```r
library(ggplot2)
#install.packages("lme4")
library(lme4)
```

```
Loading required package: Matrix
```

```r
#install.packages("DHARMa")
library(DHARMa)
```

```
This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'I
```

```r
library(quantreg)
```

```
Loading required package: SparseM


Attaching package: 'SparseM'

The following object is masked from 'package:base':

    backsolve
```

```r
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Methods

Design Matrix

The design matrix is defined to be a matrix $\mathbf{X}$ such that $\mathbf{X}_{ij}$ (the j^{th}) column of the i^{th} row of $\mathbf{X}$) represents the value of the j^th variable associated with the i^{th} variable object.

A regression model may be represent via matrix multiplication as

$$y = \mathbf{X}\beta + e$$

where X is the design matrix, β is a vector of the model's coefficient (one for each variable), e is a vector of random errors with a mean zero, and y is the vector outputs for each object.

Ordinary least squares

Ordinary least squares model or OLS, works by creating a line through the data points. Then it calculates the difference between each prediction and observation (residual). And it tries to minimize the squared value of the residuals. The ordinary least squares is defined by:

$$y_i = \alpha + \beta x_i + \varepsilon_i.$$

The least squares estimates in this case are given by simple formulas

$$\hat{\beta} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Quantile regression

In Koneker's 1978 paper, the θ^{th} Quantile regression is defined as any solution to the following problem:

$$\min_{b\in\mathbf{R}^K}\left[\sum_{t\in\{t:y_t\geqslant x_tb\}}\theta\left|y_t-x_tb\right|+\sum_{t\in\{t:y_t<x_tb\}}(1-\theta)\left|y_t-x_tb\right|\right] \tag{0.1}$$

where

$$\{x_t : t = 1, ..., T\}$$

denotes a sequence (row) of K-vectors of a known design matrix and

$$\{y_t : t = 1, ..., T\}$$

is a random sample on the regression process $u_t = y_t - x_t\beta$ [1].

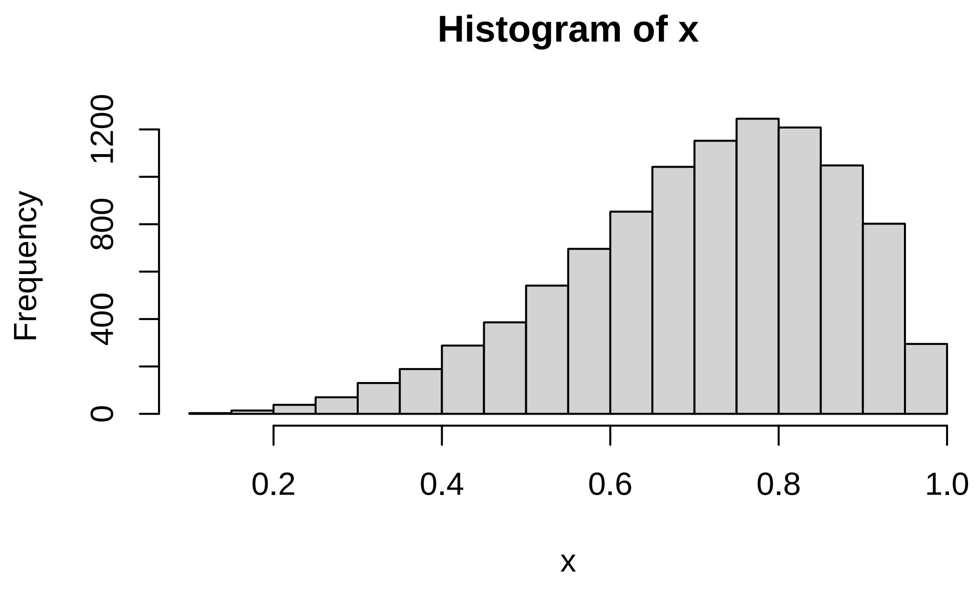
The difference

OLS regression minimizes the sum of squares of residuals, but QR minimizes the sum of weighted absolute errors. The idea behind this is if regression process underestimates or overestimates, the proportion of positive and negative residuals will shift. By using these weights, the bias of the model is reduced and the centrality of the model is able to be more accurately estimated.

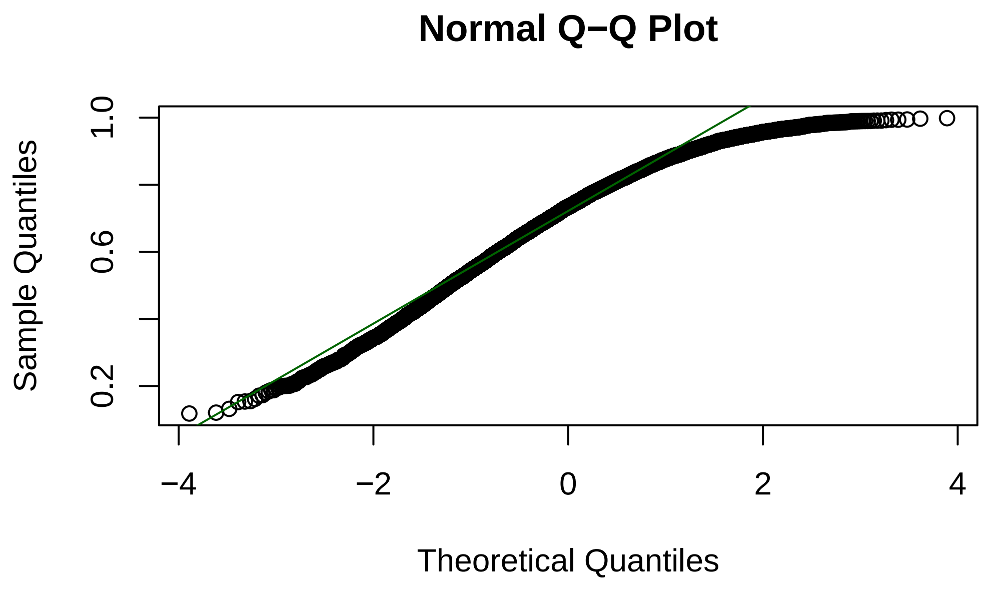
Intuition

Right Skew

