am-lab-9

Pradeep Paladugula

3/15/2020

#Introduction To Linear Regression

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 many 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.

```
download.file("http://www.openintro.org/stat/data/mlb11.RData", destfile =
"mlb11.RData")

load("mlb11.RData")
```

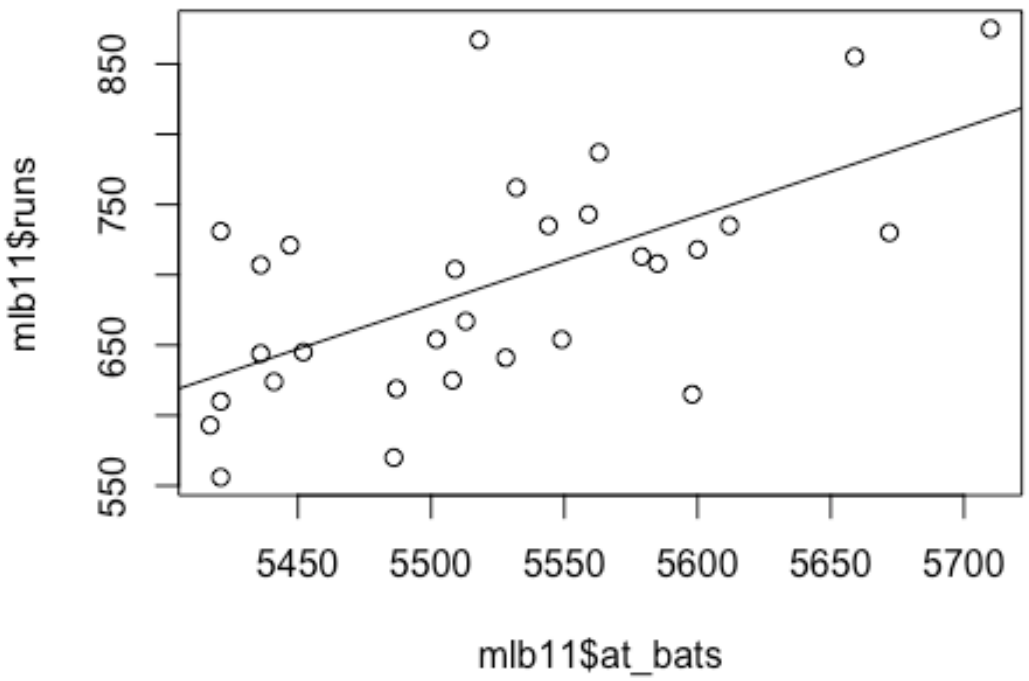
#In addition to runs scored, there are seven traditionally used variables in the data set: at-bats, hits, #home runs, batting average, strikeouts, stolen bases, and wins. #There are also three newer variables: on-base percentage, slugging percentage, and on-base plus slugging. #For the first portion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

Exercise 1:

What type of plot would you use to display the relationship between runs and one of the

other numerical variables?

```
plot(mlb11$at_bats, mlb11$runs)
abline(lm( mlb11$runs ~ mlb11$at_bats))
```



Plot this relationship using the variable at_bats as the predictor.

```
m1 = (lm( mlb11$runs ~ mlb11$at_bats, data = mlb11))
summary(m1)

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

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?

```
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.

Exercise 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.

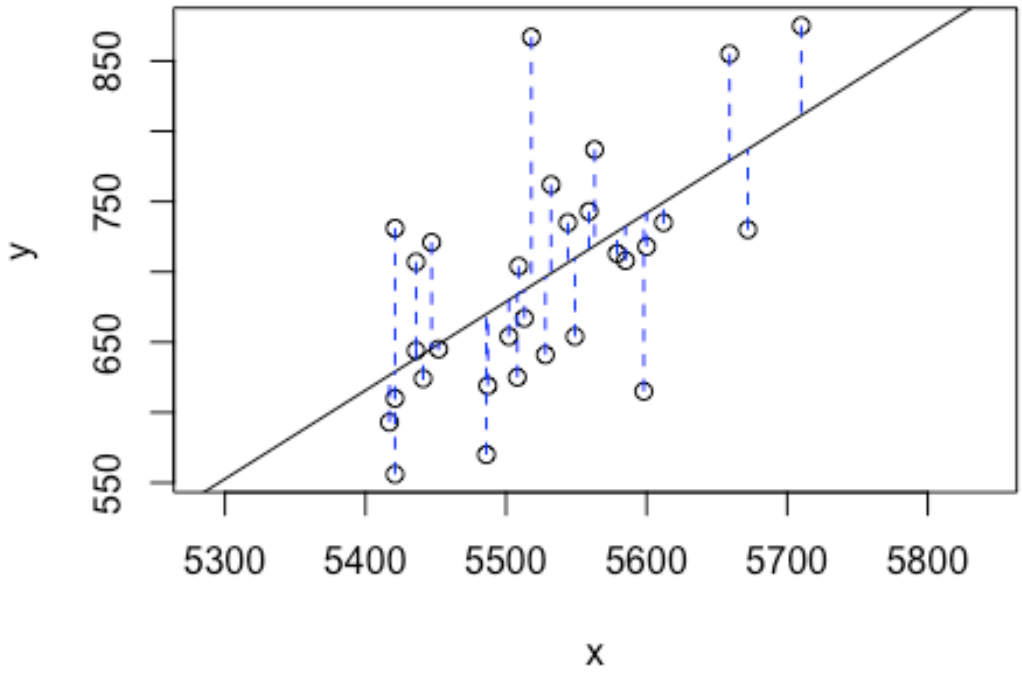
Just as we used the mean and standard deviation to summarize a single variable, we can summarize

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:

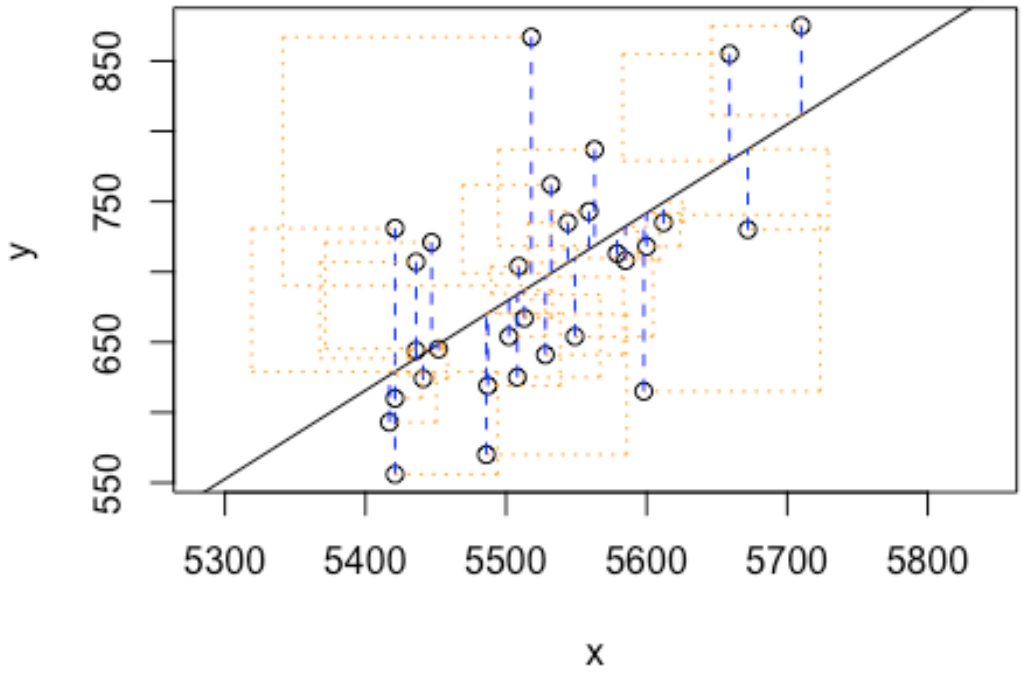
ei=yi-y^i

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.

