Data Analysis

Load Data

```
data_path <- "../data/original/train.csv"
voice <- read.csv(data_path)
head(voice)
```

```
##   id meanfreq       sd   median      Q25      Q75      IQR        skew
## 1  0 3521.667 2332.212 2997.294 1660.408 4621.867 2961.459  0.11656897
## 2  1 4189.998 2430.977 4302.741 1832.028 5901.071 4069.043  0.04560770
## 3  2 3154.455 2150.497 2609.968 1460.612 4053.928 2593.316 -0.16147499
## 4  3 4384.338 3029.302 3426.479 1596.072 7283.314 5687.242  0.02416762
## 5  4 4557.150 3158.111 4543.116 1608.165 8074.335 6466.170  0.11711588
## 6  5 4069.004 2983.199 2565.487 1305.284 6961.581 5656.297  0.13049391
##        kurt   sp.ent         sfm     mode centroid  meanfun   minfun   maxfun
## 1 0.9817728 2.308696 0.008450270 1761.333 3521.667 32.33476 153.1934 3995.790
## 2 0.9214181 3.522410 0.022862796 2095.499 4189.998 42.56545 154.0434 3993.462
## 3 0.3882481 2.027891 0.006853276 1577.728 3154.455 26.15712 153.4610 3995.524
## 4 1.4739316 4.823092 0.084471270 2192.669 4384.338 37.56627 153.6399 3994.671
## 5 1.2885699 3.820815 0.100988194 2279.075 4557.150 29.34924 153.8535 3994.646
## 6 0.7668548 3.726702 0.073939204 2035.002 4069.004 29.89368 153.2515 3995.253
##      meandom       mindom    maxdom   dfrange  modindx      age gender accent
## 1 0.06084856 9.842593e-04 194.17128 194.17029 5914.581 twenties female canada
## 2 0.04495757 7.060266e-04 102.27859 102.27788 7693.945 twenties female canada
## 3 0.08144125 2.950821e-04 164.99316 164.99287 5261.606 twenties female canada
## 4 0.01039643 3.165859e-08  29.66787  29.66787 7942.756      nan    nan    nan
## 5 0.01848914 9.267869e-07  85.19259  85.19259 8383.634      nan    nan    nan
## 6 0.01521549 6.052965e-07  32.57839  32.57839 7575.469      nan    nan    nan
```

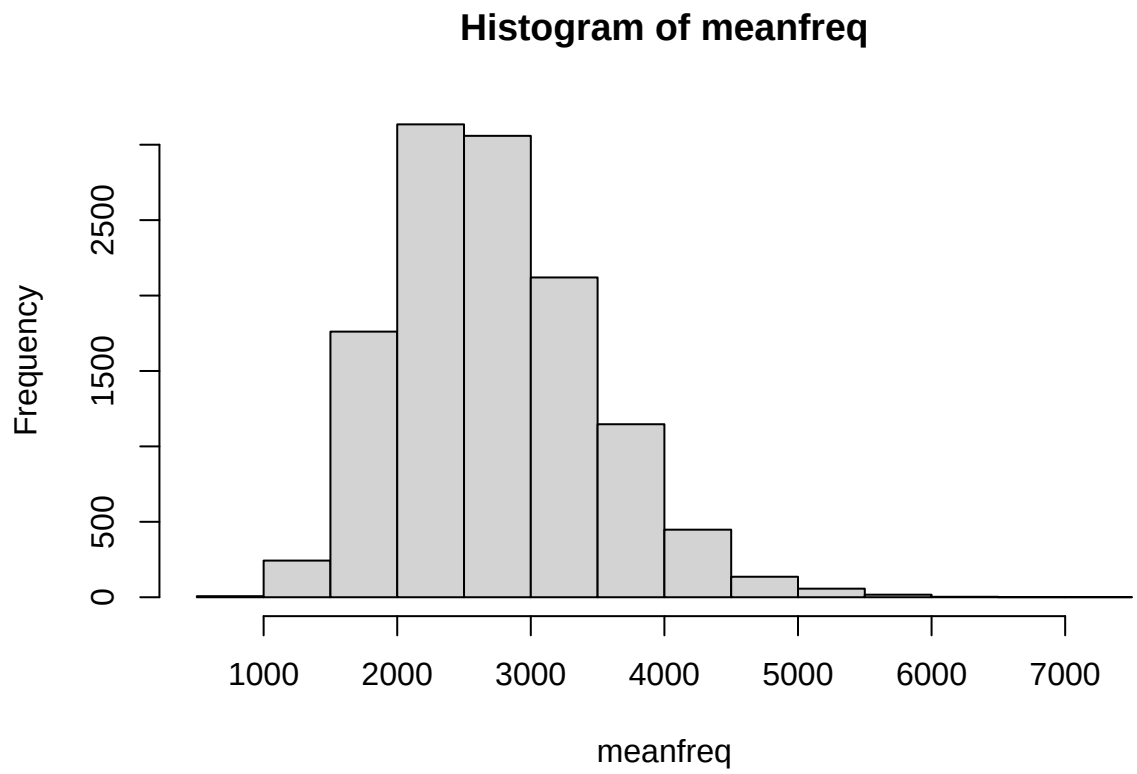
Analyze Data

MeanFreq

Basic Analysis

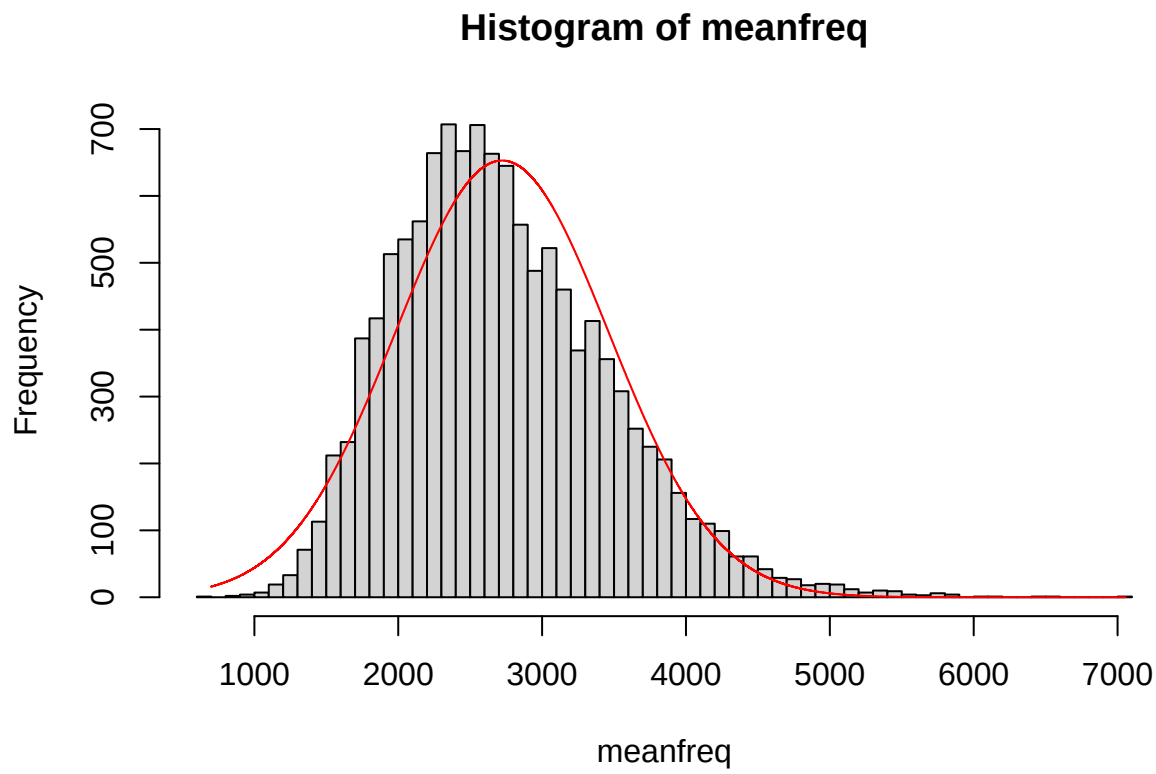
We firstly load the data and draw a histogram of the `meanfreq` column.

```
meanfreq <- voice$meanfreq
hist(meanfreq)
```

