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, betterpredict 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

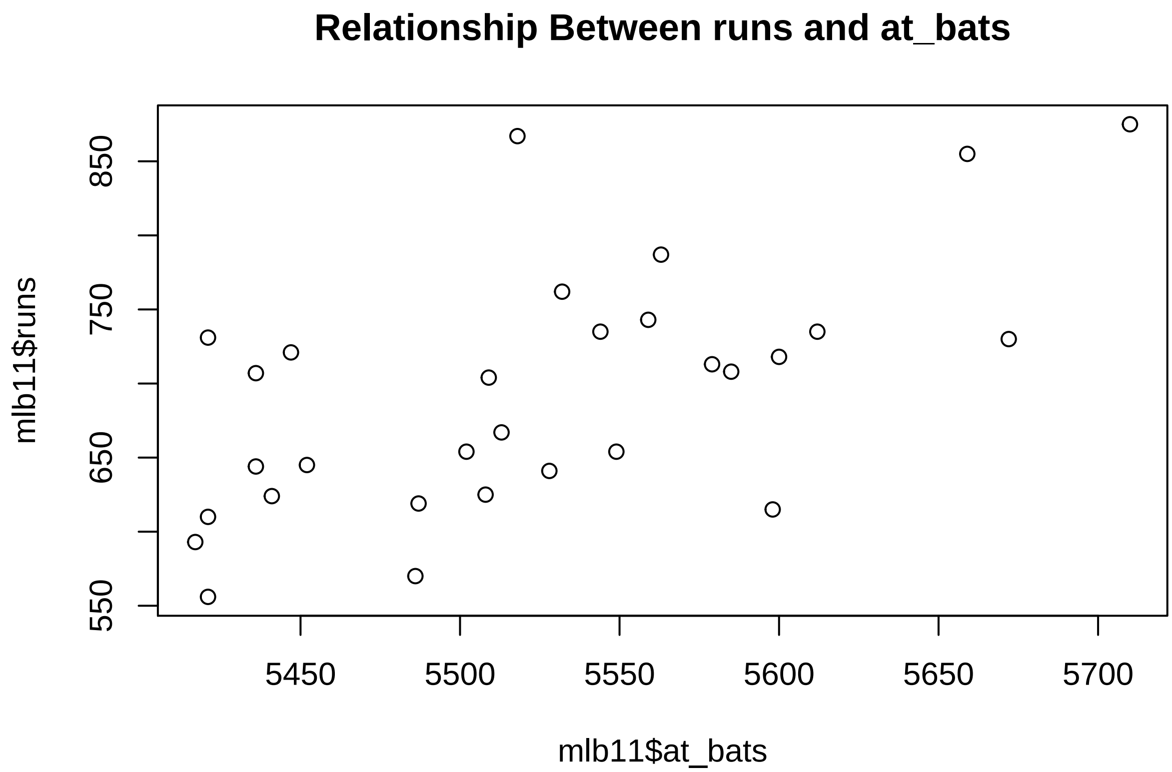
Let's load up the data for the 2011 season.

```
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?
   Answer:

```
plot(mlb11$runs ~ mlb11$at_bats, main = "Relationship Between runs and at_bats")
```



I would use scatterplot to display relationship between runs and at_bats. The relationship is positive but only moderately strong. I will not be very comfortable using a linear model to predict 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
```

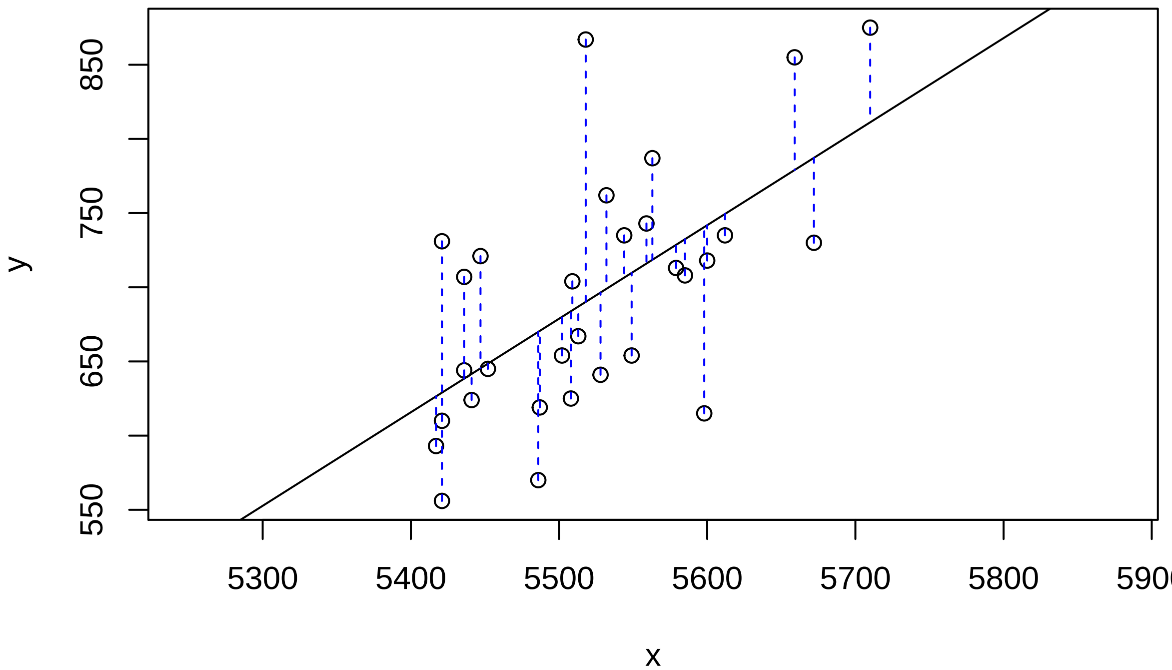
Sum of squared residuals

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.
   Answer:
   Linear relationship is positive trend and the residual distribution looks nomal with constant variability.