#Exercise 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?

The linear model Function lm()

#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).

```
m2 <- 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(m2)
```

```
##
## 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:

```
runs = 2789.2 +0.6305*mlb11$at_bats
```

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.

Exercise 4

Fit a new model that uses homeruns to predict runs.

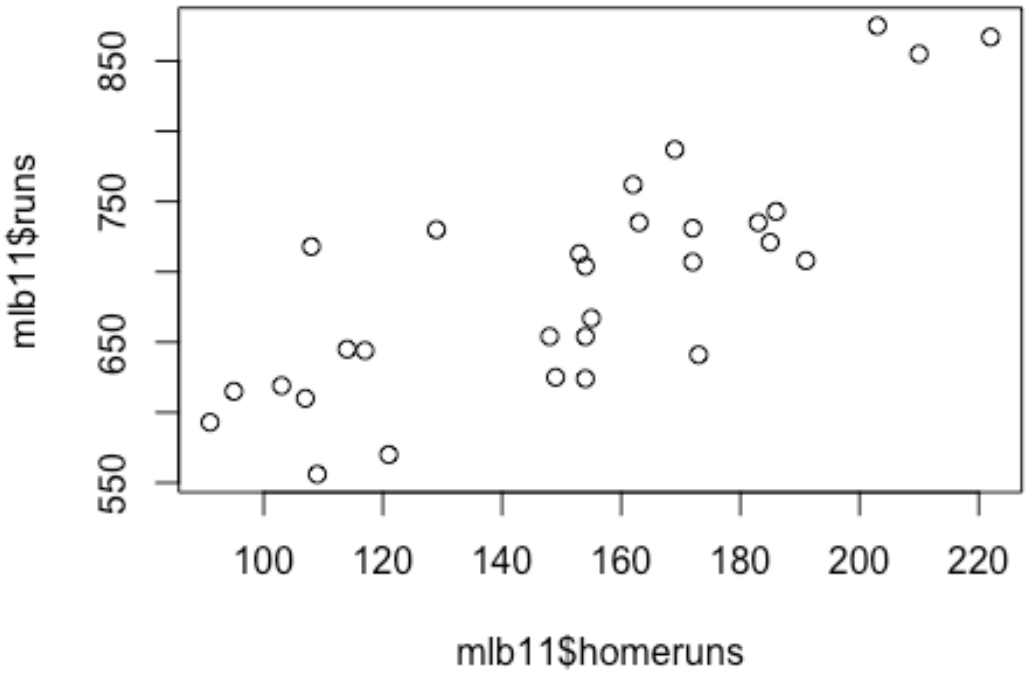
```
lmHR = (lm(runs ~ mlb11$at_bats, data = mlb11))
summary(lmHR)
```

```
##
## Call:
## lm(formula = runs ~ mlb11$at_bats, data = mlb11)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125.58  -47.05  -16.59   54.40  176.87
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2789.2429   853.6957  -3.267 0.002871 **
## mlb11$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
```

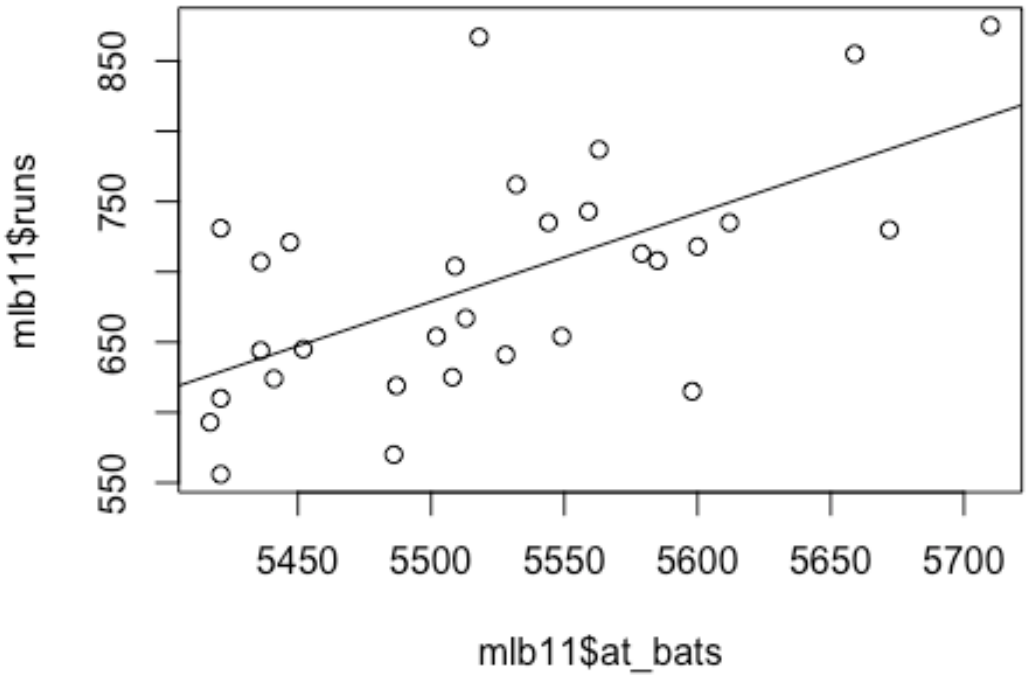
Using the estimates from the R output, write the equation of the regression line.

how does the plot look like??

