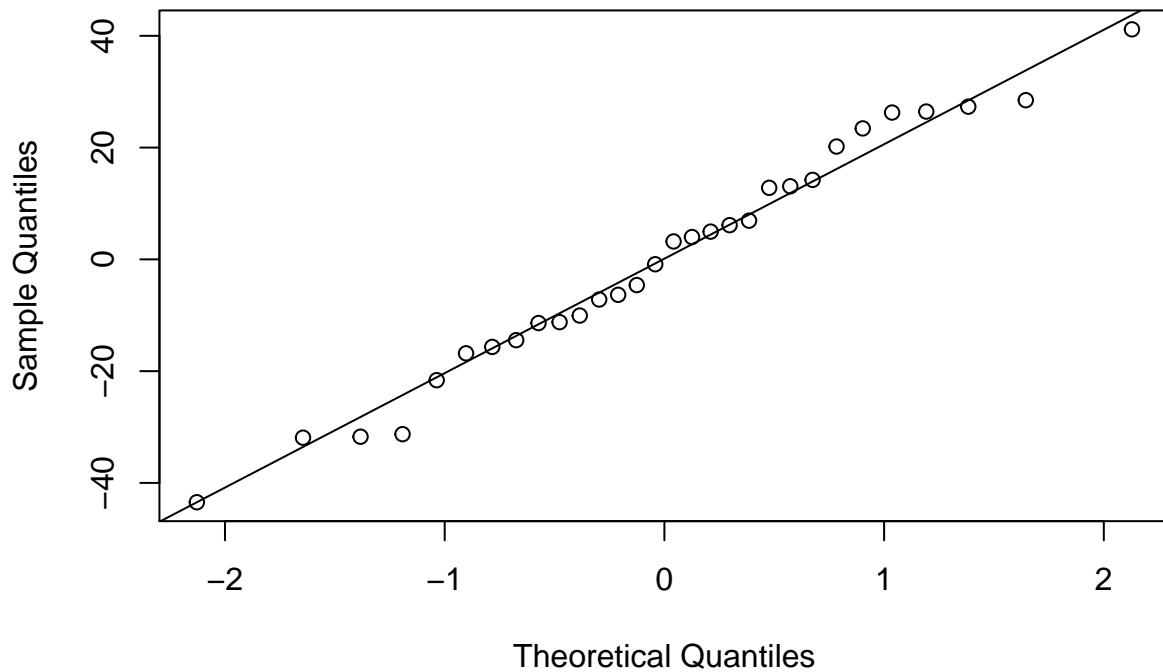

Here's a plot of the residuals for 'new_obs':

```
plot(mlb11$new_obs, m_new3$residuals)
abline(h = 0, lty = 3)
```

And here is the qqplot:

```
qqnorm(m_new3$residuals)
qqline(m_new3$residuals)
```

Normal Q–Q Plot

The linear diagnostics show linearity, nearly normal residuals (seen in the histogram which has a unimodal shape centered around 0), and the qqplot also showing residuals very close to the qqline. Finally, there seems to be constant variability, which is shown in the plot of the residuals above, where there seems to be pretty equal variability of residuals.