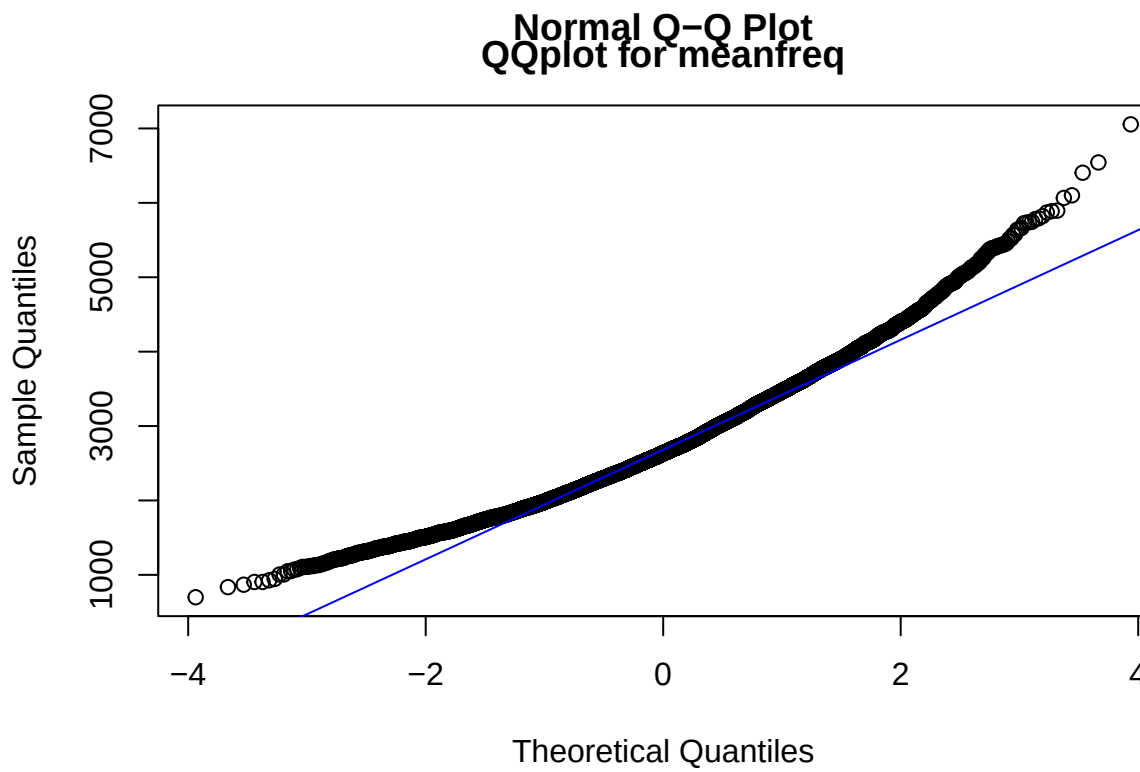


Then, we calculate the mean and standard deviation of the `meanfreq` column.

```
# hist of meanfreq
hist(meanfreq, breaks = 80, main = "Histogram of meanfreq", xlab = "meanfreq")
# impose a normal curve on the histogram
xpt <- seq(698, 7056, by = 0.1)
n_den <- dnorm(xpt, mean = mean(meanfreq), sd = sd(meanfreq))
ypt <- n_den * length(meanfreq) * 100
lines(xpt, ypt, col = "red")
```




```r
# QQplot of meanfreq
qqnorm(meanfreq)
qqline(meanfreq, col = "blue")
title("QQplot for meanfreq", line = 1) # Place title on the first line
```



QQ plot for meanfreq

shapiro test Shapiro test-sample must be less than 5000. Therefore, we will take a sample of 5000 for 100 times

```
pvalue <- numeric(100)
for (i in 1:100) {
  meanfreq_sample <- sample(meanfreq, 5000)
  shapiro.test(meanfreq_sample)
  pvalue[i] <- shapiro.test(meanfreq_sample)$p.value
}
pvalue > 0.05
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE
```

Therefore, the data is not normally distributed.

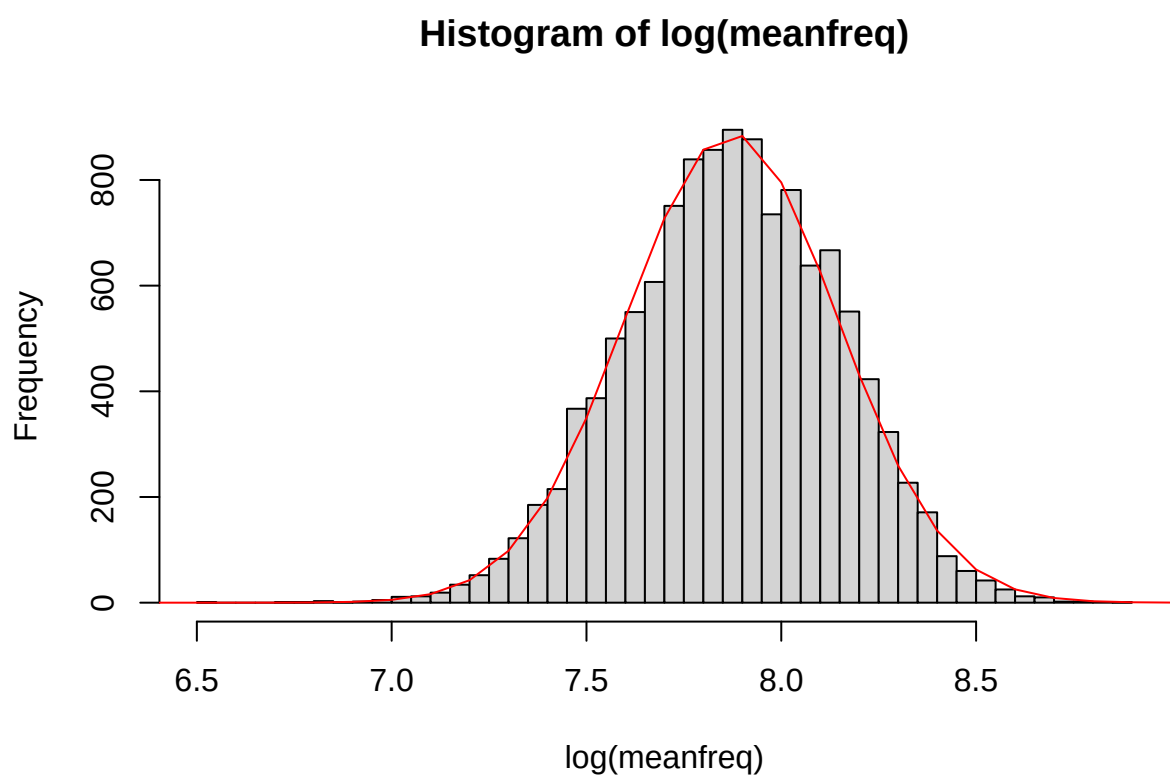
log-transformation

```
meanfreq_trans <- log(meanfreq)
hist(
```

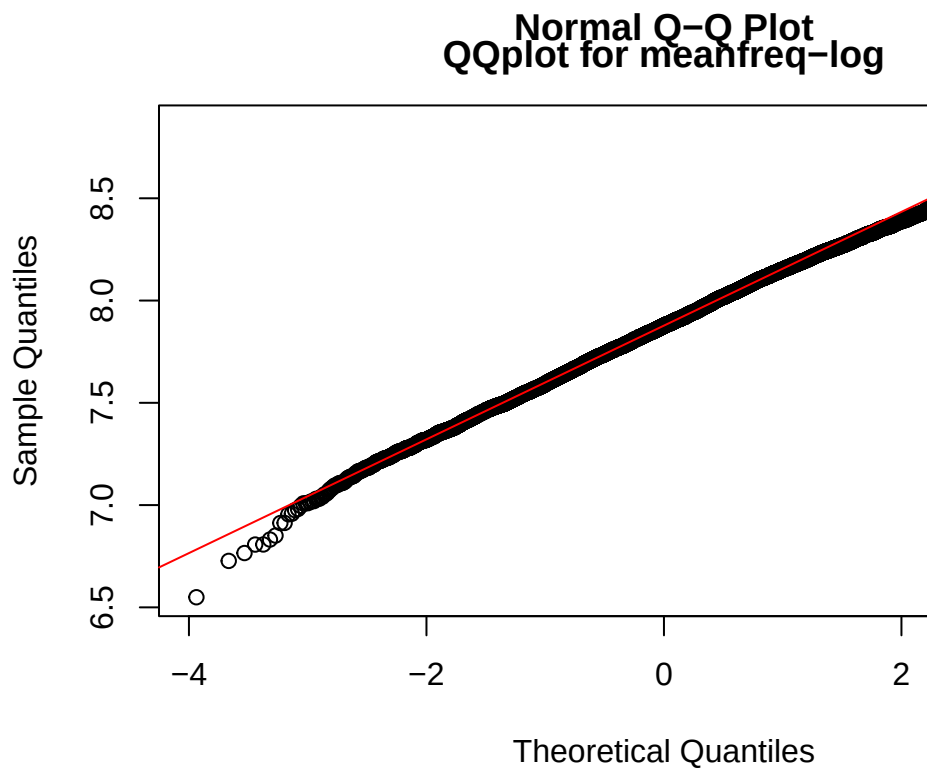
```r
  meanfreq_trans,
  breaks = 80,
  main = "Histogram of log(meanfreq)",
  xlab = "log(meanfreq)"
)
range(meanfreq_trans)
```

```
## [1] 6.549486 8.861497
```

```r
# impose a normal curve on the histogram
xpt <- seq(6, 9, by = 0.1)
n_den <- dnorm(xpt, mean = mean(meanfreq_trans), sd = sd(meanfreq_trans))
ypt <- n_den * length(meanfreq_trans) * 0.05
lines(xpt, ypt, col = "red")
```




```r
qqnorm(meanfreq_trans)
qqline(meanfreq_trans, col = "red")
title("QQplot for meanfreq-log", line = 1) # Place title on the first line
```