```
plot(mlb11$runs ~ mlb11$homeruns)
abline((lmHR))
abline(coef = coef(lmHR))
```



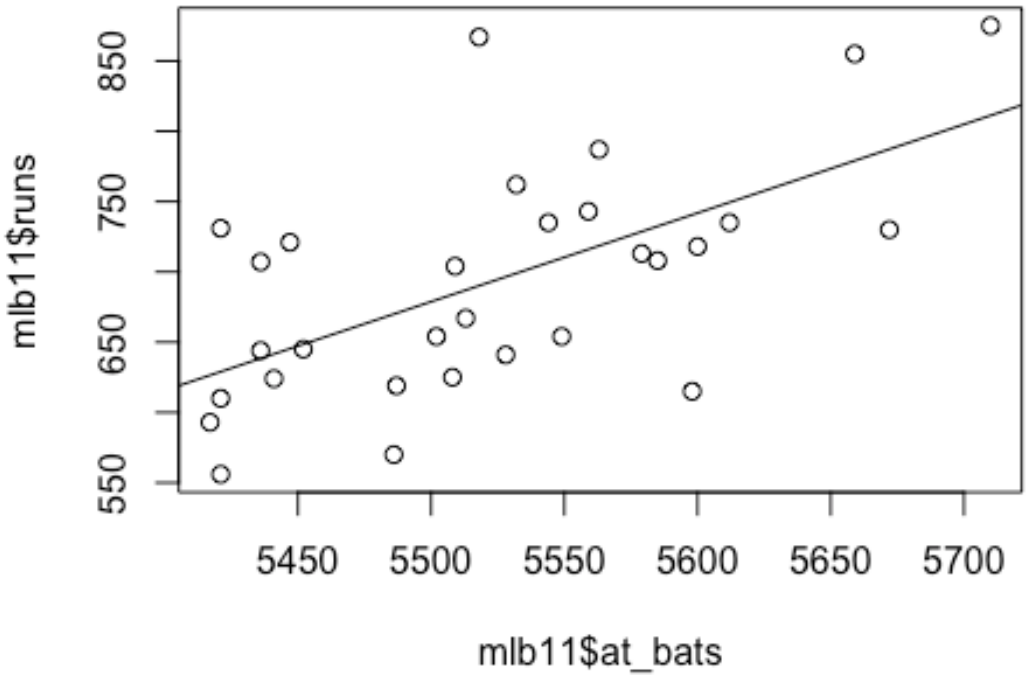

```
plot(mlb11$runs ~ mlb11$at_bats)
#abline(m1)
abline(coef = coef(m1))
```



#What does the slope tell us in the context of the relationship between #success of a team and its home runs? summary(lmHR)
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)
abline(coef = coef(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

NOTE!!!! – When predictions are made for values of x that are beyond the

range of the observed dat, 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.

#Exercise 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?

runs = -2789.2 +0.6305*atbats ## 5578 ## runs = -2789.2 +0.6305*5578

```
mlb11$runs[mlb11$at_bats==5710]
```

```
## [1] 875
```

Is this an overestimate or an underestimate, and by how much?

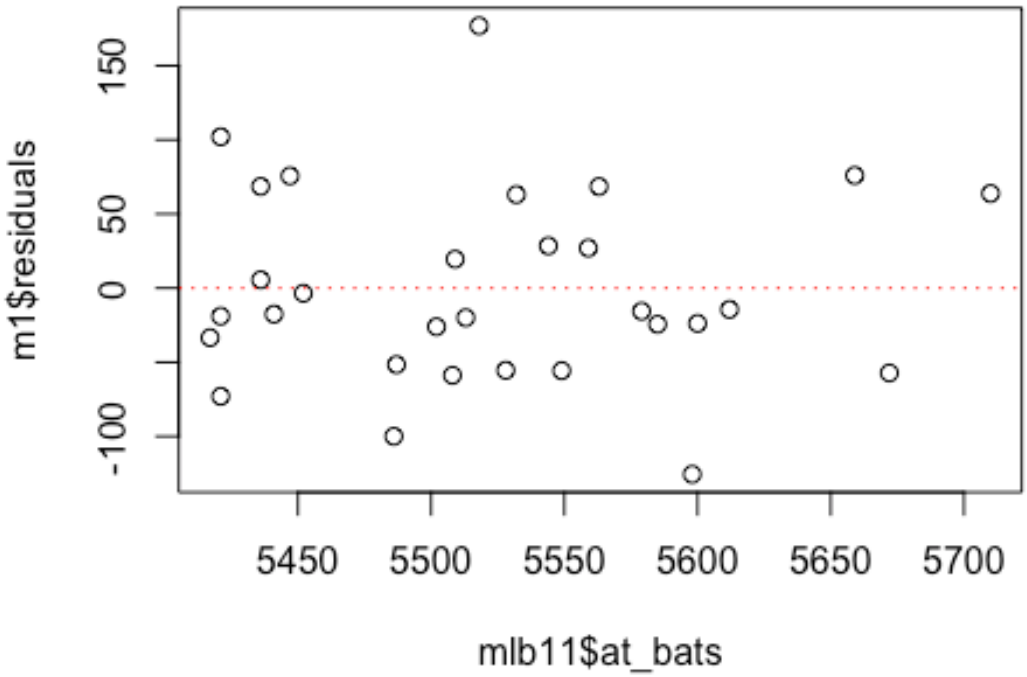
In other words, what is the residual for this prediction?

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, col = "red") # adds a horizontal line at y = 0  and
the line type = 3 for dashed line
```

