Skewed Distribution

```r
knitr::opts_chunk$set(echo = TRUE)
source("../Rcode/events.R")
```

Define 2 Groups

```r
n <- 20

#Group 1
AE.1 <- AE(200,30,20) # time to initial event , duration , gap in between
AE.2 <- AE(160,40,15)
AE.3 <- AE(180,30,4)

AEs.1 <- list(AE.1,AE.2,AE.3)

scores.1 <- c()
for(i in 1:n){
  AEs_.1  <- simulate_events(AEs.1,180)
  scores.1 <- c(scores.1,AEs_.1$Score)
}
scores.1
```

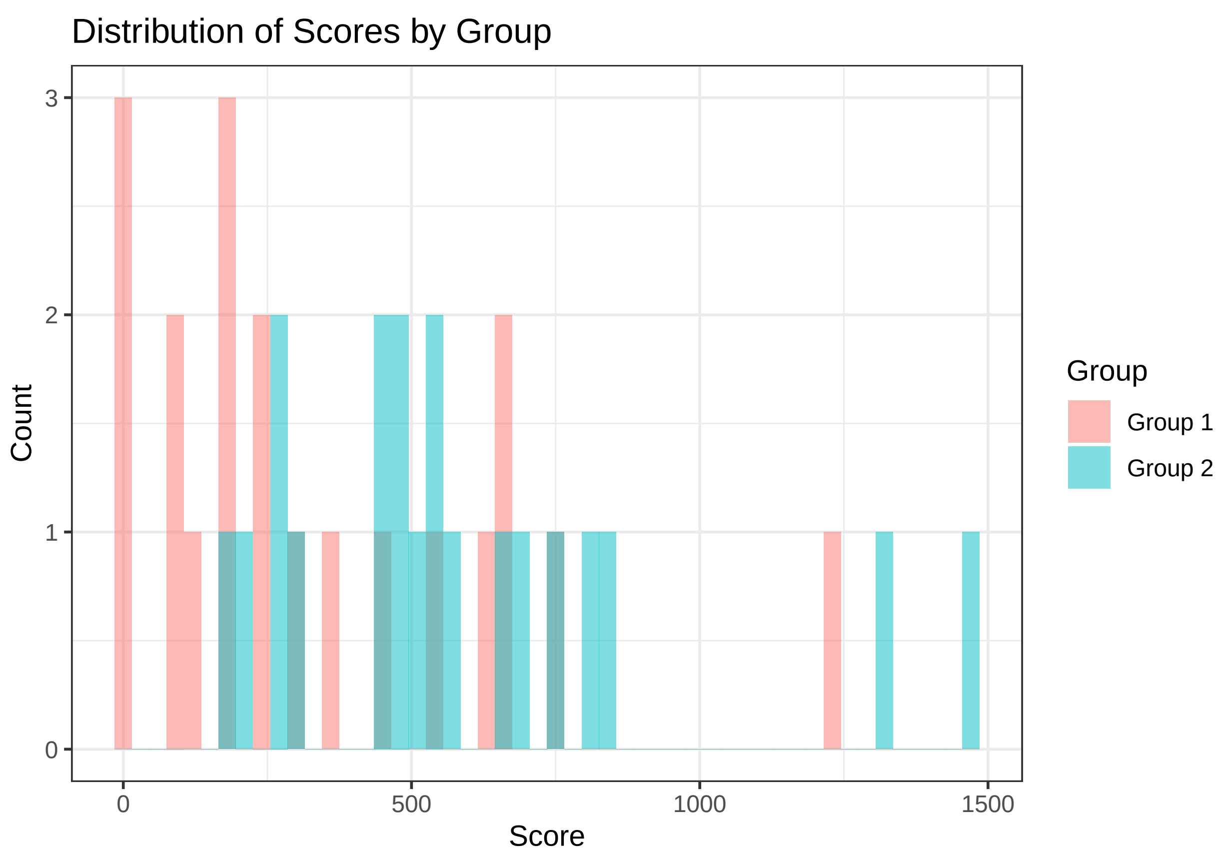
```
##  [1]  102.11500  535.69945  657.82669  347.34818   88.46190  250.23706
##  [7]  640.44639  193.35948    0.00000  438.47256 1234.83801  760.67754
## [13]  304.40449  662.69318  113.58929  194.47119    0.00000  175.34760
## [19]   13.75249  251.58692
```

```r
#Group 2
AE.1 <- AE(10,30,100)
AE.2 <- AE(20,40,90)
AE.3 <- AE(50,70,60)

AEs.2 <- list(AE.1,AE.2,AE.3)

scores.2 <- c()
for(i in 1:n){
  AEs_.2  <- simulate_events(AEs.2,180)
scores.2 <- c(scores.2,AEs_.2$Score)
}
scores.2
```

```
##  [1]  531.3950  204.3094  557.4056  435.9022  467.3668  694.7724 1473.9725
##  [8]  738.4302  649.2079  468.1019  836.5705 1305.2270  284.7441  462.5198
## [15]  314.3460  809.0746  534.6234  168.3328  255.1270  523.8731
```

```r
## put both scores in a df for with the groups as a factor variable
scores <- c(scores.1,scores.2)
groups <- factor(c(rep(1,length(scores.1)),rep(2,length(scores.2))))
score_data <- data.frame(scores,groups)

#score_data
```

```r
# plot the distriubtion of the scores

#install.packages("ggplot2")
library(ggplot2)

ggplot(score_data, aes(x = scores, fill = groups)) +
  geom_histogram(binwidth = 30, position = "identity", alpha =0.5) +
  labs(title = "Distribution of Scores by Group", x = "Score", y = "Count") +
  scale_fill_discrete(name = "Group", labels = c("Group 1", "Group 2")) +
  theme_bw()
```




```r
# First a simple linear regression model
lm_model <- lm(scores ~ groups, data = score_data)

summary(lm_model)
```

```
##
## Call:
```

```
## lm(formula = scores ~ groups, data = score_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -417.43 -237.55  -79.29  161.36  888.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   348.27      72.94   4.775 2.68e-05 ***
## groups2       237.50     103.15   2.302   0.0269 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 326.2 on 38 degrees of freedom
## Multiple R-squared:  0.1224, Adjusted R-squared:  0.09934
## F-statistic: 5.301 on 1 and 38 DF,  p-value: 0.02688
```

