Introduction to linear regression

Subhalaxmi Rout

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")
```

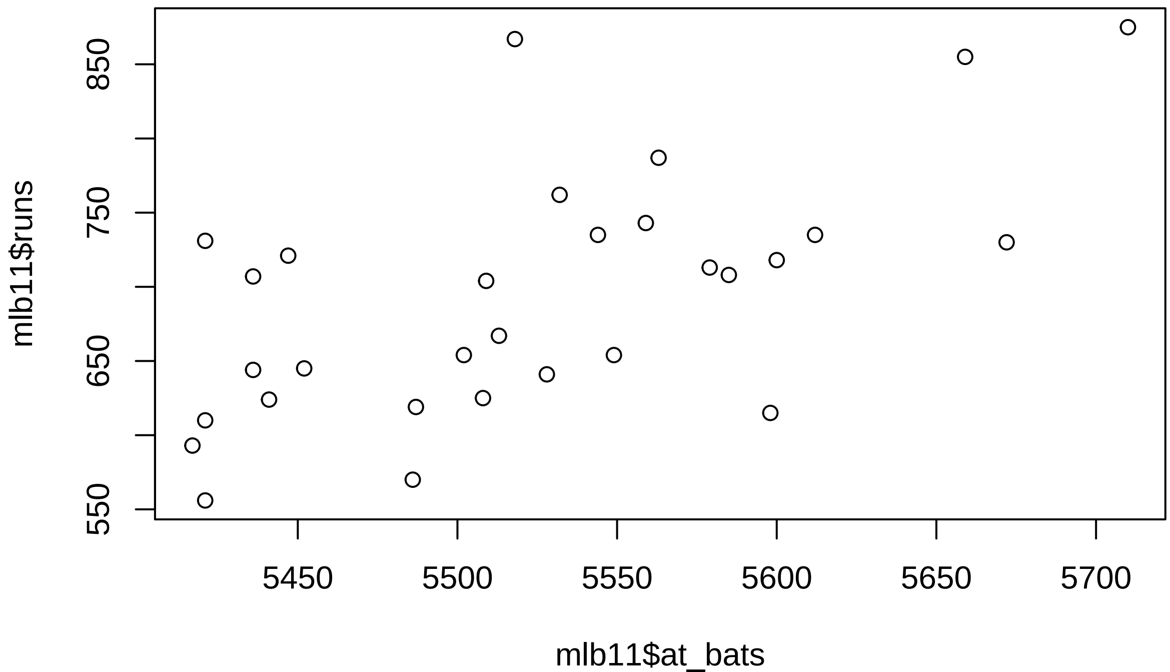
In addition to runs scored, there are seven traditionally used variables in the data set: at-bats, hits, home runs, batting average, strikeouts, stolen bases, and wins. There are also three newer variables: on-base percentage, slugging percentage, and on-base plus slugging. For the first portion of the 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

I would use a scatter plot to display the relationship between runs and at_bats. The relationship looks linear with some outliners. If I knew a team's at_bats, I would be comfortable using a linear model to predict the number of runs.

```
plot(mlb11$at_bats, mlb11$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

There appears to be a positive relationship between runs and at bats. Correlation coefficient(0.610627) indicates a positive correlation but not very strong. The at bats increases number of runs also increases. There are some outliners present in the data.

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.