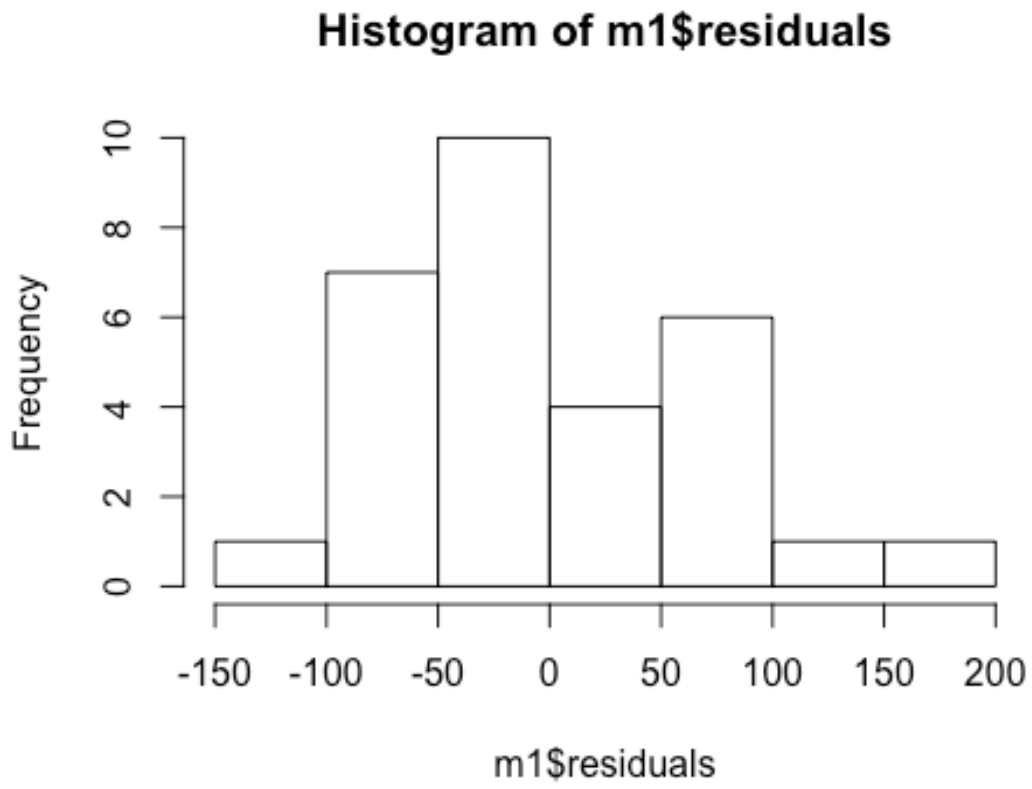


#Exercise 6

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

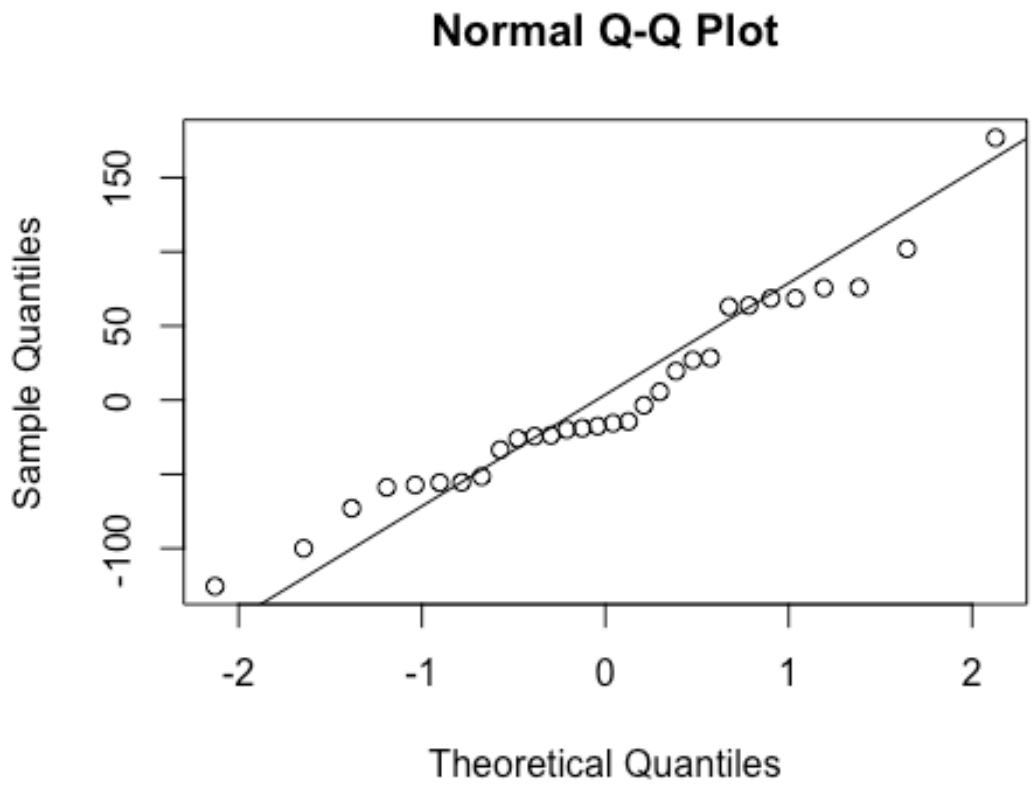
#Nearly normal residuals: To check this condition, we can look at a histogram

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



or a normal probability plot of the residuals.

```
qqnorm(m1$residuals)
#qqline(m1$residuals)
qqline(m1$residuals, names = FALSE, type = 'qtype', na.rm = TRUE) # adds
diagonal line to the normal prob plot

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
"names" is
## not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
graphical
## parameter "type" is obsolete

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
"na.rm" is
## not a graphical parameter
```



#

Exercise 7

Based on the histogram and the normal probability plot,

does the nearly normal residuals condition appear to be met?

Yes, the nearly normal residuals condition appear to be met

Constant variability:

Exercise 8

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

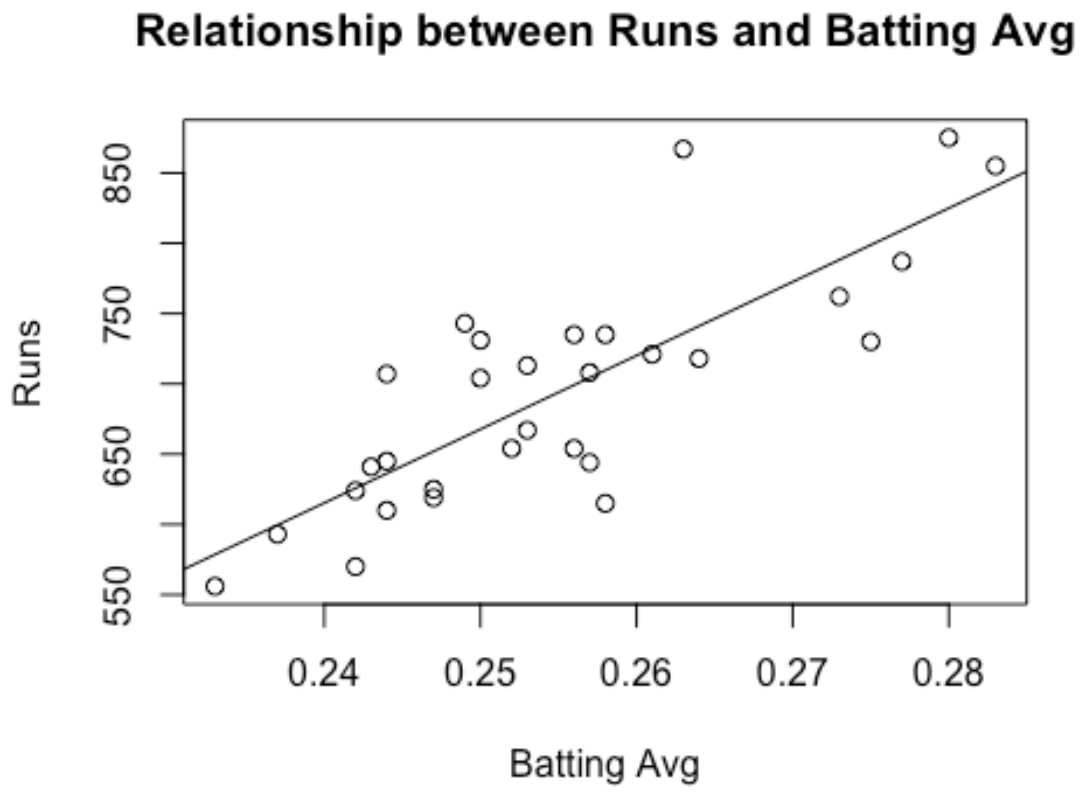
Yes, the constant variability condition appear to be met.