```r
# Fit a Tweedie regression model
#install.packages("tweedie")
#install.packages("statmod")

library(tweedie)
library(statmod)

tweedie_model <- glm(score_data$scores~score_data$groups, family =
           tweedie(var.power=1,
           link.power=0), control = glm.control(maxit = 100))

tweedie_model.2 <- glm(score_data$scores~score_data$groups, family =
           tweedie(var.power=1.5,
           link.power=0), control = glm.control(maxit = 100))

tweedie_model.3 <- glm(score_data$scores~score_data$groups, family =
           tweedie(var.power=2,
           link.power=0), control = glm.control(maxit = 100))

tweedie_model.4 <- glm(score_data$scores~score_data$groups, family =
           tweedie(var.power=1,
           link.power=1), control = glm.control(maxit = 100))
summary(tweedie_model.4)
```

```
##
## Call:
## glm(formula = score_data$scores ~ score_data$groups, family = tweedie(var.power = 1,
##     link.power = 1), control = glm.control(maxit = 100))
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         348.27      64.69   5.384 3.98e-06 ***
## score_data$groups2  237.50     105.94   2.242   0.0309 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Tweedie family taken to be 240.3226)
```

```
##
##     Null deviance: 10021.0  on 39  degrees of freedom
## Residual deviance:  8799.9  on 38  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 3
```

```
# Fit quantile regression model
#install.packages("quantreg")

library(quantreg)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
quantile_regression_model <- rq(score_data$scores~score_data$groups)
```

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

```
summary(quantile_regression_model, se="nid") #https://cran.r-project.org/web/packages/quantreg/vignette
```

```
##
## Call: rq(formula = score_data$scores ~ score_data$groups)
##
## tau: [1] 0.5
##
## Coefficients:
##                    Value     Std. Error t value   Pr(>|t|)
## (Intercept)        251.58692 105.93160    2.37499   0.02270
## score_data$groups2 272.28614 134.85345    2.01913   0.05057
```