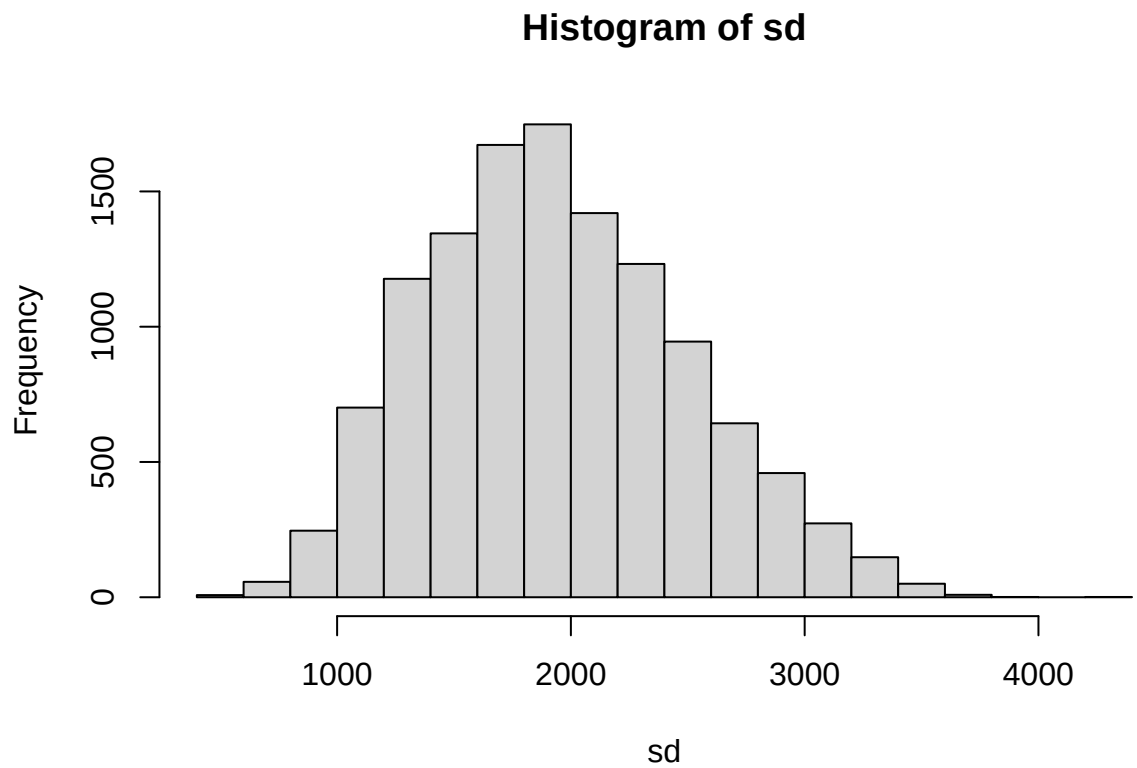
QQ plot for log-transformed meanfreq

SD

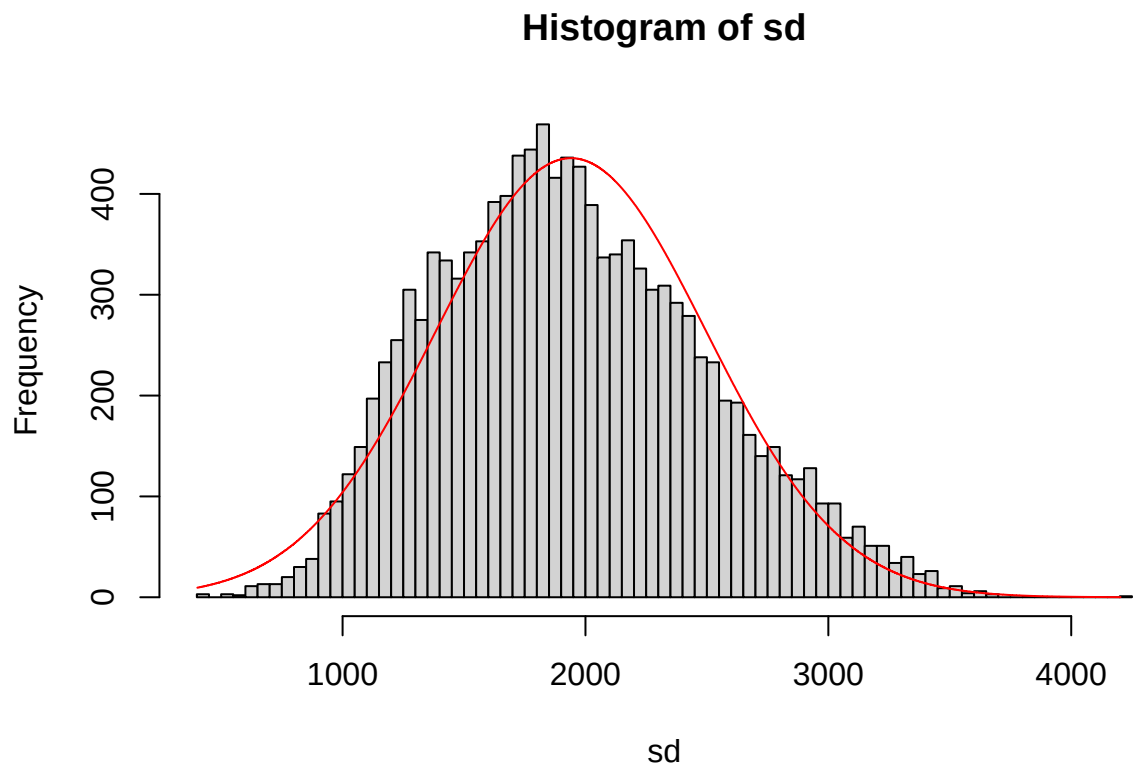
Basic Analysis

We firstly load the data and draw a histogram of the `sd` column.

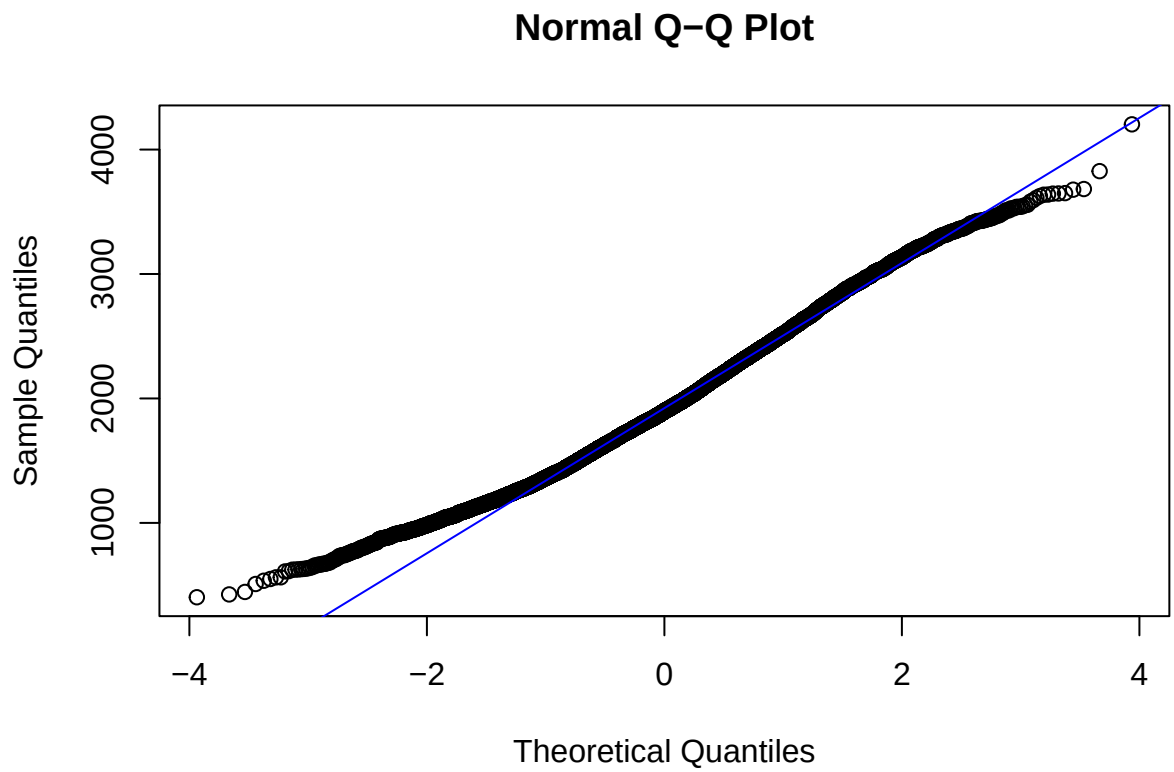
```
sd <- voice$sd
hist(sd)
```




```r
hist(sd, breaks = 80, main = "Histogram of sd", xlab = "sd")
# impose a normal curve on the histogram
xpt <- seq(402, 4203, by = 0.1)
n_den <- dnorm(xpt, mean = mean(sd), sd = sd(sd))
ypt <- n_den * length(sd) * 50
lines(xpt, ypt, col = "red")
```