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)
```



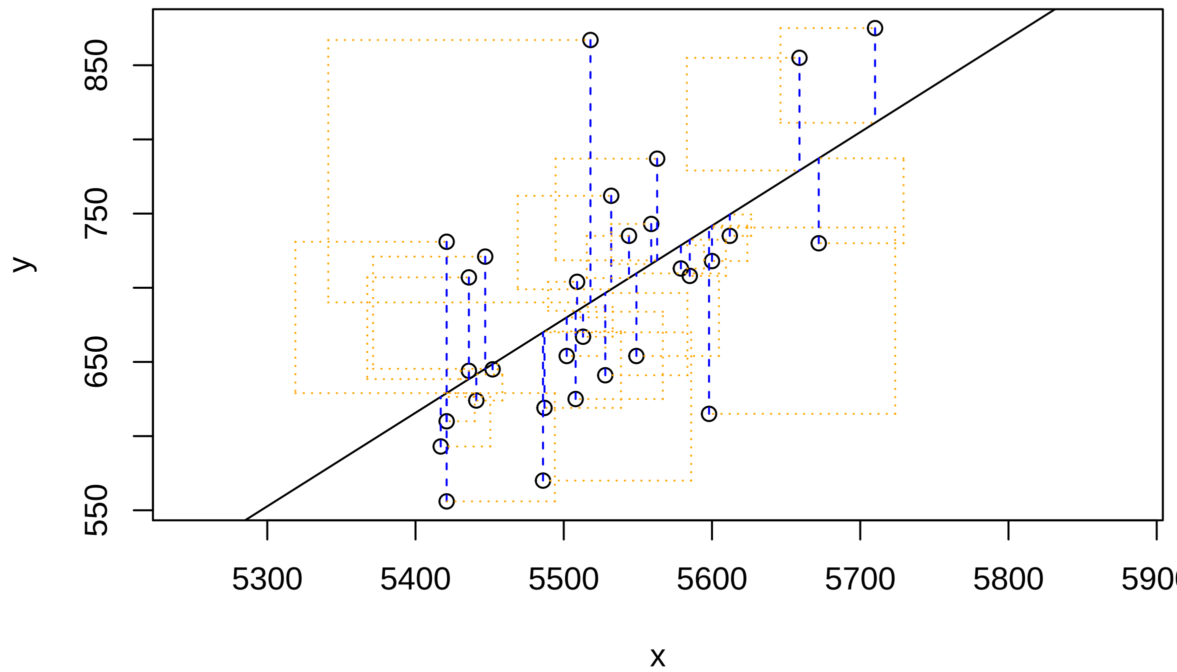

```
## Click two points to make a line.

## 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`.

```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
```

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?
   Answer: I ran the plot using plot_ss 5 times and the best result for the sum of squares i got was 127,559. I can compare the result with the R generated sum of squares which is not too terribly far apart.

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.

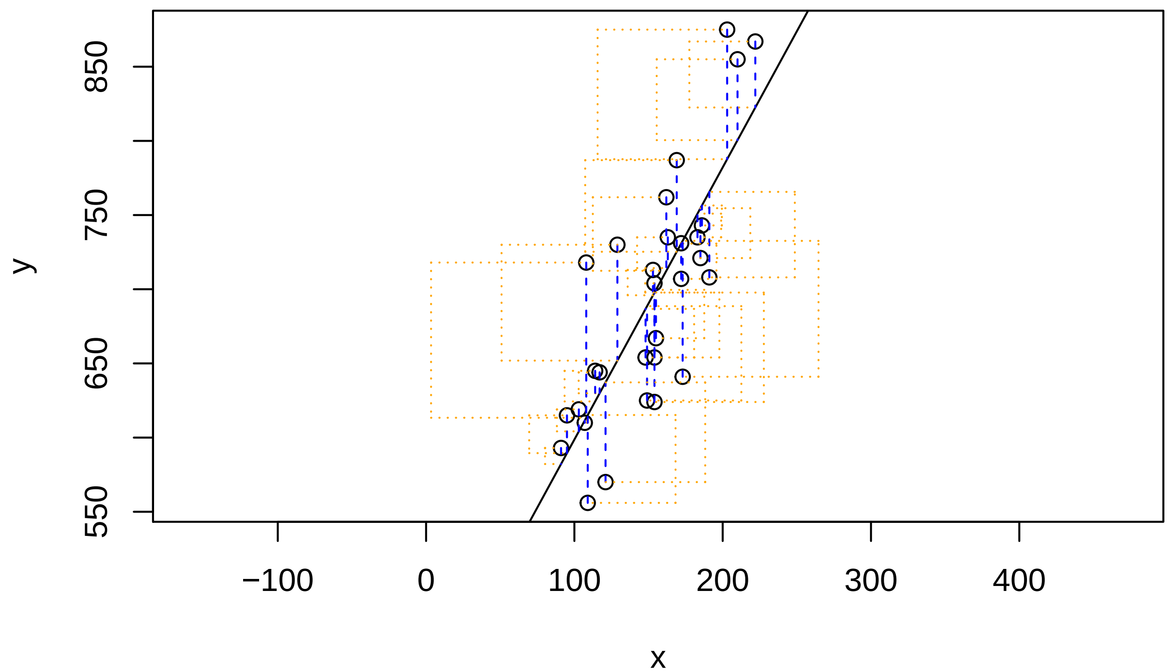
4. Fit a new model that uses `homeruns` to predict `runs`. Using the estimates from the R output, write the equation of the regression line. What does the slope tell us in the context of the relationship between success of a team and its home runs?