AIC analysis

```
AIC(lm_model)
```

```
## [1] 580.4607
```

```
AICtweedie(tweedie_model,dispersion=1) # simple AIC is NAN
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 102.114996
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 535.699451
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 657.826693

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 347.348183

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 88.461898

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 250.237062

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 640.446391

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 193.359475

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 438.472564

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1234.838015

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 760.677543

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 304.404493

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 662.693175

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 113.589292

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 194.471188

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 175.347599

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 13.752494

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 251.586918

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 531.394983

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 204.309444

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 557.405601

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 435.902159

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 467.366828

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 694.772390

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1473.972491

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 738.430175
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 649.207943

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 468.101903

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 836.570509

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1305.226973

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 284.744135

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 462.519766

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 314.346048

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 809.074590

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 534.623423

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 168.332821

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 255.127045

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 523.873059

## [1] Inf
```

```
AICtweedie(tweedie_model.2,dispersion=1)
```

```
## [1] 923.5948
```

```
AICtweedie(tweedie_model.3,dispersion=1)
```

```
## [1] 573.0355
```

```
AICtweedie(tweedie_model.4,dispersion=1)
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 102.114996

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 535.699451

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 657.826693

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 347.348183

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 88.461898

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 250.237062
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 640.446391

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 193.359475

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 438.472564

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1234.838015

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 760.677543

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 304.404493

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 662.693175

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 113.589292

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 194.471188

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 175.347599

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 13.752494

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 251.586918

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 531.394983

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 204.309444

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 557.405601

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 435.902159

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 467.366828

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 694.772390

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1473.972491

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 738.430175

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 649.207943

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 468.101903

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 836.570509

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 1305.226973
```

```
## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 284.744135

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 462.519766

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 314.346048

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 809.074590

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 534.623423

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 168.332821

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 255.127045

## Warning in dpois(x = y/phi, lambda = mu/phi): non-integer x = 523.873059

## [1] Inf
```

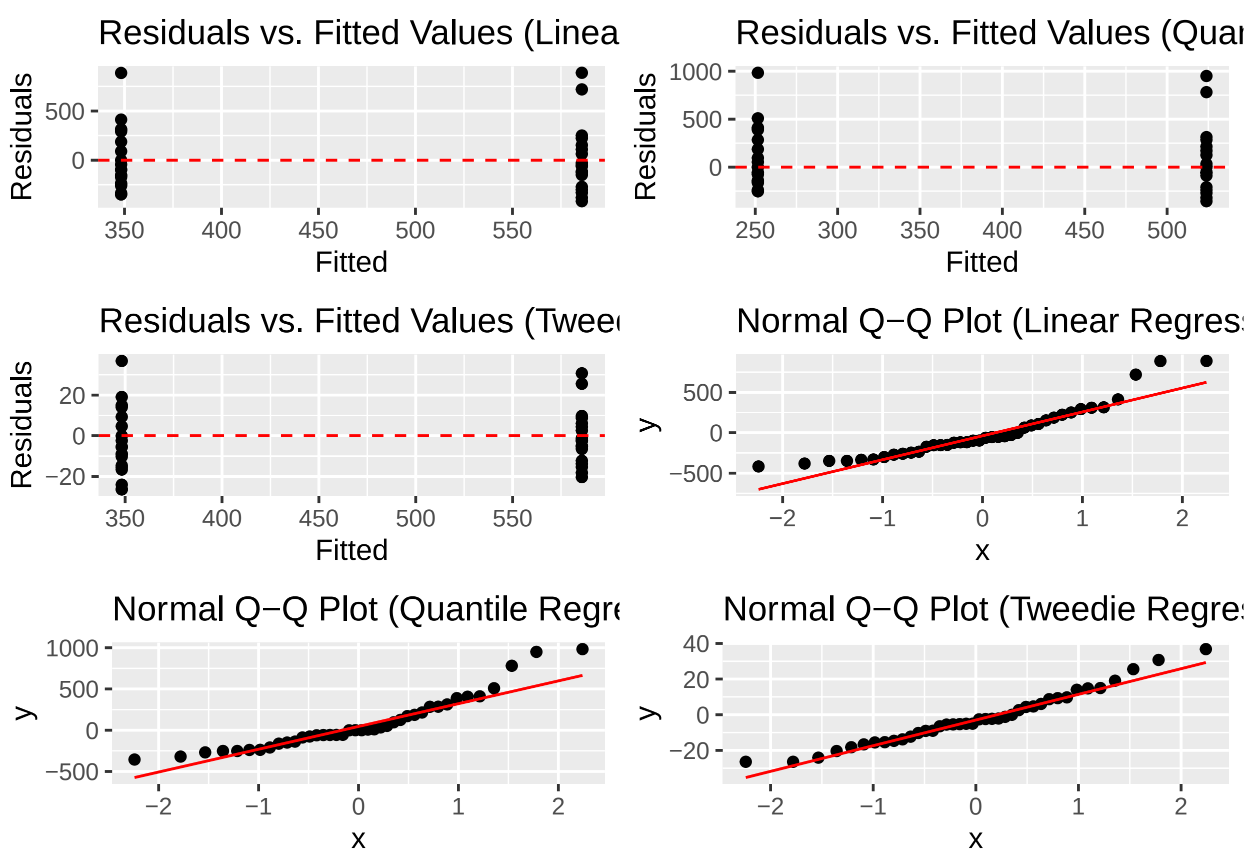
```
AIC(quantile_regression_model)[1]
```

```
## [1] 574.9206
```