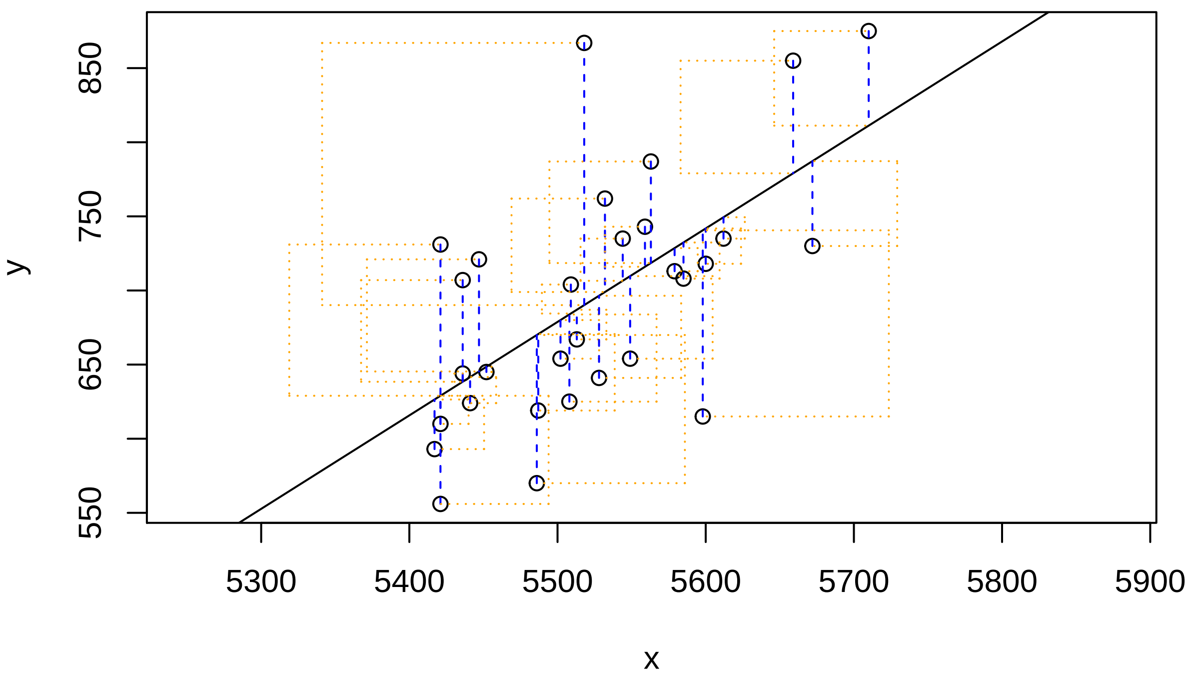
```
# Note that this chunk will only run in interactive mode
plot_ss(x = mlb11$at_bats, y = mlb11$runs)
```

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`.

```
# Note that this chunk will only run in interactive mode
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 function many times and every time I got the same sum of squares: 123721.9.

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
lm(mlb11$homeruns~mlb11$runs)
```

```
##
## Call:
## lm(formula = mlb11$homeruns ~ mlb11$runs)
##
## Coefficients:
## (Intercept)   mlb11$runs
##    -85.1566       0.3415
```

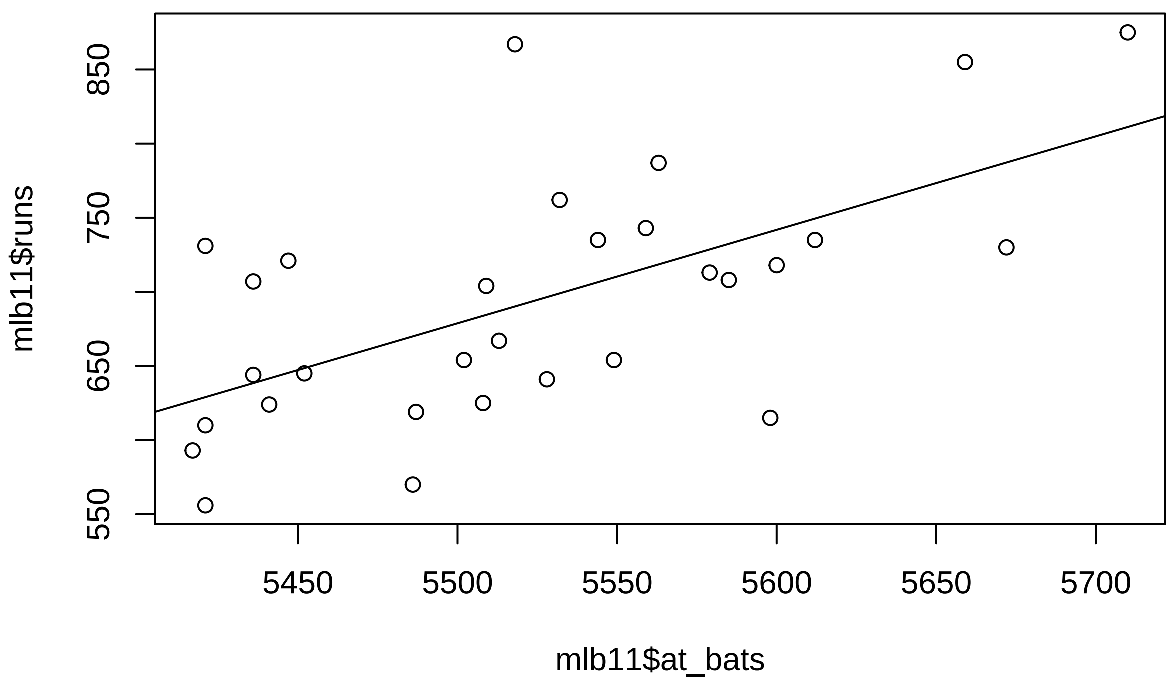
Regression line equation:

$$\hat{y} = -85.1566 + 0.3415 * homeruns$$

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

```
# predict runs with 5578 at-bats
y <- round(-2789.2429 + 0.6305 * 5578)
y
```

```
## [1] 728
```

```
data <- DT::datatable(mlb11)
data
```