Answer:

```r
cor(mlb11$runs, mlb11$homeruns)
```

```
## [1] 0.7915577
```

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




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

## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##     415.239        1.835
##
## Sum of Squares:  73671.99
```

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

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

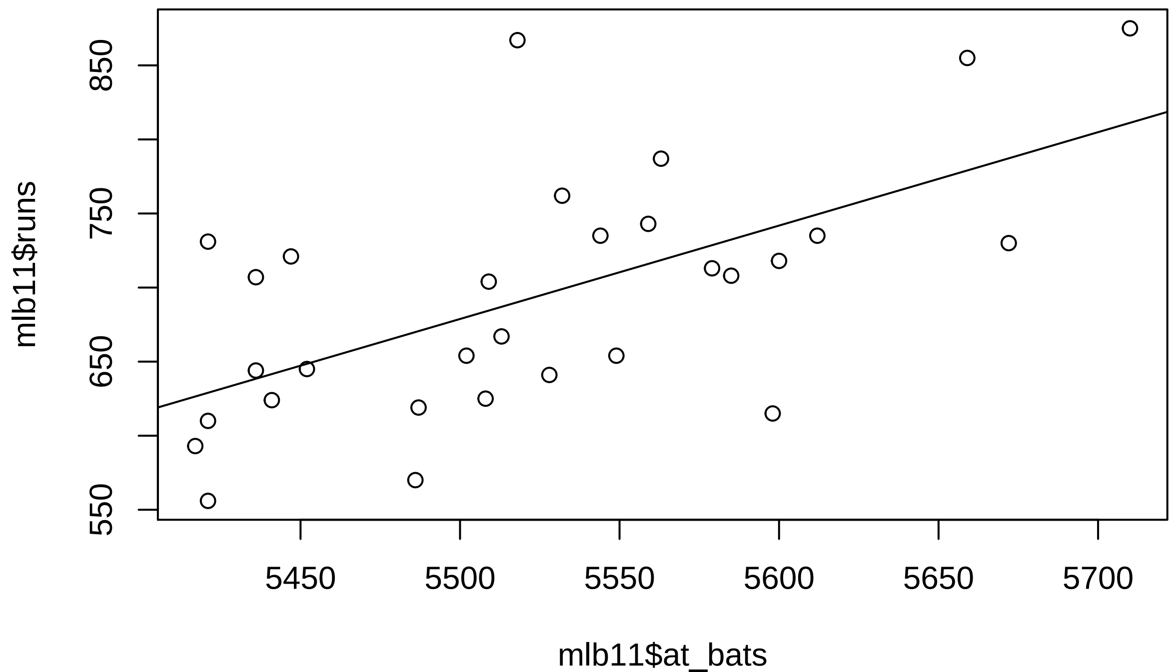
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -91.615 -33.410   3.231  24.292 104.631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 415.2389    41.6779   9.963 1.04e-10 ***
## homeruns      1.8345     0.2677   6.854 1.90e-07 ***
## ---
##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

In term of the relationship between success of a team and it home run, it seems that for every home run a team has the average number of total runs will also increase by 1.83. This is a positive relationship with a correlation coefficient of 0.7916, which is relatively strong.

Prediction and prediction errors

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

```r
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?
   Answer:
   Based on the formula for least squares regression line for the linear model below the estimated runs for a team with 5578 at_bats are 730.5. Looking at the actual observed data there is no team with 5578 at_bats, but Philadelphia Phillies has a at_bats of 5,579 with 713 runs. Using these two numbers we can see that the model overestimated the runs by 730.5 - 713 = 17.5.

```
b0 <- -2789.243
b1 <- 0.631
x <- 5578
Yhat <- b0 + b1*x
Yhat
```

```
## [1] 730.475
```

```
mlb11[order(mlb11$runs,mlb11$at_bats),]
```

```
##                       team runs at_bats hits homeruns bat_avg strikeouts
## 30      Seattle Mariners  556    5421 1263      109   0.233       1280
## 28  San Francisco Giants  570    5486 1327      121   0.242       1122
## 29      San Diego Padres  593    5417 1284       91   0.237       1320
## 23    Pittsburgh Pirates  610    5421 1325      107   0.244       1308
## 10        Houston Astros  615    5598 1442       95   0.258       1164
## 21       Minnesota Twins  619    5487 1357      103   0.247       1048
## 27  Washington Nationals  624    5441 1319      154   0.242       1323
## 22       Florida Marlins  625    5508 1358      149   0.247       1244
## 26        Atlanta Braves  641    5528 1345      173   0.243       1260
## 12   Los Angeles Dodgers  644    5436 1395      117   0.257       1087
## 24     Oakland Athletics  645    5452 1330      114   0.244       1094
## 17     Chicago White Sox  654    5502 1387      154   0.252        989
## 13          Chicago Cubs  654    5549 1423      148   0.256       1202
## 15    Los Angeles Angels  667    5513 1394      155   0.253       1086
## 18     Cleveland Indians  704    5509 1380      154   0.250       1269
## 25        Tampa Bay Rays  707    5436 1324      172   0.244       1193
## 11     Baltimore Orioles  708    5585 1434      191   0.257       1120
## 16 Philadelphia Phillies  713    5579 1409      153   0.253       1024
## 6          New York Mets  718    5600 1477      108   0.264       1085
## 8      Milwaukee Brewers  721    5447 1422      185   0.261       1083
## 4     Kansas City Royals  730    5672 1560      129   0.275       1006
## 19  Arizona Diamondbacks  731    5421 1357      172   0.250       1249
## 9       Colorado Rockies  735    5544 1429      163   0.258       1201
## 14       Cincinnati Reds  735    5612 1438      183   0.256       1250
## 20     Toronto Blue Jays  743    5559 1384      186   0.249       1184
## 5    St. Louis Cardinals  762    5532 1513      162   0.273        978
## 3         Detroit Tigers  787    5563 1540      169   0.277       1143
## 1          Texas Rangers  855    5659 1599      210   0.283        930
## 7       New York Yankees  867    5518 1452      222   0.263       1138
## 2         Boston Red Sox  875    5710 1600      203   0.280       1108
##    stolen_bases wins new_onbase new_slug new_obs
## 30          125   67      0.292    0.348   0.640
## 28           85   86      0.303    0.368   0.671
## 29          170   71      0.305    0.349   0.653
## 23          108   72      0.309    0.368   0.676
## 10          118   56      0.311    0.374   0.684
## 21           92   63      0.306    0.360   0.666
## 27          106   80      0.309    0.383   0.691
## 22           95   72      0.318    0.388   0.706
## 26           77   89      0.308    0.387   0.695
## 12          126   82      0.322    0.375   0.697
## 24          117   74      0.311    0.369   0.680
## 17           81   79      0.319    0.388   0.706
## 13           69   71      0.314    0.401   0.715
## 15          135   86      0.313    0.402   0.714
## 18           89   80      0.317    0.396   0.714
## 25          155   91      0.322    0.402   0.724
## 11           81   69      0.316    0.413   0.729
## 16           96  102      0.323    0.395   0.717
## 6           130   77      0.335    0.391   0.725
## 8            94   96      0.325    0.425   0.750
## 4           153   71      0.329    0.415   0.744
## 19          133   94      0.322    0.413   0.736
```

