

Documentation on using ioslides is available here: http://rmarkdown.rstudio.com/ioslides_presentation_format.html Some slides are adopted (or copied) from OpenIntro: <https://www.openintro.org/>

SAT Scores

We will use the SAT data for 162 students which includes their verbal and math scores. We will model math from verbal. Recall that the linear model can be expressed as:

$$y = mx + b$$

Or alternatively as:

$$y = b_1x + b_0$$

Where m (or b_1) is the slope and b (or b_0) is the intercept. Therefore, we wish to model:

$$SAT_{math} = b_1 SAT_{verbal} + b_0$$

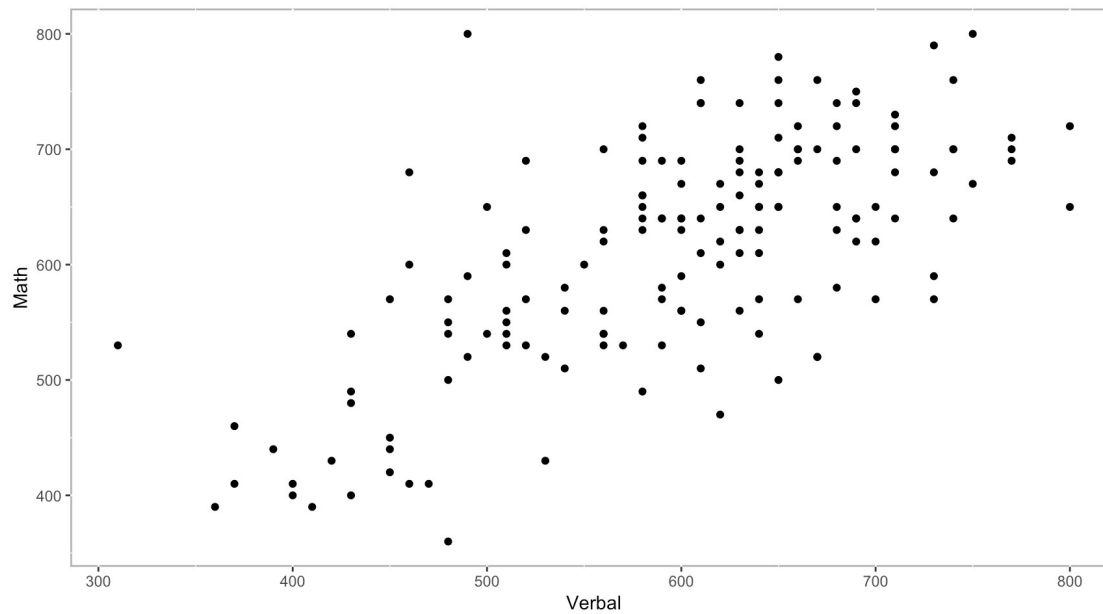
Data Prep

To begin, we read in the CSV file and convert the `Verbal` and `Math` columns to integers. The data file uses `.` (i.e. a period) to denote missing values. The `as.integer` function will automatically convert those to `NA` (the indicator for a missing value in R). Finally, we use the `complete.cases` eliminate any rows with any missing values.

```
sat <- read.csv('../Data/SAT_scores.csv', stringsAsFactors=FALSE)
names(sat) <- c('Verbal', 'Math', 'Sex')
sat$Verbal <- as.integer(sat$Verbal)
sat$Math <- as.integer(sat$Math)
sat <- sat[complete.cases(sat),]
```

Scatter Plot

The first step is to draw a scatter plot. We see that the relationship appears to be fairly linear.



Descriptive Statistics

Next, we will calculate the means and standard deviations.

```
( verbalMean <- mean(sat$Verbal) )
```

```
## [1] 596.2963
```

```
( mathMean <- mean(sat$Math) )
```

```
## [1] 612.0988
```

```
( verbalSD <- sd(sat$Verbal) )
```

```
## [1] 99.5199
```

```
( mathSD <- sd(sat$Math) )
```

```
## [1] 98.13435
```

```
( n <- nrow(sat) )
```

```
## [1] 162
```

z-Scores

Calculate z-scores (standard scores) for the verbal and math scores.

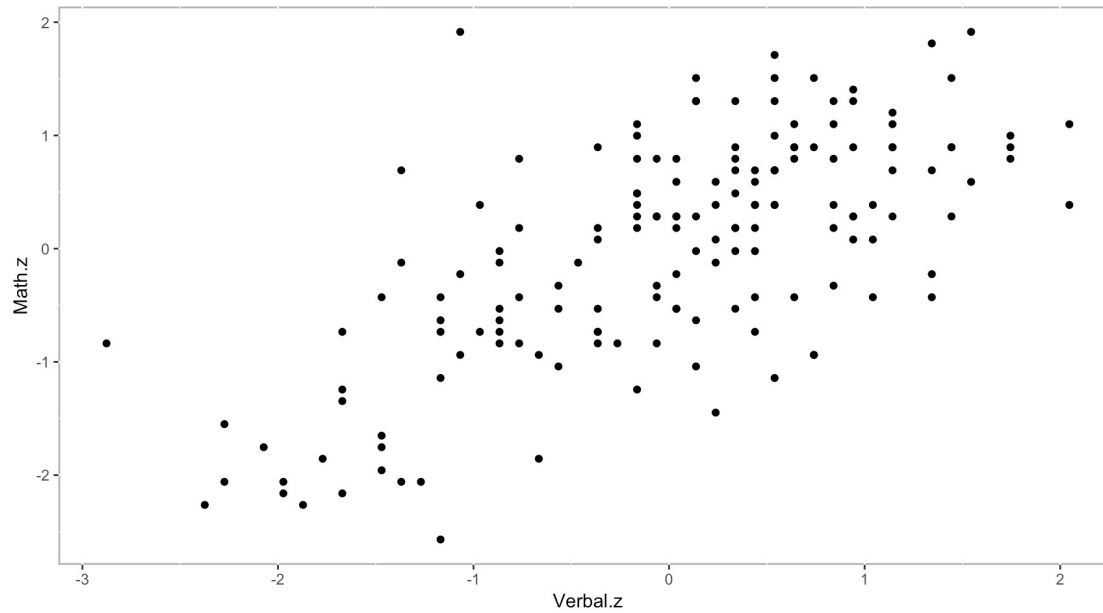
$$z = \frac{y - \bar{y}}{s}$$

```
sat$Verbal.z <- (sat$Verbal - verbalMean) / verbalSD
sat$Math.z <- (sat$Math - mathMean) / mathSD
head(sat)
```

##	Verbal	Math	Sex	Verbal.z	Math.z
## 1	450	450	F	-1.47002058	-1.65180456
## 2	640	540	F	0.43914539	-0.73469449
## 3	590	570	M	-0.06326671	-0.42899113
## 4	400	400	M	-1.97243268	-2.16131016
## 5	600	590	M	0.03721571	-0.22518889
## 6	610	610	M	0.13769813	-0.02138665

Scatter Plot of z-Scores

Scatter plot of z-scores. Note that the pattern is the same but the scales on the x- and y-axes are different.