Show 10 entries Search:

	team	runs	at_bats	hits	homeruns	bat_avg	strikeouts	stolen_bases	wins	new_onbase	new_slug	new_obs
1	Texas Rangers	855	5659	1599	210	0.283	930	143	96	0.34	0.46	0.8
2	Boston Red Sox	875	5710	1600	203	0.28	1108	102	90	0.349	0.461	0.81
3	Detroit Tigers	787	5563	1540	169	0.277	1143	49	95	0.34	0.434	0.773
4	Kansas City Royals	730	5672	1560	129	0.275	1006	153	71	0.329	0.415	0.744
5	St. Louis Cardinals	762	5532	1513	162	0.273	978	57	90	0.341	0.425	0.766
6	New York Mets	718	5600	1477	108	0.264	1085	130	77	0.335	0.391	0.725
7	New York Yankees	867	5518	1452	222	0.263	1138	147	97	0.343	0.444	0.788
8	Milwaukee Brewers	721	5447	1422	185	0.261	1083	94	96	0.325	0.425	0.75
9	Colorado Rockies	735	5544	1429	163	0.258	1201	118	73	0.329	0.41	0.739
10	Houston Astros	615	5598	1442	95	0.258	1164	118	56	0.311	0.374	0.684

Showing 1 to 10 of 30 entries Previous 1 2 3 Next

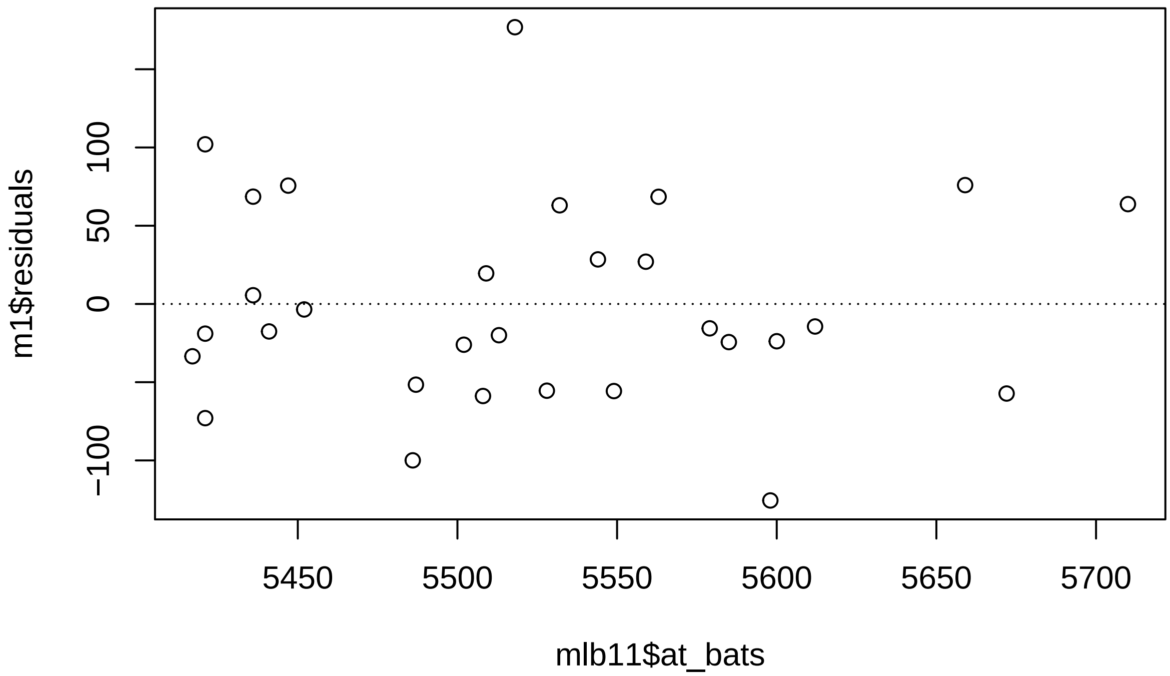
According to regression line, the manager would predict 728 runs for a team with 5578 at-bats. For actual data, Philadelphia Phillies has a close at-bats of 5579 and runs of 713, which is 15 runs less than the prediction. So this is an overestimation by 15 runs.