#On Your Own

#1.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 am considering the batting average for the current exercize

```
lm1 <- lm(runs ~ bat_avg, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$bat_avg, main = "Relationship between Runs and
Batting Avg", xlab = "Batting Avg", ylab = "Runs")
abline(lm1)
```




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

## [1] 0.8099859

summary(lm1)

##
## 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
```

#2.How does this relationship compare to the relationship between runs and at_bats? Use the R2

```
runs = -642.8+5242.2*mlb11$bat_avg
runs
```

```
##  [1] 840.7426 825.0160 809.2894 798.8050 788.3206 741.1408 735.8986
725.4142
##  [9] 709.6876 709.6876 704.4454 704.4454 699.2032 699.2032 683.4766
683.4766
## [17] 678.2344 667.7500 667.7500 662.5078 652.0234 652.0234 636.2968
636.2968
## [25] 636.2968 631.0546 625.8124 625.8124 599.6014 578.6326
```

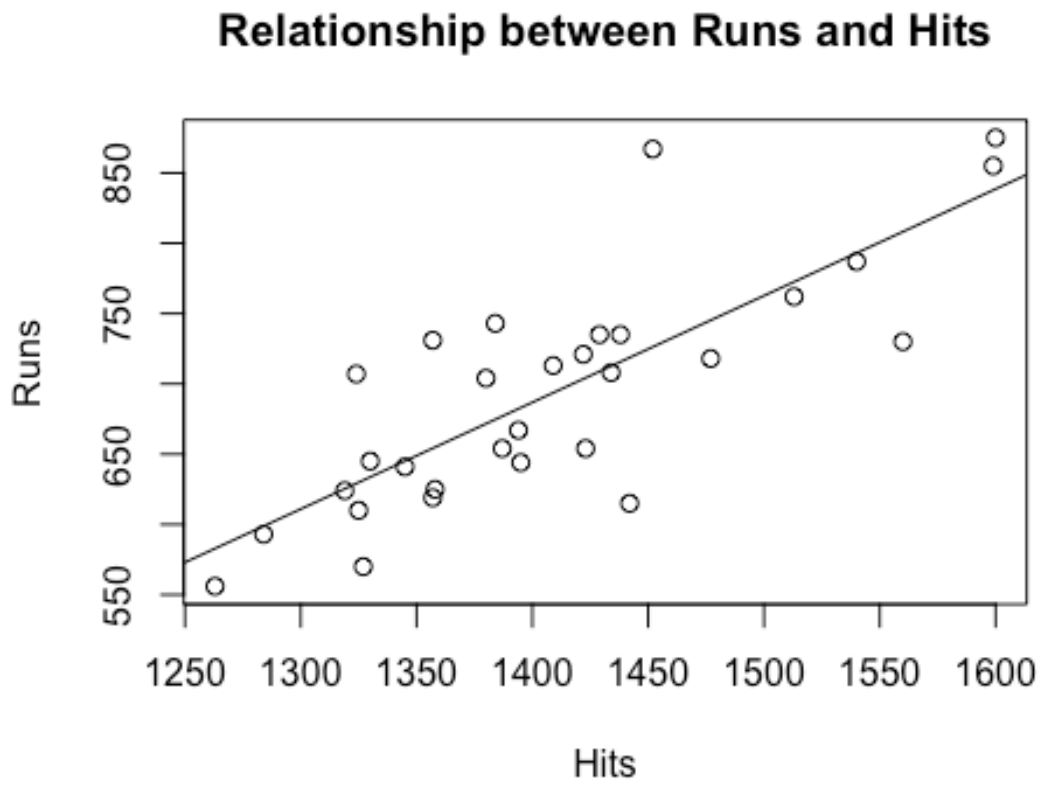
The linear model statistics and correlation coefficient for the relationship between runs and the batting average, it is evident that the relationship is positive, linear and relatively strong.

#values from the two model summaries to compare. Does your variable seem to predict runs better than at_bats? How can you tell?

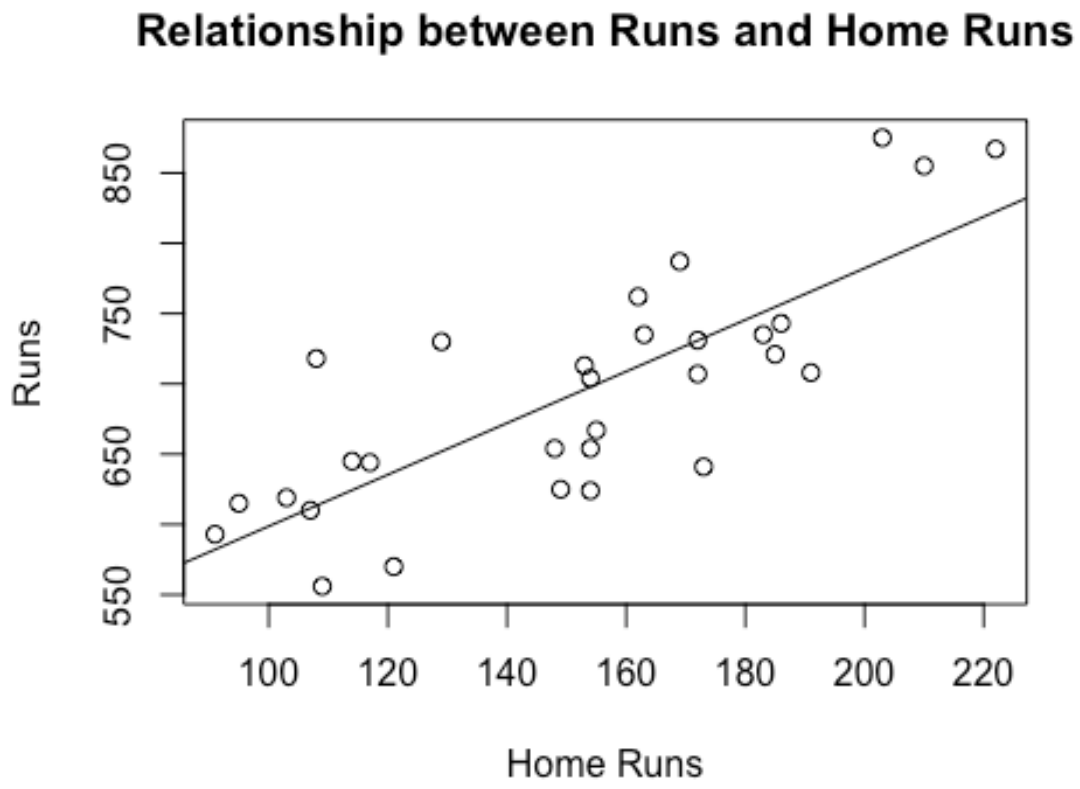
#3.Now that you can summarize the linear relationship between two variables, investigate the relationships between runs and each of the other five traditional variables. Which variable best predicts runs? Support your conclusion using the graphical and numerical methods we've discussed (for the sake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

#at-bats, hits, home runs, batting average, strikeouts, stolen bases, and wins.

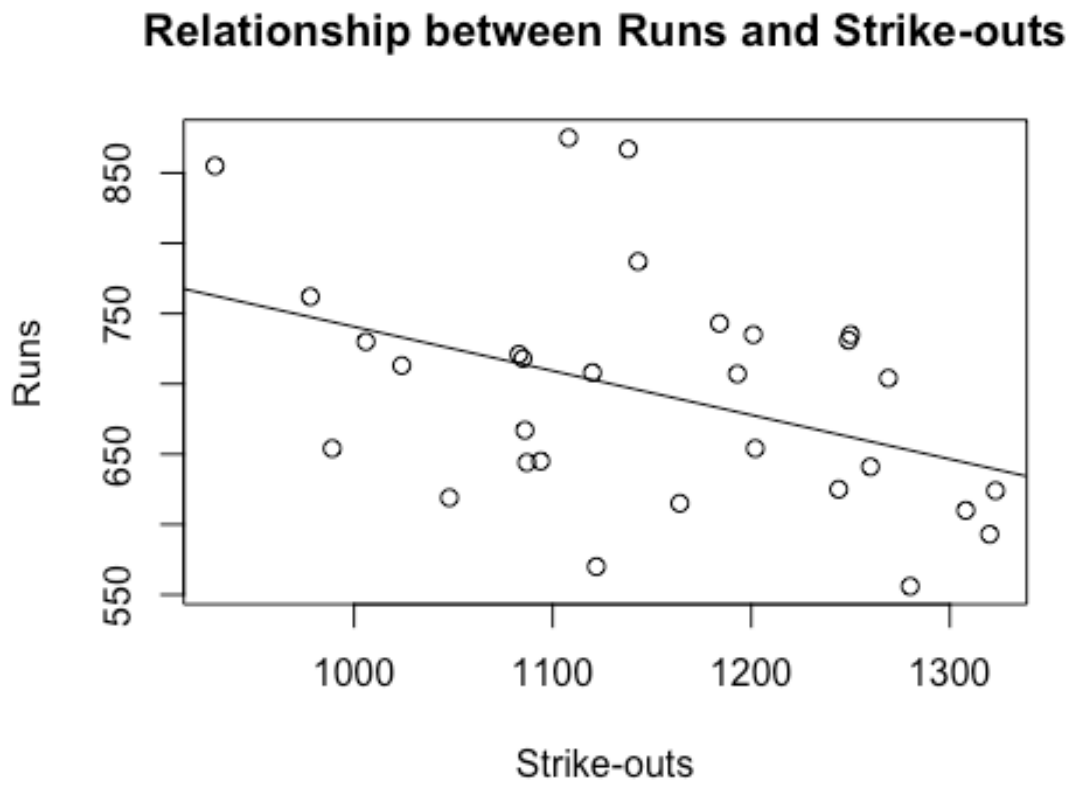
```
lmhits <- lm(runs ~ hits, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$hits, main = "Relationship between Runs and Hits",
xlab = "Hits", ylab = "Runs")
abline(lmhits)
```



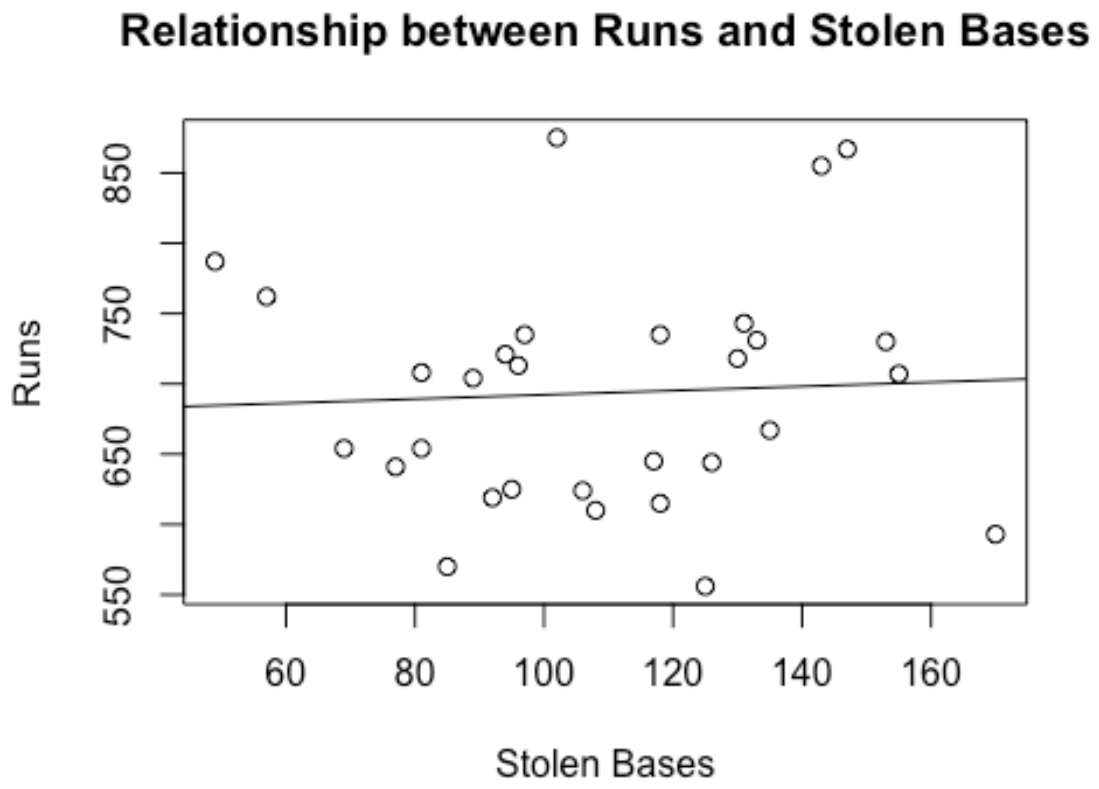