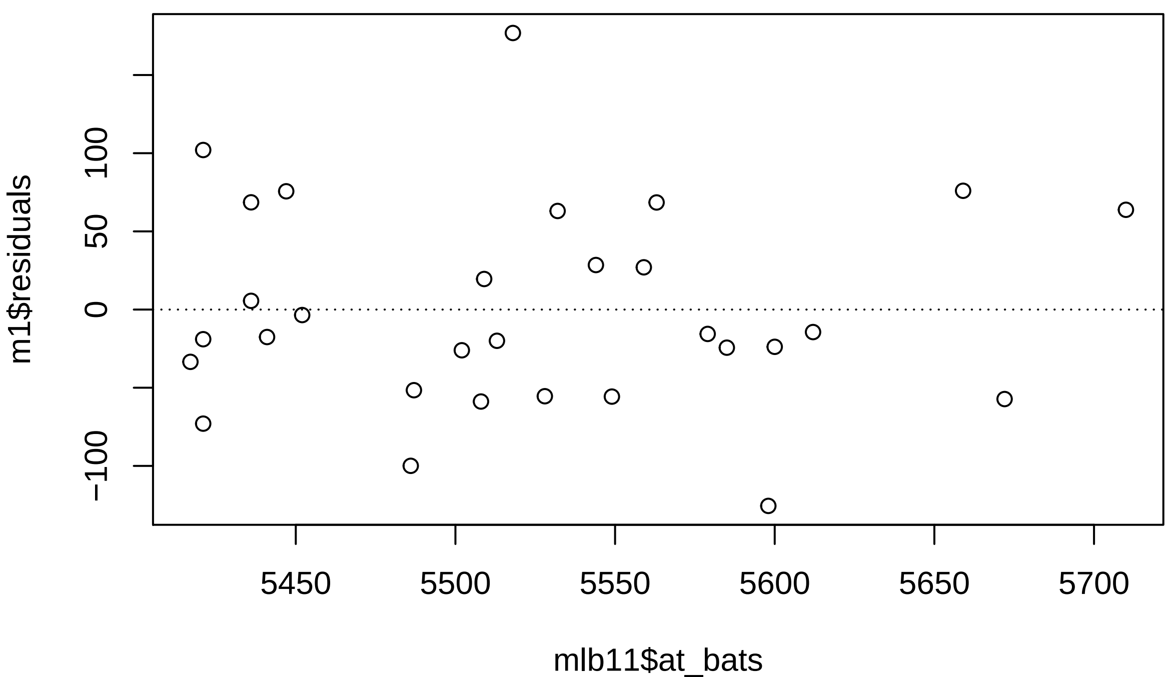
```
## 9           118   73      0.329    0.410   0.739
## 14           97   79      0.326    0.408   0.734
## 20          131   81      0.317    0.413   0.730
## 5            57   90      0.341    0.425   0.766
## 3            49   95      0.340    0.434   0.773
## 1           143   96      0.340    0.460   0.800
## 7           147   97      0.343    0.444   0.788
## 2           102   90      0.349    0.461   0.810
```

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.