```
x <- rbeta(10000,5,2)
hist(x)
```



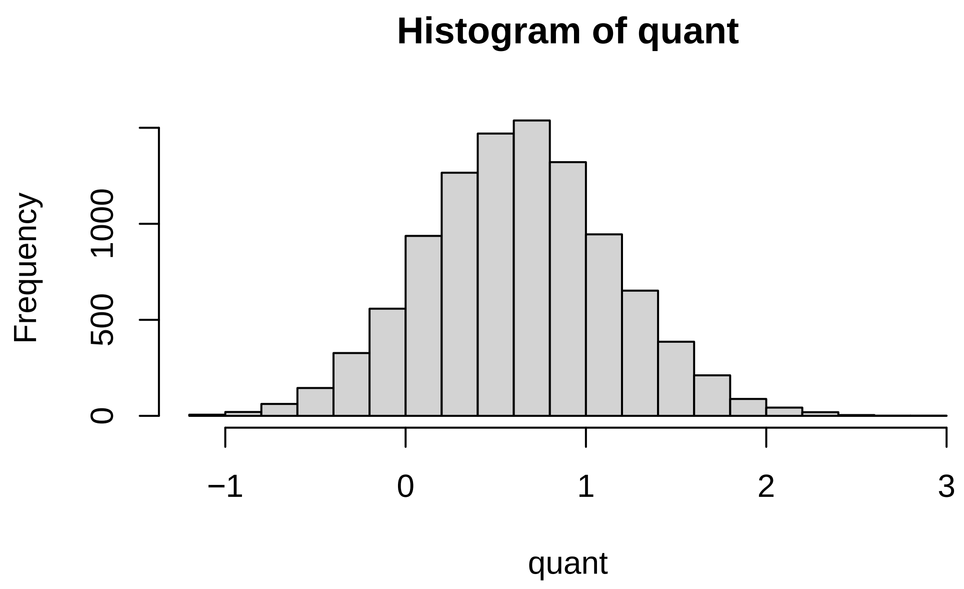
```
qqnorm(x)
qqline(x, col= "darkgreen")
```