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
```



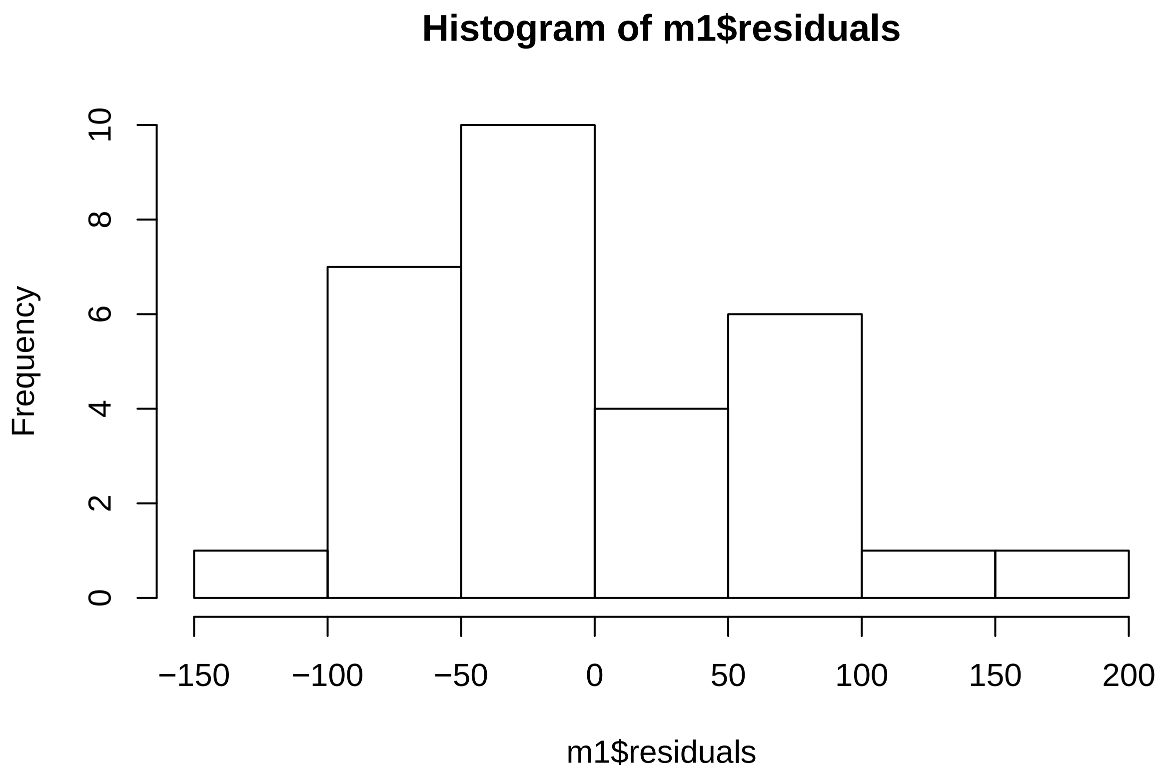
6. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between runs and at-bats?

Answer

There is no such pattern in the residuals plot. There is probably a linear relationship between runs and at-bats.

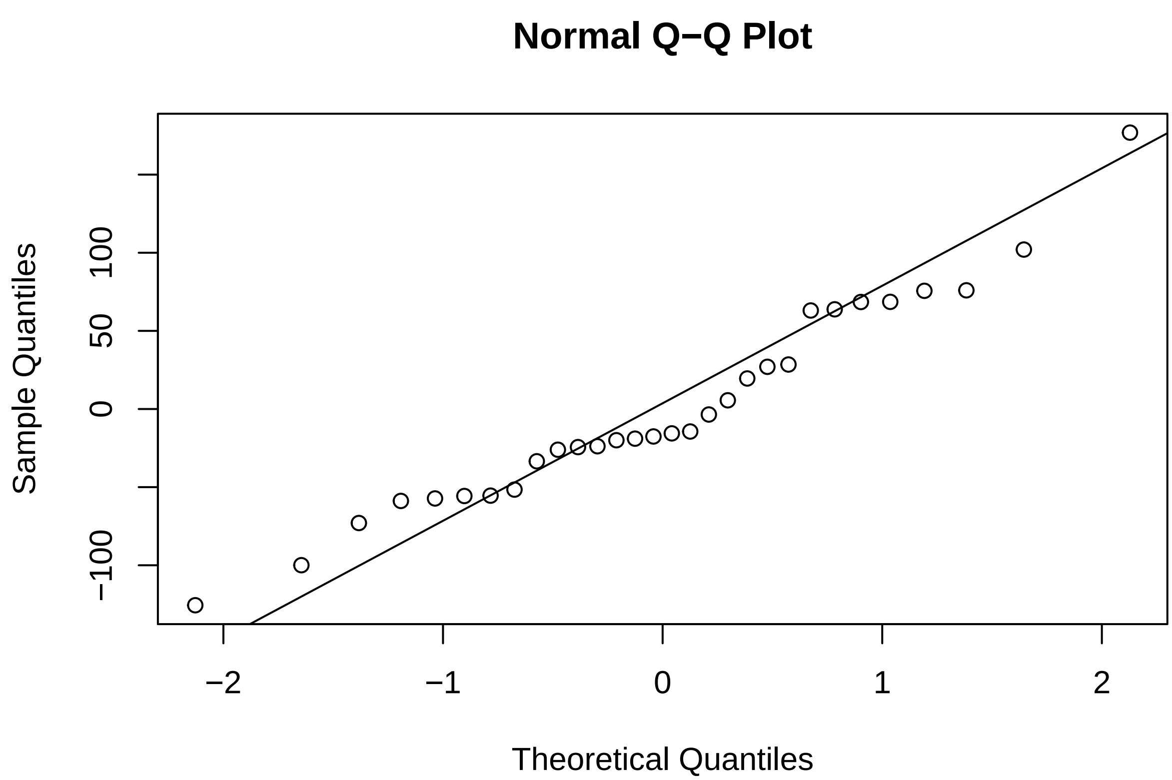
Nearly normal residuals: To check this condition, we can look at a histogram