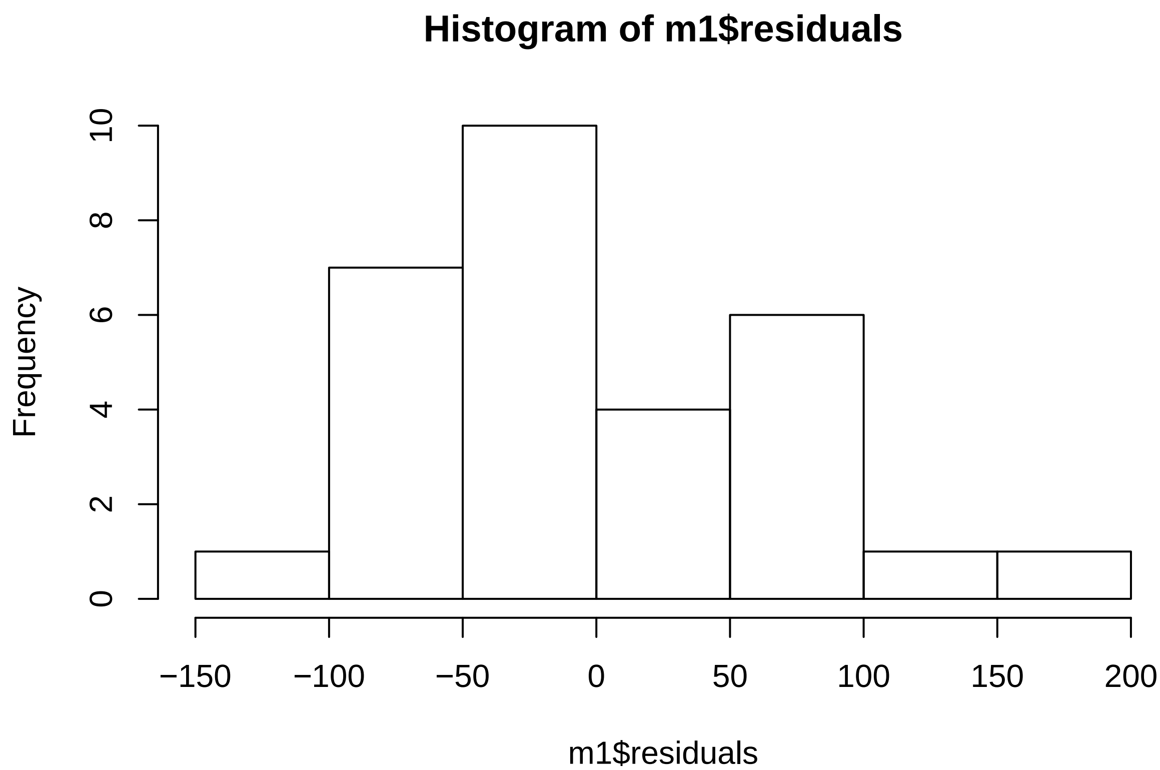
```
plot(m1$residuals ~ mlb11$at_bats)
abline(h = 0, lty = 3)  # adds a horizontal dashed line at y = 0
```



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?
   Answwer: The residuals show no obvious patterns and appear to be scattered randomly around the dashed line that represents 0. I would say that the relationship is linear.

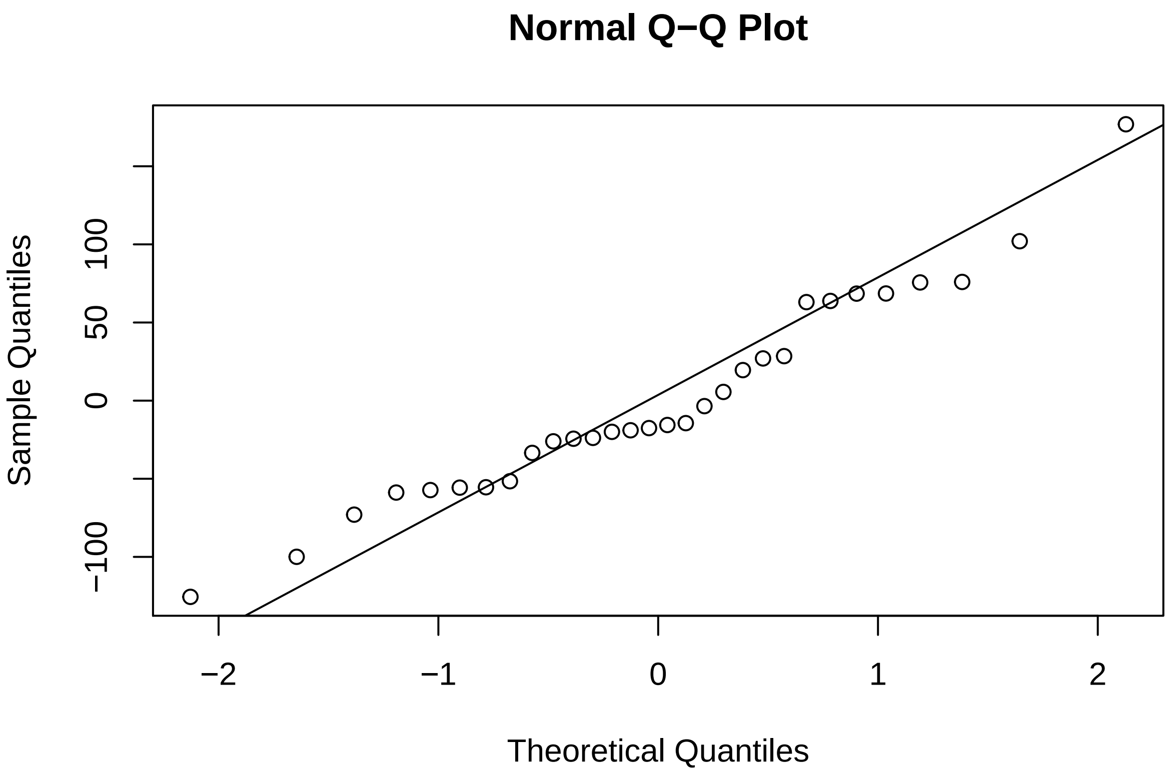
Nearly normal residuals: To check this condition, we can look at a histogram

```r
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?
   Answer: It looks nearly normal.

Constant variability:

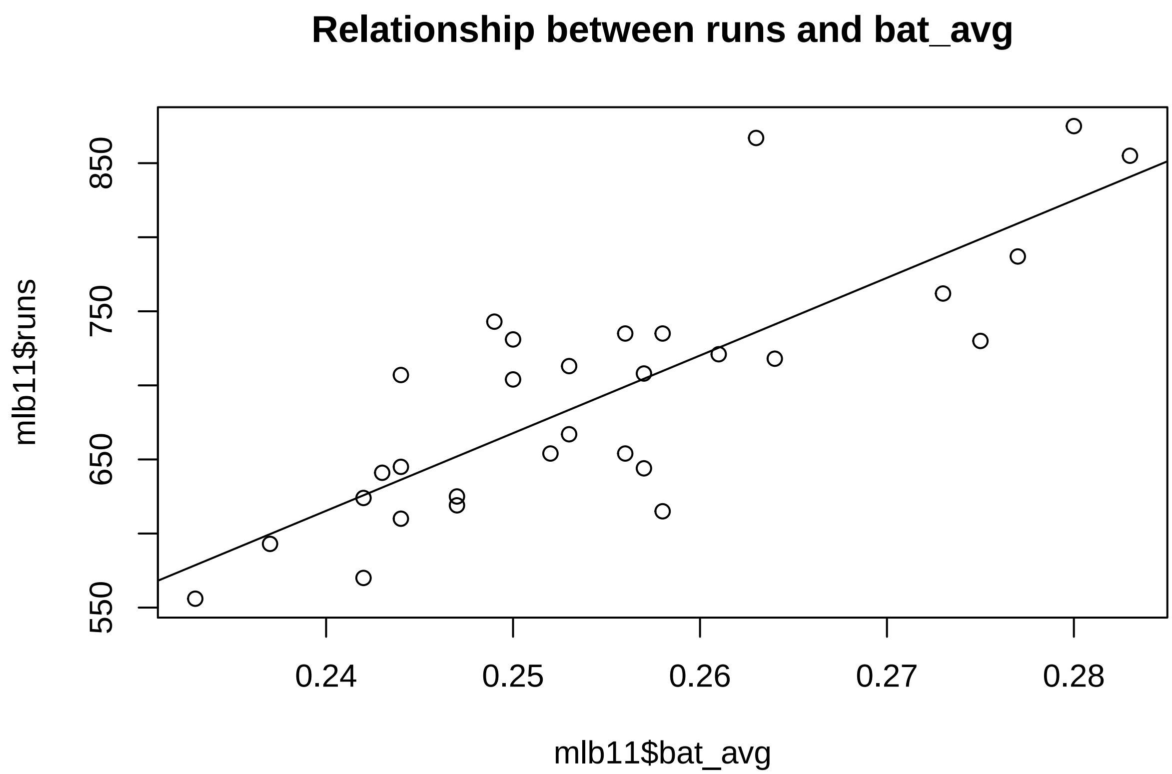
8. Based on the plot in (1), does the constant variability condition appear to be met?
   Answer: Based on the plots we did, it looks to me this condition has been met.

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?
  Answer:
  Since we already looked at the relationship between runs and homeruns and runs and at_bat I chose runs and bat_avg to see if it is a good predictor. From the plot and summary statistics below it looks to me that the two variables fit a liner model. Also, for this model, 65.6% of the variability in runs is explained by bat-avg.

y = b0 + b1X = -642.8+5242.2*bat_avg