```r
lmhr <- lm(runs ~ homeruns, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$homeruns, main = "Relationship between Runs and Home
Runs", xlab = "Home Runs", ylab = "Runs")
abline(lmhr)
```



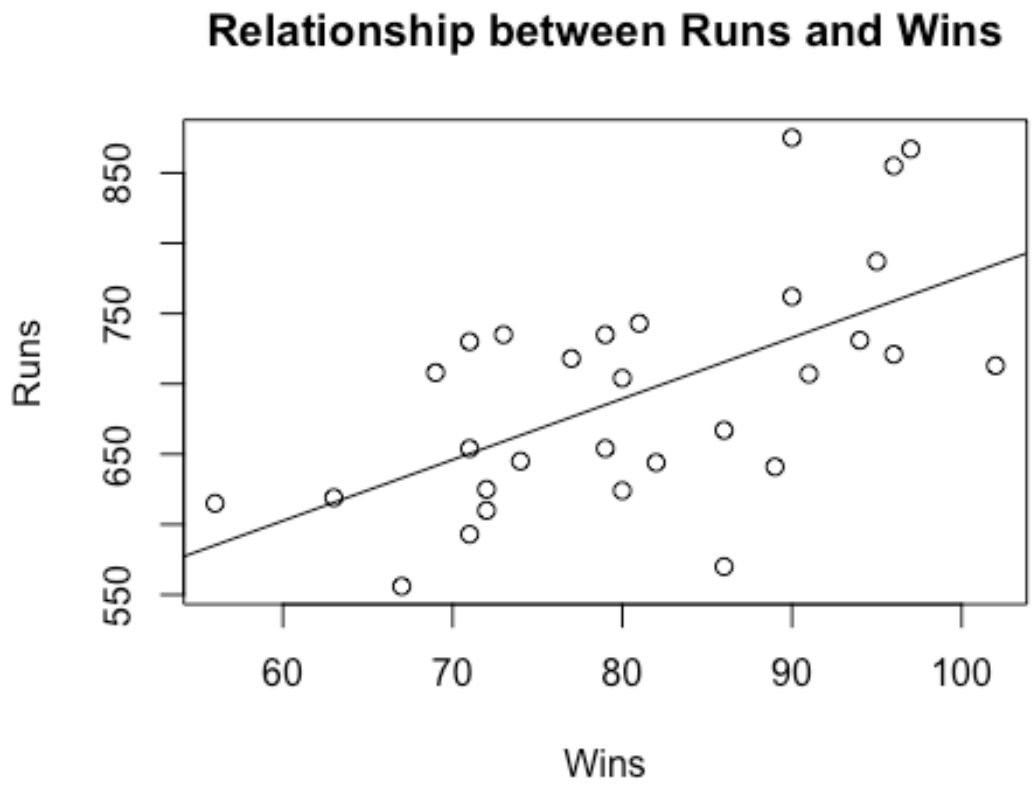