```
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?

Answer

Based on the histogram and normal probability plot, the residuals appear to be normal.

Constant variability:

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

Answer

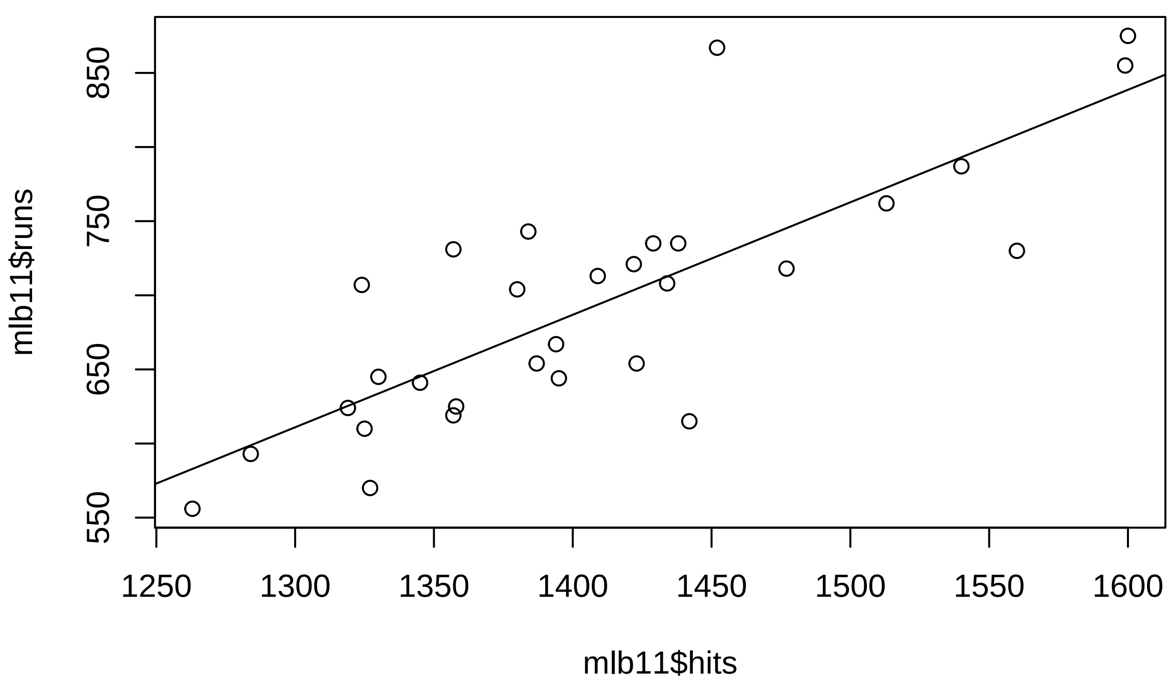
Yes, the constant variability condition appears to be met.