Correlation

<http://shiny.albany.edu/stat/rectangles/>

Calculate the correlation manually using the z-score formula:

$$r = \frac{\sum z_x z_y}{n - 1}$$

```
r <- sum( sat$Verbal.z * sat$Math.z ) / ( n - 1 )  
r
```

```
## [1] 0.6846061
```

Or the `cor` function in R is probably simpler.

```
cor(sat$Verbal, sat$Math)
```

```
## [1] 0.6846061
```

And to show that the units don't matter, calculate the correlation with the z-scores.

```
cor(sat$Verbal.z, sat$Math.z)
```

```
## [1] 0.6846061
```


Calculate the slope.

$$m = r \frac{S_y}{S_x} = r \frac{S_{math}}{S_{verbal}}$$

```
m <- r * (mathSD / verbalSD)
m
```

```
## [1] 0.6750748
```

Calculate the intercept

Recall that the point where the mean of x and mean of y intersect will be on the line of best fit). Therefore,

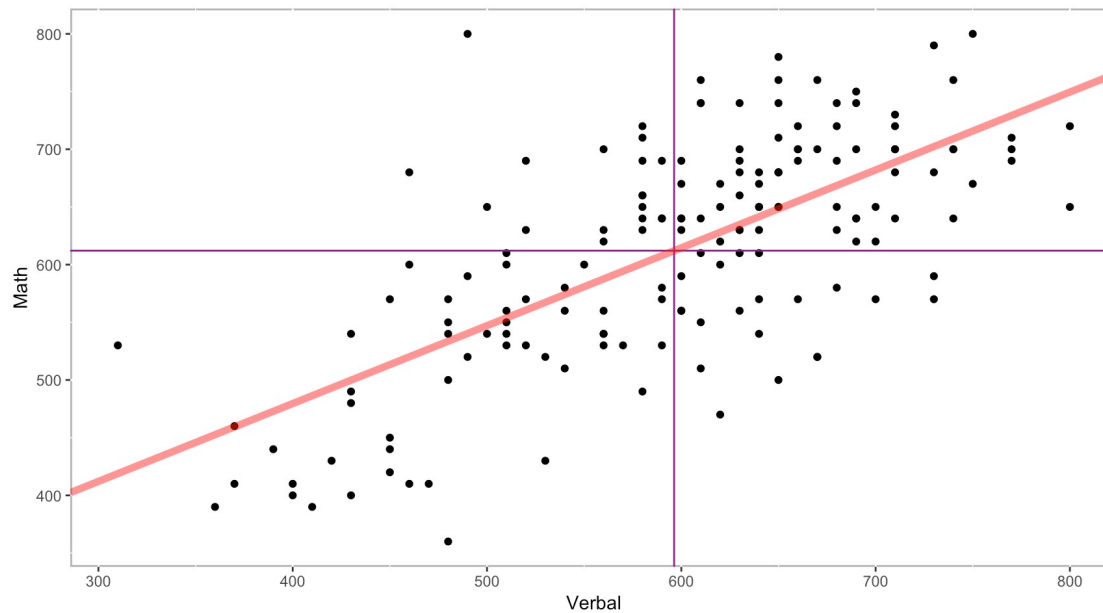
$$b = \bar{y} - m\bar{x} = \overline{SAT_{math}} - m\overline{SAT_{verbal}}$$

```
b <- mathMean - m * verbalMean  
b
```

```
## [1] 209.5542
```

Scatter Plot with Regression Line

We can now add the regression line to the scatter plot. The vertical and horizontal lines represent the mean Verbal and Math SAT scores, respectively.



Examine the Residuals

To examine the residuals, we first need to calculate the predicted values of y (Math scores in this example).

```
sat$Math.predicted <- m * sat$Verbal + b
head(sat, n=4)
```

```
##   Verbal Math Sex   Verbal.z   Math.z Math.predicted
## 1    450  450  F -1.47002058 -1.6518046      513.3378
## 2    640  540  F  0.43914539 -0.7346945      641.6020
## 3    590  570  M -0.06326671 -0.4289911      607.8483
## 4    400  400  M -1.97243268 -2.1613102      479.5841
```