```r
m3 <- lm(runs ~ bat_avg, data = mlb11)
plot(mlb11$runs ~ mlb11$bat_avg, main = "Relationship between runs and bat_avg")
abline(m3)
```



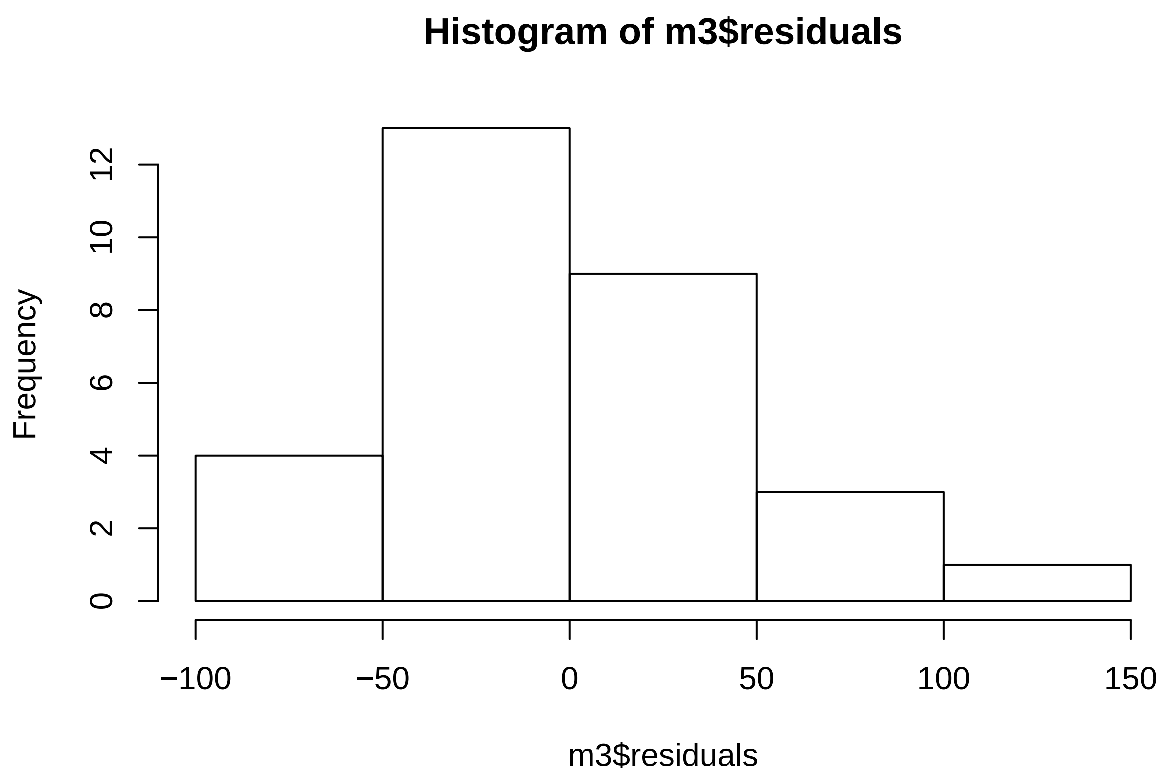