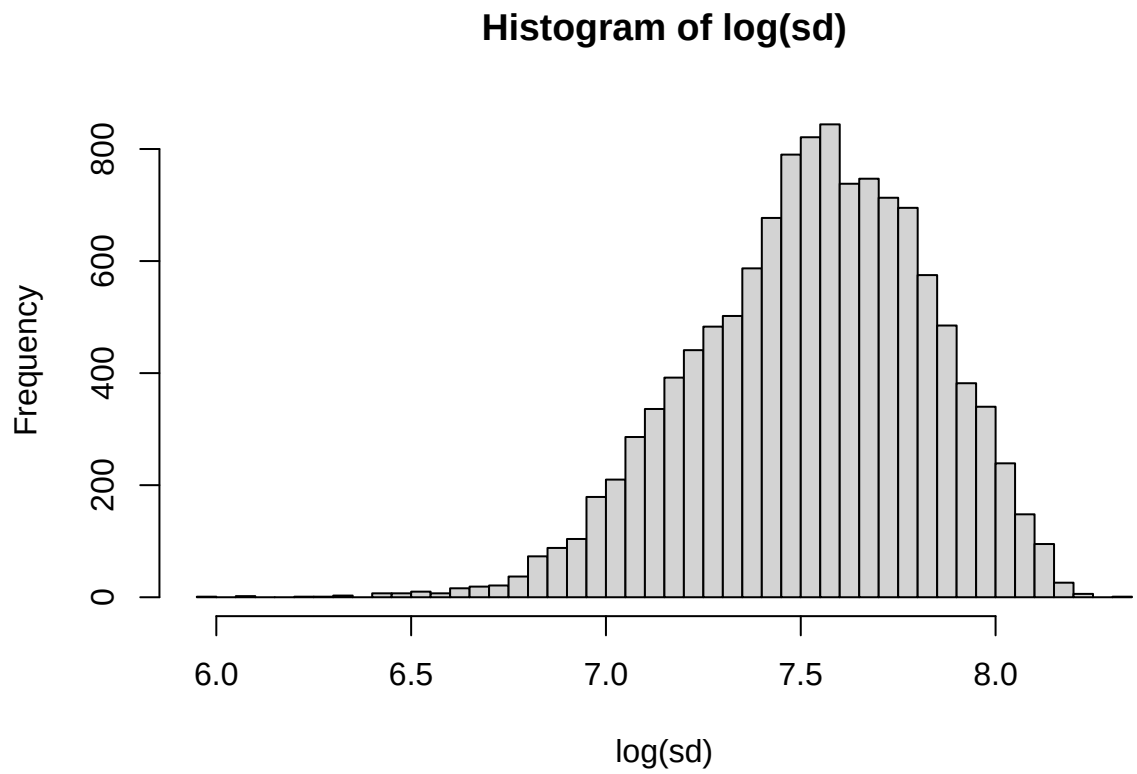



```r
qqnorm(sd)
qqline(sd, col = "blue")
```

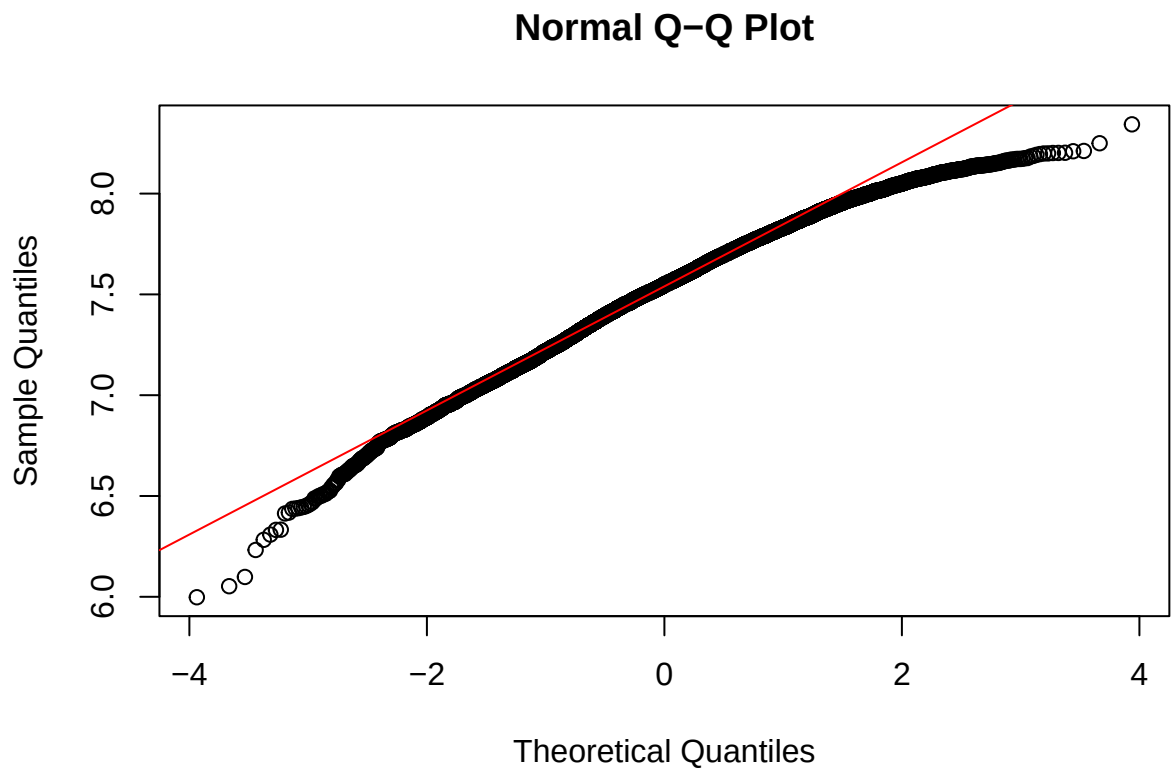


log-transformation

```
# log-transformation of sd
sd_trans <- log(sd)
hist(sd_trans, breaks = 80, main = "Histogram of log(sd)", xlab = "log(sd)")
```



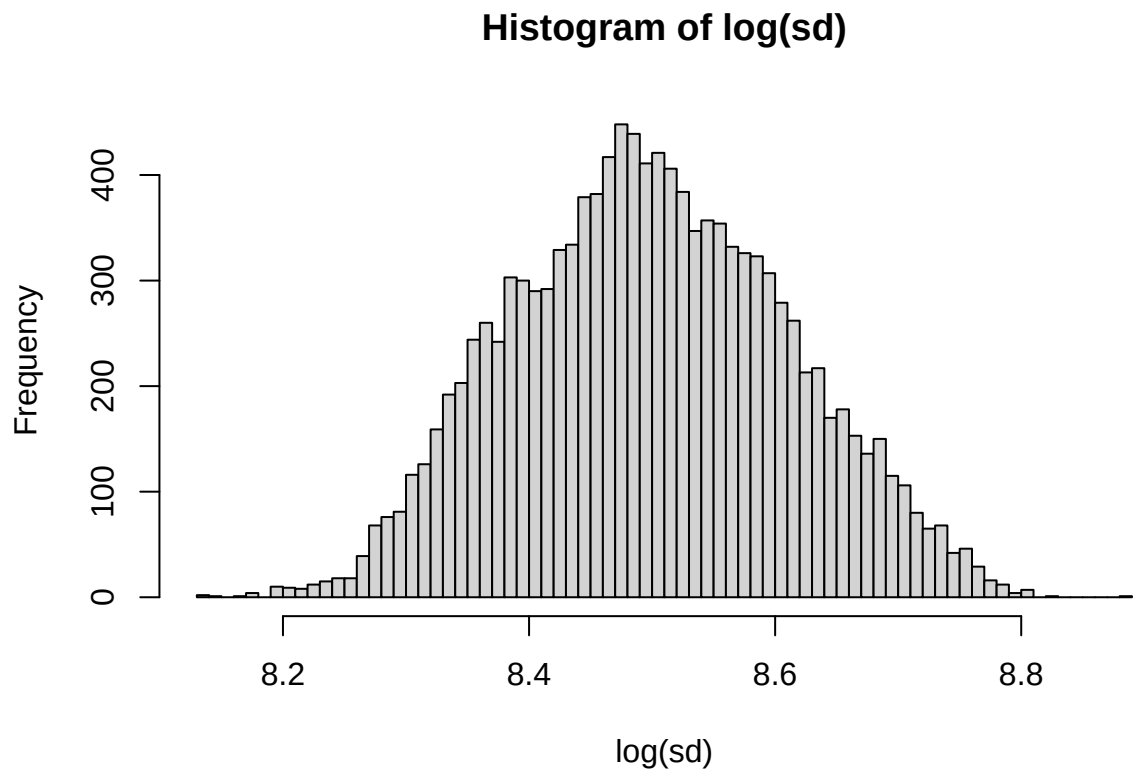

```
qqnorm(sd_trans)
qqline(sd_trans, col = "red")
```