```r
quant <- qnorm(x)
#quant
table(sign(quant))
```

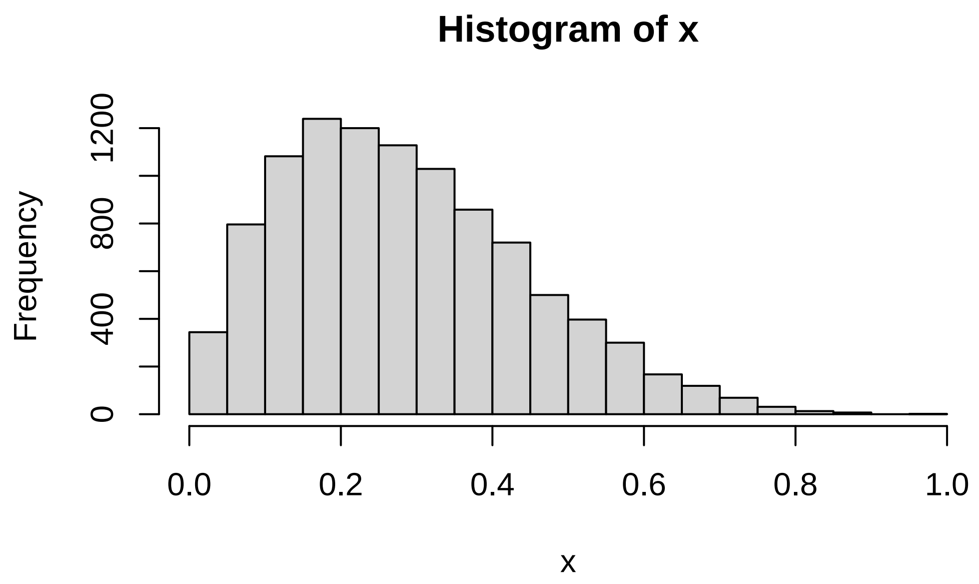
```
  -1    1
1118 8882
```

```r
hist(quant)
```



Left Skew

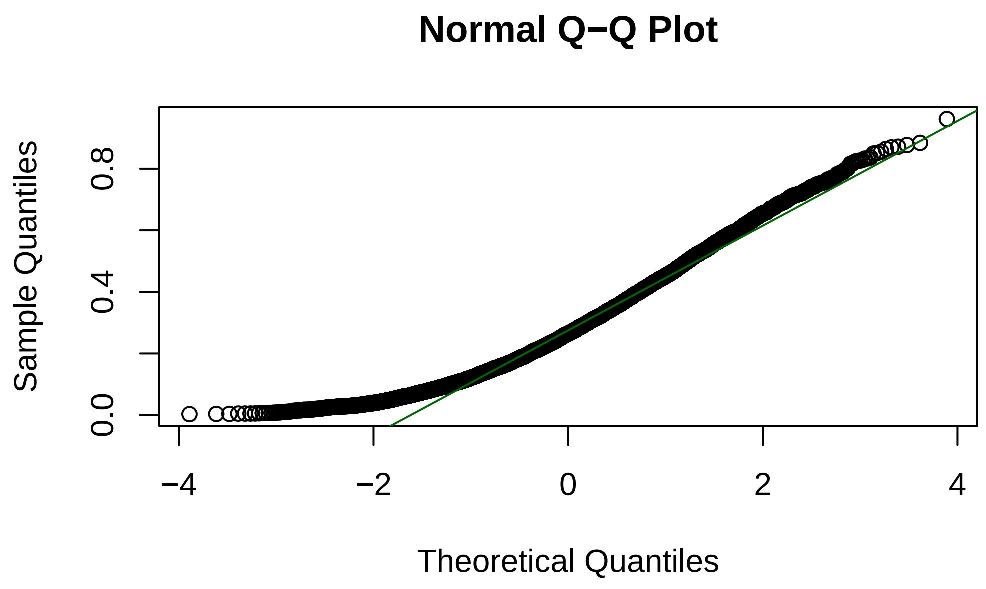
```r
x <- rbeta(10000,2,5)
hist(x)
```