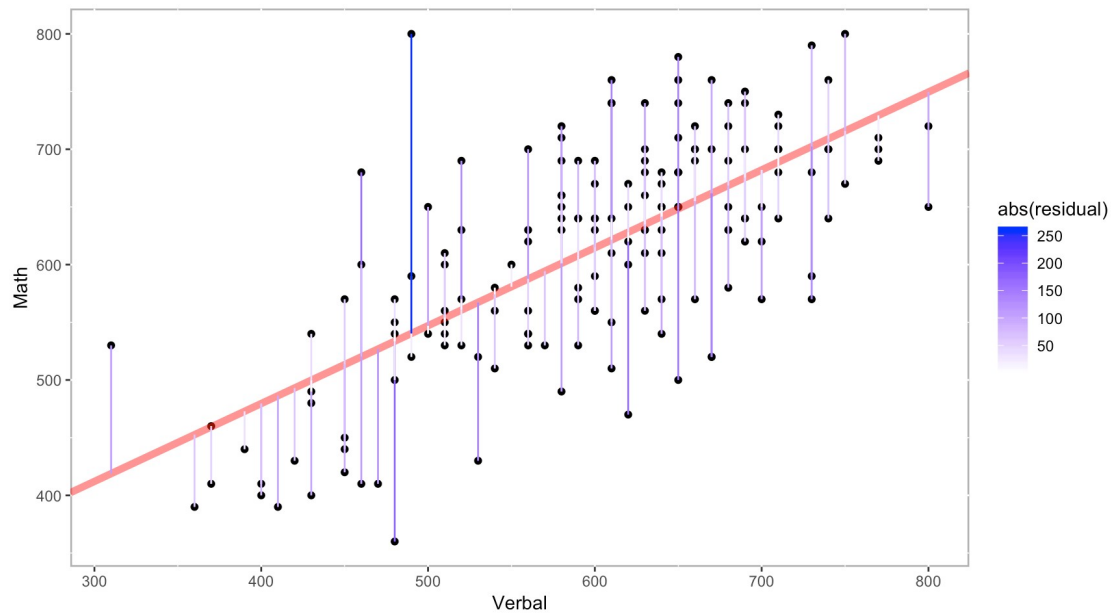
The residuals are simply the difference between the observed and predicted values.

```
sat$residual <- sat$Math - sat$Math.predicted
head(sat, n=4)
```

```
##   Verbal Math Sex   Verbal.z   Math.z Math.predicted  residual
## 1    450  450  F -1.47002058 -1.6518046      513.3378   -63.33782
## 2    640  540  F  0.43914539 -0.7346945      641.6020  -101.60203
## 3    590  570  M -0.06326671 -0.4289911      607.8483   -37.84829
## 4    400  400  M -1.97243268 -2.1613102      479.5841   -79.58408
```

Scatter Plot with Residuals

Plot our regression line with lines representing the residuals. The line of best fit minimizes the residuals.

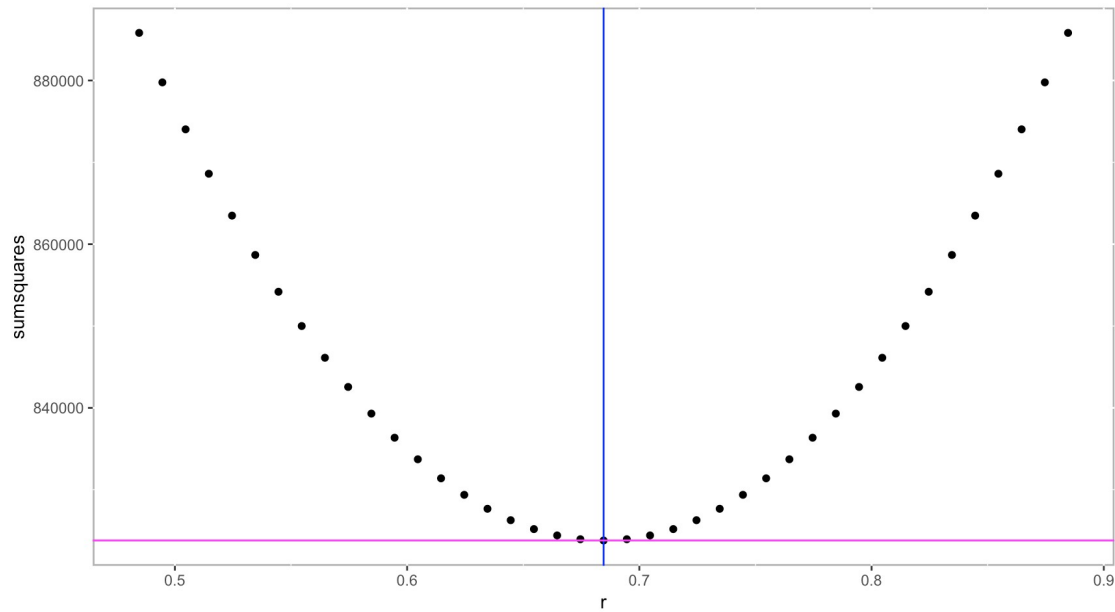


What does it mean to minimize the sum of squared residuals?

To show that $m = r \frac{S_y}{S_x}$ minimizes the sum of squared residuals, this loop will calculate the sum of squared residuals for varying values of r above and below the calculated value.

```
results <- data.frame(r=seq(r - .2, r + .2, by=.01),
                      m=as.numeric(NA),
                      b=as.numeric(NA),
                      sumsquares=as.numeric(NA))
for(i in 1:nrow(results)) {
  results[i,]$m <- results[i,]$r * (mathSD / verbalSD)
  results[i,]$b <- mathMean - results[i,]$m * verbalMean
  predicted <- results[i,]$m * sat$Verbal + results[i,]$b
  residual <- sat$Math - predicted
  sumsquares <- sum(residual^2)
  results[i,]$sumsquares <- sum(residual^2)
}
```

Plot the sum of squared residuals for different slopes (i.e. r 's). The vertical line corresponds to the r (slope) calculated above and the horizontal line corresponds the sum of squared residuals for that r . This should have the smallest sum of squared residuals.