```r
summary(m3)
```

```
##
## 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
```

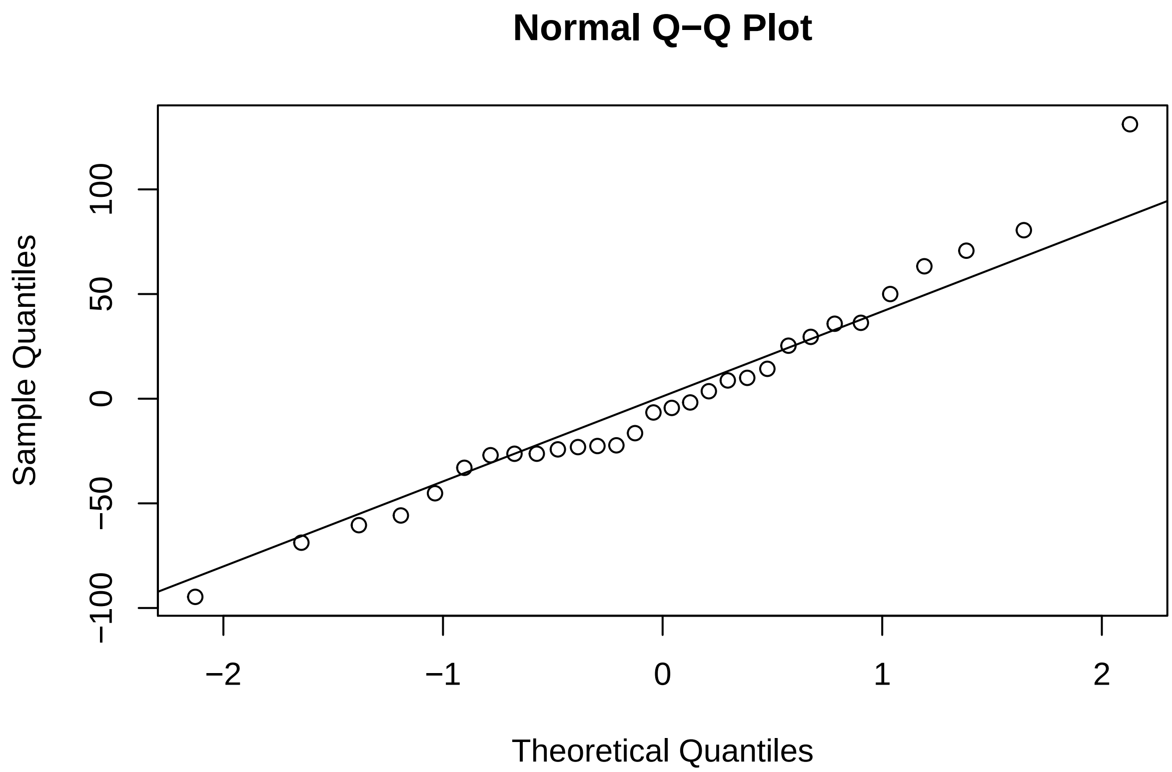
- How does this relationship compare to the relationship between `runs` and `at_bats`? Use the R^2 values from the two model summaries to compare. Does your variable seem to predict `runs` better than `at_bats`? How can you tell?
  Answer: R2 measure of how close the data are to least squares line. 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. comparing the R2 data for runs and at-bats and runs and bat_avg it seems that the latter predict runs better because the R2 for bat_avg is 0.6561 vs. 0.3729 forat_abts. This indicates that 65.61% of variability can be explained by the model.

- 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).
  Answer: after running summary statistics for all other traditional variables it turns out that the best variable to predict the runs is bat_avg. It has the highest r2 value.

```
m3 <- lm(runs ~ bat_avg, data = mlb11)
hist(m3$residuals)
```




```
qqnorm(m3$residuals)
qqline(m3$residuals) # adds diagonal line to the normal prob plot
```



- 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?
Answer: If I don't know anything about baseball but only have the following summary statistics to predict which new variable is the most effective at predicting run I would pick new_obs. The R-squared for new_obs is at a high 93.5%.

```
names(mlb11)
```

```
##  [1] "team"         "runs"         "at_bats"      "hits"
##  [5] "homeruns"     "bat_avg"      "strikeouts"   "stolen_bases"
##  [9] "wins"         "new_onbase"   "new_slug"     "new_obs"
```

```
model_new_obs <- lm(runs ~ new_obs, data = mlb11)
model_new_slug <- lm(runs ~ new_slug, data = mlb11)
model_new_onbase <- lm(runs ~ new_onbase, data = mlb11)
summary(model_new_obs)
```

