Introduction to linear regression

Ilya Kats

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.

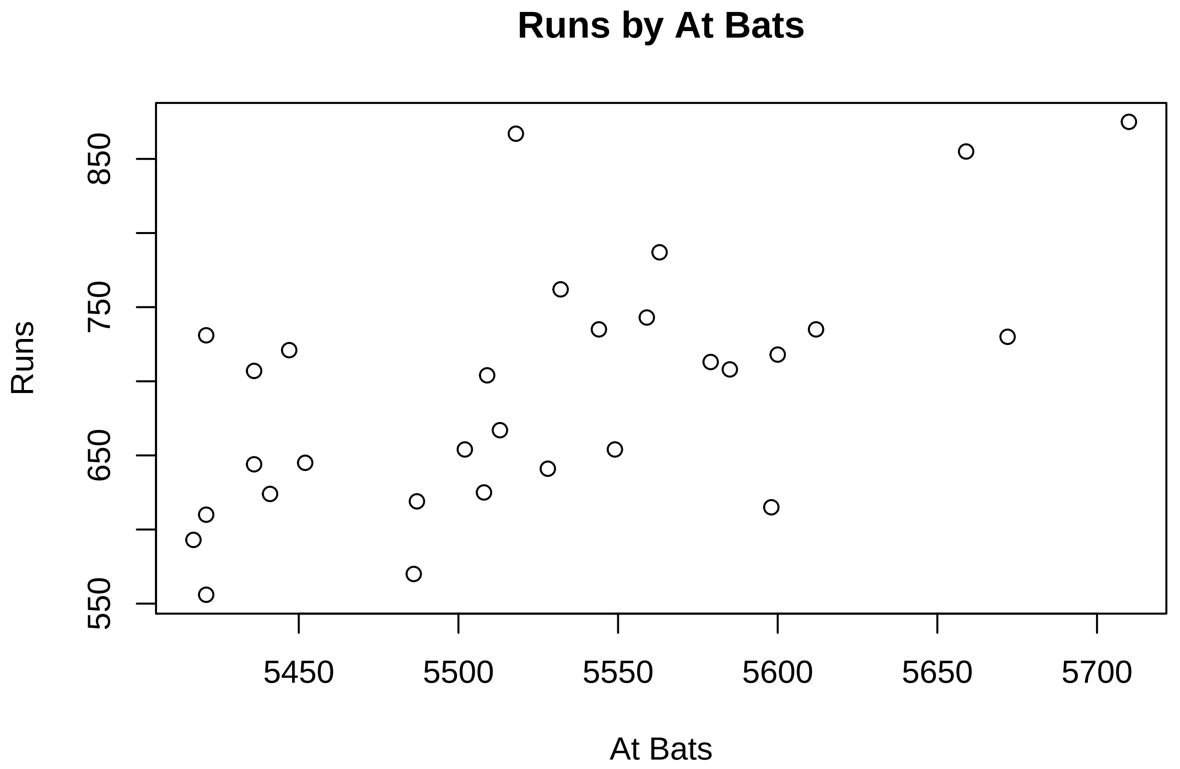
```
load("more/mlb11.RData")
```

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?

We can use a scatterplot to show relationship between `runs` and another numerical variable, for example`at_bats`.

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



Relationship does appear linear and positive. However, I believe it is not very strong as there are a few teams far removed from the possible regression line. I would be curious to see what the linear model predicts for the number of runs, but I would also remain a bit skeptical.