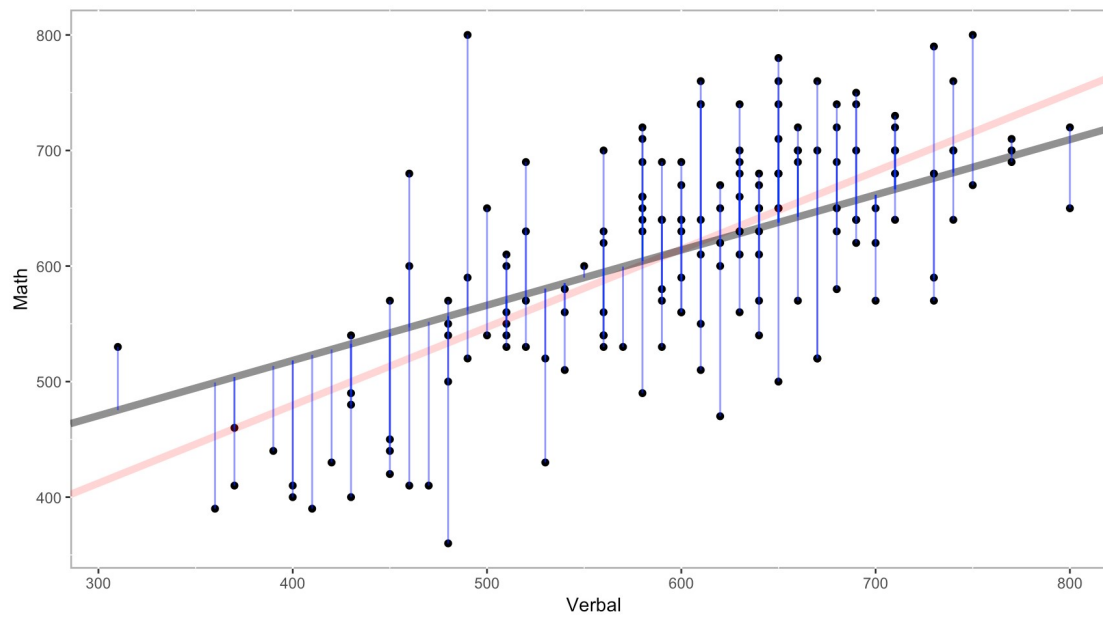


Example of a "bad" model

To exemplify how the residuals change, the following scatter plot picks one of the "bad" models and plot that regression line with the original, best fitting line. Take particular note how the residuals would be less if they ended on the red line (i.e. the better fitting model). This is particularly evident on the far left and far right, but is true across the entire range of values.

```
b.bad <- results[1,]$b  
m.bad <- results[1,]$m  
sat$predicted.bad <- m.bad * sat$Verbal + b.bad
```

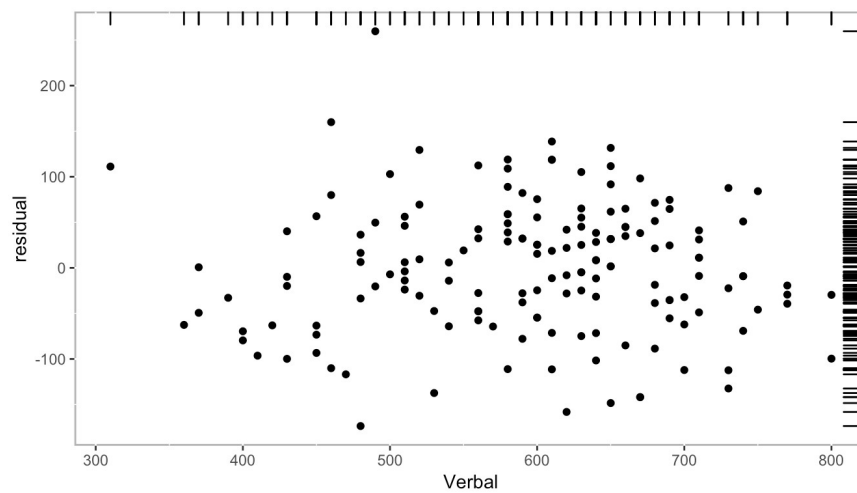

Example of a "bad" model



Residual Plot

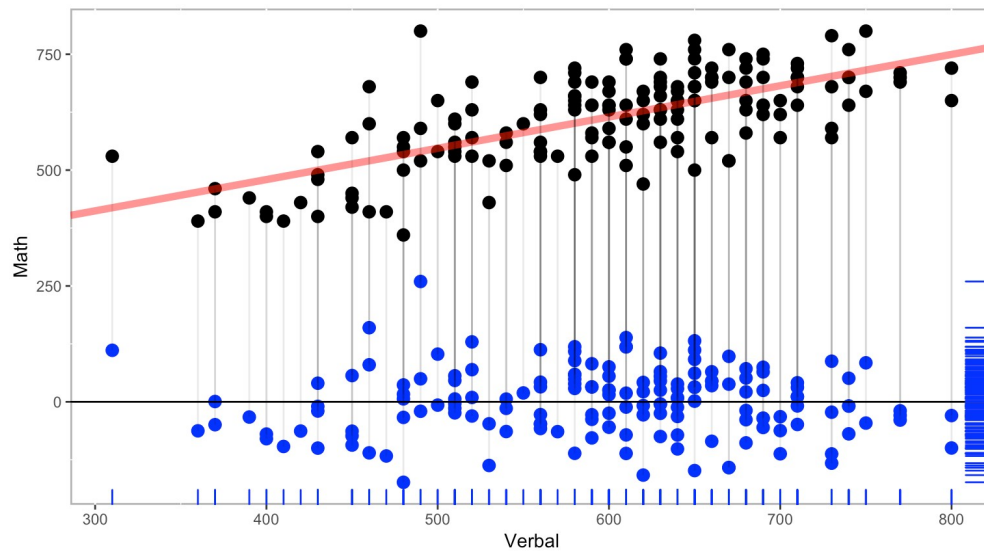
Next, we'll plot the residuals with the independent variable. In this plot we expect to see no pattern, bending, or clustering if the model fits well. The rug plot on the right and top given an indication of the distribution. Below, we will also examine the histogram of residuals.

```
ggplot(sat, aes(x=Verbal, y=residual)) + geom_point() + geom_rug(sides='rt')
```



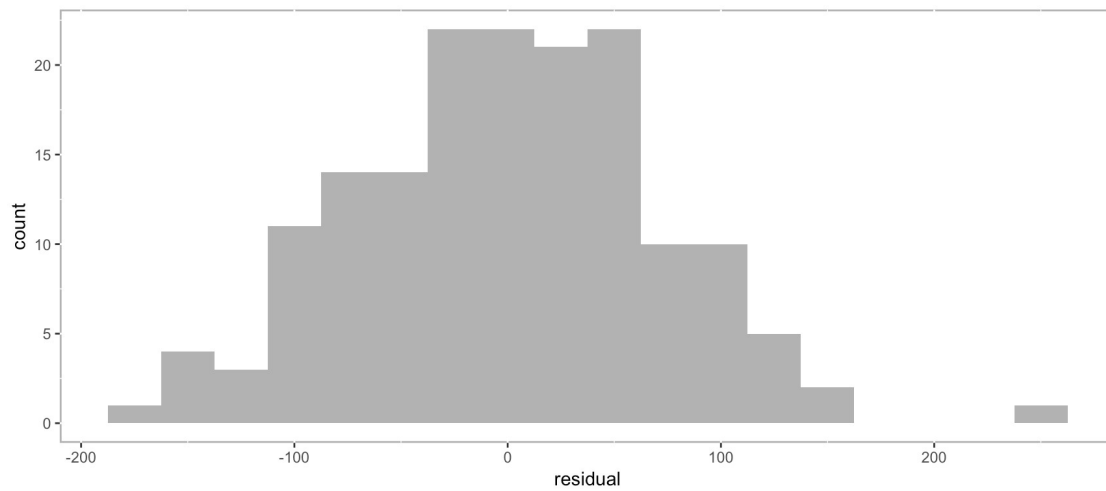
Scatter and Residual Plot, Together

In an attempt to show the relationship between the predicted value and the residuals, this figure combines both the basic scatter plot with the residuals. Each Math score is connected with the corresponding residual point.