residual analysis

```
lm_resid <- resid(lm_model)
quantile_resid <- resid(quantile_regression_model)
tweeie_resid <- resid(tweedie_model)

#install.packages("ggplot2")
#install.packages("gridExtra")

library(ggplot2)
library(gridExtra)

# Create residual vs. fitted value plots
residual_vs_fitted_lm <- ggplot(data = data.frame(Fitted = fitted(lm_model), Residuals = lm_resid), aes
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Fitted Values (Linear Regression)")

residual_vs_fitted_quantile <- ggplot(data = data.frame(Fitted = fitted(quantile_regression_model), Resi
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Fitted Values (Quantile Regression)")

residual_vs_fitted_tweedie <- ggplot(data = data.frame(Fitted = fitted(tweedie_model), Residuals = twee
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs. Fitted Values (Tweedie Regression)")
```

```r
# Create normal Q-Q plots
qq_plot_lm <- ggplot(data = data.frame(Residuals = lm_resid), aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Linear Regression)")

qq_plot_quantile <- ggplot(data = data.frame(Residuals = quantile_resid), aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Quantile Regression)")

qq_plot_tweedie <- ggplot(data = data.frame(Residuals = tweeie_resid), aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (Tweedie Regression)")

# Combine the plots
grid.arrange(residual_vs_fitted_lm, residual_vs_fitted_quantile, residual_vs_fitted_tweedie, qq_plot_lm
```




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

```
## Loading required package: zoo

##
```

```
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Perform a Likelihood Ratio Test
lrt_result <- lrtest(lm_model, tweedie_model.2) #quantile_regression_model not applicable
print("Likelihood Ratio Test:")
```

```
## [1] "Likelihood Ratio Test:"
```

```
print(lrt_result)
```

```
## Likelihood ratio test
##
## Model 1: scores ~ groups
## Model 2: score_data$scores ~ score_data$groups
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1   3 -287.23
## 2   2          -1
```

```
# for nested models only
#deviance_lm <- deviance(lm_model)
#deviance_tweedie <- deviance(tweedie_model.2)

#print("Deviance for Linear Model:")
#print(deviance_lm)
#print("Deviance for Tweedie Model:")
#print(deviance_tweedie)
```