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
```

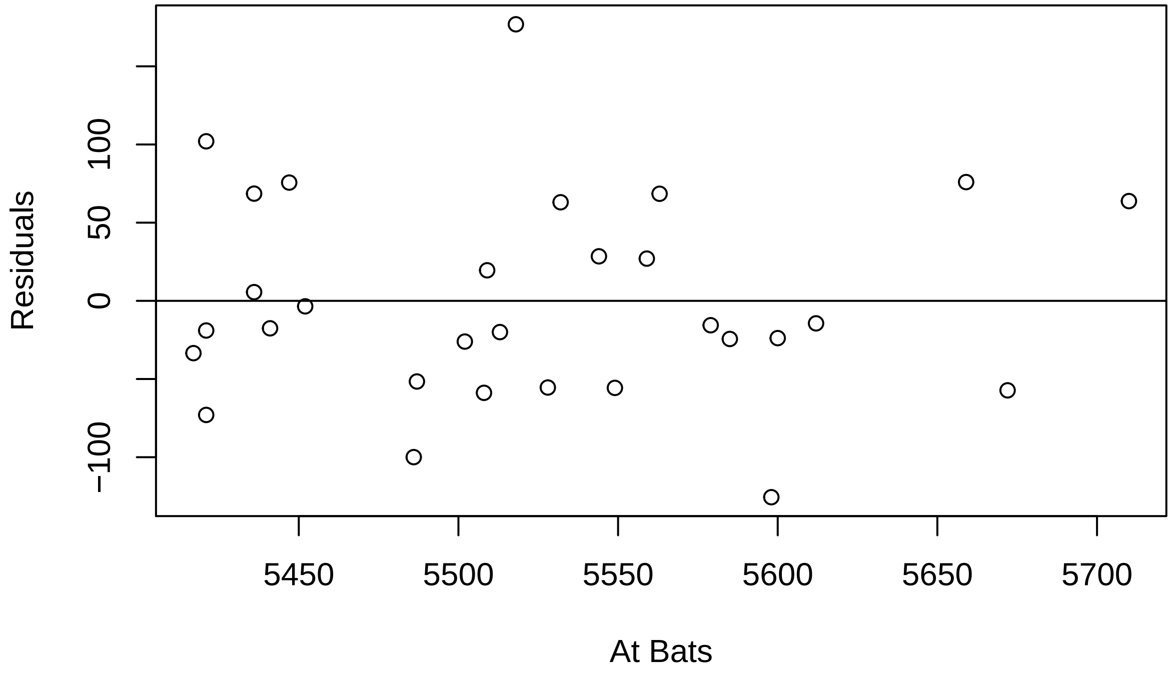
Sum of squared residuals

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.

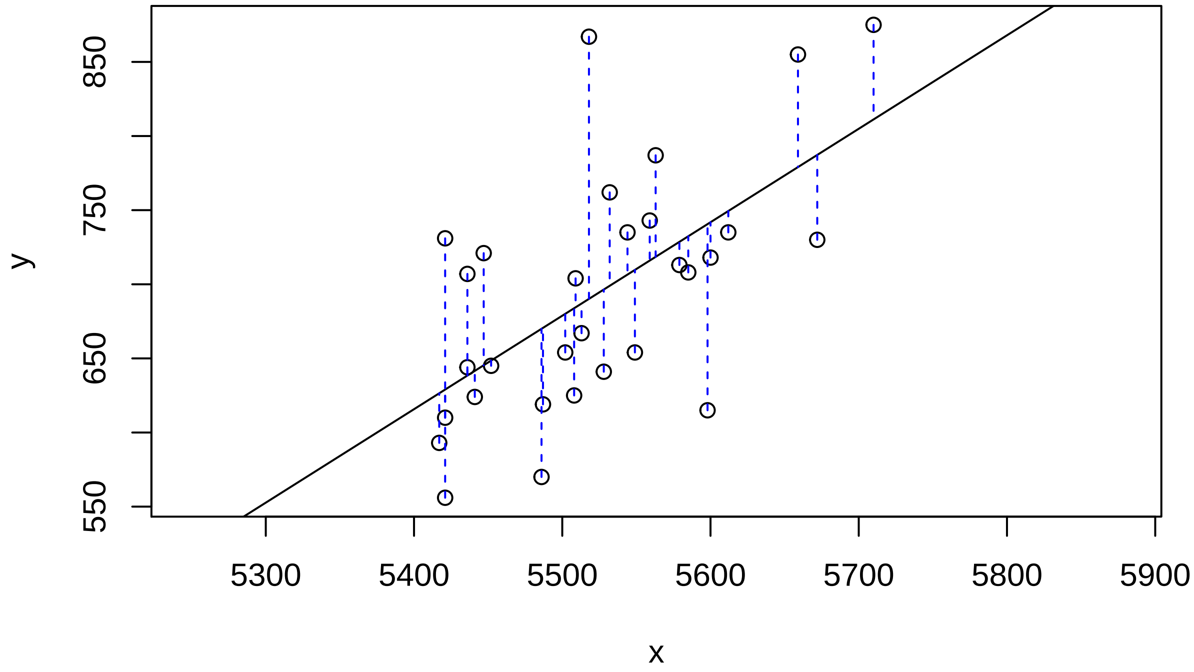
As I mentioned above, the relationship between `runs` and `at_bats` appears to be linear and positive. It has perhaps medium strength. Looking at the residual plot below I can see a few teams far removed from the regression line. A 100, 150 runs is a signifcant difference.

```
runs_lm <- lm(runs ~ at_bats, data=mlb11)
plot(mlb11$at_bats, resid(runs_lm),
     ylab="Residuals", xlab="At Bats", main="")
abline(0, 0)
```



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.

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



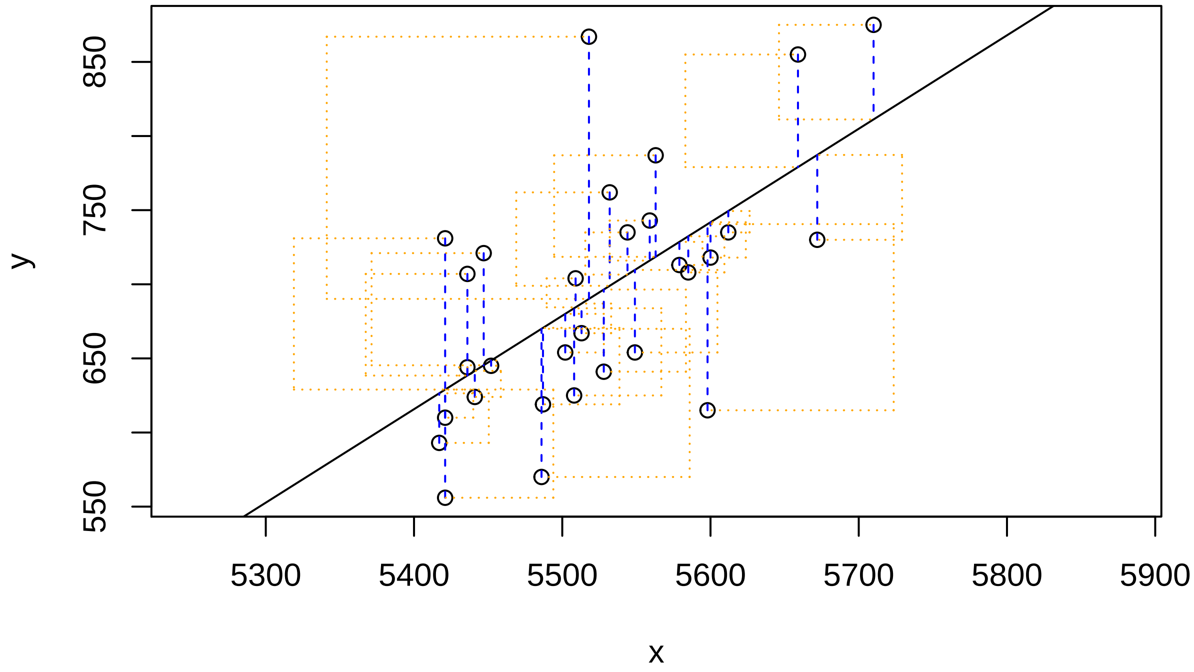
```
## Click two points to make a line.