Histogram of residuals

```
ggplot(sat, aes(x=residual)) + geom_histogram(alpha=.5, binwidth=25)
```



Calculate R^2

```
r ^ 2
```

```
## [1] 0.4686855
```

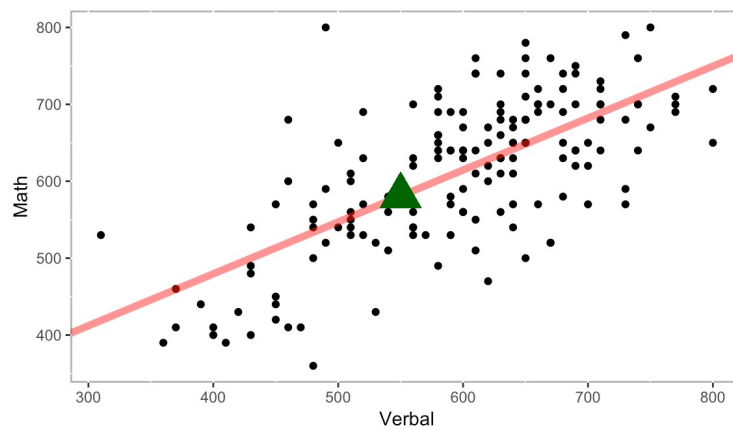
This model accounts for 46.9% of the variance math score predicted from verbal score.

Prediction

Now we can predict Math scores from new Verbal.

```
newX <- 550  
(newY <- newX * m + b)
```

```
## [1] 580.8453
```



Using R's built in functionality for linear modeling

The `lm` function in R will calculate everything above for us in one command.

```
sat.lm <- lm(Math ~ Verbal, data=sat)
summary(sat.lm)

##
## Call:
## lm(formula = Math ~ Verbal, data = sat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -173.590  -47.596    1.158   45.086  259.659
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  209.55417   34.34935   6.101 7.66e-09 ***
## Verbal        0.67507    0.05682  11.880 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.75 on 160 degrees of freedom
## Multiple R-squared:  0.4687, Adjusted R-squared:  0.4654
## F-statistic: 141.1 on 1 and 160 DF,  p-value: < 2.2e-16
```

Predicted Values, Revisited

We can get the predicted values and residuals from the `lm` function

```
sat.lm.predicted <- predict(sat.lm)
sat.lm.residuals <- resid(sat.lm)
```

Confirm that they are the same as what we calculated above.

```
head(cbind(sat.lm.predicted, sat$Math.predicted), n=4)
```

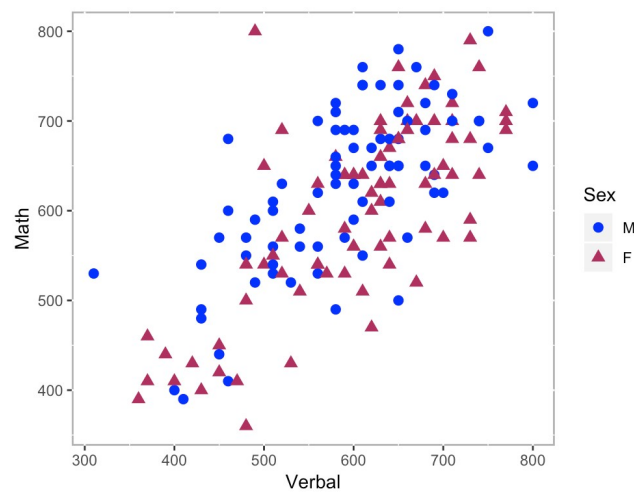
```
##    sat.lm.predicted
## 1      513.3378 513.3378
## 2      641.6020 641.6020
## 3      607.8483 607.8483
## 4      479.5841 479.5841
```

```
head(cbind(sat.lm.residuals, sat$residual), n=4)
```

```
##    sat.lm.residuals
## 1      -63.33782 -63.33782
## 2     -101.60203 -101.60203
## 3     -37.84829 -37.84829
## 4     -79.58408 -79.58408
```

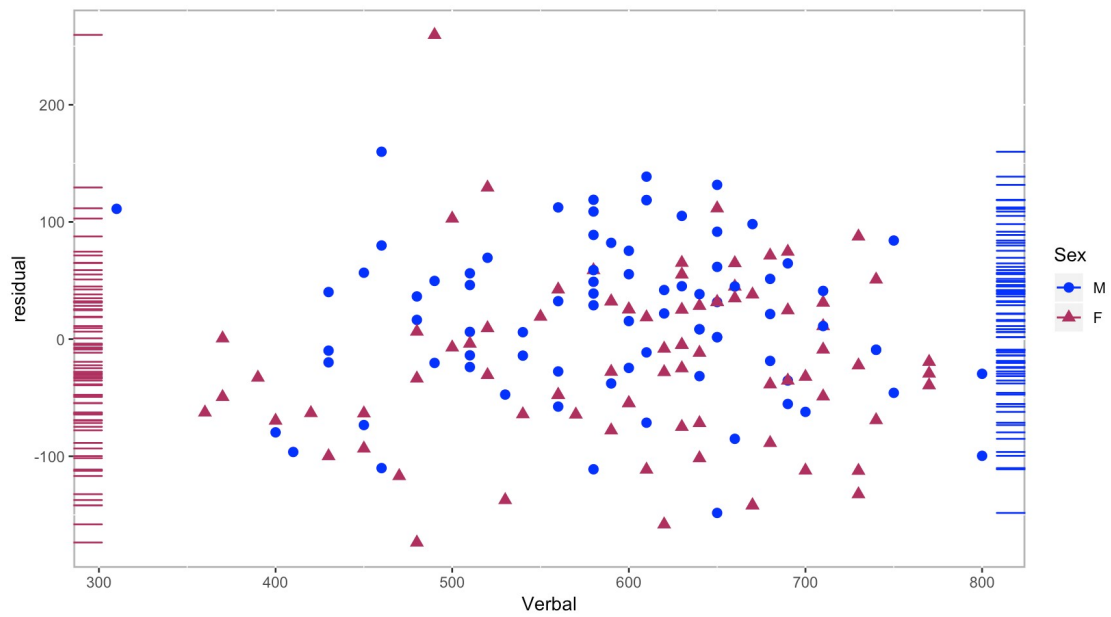

Re-evaluating the Residuals – Implications for Grouping Variables

First, let's look at the scatter plot but with a gender indicator.