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

```
plot(mlb11$runs~mlb11$hits)
abline(lm(mlb11$runs~mlb11$hits))
```



The above scatter plot shows linear relationship between hits and runs.

- 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

```r
m2 <- lm(mlb11$runs~mlb11$hits)
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
summary(m2)
```

```
##
## Call:
## lm(formula = mlb11$runs ~ mlb11$hits)
##
## 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 *
## mlb11$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
```

Comparing R2 values from 2 models i.e m1 and m2, hits seems to predict runs better than at_bats because m2 has higher R2 value than m1.

- 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

```r
m3 = lm(runs ~ bat_avg, data = mlb11)
m4 = lm(runs ~ strikeouts, data = mlb11)
m5 = lm(runs ~ homeruns, data = mlb11)
m6 = lm(runs ~ wins, data = mlb11)
m7 = lm(runs ~ stolen_bases, data = mlb11)
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
```

```r
summary(m4)
```

```
##
## 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
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
summary(m6)
```

```
##
## 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
```

```r
summary(m7)
```

```
##
## 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
```

m3 model (bat_avg vs runs) has best predicting value for the runs because this model has higher R^2 value.

- 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

```r
m8 = lm(runs ~ new_onbase, data = mlb11)
m9 = lm(runs ~ new_obs, data = mlb11)
m10 =  lm(runs ~ new_slug, data = mlb11)
summary(m8)
```

```
##
## 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
summary(m9)
```

```
##
## 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(m10)
```

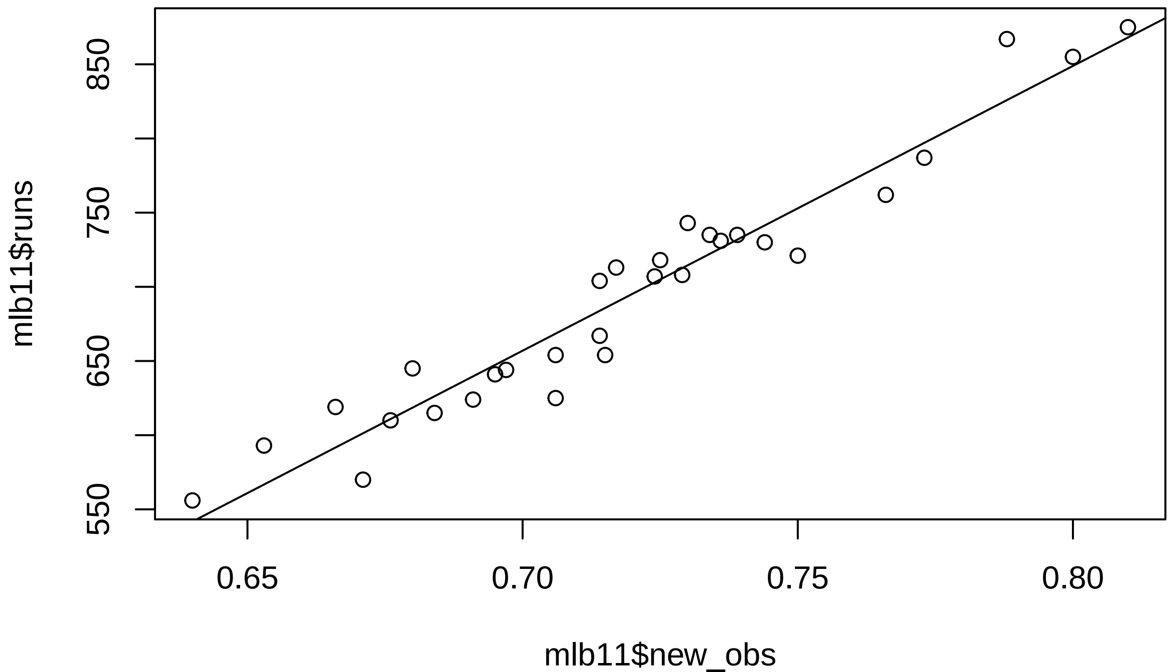
```
##
## Call:
## lm(formula = runs ~ new_slug, data = mlb11)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45.41 -18.66  -0.91  16.29  52.29
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -375.80      68.71   -5.47 7.70e-06 ***
## new_slug     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
```

All 3 models are better predictions for runs. m9 (runs ~ new_obs) has higher R^2 than among all. m9 has 93.5% variability accounted for.