basic tests

```
t_test_result <- t.test(scores ~ groups, data = score_data)
anova_result <- aov(scores ~ groups, data = score_data)
wilcoxon_result <- wilcox.test(scores ~ groups, data = score_data)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
"t_test"
```

```
## [1] "t_test"
```

```
t_test_result
```

```
##
##  Welch Two Sample t-test
##
## data:  scores by groups
## t = -2.3025, df = 37.896, p-value = 0.02689
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -446.33193  -28.66556
## sample estimates:
## mean in group 1 mean in group 2
##        348.2664        585.7651
```

```
"wilcoxon"
```

```
## [1] "wilcoxon"
```

```
wilcoxon_result
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  scores by groups
## W = 109, p-value = 0.01436
## alternative hypothesis: true location shift is not equal to 0
```

```
"anova"
```

```
## [1] "anova"
```

```
summary(anova_result)
```

```
##             Df  Sum Sq Mean Sq F value Pr(>F)
## groups       1  564057  564057   5.301 0.0269 *
## Residuals   38 4043095  106397
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## anova with ln
delta = 1
```

```
score_data.ln <- score_data
score_data.ln$scores <- log(score_data.ln$scores+delta)
anova_result.ln <- aov(scores ~ groups, data = score_data.ln)
"anova log"
```

```
## [1] "anova log"
```

```
summary(anova_result.ln)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## groups       1  14.54  14.538   6.882 0.0125 *
## Residuals   38  80.28   2.113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-values

```
lm_p_value <- summary(lm_model)$coefficients["groups2", "Pr(>|t|)"]
tweedy_p_value <- summary(tweedie_model)$coefficients["score_data$groups2", "Pr(>|t|)"]
p_value_groups2_quantile <- summary(quantile_regression_model, se="nid")$coefficients["score_data$groups

# Print model the p-values
print("P-values:")
```

```
## [1] "P-values:"
```

```
print(paste("Linear Regression:", lm_p_value))
```

```
## [1] "Linear Regression: 0.0268781896144321"
```

```
print(paste("Tweedie Regression:", tweedy_p_value))
```

```
## [1] "Tweedie Regression: 0.0326991739072926"
```

```
print(paste("Tweedie Regression 2 :", summary(tweedie_model.2)$coefficients["score_data$groups2", "Pr(>
```

```
## [1] "Tweedie Regression 2 : 0.0334868758849377"
```

```
print(paste("Tweedie Regression 3:", summary(tweedie_model.3)$coefficients["score_data$groups2", "Pr(>|
```

```
## [1] "Tweedie Regression 3: 0.037041669536625"
```

```
print(paste("Tweedie Regression 4:", summary(tweedie_model.4)$coefficients["score_data$groups2", "Pr(>|
```

```
## [1] "Tweedie Regression 4: 0.0308914322292152"
```

```
print(paste("Quantile Regression:", p_value_groups2_quantile))
```

```
## [1] "Quantile Regression: 0.0505695327150513"
```

```
print(paste("wilcoxon:",wilcoxon_result$p.value))
```

```
## [1] "wilcoxon: 0.0143592703433878"
```

Different Runs

[1] "P-values:" [1] "Linear Regression: 0.131811448264589" [1] "Tweedie Regression: 0.128300461805298" [1] "Tweedie Regression 2 : 0.124800031276357" [1] "Tweedie Regression 3: 0.123348061888567" [1] "Tweedie Regression 4: 0.126926746066999" [1] "Quantile Regression: 0.332359002537367" [1] "wilcoxon: 0.180556930110569"

[1] "P-values:" [1] "Linear Regression: 0.0038693137098985" [1] "Tweedie Regression: 0.00301175316866842" [1] "Tweedie Regression 2 : 0.00214240749103708" [1] "Tweedie Regression 3: 0.00189515702769465" [1] "Tweedie Regression 4: 0.0025921108813993" [1] "Quantile Regression: 0.0848589332913616" [1] "wilcoxon: 0.00195712449424303"

[1] "P-values:" [1] "Linear Regression: 0.0171658845281005" [1] "Tweedie Regression: 0.0179279882723774" [1] "Tweedie Regression 2 : 0.0171414976224537" [1] "Tweedie Regression 3: 0.0177462914377656" [1] "Tweedie Regression 4: 0.0169905739634292" [1] "Quantile Regression: 0.0142087635670598" [1] "wilcoxon: 0.0210767943514473"