## 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:

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

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?

I ran the function 10 times. The smallest sum of squares that I got, 131799.7, was on the eighth run.

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:
##               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?

```
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
```

Equation of the regression line:

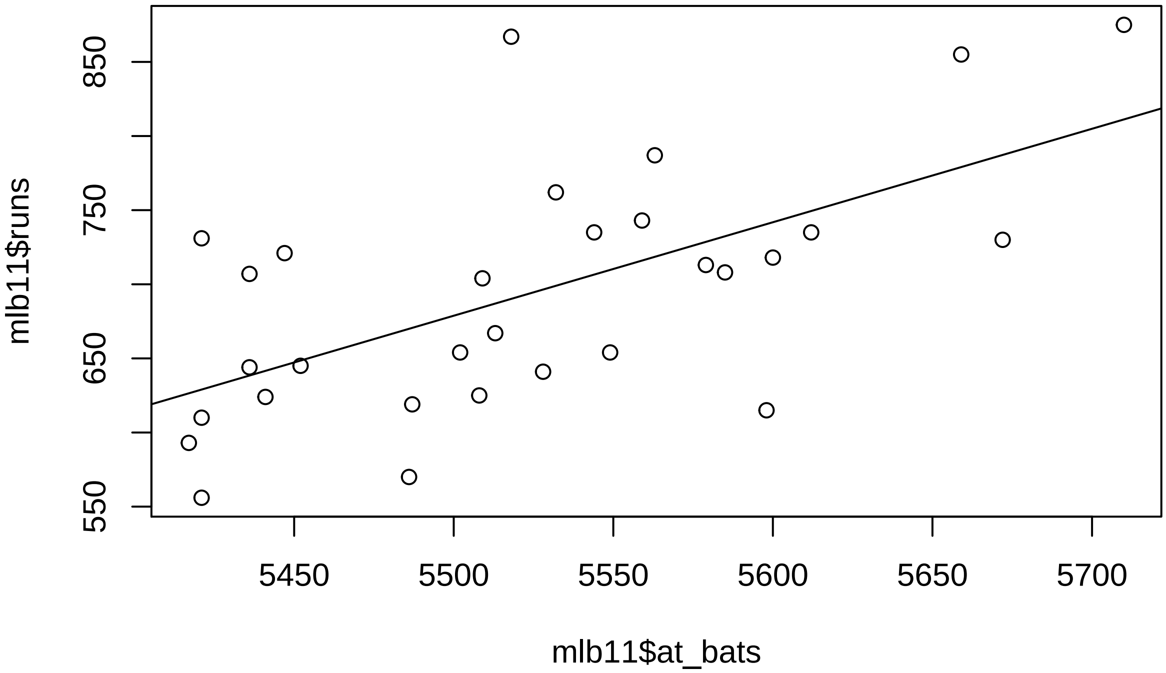
$\widehat{runs} = 415.2389 + 1.8345 * homeruns$

The slope tells us that for every additional homerun the average number of runs the team scores increases by 1.8345.

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?

```
m1$coefficients[1] + m1$coefficients[2] * 5578
```

```
## (Intercept)
##     727.965
```

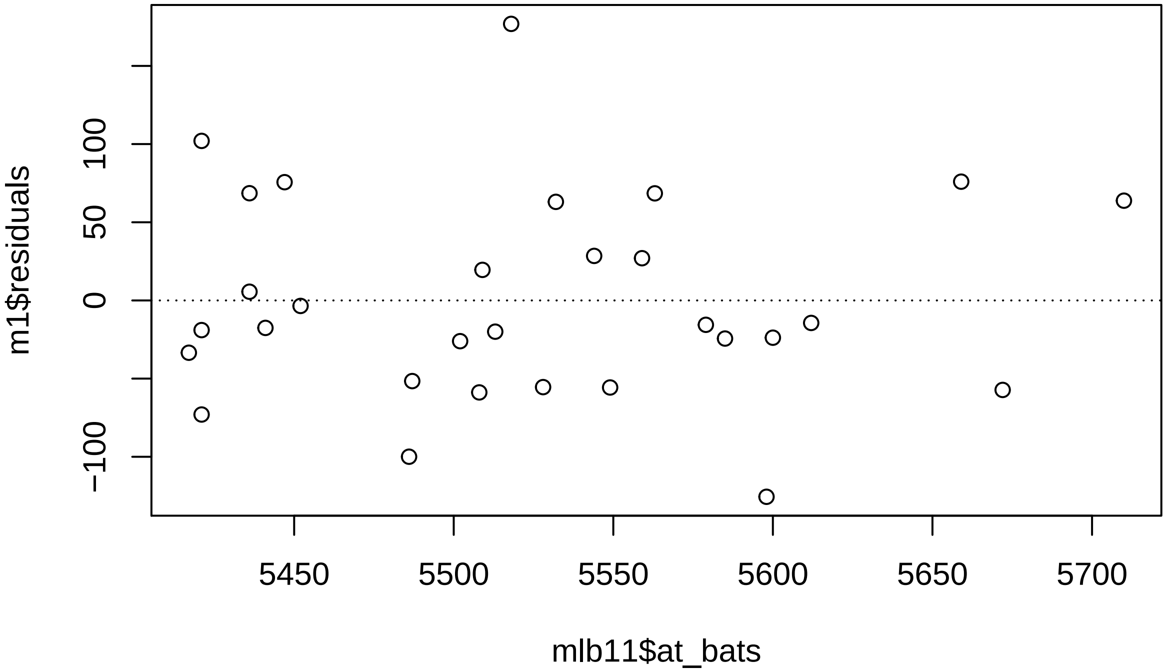
He or she would predict about 728 runs. The residual of this prediction is zero since it lies exactly on the regression line, so it is simply an estimate and not an underestimate or an overestimate.