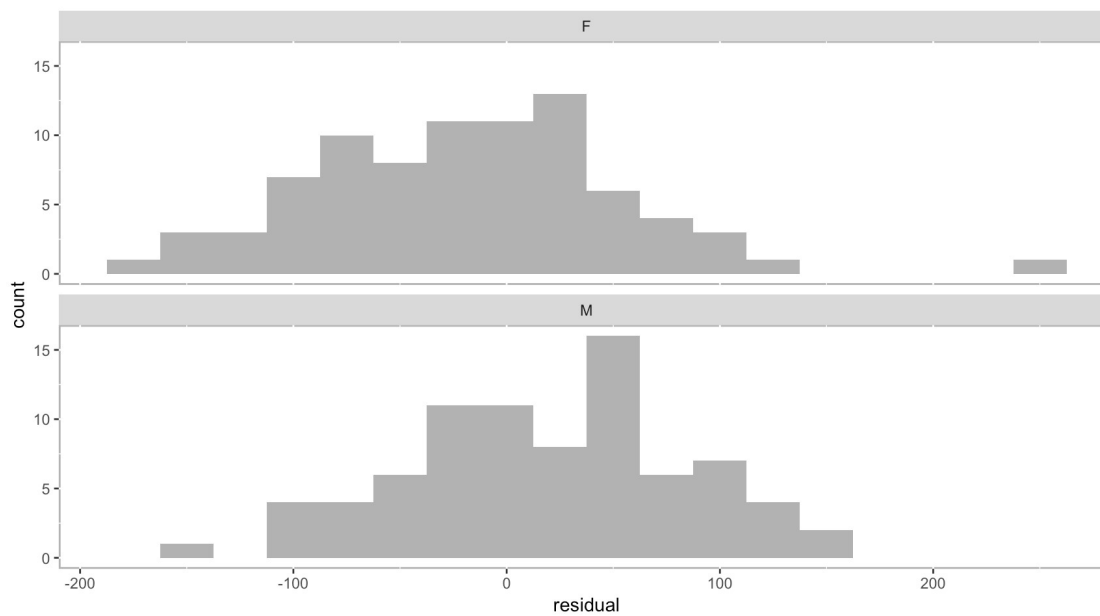
Residual Plot by Gender

And also the residual plot with an indicator for gender.



Histograms

The histograms also show that the distribution are different across gender.



Upon careful examination of these two figures, there is some indication there may be a difference between genders. In the scatter plot, it appears that there is a cluster of males towards the top left and a cluster of females towards the right. The residual plot also shows a cluster of males on the upper left of the cluster as well as a cluster of females to the lower right. Perhaps estimating two separate models would be more appropriate.

To start, we create two data frames for each gender.

```
sat.male <- sat[sat$Sex == 'M',]  
sat.female <- sat[sat$Sex == 'F',]
```

Descriptive Statistics

Calculate the mean for Math and Verbal for both males and females.

```
(male.verbal.mean <- mean(sat.male$Verbal))  
  
## [1] 590.375  
  
(male.math.mean <- mean(sat.male$Math))  
  
## [1] 626.875  
  
(female.verbal.mean <- mean(sat.female$Verbal))  
  
## [1] 602.0732  
  
(female.math.mean <- mean(sat.female$Math))  
  
## [1] 597.6829
```

Two Regression Models

Estimate two linear models for each gender.

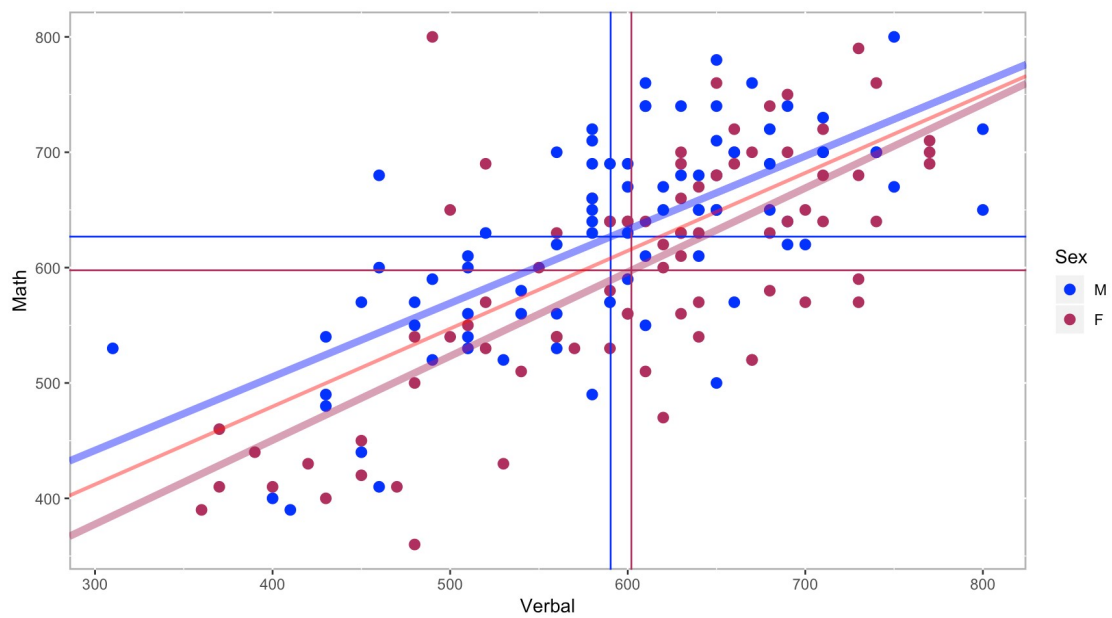
```
sat.male.lm <- lm(Math ~ Verbal, data=sat.male)
sat.female.lm <- lm(Math ~ Verbal, data=sat.female)
sat.male.lm
```

```
##
## Call:
## lm(formula = Math ~ Verbal, data = sat.male)
##
## Coefficients:
## (Intercept)      Verbal
##    250.1452      0.6381
```

```
sat.female.lm
```

```
##
## Call:
## lm(formula = Math ~ Verbal, data = sat.female)
##
## Coefficients:
## (Intercept)      Verbal
##    158.9965      0.7286
```