```
lmso <- lm(runs ~ strikeouts, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$strikeouts, main = "Relationship between Runs and
Strike-outs", xlab = "Strike-outs", ylab = "Runs")
abline(lmso)
```




```r
lmsb <- lm(runs ~ stolen_bases, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$stolen_bases, main = "Relationship between Runs and
Stolen Bases", xlab = "Stolen Bases", ylab = "Runs")
abline(lmsb)
```

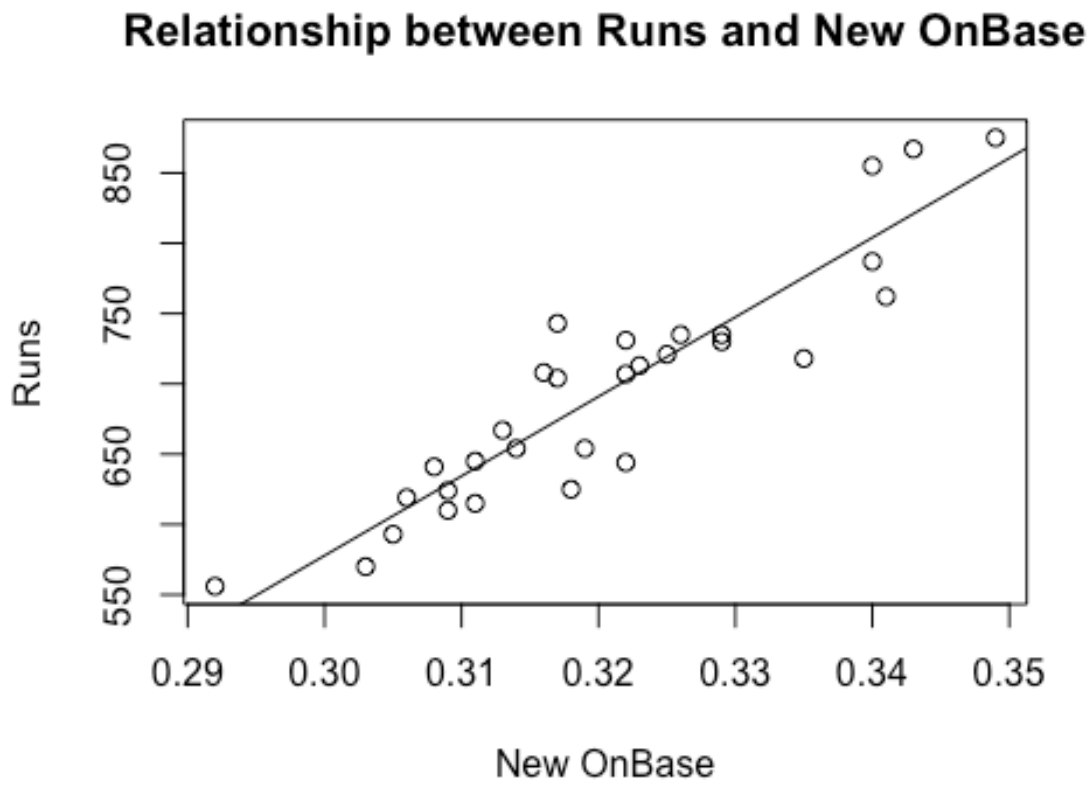



```r
lmwin <- lm(runs ~ wins, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$wins, main = "Relationship between Runs and Wins",
xlab = "Wins", ylab = "Runs")
abline(lmwin)
```



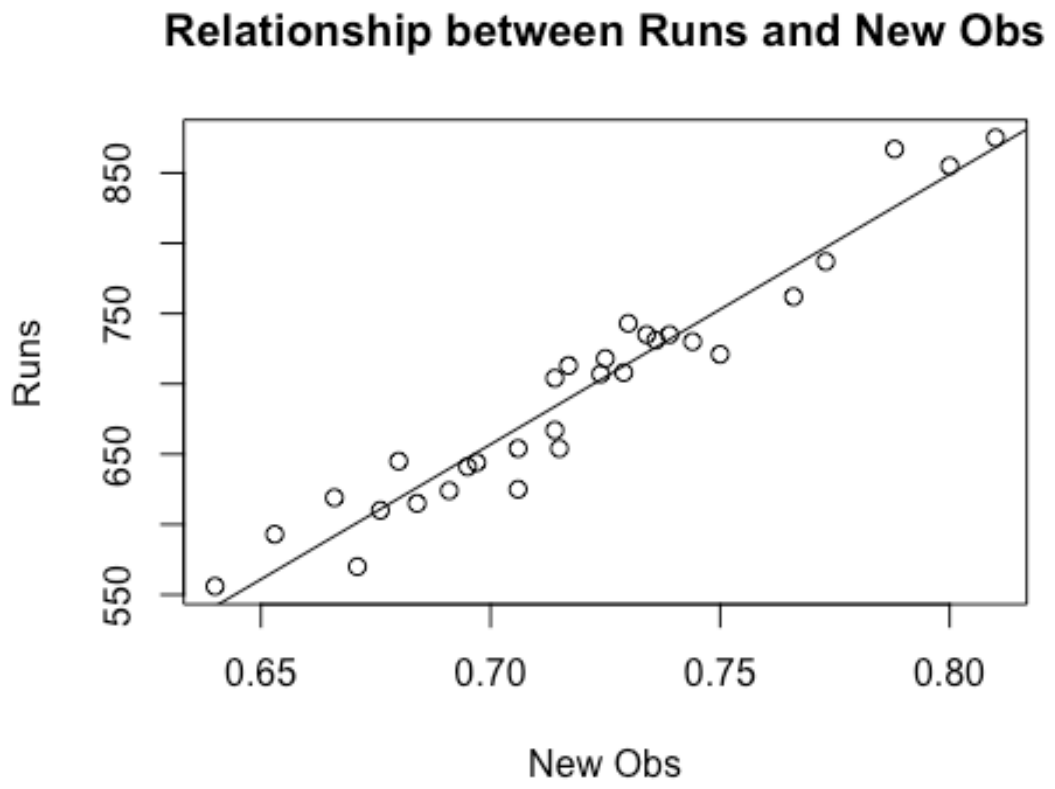
#4.Now examine the three newer variables. These are the statistics used by the author of Moneyball to predict a teams success. In general, are they more or less effective at predicting runs that the old variables? Explain using appropriate graphical and numerical evidence. Of all ten variables we've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