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
```

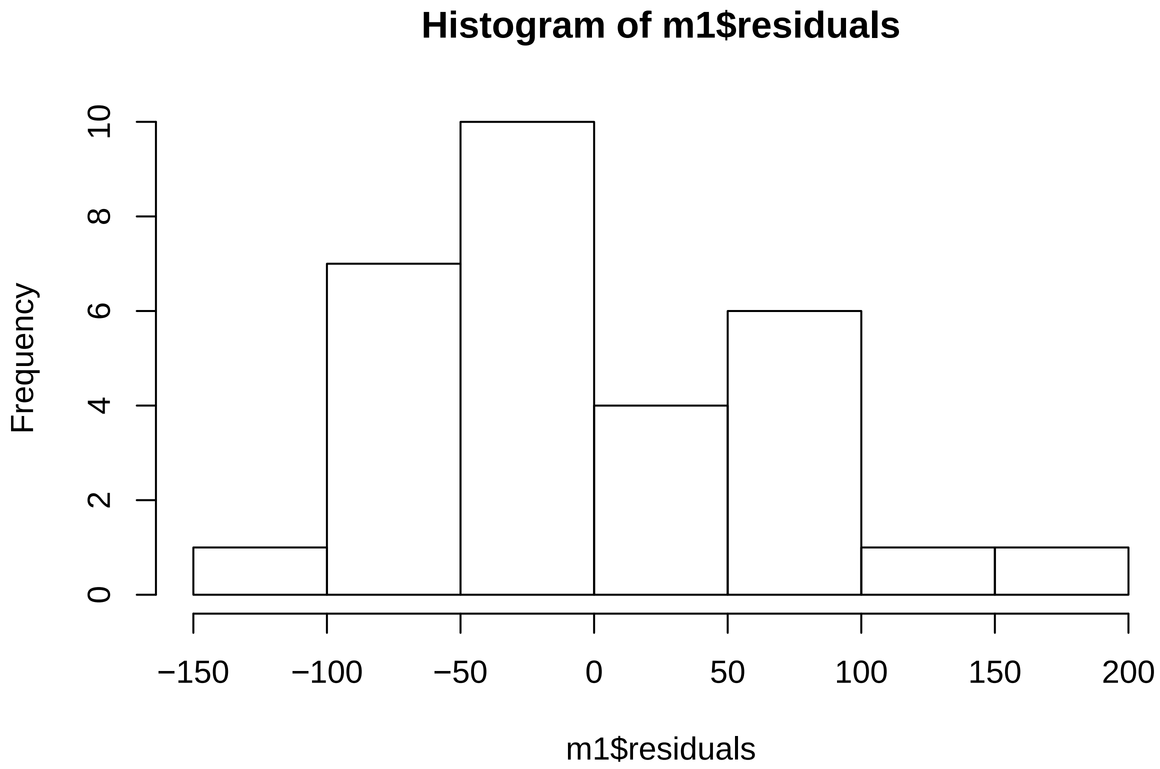


6. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between runs and at-bats?

There is no apparent pattern in the residuals plot, so the relationship appears to be linear.

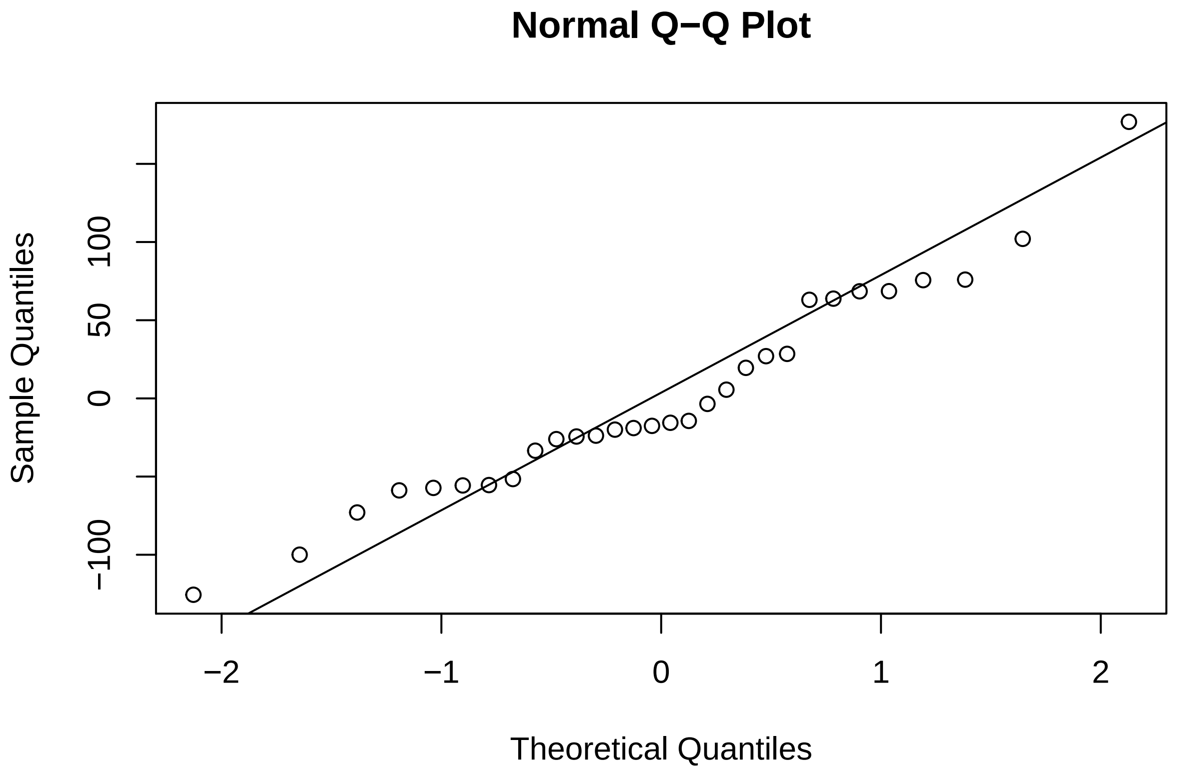
Nearly normal residuals: To check this condition, we can look at a histogram