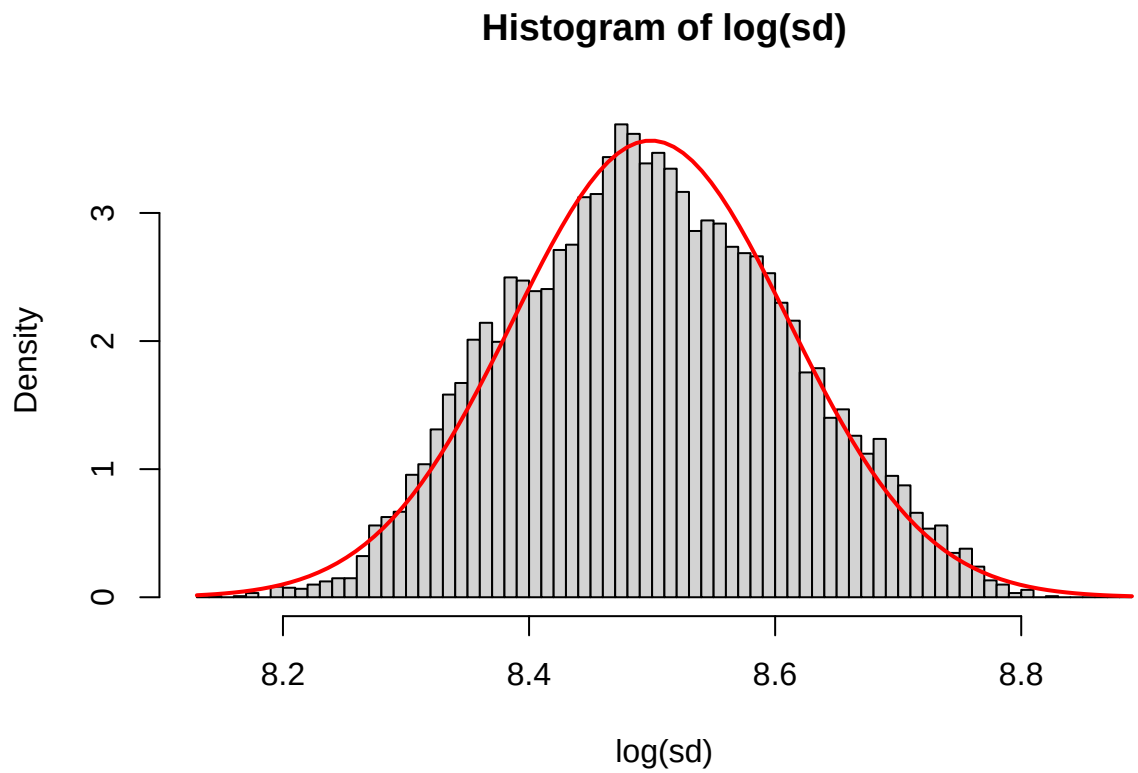
We found that after log-transformation, the data is even worse than before.

Therefore, we tried another way: $y = \log(x + 3000)$

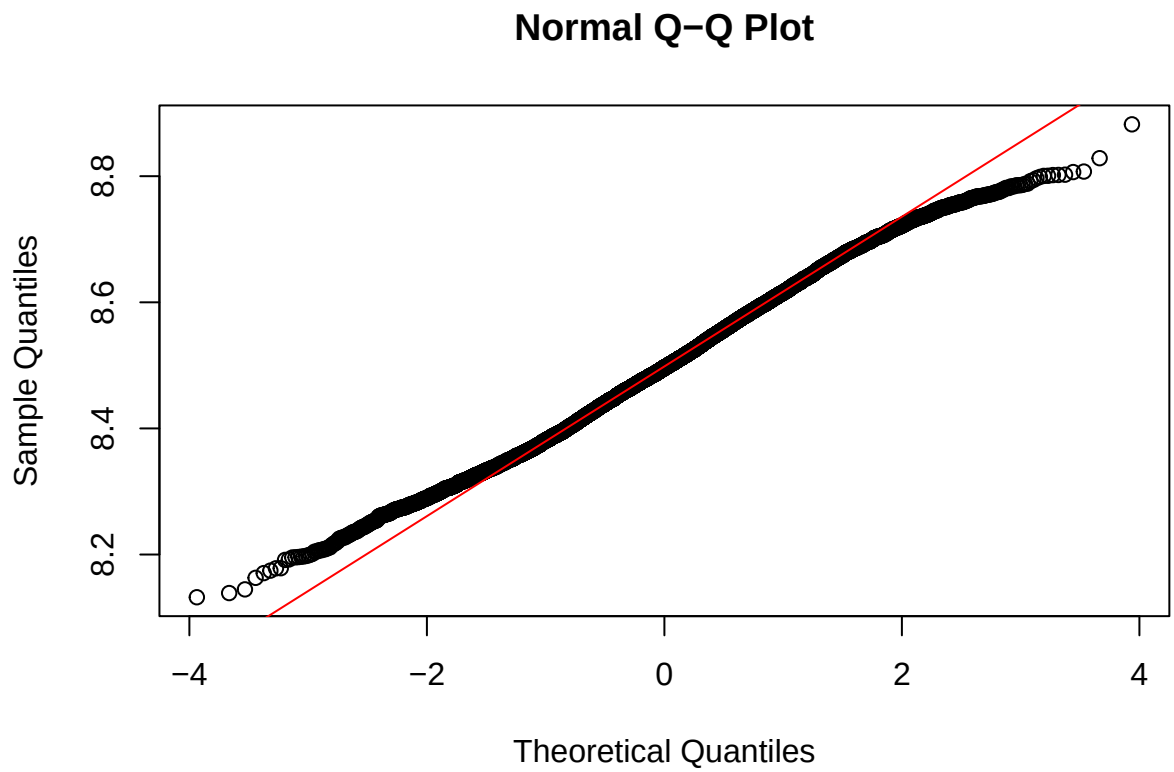
```
sd_trans <- log(sd + 3000)
hist(sd_trans, breaks = 80, main = "Histogram of log(sd)", xlab = "log(sd)")
```




```r
hist(
  sd_trans,
  breaks = 80,
  main = "Histogram of log(sd)",
  xlab = "log(sd)",
  probability = TRUE
)
curve(
  dnorm(x, mean = mean(sd_trans), sd = sd(sd_trans)),
  col = "red",
  lwd = 2,
  add = TRUE
)
```




```
qqnorm(sd_trans)
qqline(sd_trans, col = "red")
```

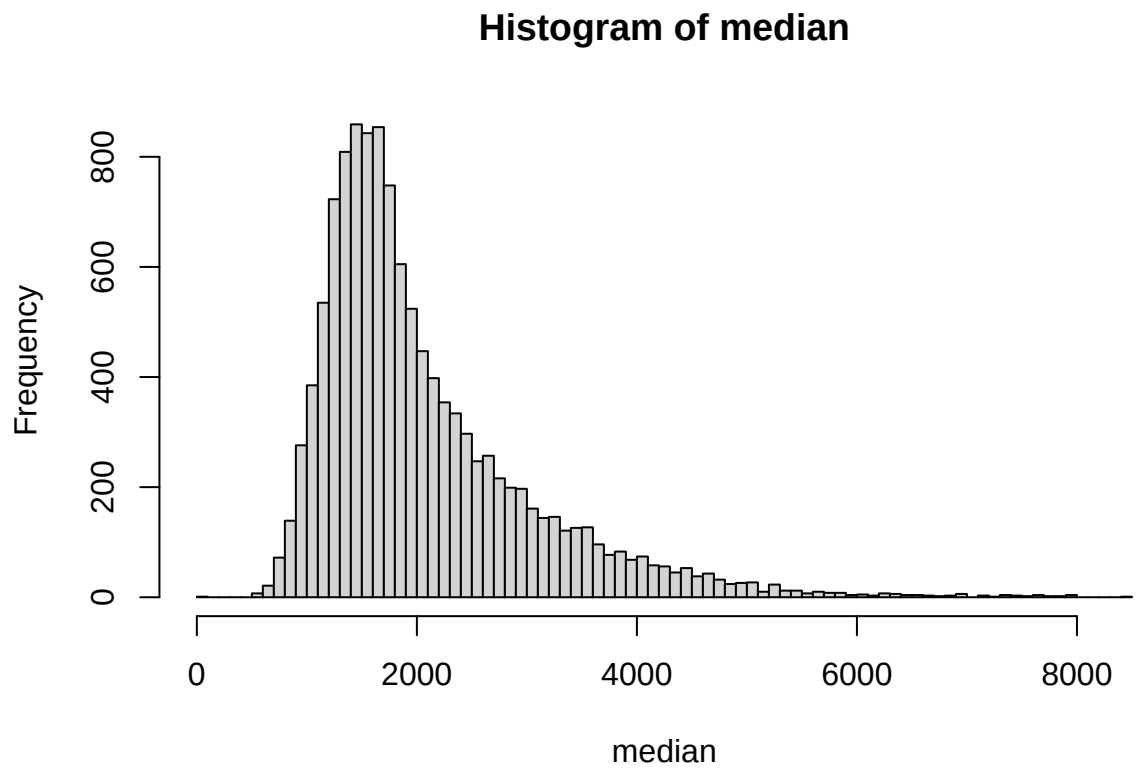


Median

Basic Analysis

We firstly load the data and draw a histogram of the `median` column.