```r
qqnorm(x)
table(sign(x))
```

```
    1
10000
```

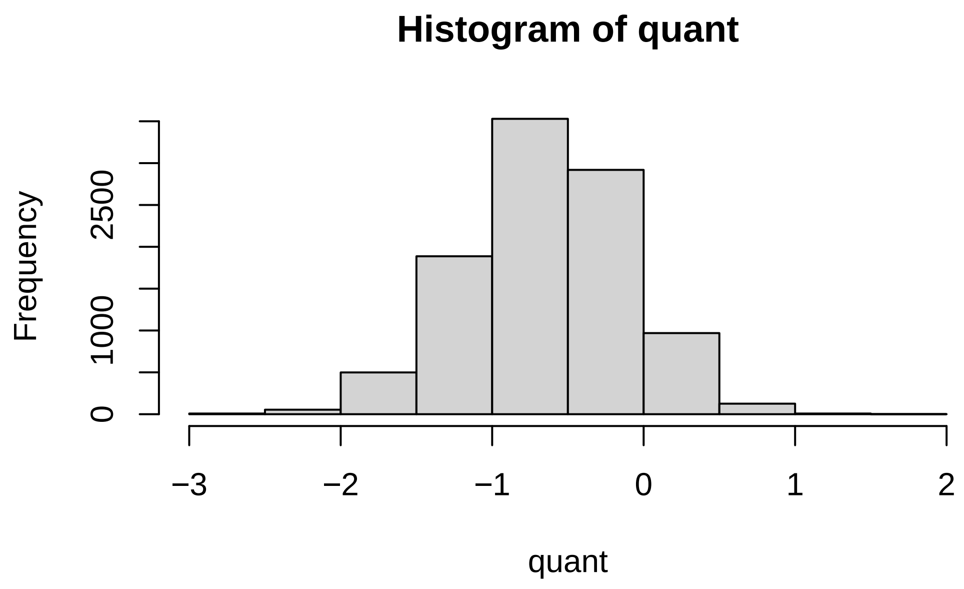
```r
qqline(x, col= "darkgreen")
```



```r
quant <- qnorm(x)
#quant
table(sign(quant))
```

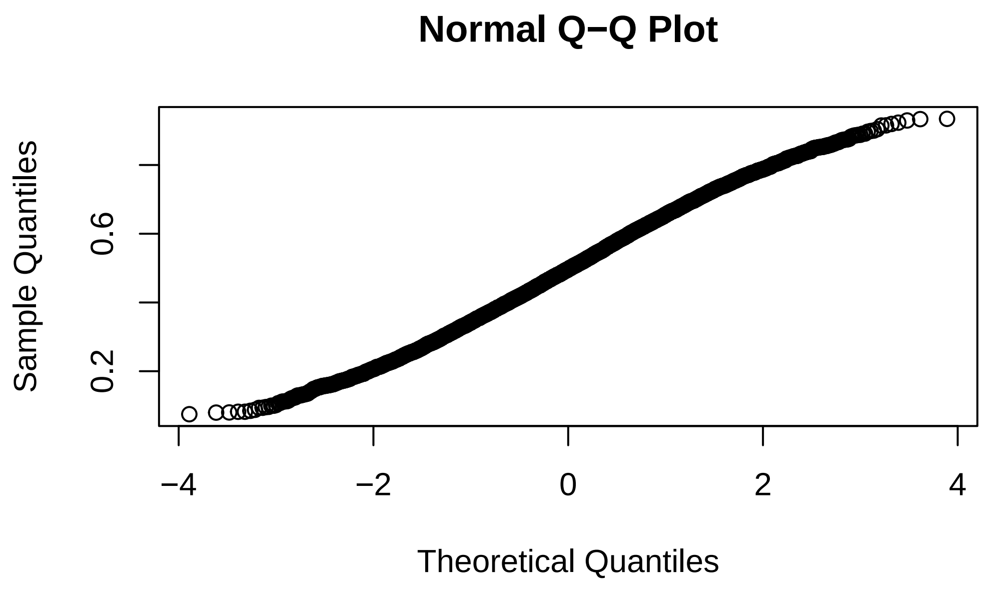
```
  -1    1
8896 1104
```