```
hist(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
```



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

The histogram shows a bit of right-skew and the points do not follow the line perfectly on the normal probability plot; however, both look very close, so I would say that the normal residuals condition is satisfied.

Constant variability:

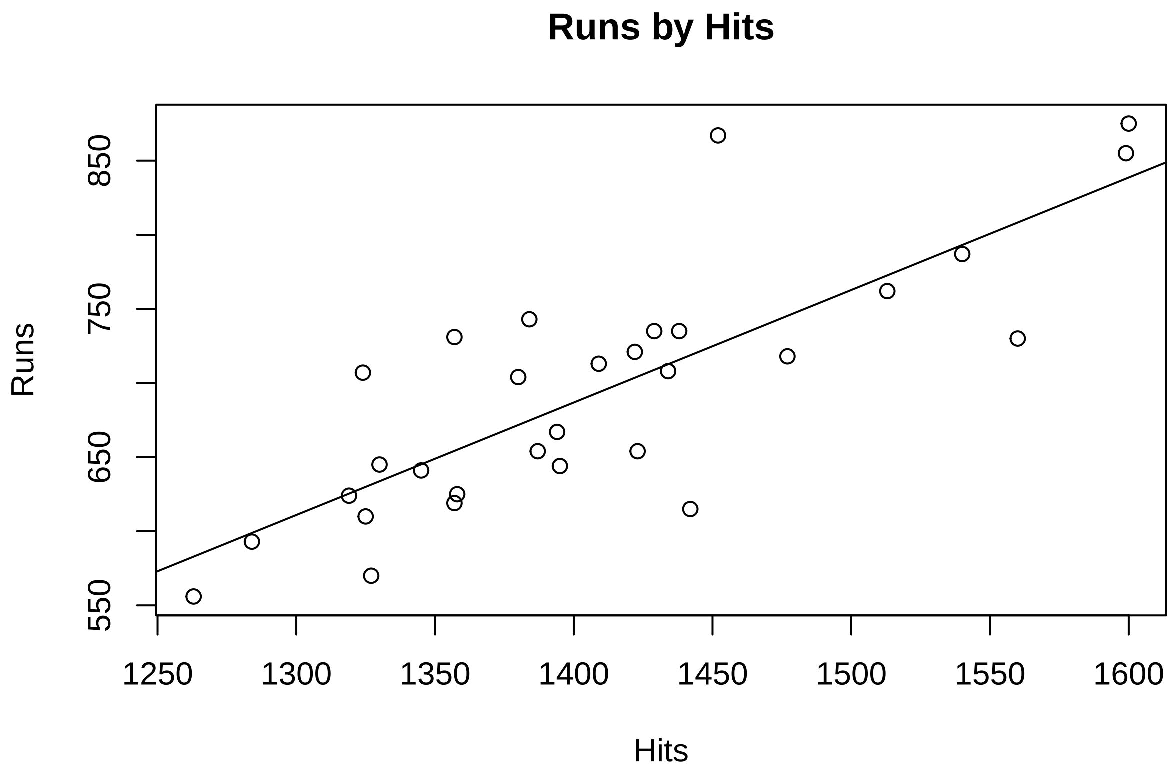
8. Based on the plot in (1), does the constant variability condition appear to be met?

The points are a little sparse on the high end, but it still appears that the constant variability condition is met. We are only dealing with 30 points, so it is a fairly small set.

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?

```
m2 <- lm(runs ~ hits, data = mlb11)
plot(mlb11$hits, mlb11$runs,
     main="Runs by Hits", xlab="Hits", ylab="Runs")
abline(m2)
```



```
summary(m2)
```

```
##
## 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
```

I have chosen hits to try and predict runs. I think based on the scatterplot there is clearly a

linear relationship.

- 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?

For the relationship between `runs` and `at_bats`, the R^2 is 0.3729. For the relationship between `runs` and `hits`, the R^2 is 0.6419. In other words, 64.9% of the variability in runs is explained by hits. `hits` is a better predictor of `runs` than `at_bats`.

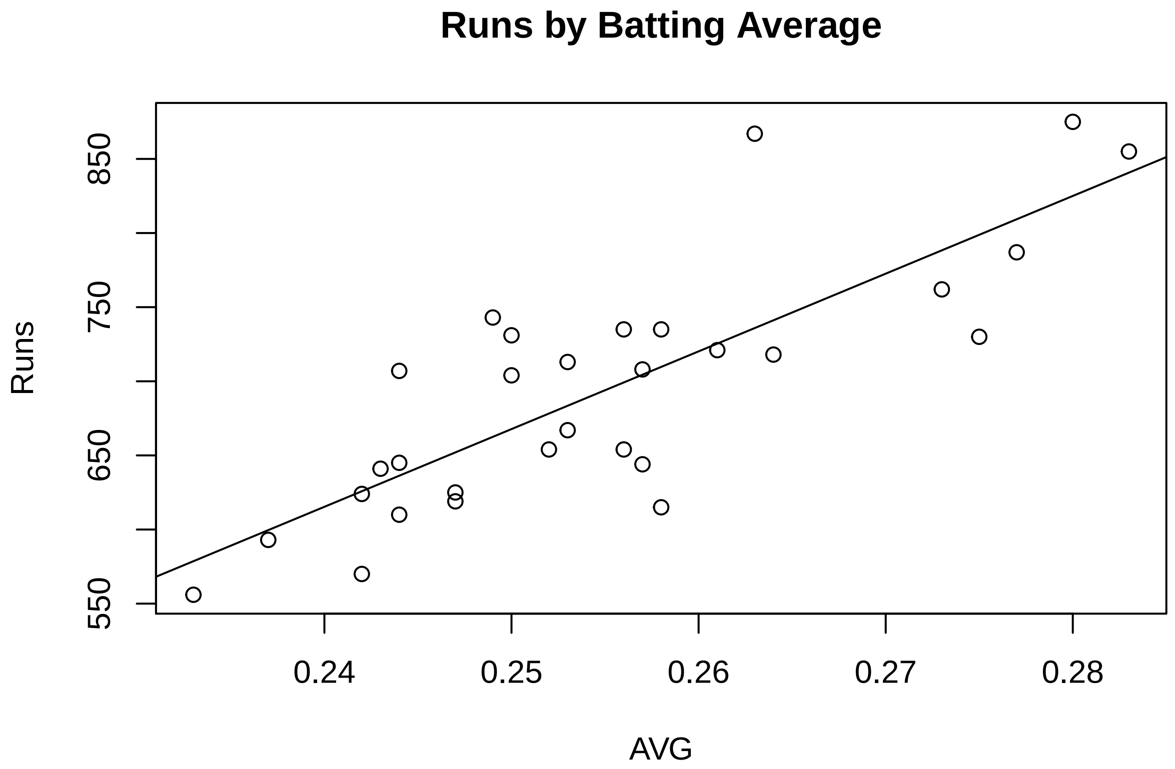
- Now that you can summarize the linear relationship between two variables, investigate the relationships between `runs` and each of the other five traditional variables. Which variable best predicts `runs`? Support your conclusion using the graphical and numerical methods we've discussed (for the sake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