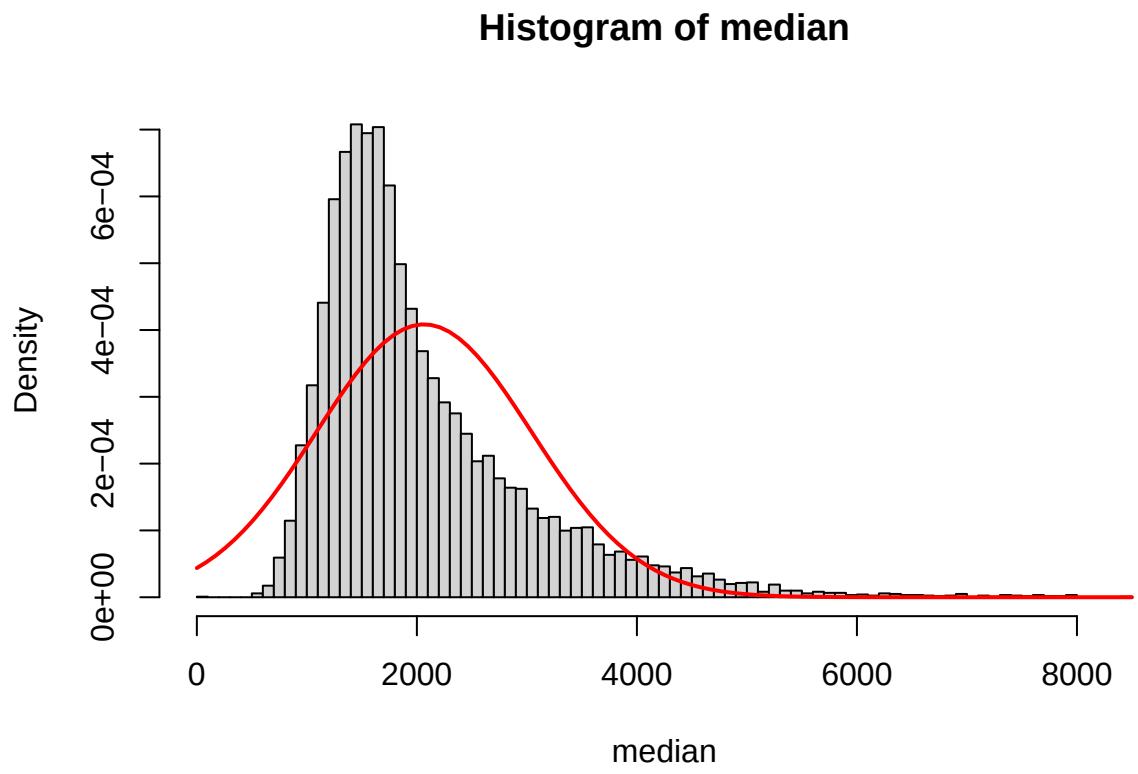
```
median <- voice$median
hist(median, breaks = 80, main = "Histogram of median", xlab = "median")
```



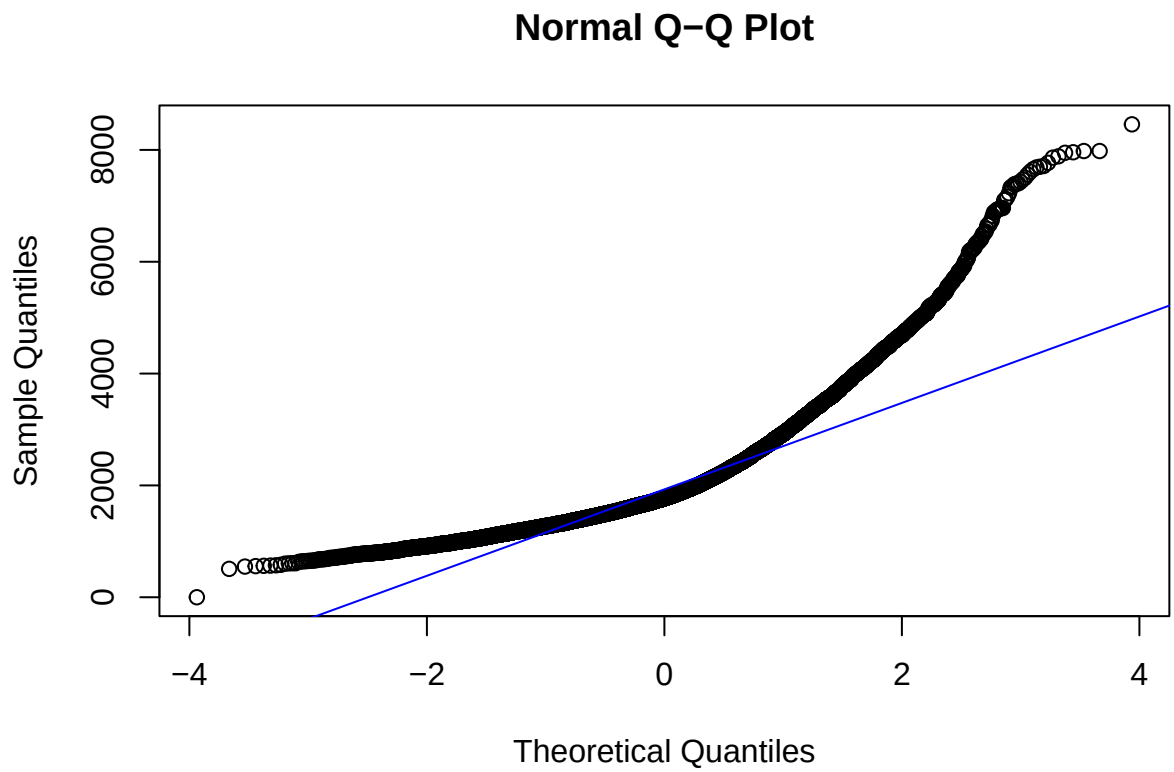
Then, we calculate the mean and standard deviation of the `median` column.

```
median_mean <- mean(median)
median_sd <- sd(median)
```

```
hist(median, breaks = 80, main = "Histogram of median", xlab = "median", probability = TRUE)
curve(
  dnorm(x, mean = median_mean, sd = median_sd),
  col = "red",
  lwd = 2,
  add = TRUE
)
```

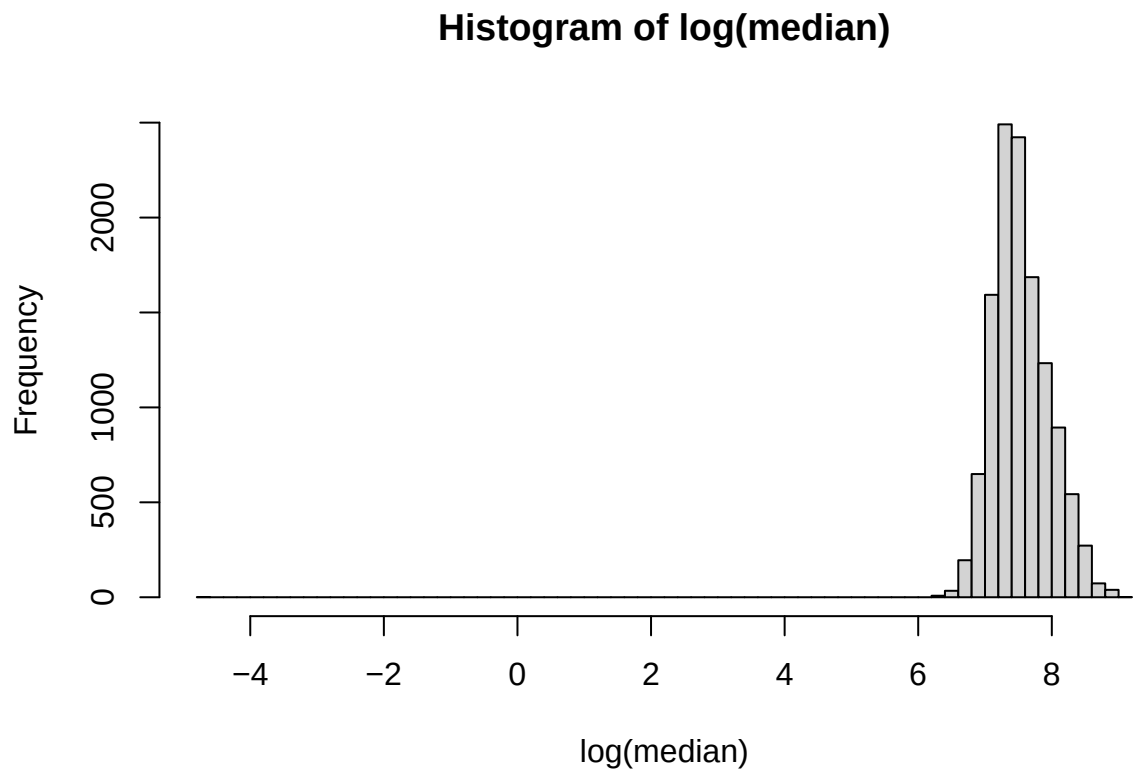



```r
qqnorm(median)
qqline(median, col = "blue")
```