We do in fact find that the intercepts and slopes are both fairly different. The figure below adds the regression lines to the scatter plot.



$$R^2$$

Let's compare the R^2 for the three models.

```
cor(sat$Verbal, sat$Math) ^ 2
```

```
## [1] 0.4686855
```

```
cor(sat.male$Verbal, sat.male$Math) ^ 2
```

```
## [1] 0.4710744
```

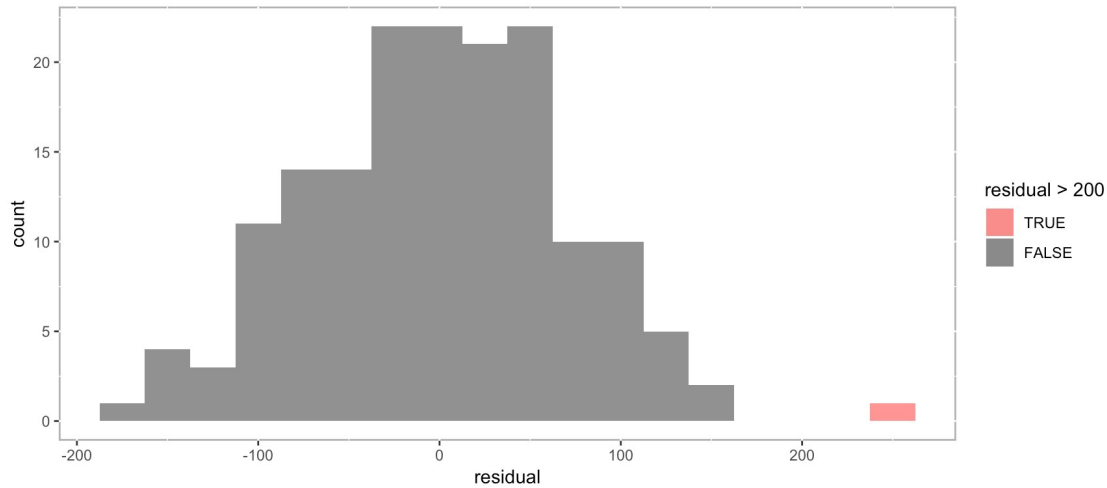
```
cor(sat.female$Verbal, sat.female$Math) ^ 2
```

```
## [1] 0.5137626
```

The R^2 for the full model accounts for approximately 46.9% of the variance. By estimating separate models for each gender we can account for 47.1% and 51.4% of the variance for males and females, respectively.

Examining Possible Outliers

Re-examining the histogram of residuals, there is one data point with a residual higher than the rest. This is a possible outlier. In this section we'll examine how that outlier may impact our linear model.

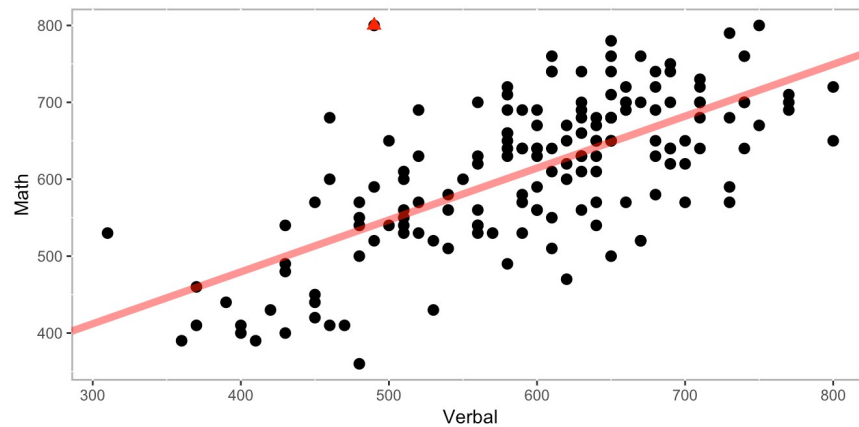


Possible Outlier

We can extract that record from our data frame. We can also highlight that point on the scatter plot.

```
sat.outlier <- sat[sat$residual > 200,]  
sat.outlier
```

```
##      Verbal Math Sex  Verbal.z   Math.z Math.predicted residual predicted.bad  
## 162    490  800   F -1.068091  1.914735    540.3408  259.6592    561.3041
```



We see that excluding this point changes model slightly. With the outlier included we can account for 45.5% of the variance and by excluding it we can account for 47.9% of the variance. Although excluding this point improves our model, this is an insufficient enough reason to do so. Further explanation is necessary.

```
(sat.lm <- lm(Math ~ Verbal, data=sat))

##
## Call:
## lm(formula = Math ~ Verbal, data = sat)
##
## Coefficients:
## (Intercept)      Verbal
##    209.5542      0.6751

(sat.lm2 <- lm(Math ~ Verbal, data=sat[sat$residual < 200,]))

##
## Call:
## lm(formula = Math ~ Verbal, data = sat[sat$residual < 200, ])
##
## Coefficients:
## (Intercept)      Verbal
##    197.4697      0.6926
```

R^2 with and without the outlier

```
sat.lm$coefficients[2] ^ 2
```

```
##      Verbal  
## 0.455726
```

```
sat.lm2$coefficients[2] ^ 2
```

```
##      Verbal  
## 0.4797177
```

More outliers

For the following two examples, we will add outliers to examine how they would effect our models. In the first example, we will add an outlier that is close to our fitted model (i.e. a small residual) but lies far away from the cluster of points. As we can see below, this single point increases our R^2 by more than 5%.

```
outX <- 1200
outY <- 1150
sat.outlier <- rbind(sat[,c('Verbal', 'Math')], c(Verbal=outX, Math=outY))
```