```r
# Get linear models for remaining 5 traditional variables
m3 <- lm(runs ~ homeruns, data = mlb11)
m4 <- lm(runs ~ bat_avg, data = mlb11)
m5 <- lm(runs ~ strikeouts, data = mlb11)
m6 <- lm(runs ~ stolen_bases, data = mlb11)
m7 <- lm(runs ~ wins, data = mlb11)

# Get R-squared for all models
R2 <- data.frame(c(summary(m1)$r.squared,
                   summary(m2)$r.squared,
                   summary(m3)$r.squared,
                   summary(m4)$r.squared,
                   summary(m5)$r.squared,
                   summary(m6)$r.squared,
                   summary(m7)$r.squared))

# Adjust columns and rows for display
colnames(R2) <- c("R2")
rownames(R2) <- c("at_bats", "hits", "homeruns", "bat_avg",
                  "strikeouts", "stolen_bases", "wins")

R2
```

```
##                         R2
## at_bats      0.372865390
## hits         0.641938767
## homeruns     0.626563570
## bat_avg      0.656077135
## strikeouts   0.169357932
## stolen_bases 0.002913993
## wins         0.360971179
```

Based on R^2 values it appears that batting average explains the most variability in runs, so it is the best predictor. Stolen bases is the worst predictor.

```r
plot(mlb11$bat_avg, mlb11$runs,
     main="Runs by Batting Average", xlab="AVG", ylab="Runs")
abline(m4)
```



```
summary(m4)
```

```
##
## 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
```

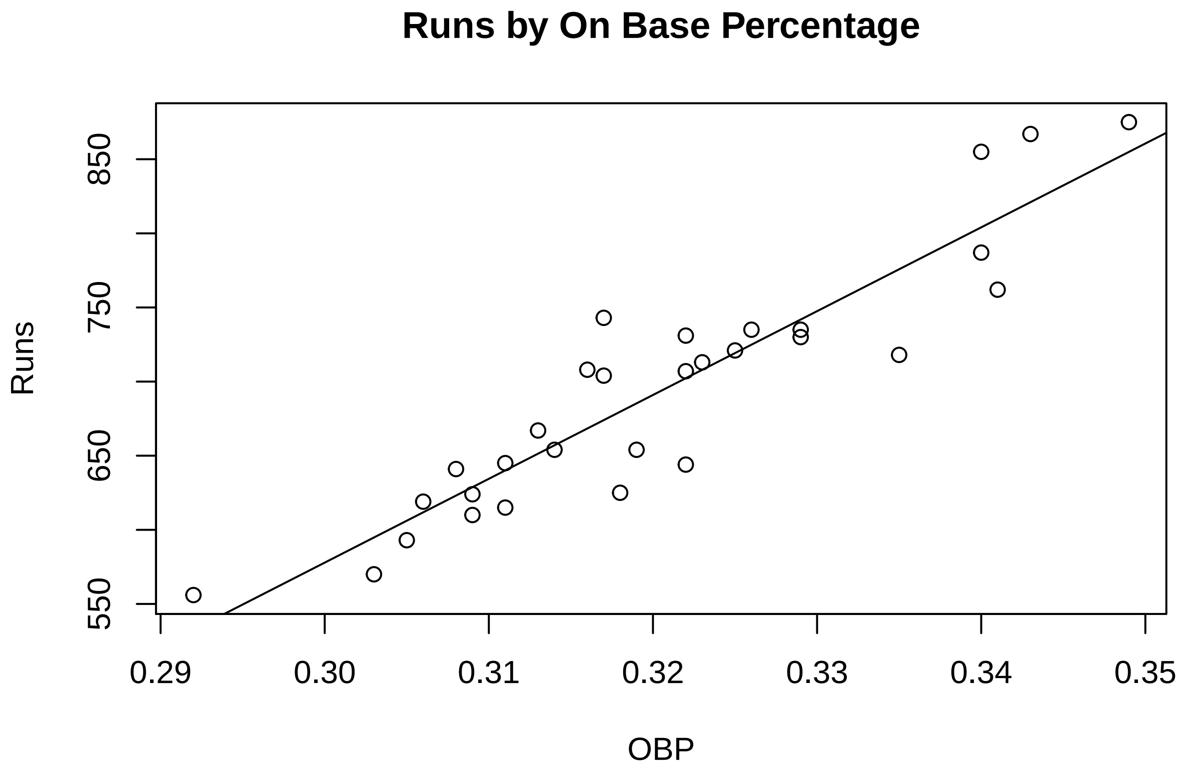
- Now examine the three newer variables. These are the statistics used by the author of *Moneyball* to predict a teams success. In general, are they more or less effective at predicting runs that the old variables? Explain using appropriate graphical and numerical evidence. Of all ten variables we've 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?