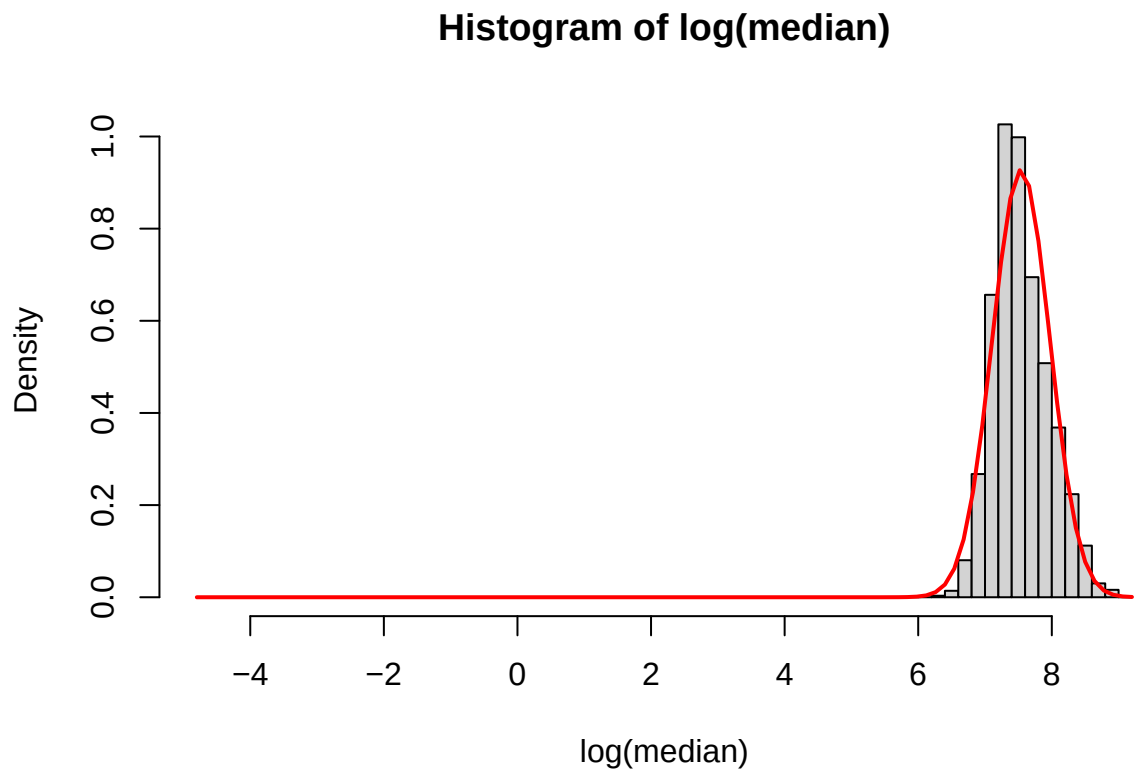


log-transformation

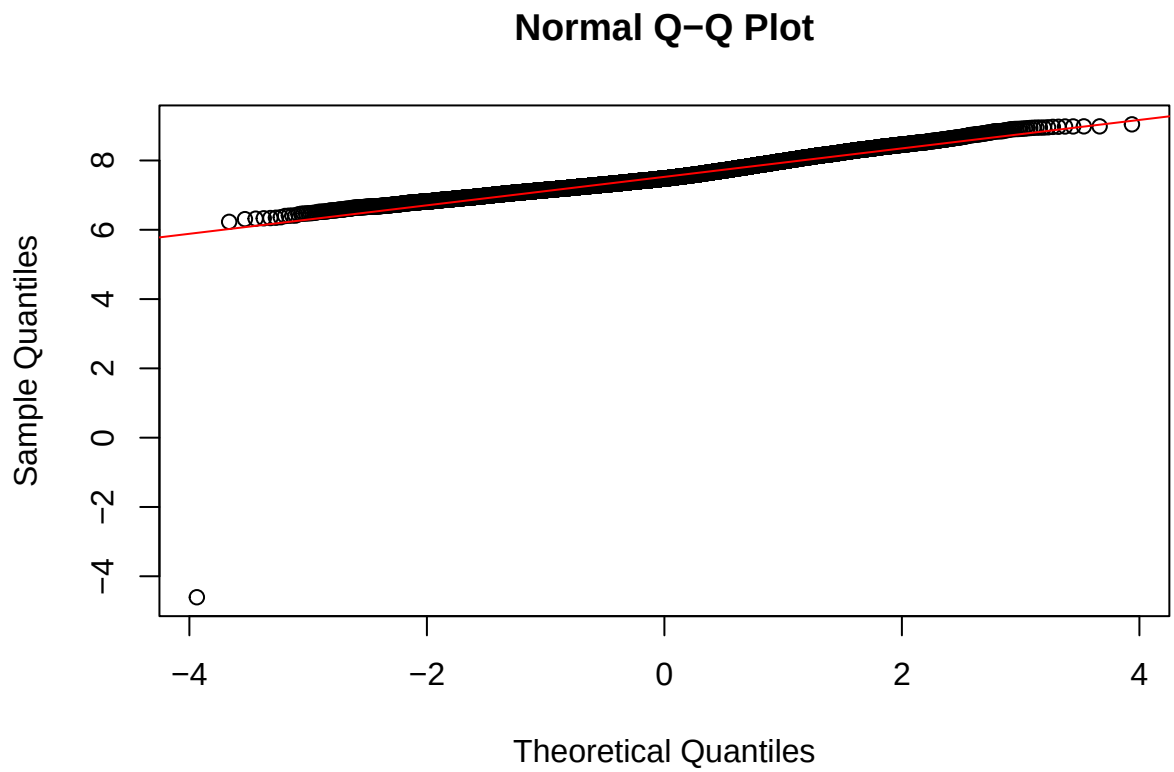
```
median_trans <- log(median + 0.01)
hist(median_trans, breaks = 80, main = "Histogram of log(median)", xlab = "log(median)")
```




```r
hist(
  median_trans,
  breaks = 90,
  main = "Histogram of log(median)",
  xlab = "log(median)",
  probability = TRUE
)
curve(
  dnorm(x, mean = mean(median_trans), sd = sd(median_trans)),
  col = "red", lwd = 2,
  add = TRUE
)
```




```
qqnorm(median_trans)
qqline(median_trans, col = "red")
```

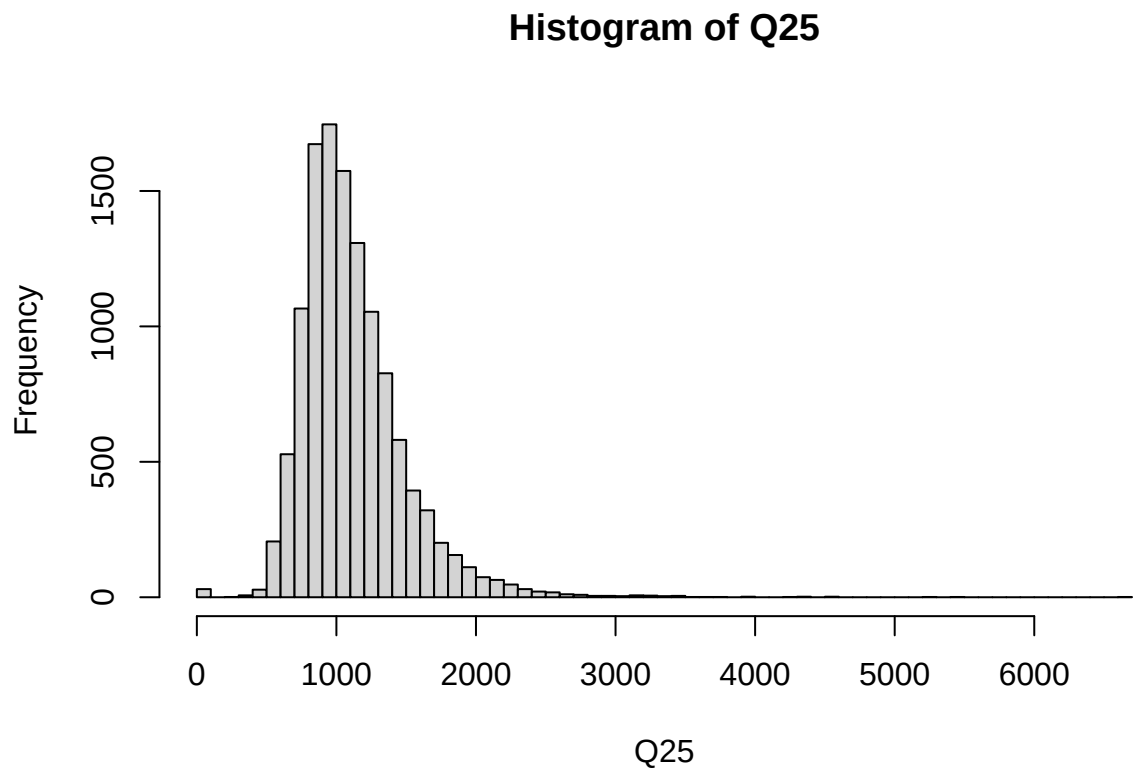


Q25

Basic Analysis

We firstly load the data and draw a histogram of the `Q25` column.