```r
hist(quant)
```



No Skew

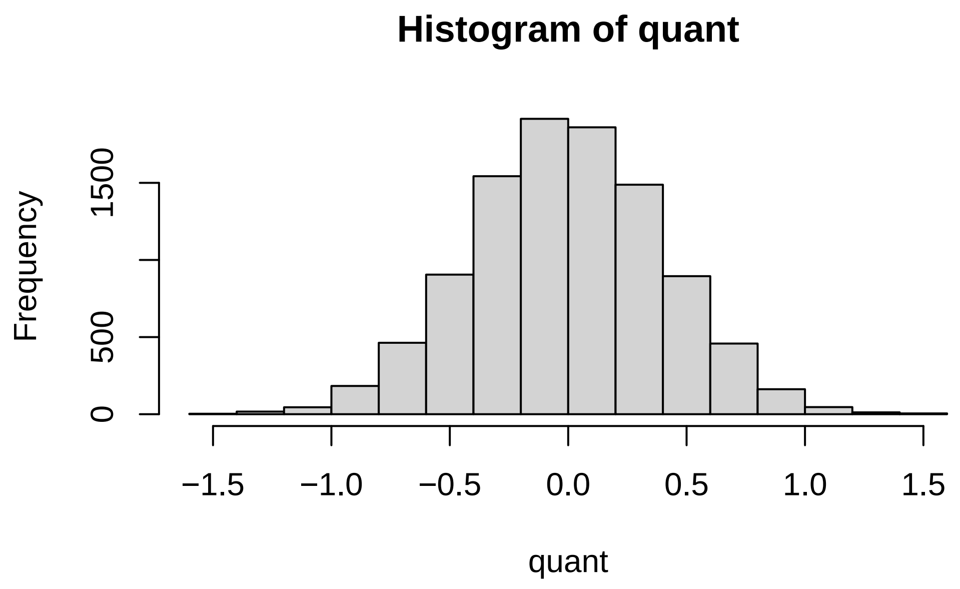
```
x <- rbeta(10000,5,5)
qqnorm(x)
```



```r
quant <- qnorm(x)
#quant
table(sign(quant))
```

```
  -1    1 
5074 4926 
```

```r
hist(quant)
```



As we can see, depending on the distribution of the residuals, the theoretical quantiles will have differing proportions of positive and negative values. In the right skew example, there are 1130 negative data points and 8870 positive data points. For instance, let's look to see if we were interested in the 90th percentile, there would be 8870 points that would be multiplied by 0.9, and 1130 points that would be multiplied by 0.1. Thus the burden of minimization is going to be cenetered on finding betas that reduce the sum of those 8870 points to the lowest possible number.

Example

Create Data

```
#make this example reproducible
set.seed(0)

#create data frame
hours <- runif(100, 1, 10)
score <- 60 + 2*hours + rnorm(100, mean=0, sd=.45*hours)
df <- data.frame(hours, score)
```

```r
#view first six rows
head(df)
```

```
     hours    score
1 9.070275 79.22682
2 3.389578 66.20457
3 4.349115 73.47623
4 6.155680 70.10823
5 9.173870 78.12119
6 2.815137 65.94716
```

Create Model

```r
#fit model
model_90 <- rq(score ~ hours, data = df, tau = 0.9)
summary(model_90)
```

```
Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be
nonunique


Call: rq(formula = score ~ hours, tau = 0.9, data = df)

tau: [1] 0.9

Coefficients:
            coefficients lower bd upper bd
(Intercept) 60.25185     59.27193 62.56459
hours        2.43746      1.98094  2.76989
```

```r
model_50 <- rq(score ~ hours, data = df, tau = 0.5)
summary(model_50)
```

```
Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be
nonunique


Call: rq(formula = score ~ hours, tau = 0.5, data = df)

tau: [1] 0.5
```

```
Coefficients:
            coefficients lower bd upper bd
(Intercept) 60.20392     59.20769 60.46949
hours        1.92357      1.87038  2.10630
```