```r
# Get linear models for 3 newer variables
m8 <- lm(runs ~ new_onbase, data = mlb11)
m9 <- lm(runs ~ new_slug, data = mlb11)
m10 <- lm(runs ~ new_obs, data = mlb11)

# Get R-squared for all models
R2_new <- data.frame(c(summary(m8)$r.squared,
                       summary(m9)$r.squared,
                       summary(m10)$r.squared))

# Adjust columns and rows for display
colnames(R2_new) <- c("R2")
rownames(R2_new) <- c("onbase", "slug", "obs")

R2_new
```

```
##               R2
## onbase 0.8491053
## slug   0.8968704
## obs    0.9349271
```

All three newer variables are noticeably better at predicting runs than all traditional variables. Consider the worst predictor among newer variables, `new_onbase`.

```r
plot(mlb11$new_onbase, mlb11$runs,
     main="Runs by On Base Percentage", xlab="OBP", ylab="Runs")
abline(m8)
```

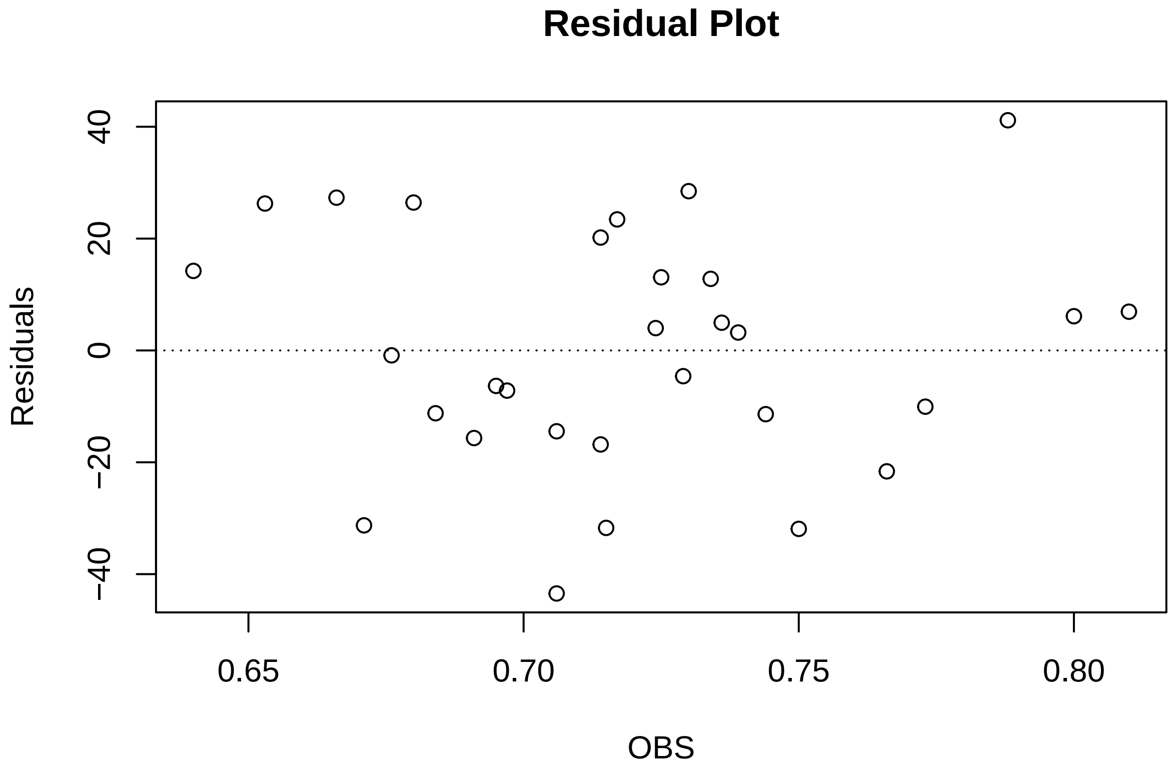


The scatterplot displays noticeably stronger relationship than the most effective traditional predictor, `bat_avg`. OBP explains 84.9% of variability in runs, while BA explains only 65.6%. It seems to make a lot of sense that getting on base or getting extra bases is a better predictor of number of runs. A batter