Regression Models

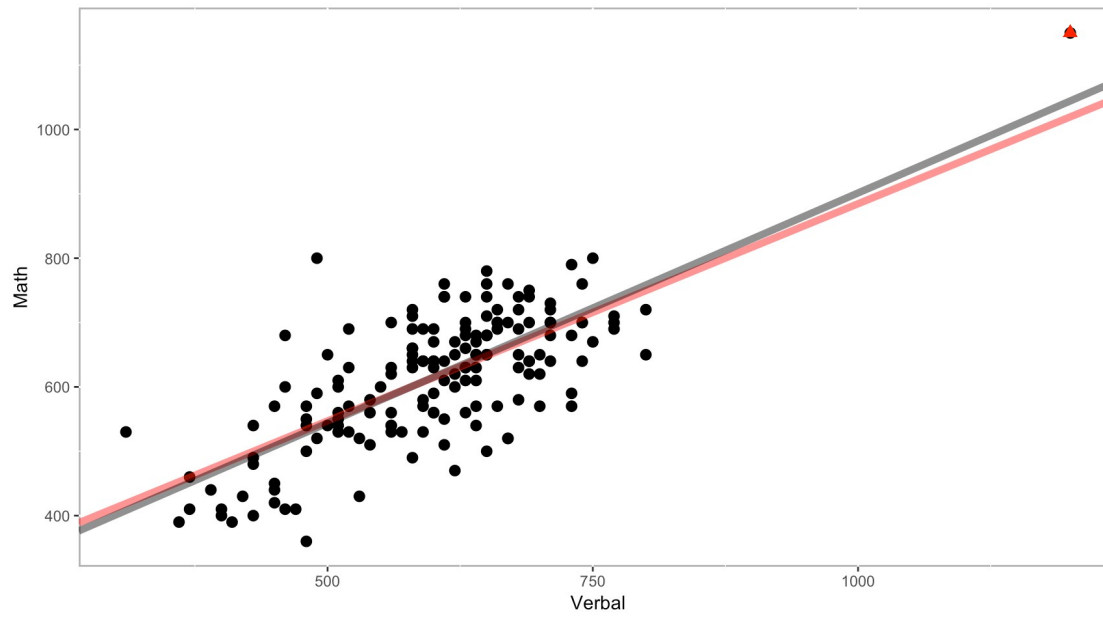
```
(sat.lm <- lm(Math ~ Verbal, data=sat))

##
## Call:
## lm(formula = Math ~ Verbal, data = sat)
##
## Coefficients:
## (Intercept)      Verbal
##    209.5542      0.6751

(sat.lm2 <- lm(Math ~ Verbal, data=sat.outlier))

##
## Call:
## lm(formula = Math ~ Verbal, data = sat.outlier)
##
## Coefficients:
## (Intercept)      Verbal
##    186.372      0.715
```

Scatter Plot



$$R^2$$

```
unname(sat.lm$coefficients[2] ^ 2)
```

```
## [1] 0.455726
```

```
unname(sat.lm2$coefficients[2] ^ 2)
```

```
## [1] 0.5112893
```

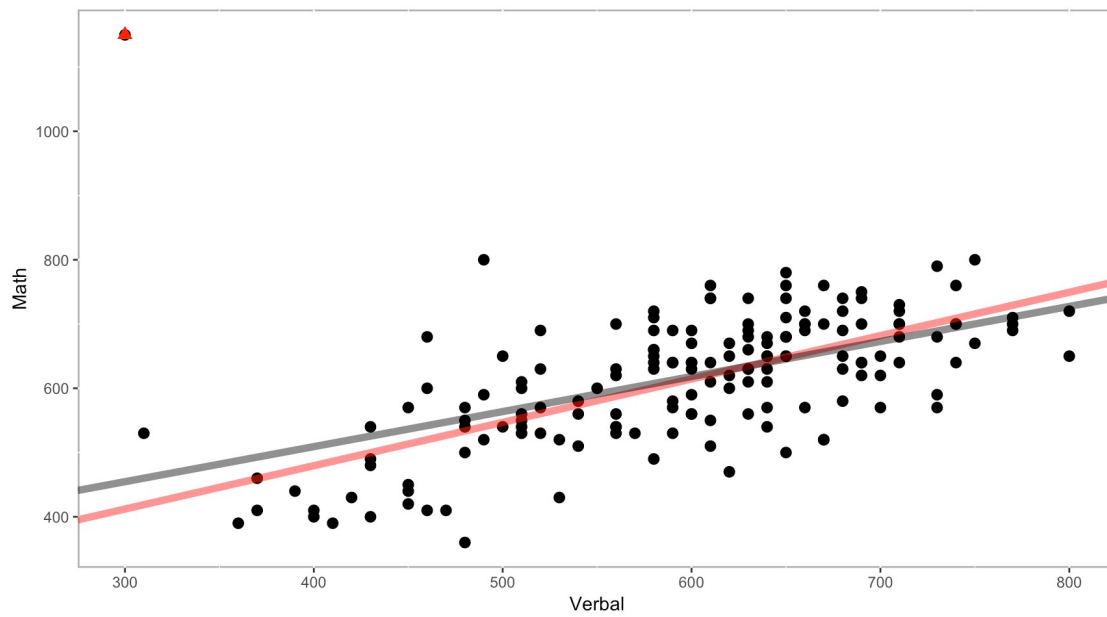

Outliers can have the opposite effect too. In this example, our R^2 is decreased by almost 16%.

```
outX <- 300
outY <- 1150
sat.outlier <- rbind(sat[,c('Verbal', 'Math')], c(Verbal=outX, Math=outY))
(sat.lm <- lm(Math ~ Verbal, data=sat))
```

```
##
## Call:
## lm(formula = Math ~ Verbal, data = sat)
##
## Coefficients:
## (Intercept)      Verbal
##    209.5542      0.6751
```

```
(sat.lm2 <- lm(Math ~ Verbal, data=sat.outlier))
```

```
##
## Call:
## lm(formula = Math ~ Verbal, data = sat.outlier)
##
## Coefficients:
## (Intercept)      Verbal
##    290.8915      0.5459
```



$$R^2$$

```
unname(sat.lm$coefficients[2] ^ 2)
```

```
## [1] 0.455726
```

```
unname(sat.lm2$coefficients[2] ^ 2)
```

```
## [1] 0.2979727
```