```
##
## 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
```

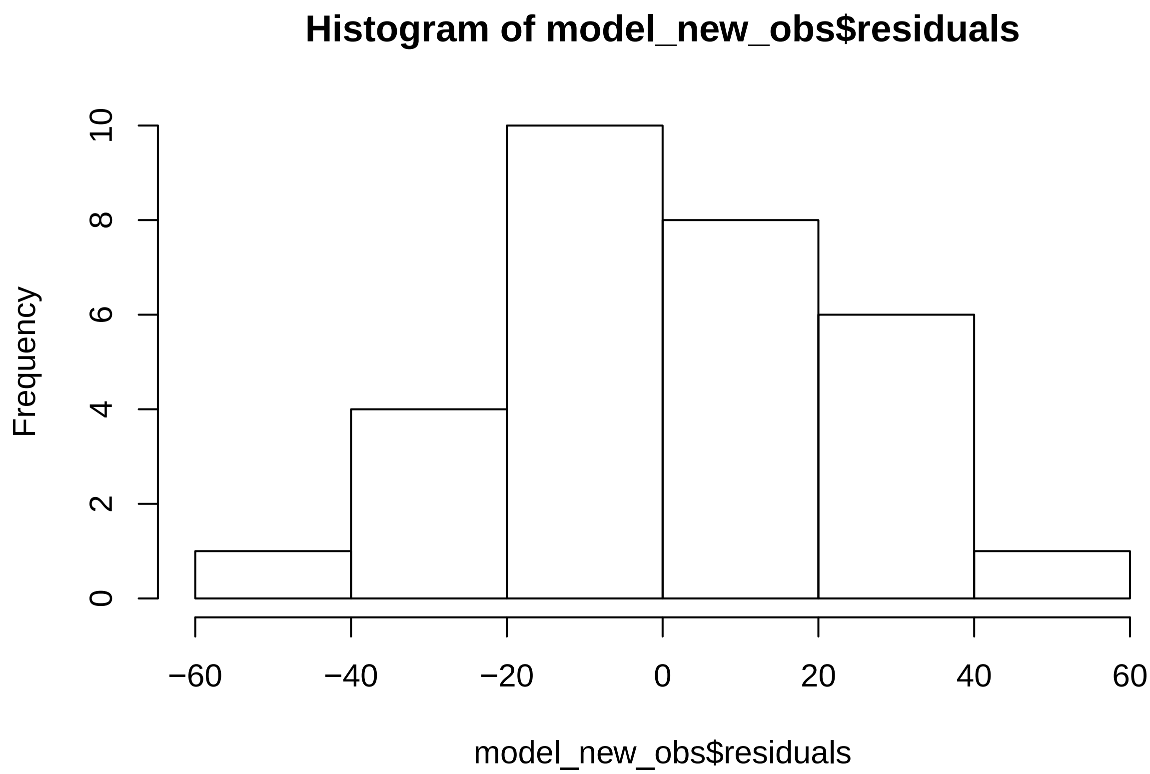
```r
summary(model_new_slug)
```

```
##
## 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
```

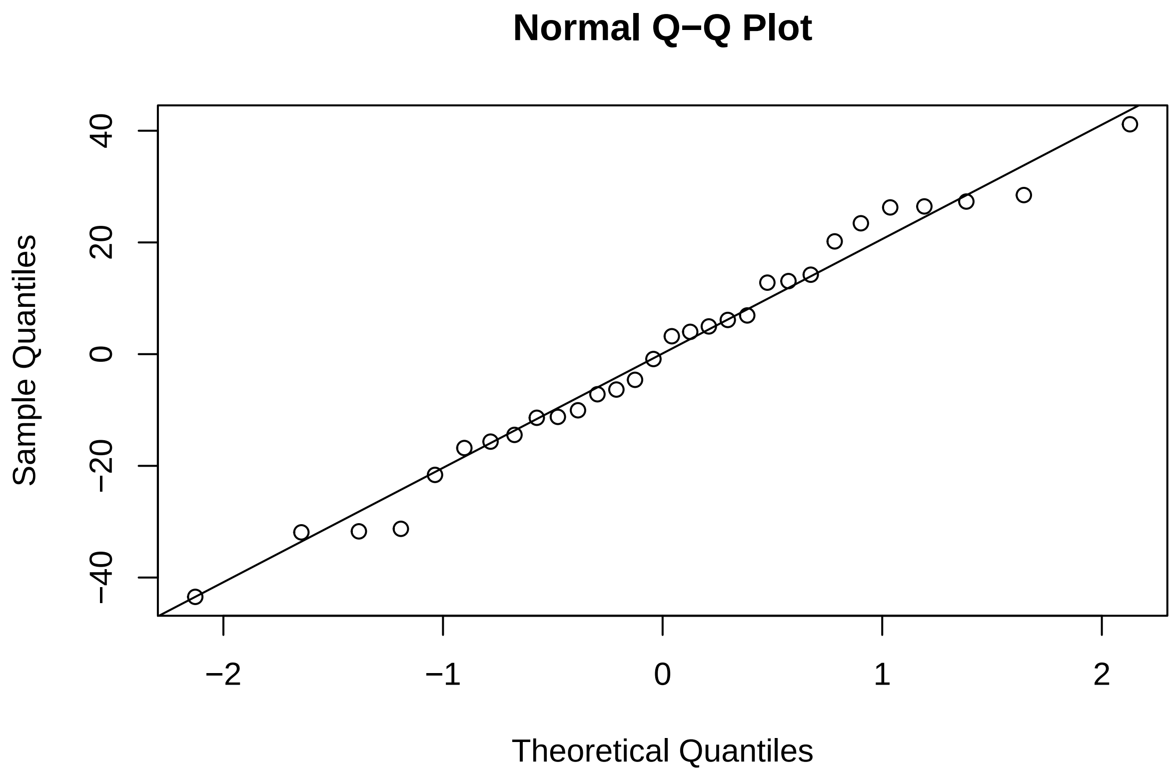
```r
summary(model_new_onbase)
```

```
##
## 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 ***
## ---
## 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
model_new_obs <- lm(runs ~ new_obs, data = mlb11)
hist(model_new_obs$residuals)
```




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



- Check the model diagnostics for the regression model with the variable you decided was the best predictor for runs.
  Answer: The variabale new_obs is the best predictor for runs. The model built using new_obs has R2 value of 0.93 which is higher than the models built using other variable. The residual sum of errors is 20345.54 which is lowest compared to models built using other variables

```
summary(model_new_obs)
```

```
##
## 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
```