that gets hits, but also strikes out a lot will not score as many runs as a batter that gets on base by any means possible and advances as much as possible. I am a bit surprised that the relationship between wins and runs is not stronger, but I guess in baseball you can score a lot of runs and still lose (or you can win with a low scoring game).

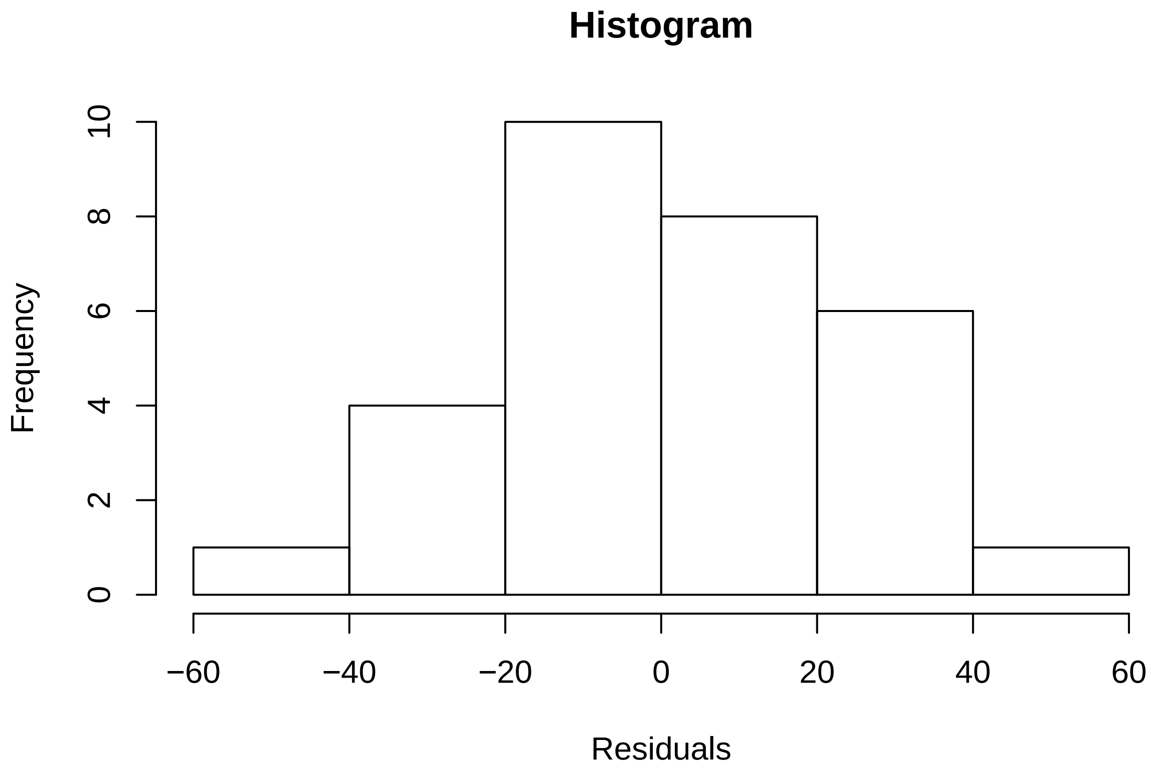
- Check the model diagnostics for the regression model with the variable you decided was the best predictor for runs.

Consider `new_obs` and graph residual plot, histogram of residuals and normal probability plot of residuals.

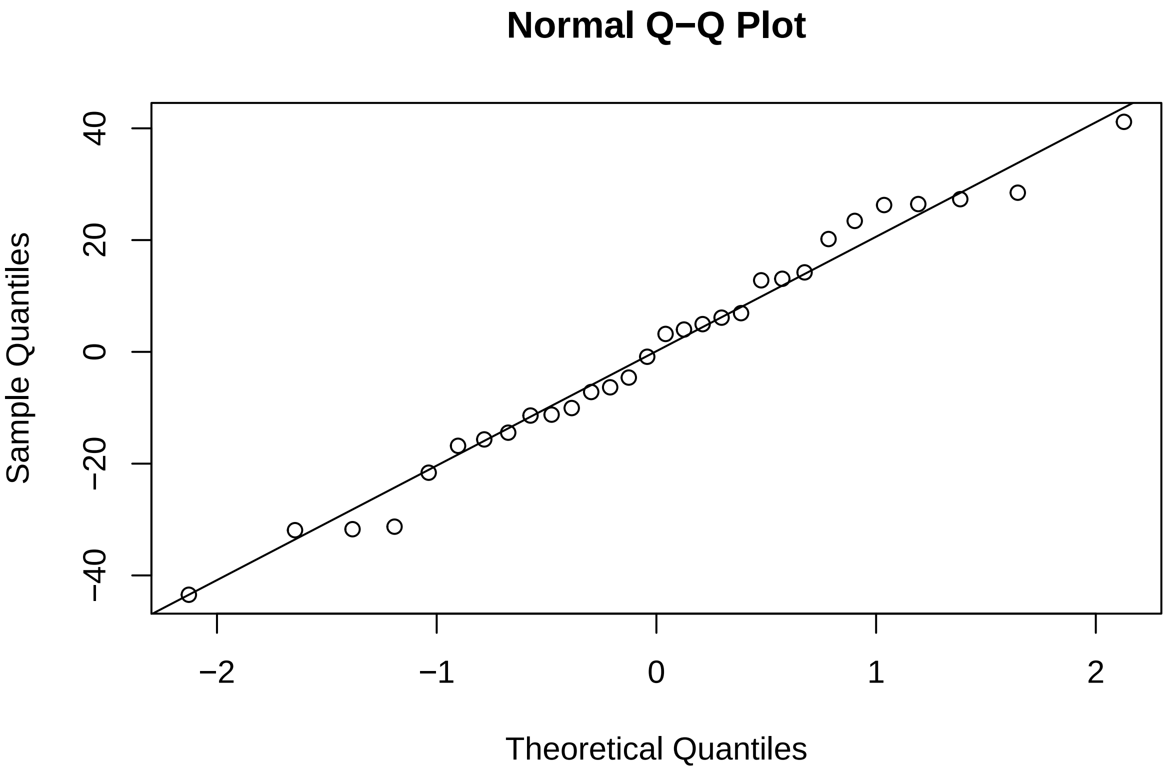
```r
plot(m10$residuals ~ mlb11$new_obs,
     main = "Residual Plot", xlab = "OBS", ylab = "Residuals")
abline(h = 0, lty = 3)
```



```r
hist(m10$residuals, main = "Histogram", xlab = "Residuals")
```



```
qqnorm(m10$residuals)
qqline(m10$residuals)
```



Based on the residual plot, there is no noticeable pattern and variability appears to be constant. Based on the histogram and normal probability plot, residuals appear to be nearly normal. I think the linear model for relationship between runs and OBS can be considered reliable.