```r
model_10 <- rq(score ~ hours, data = df, tau = 0.1)
summary(model_10)
```

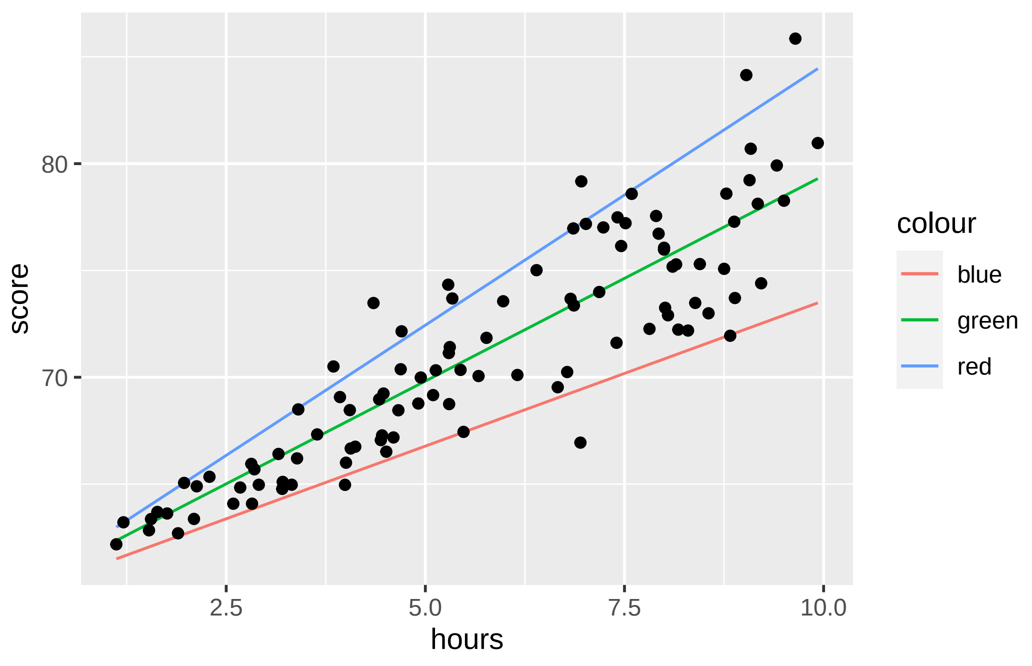
```
Warning in rq.fit.br(x, y, tau = tau, ci = TRUE, ...): Solution may be
nonunique
```

```
Call: rq(formula = score ~ hours, tau = 0.1, data = df)

tau: [1] 0.1

Coefficients:
            coefficients lower bd upper bd
(Intercept) 59.97472     59.34367 60.30126
hours        1.49922      1.36072  1.59683
```

```r
df %>% ggplot(aes(x=hours, y=score)) +
  geom_line(aes(color="red", y = 60.25185 + 2.43746*hours)) +
  geom_line(aes(color="green", y= 60.20392 + 1.92357*hours)) +
  geom_line(aes(color="blue", y=59.97472 + 1.36072*hours)) +
  geom_point()
```



```
#view summary of model
#summary(model)
#table(sign(resid(model)))
```

Graphic

Evaluation metrics

Mean absolute error

The mean absolute error (MAE) is the average magnitude of the errors of the values predicted by the regression and the actual observed values for the response variable. Because it is a simple average, all errors have the same weight, there are no penalties for different magnitude deviations [2]. MAE assumes that the errors are normally distributed, if the error distribution was non-normal, the average may not be a good measure of centrality and can paint a false picture of the goodness-of-fit of the regression curve. MAE also assumes that the errors are unbiased. While the average magnitude of the errors is expected to be non-zero (unless the regression is a perfect fit) the average of the residuals, i.e., the deviation of the predicted value from the actual value, considering underestimation and overestimation. This means on average the regression curve does not over or underestimate.

$$\text{MAE } = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

Root mean squared error

It calculates the differences between the predictions and the actual observations (residuals) and then gets their quadratic mean for each. This type of error gives a larger penalty for larger errors [2]. This error also assumes that the errors are unbiased and that they follow a normal distribution. This gives a picture of the size of residuals in comparison to the regression line.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}e_i^2}$$

Variance of error

It is a measure of how spread all the errors are from the mean of all errors.

$$\text{Var}(e) = \frac{1}{n}\sum_{i=1}^{n}(e_i - \bar{e})^2$$

Min/max error

A measure of the maximum residual for a prediction and the minimum residual.

$$f : X \to \mathbb{R}, \text{ if } (\forall e \in X_{error}) f(e_i) \geq f(e)$$

$$f : X \to \mathbb{R}, \text{ if } (\forall e \in X_{error}) f(e_i) \leq f(e)$$

ANOVA test??