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

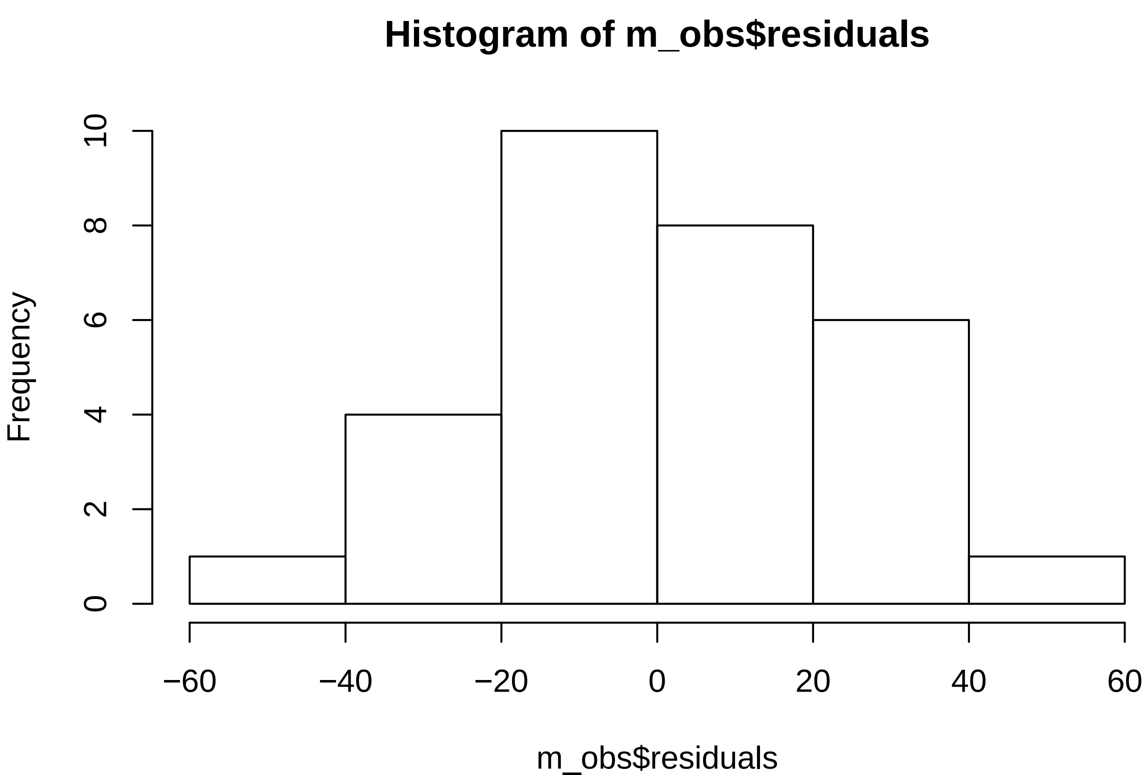
Answer

```r
m_obs <- lm(runs ~ new_obs, data = mlb11)
plot(mlb11$runs ~ mlb11$new_obs)
abline(m_obs)
```



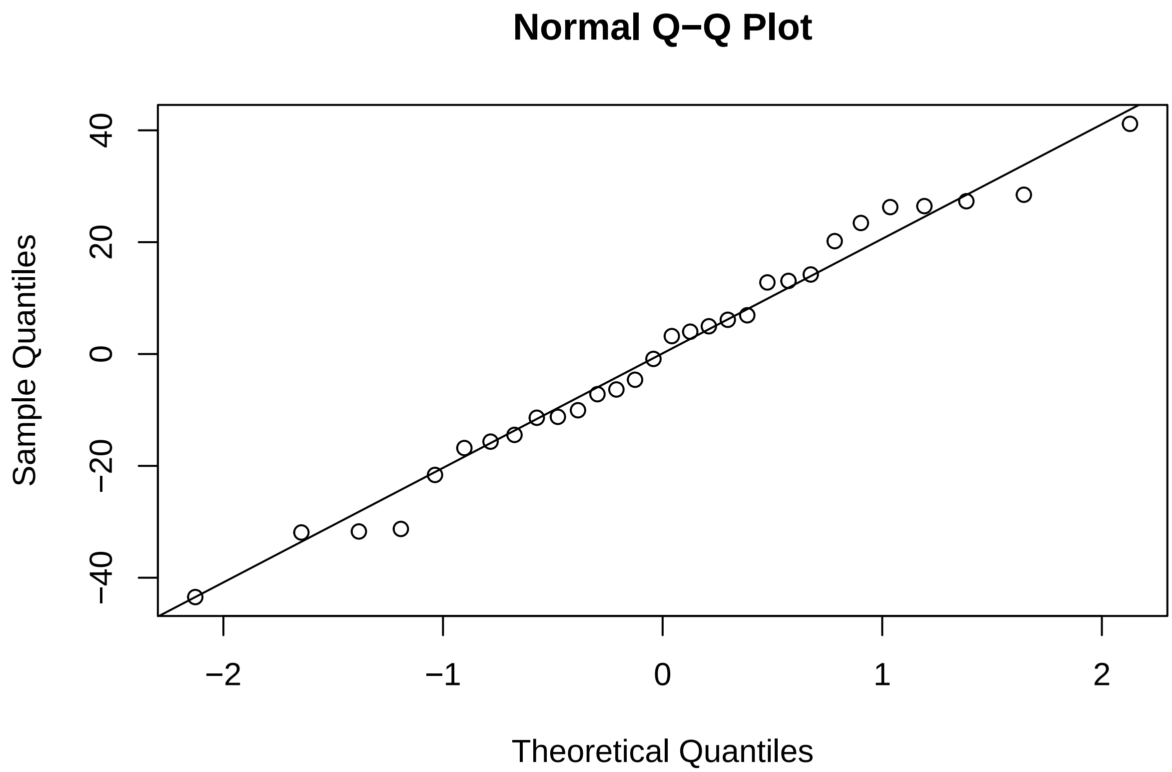
The relationship looks linear. This indicates a linear relationship between the two variables.

```r
hist(m_obs$residuals)
```