```
lmNOB <- lm(runs ~ new_onbase, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$new_onbase, main = "Relationship between Runs and New
OnBase", xlab = "New OnBase", ylab = "Runs")
abline(lmNOB)
```




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

## [1] 0.9214691

lmnslug <- lm(runs ~ new_slug, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$new_slug, main = "Relationship between Runs and New
Slugger", xlab = "New Slugger", ylab = "Runs")
abline(lmnslug)
```


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

## [1] 0.9470324

lmnobs <- lm(runs ~ new_obs, data = mlb11)   #linear modal line
plot(mlb11$runs ~ mlb11$new_obs, main = "Relationship between Runs and New
Obs", xlab = "New Obs", ylab = "Runs")
abline(lmnobs)
```



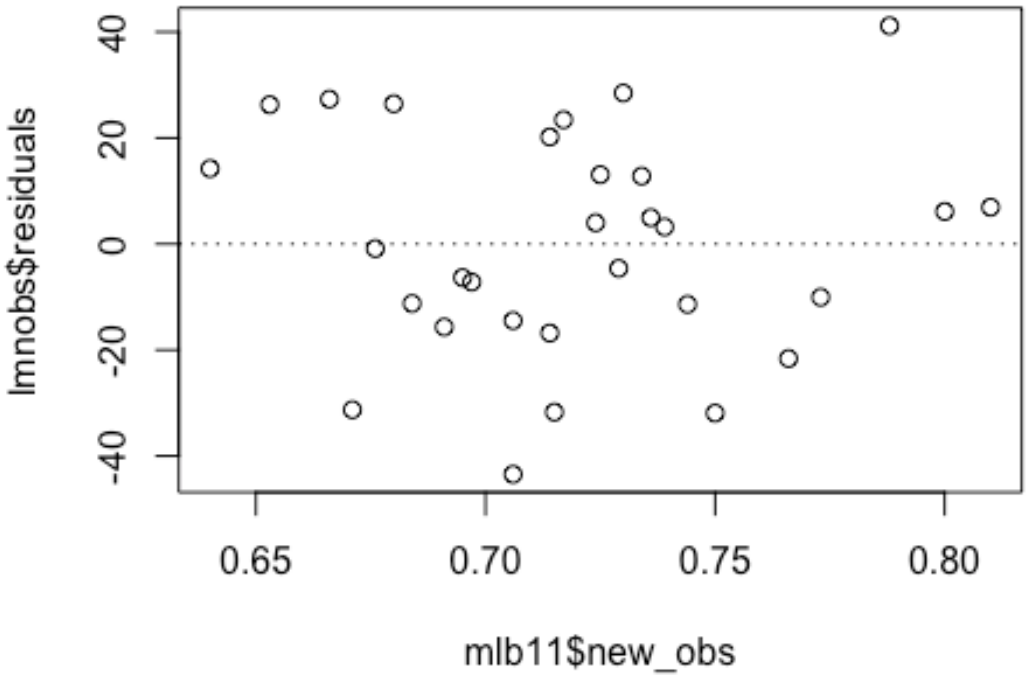

```
cor(mlb11$new_obs, mlb11$runs)

## [1] 0.9669163

summary(lmnobs)

##
## 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
```

```
summary(lmnslug)

##
## 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

summary(lmNOB)

##
## 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
```

From above graphs and the values, new_obs variable has the highest R-squared value and coefficient correlation values and appears to be the best and most effective predictor of the runs.

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

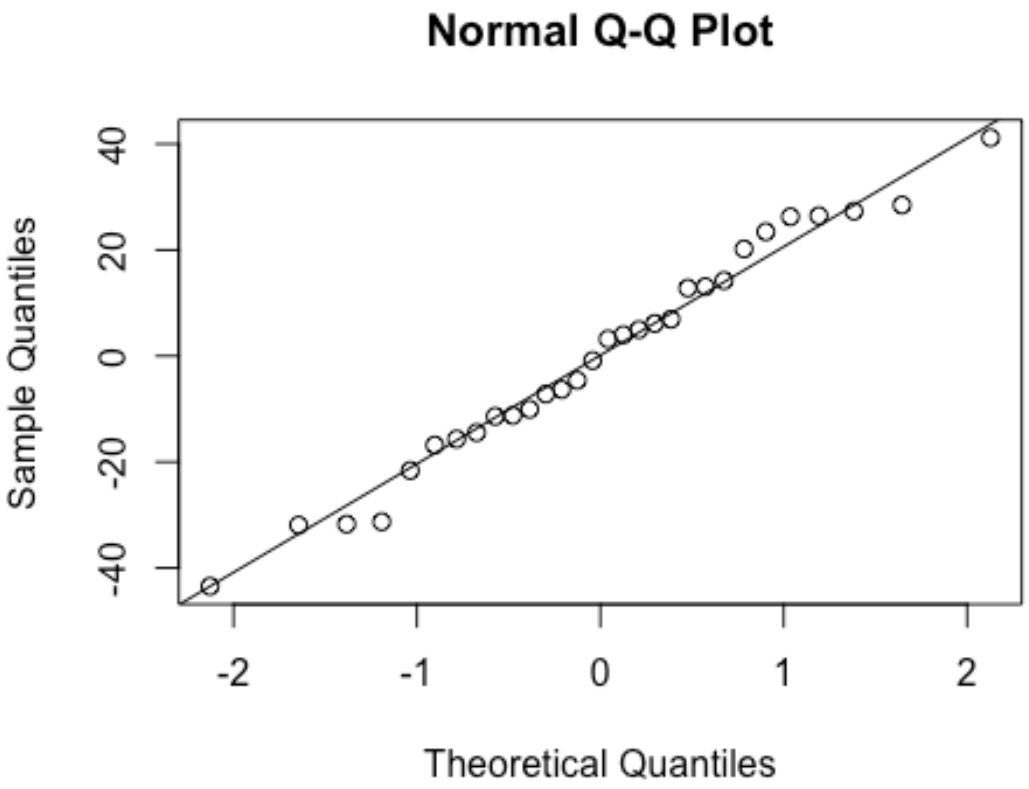
```
plot(lmnobs$residuals ~ mlb11$new_obs)
abline(h = 0, lty = 3)
```



The Linearity of residuals is approximately constant across the distribution, but does not indicate any curvatures or any indication of non-normality

```
hist(lmnobs$residuals)
```


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



from the above graph we can notify that residuals are approximately normaly ditributed and the modal meets the nearly normal residual condition.

3) Constatn Variablity: Based on the plot the variability of points around the least squares line are roughly constant so the condition constant variability has been met.