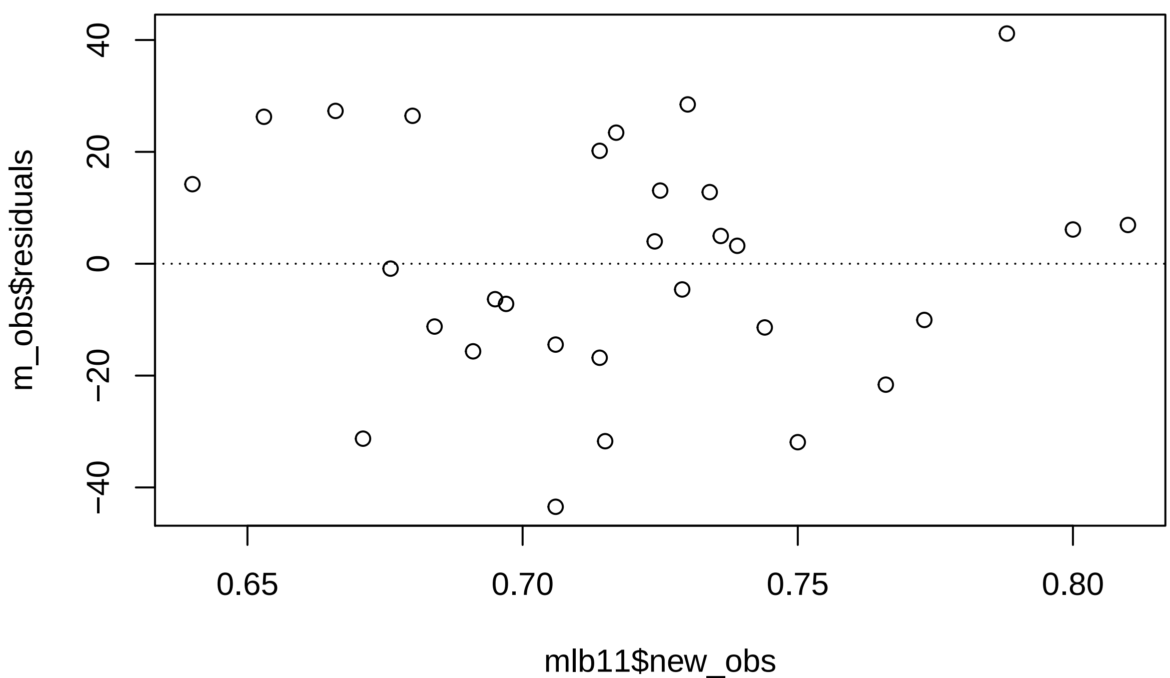
The residuals seems mostly normal according to histogram plot.

```r
qqnorm(m_obs$residuals)
qqline(m_obs$residuals)
```



The points seem to fit the qq plot line well.

```r
plot(m_obs$residuals ~ mlb11$new_obs)
abline(h = 0, lty = 3)
```



The points fall above and below the line constantly. This seems to be constant variability.