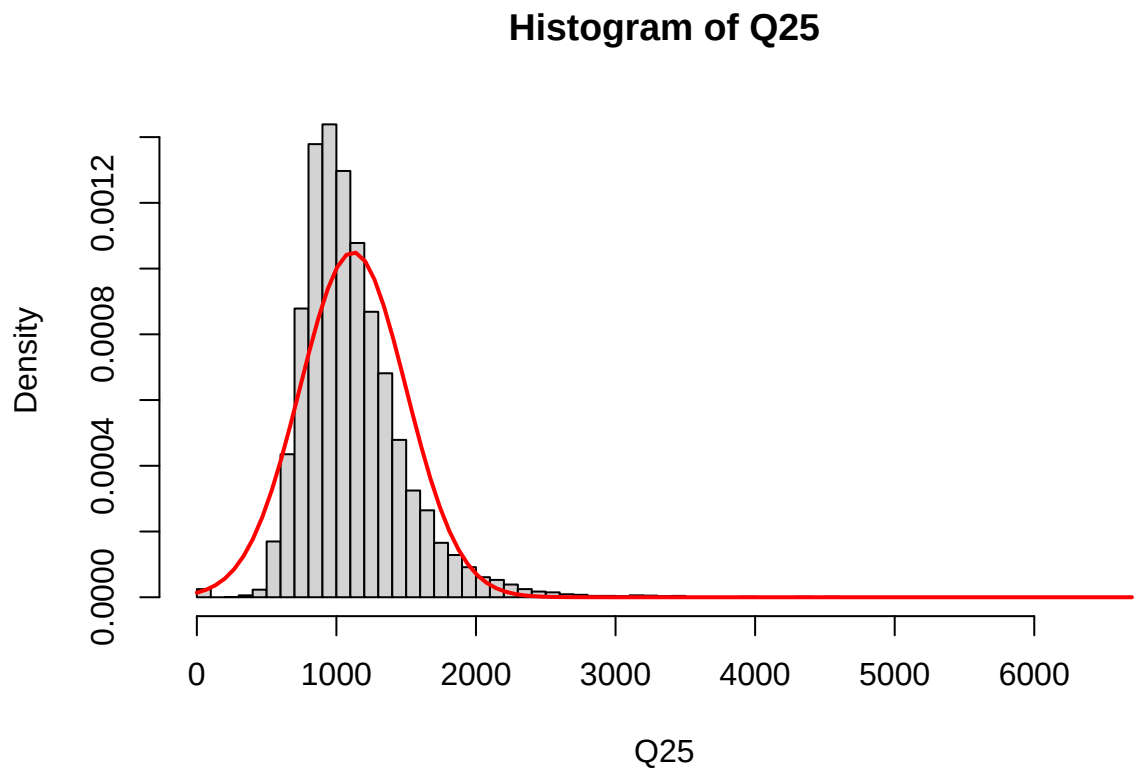
```
Q25 <- voice$Q25
hist(Q25, breaks = 80, main = "Histogram of Q25", xlab = "Q25")
```



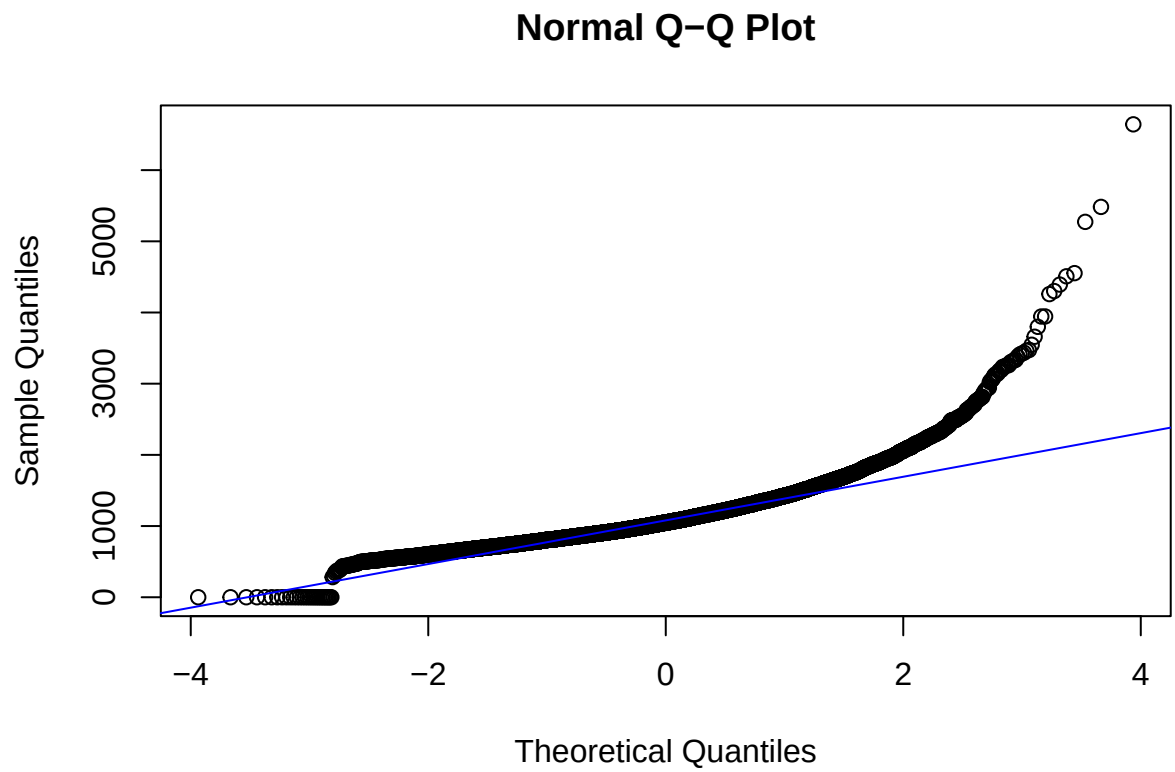
Then, we calculate the mean and standard deviation of the `Q25` column.

```
Q25_mean <- mean(Q25)
Q25_sd <- sd(Q25)
```

```
hist(Q25, breaks = 80, main = "Histogram of Q25", xlab = "Q25", probability = TRUE)
curve(dnorm(x, mean = Q25_mean, sd = Q25_sd), col = "red", lwd = 2, add = TRUE)
```




```r
qqnorm(Q25)
qqline(Q25, col = "blue")
```

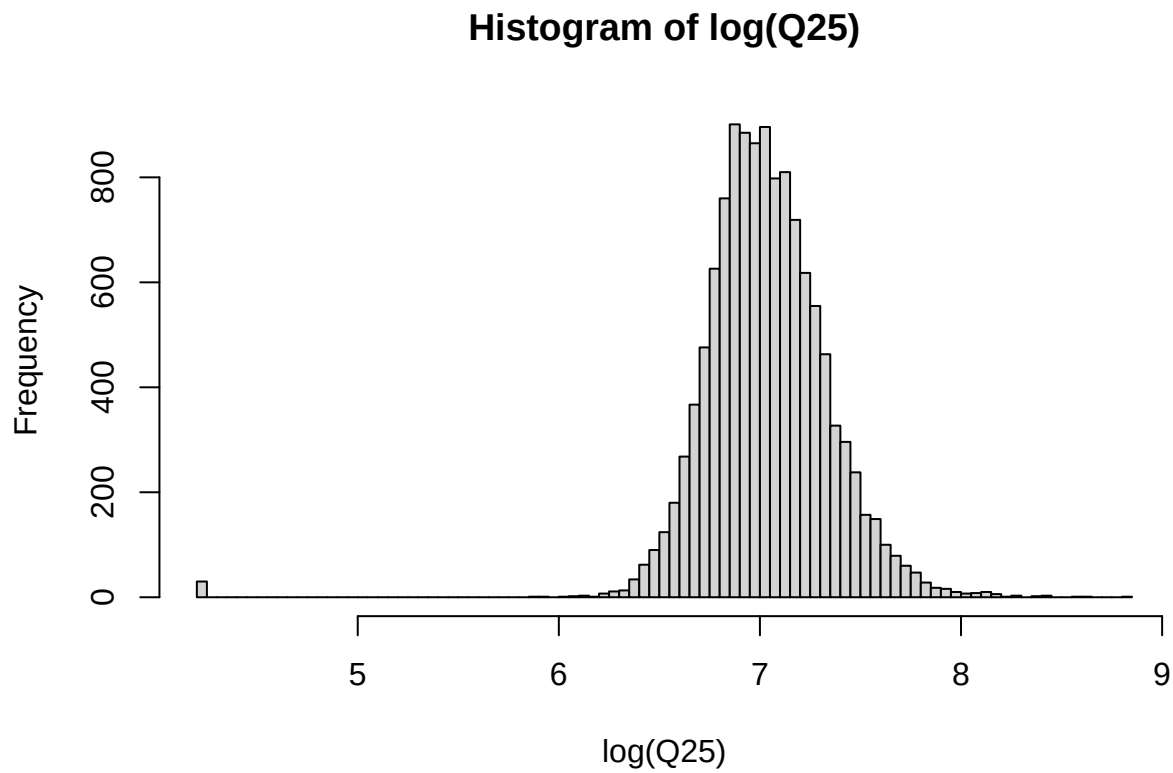


QQ plot for Q25

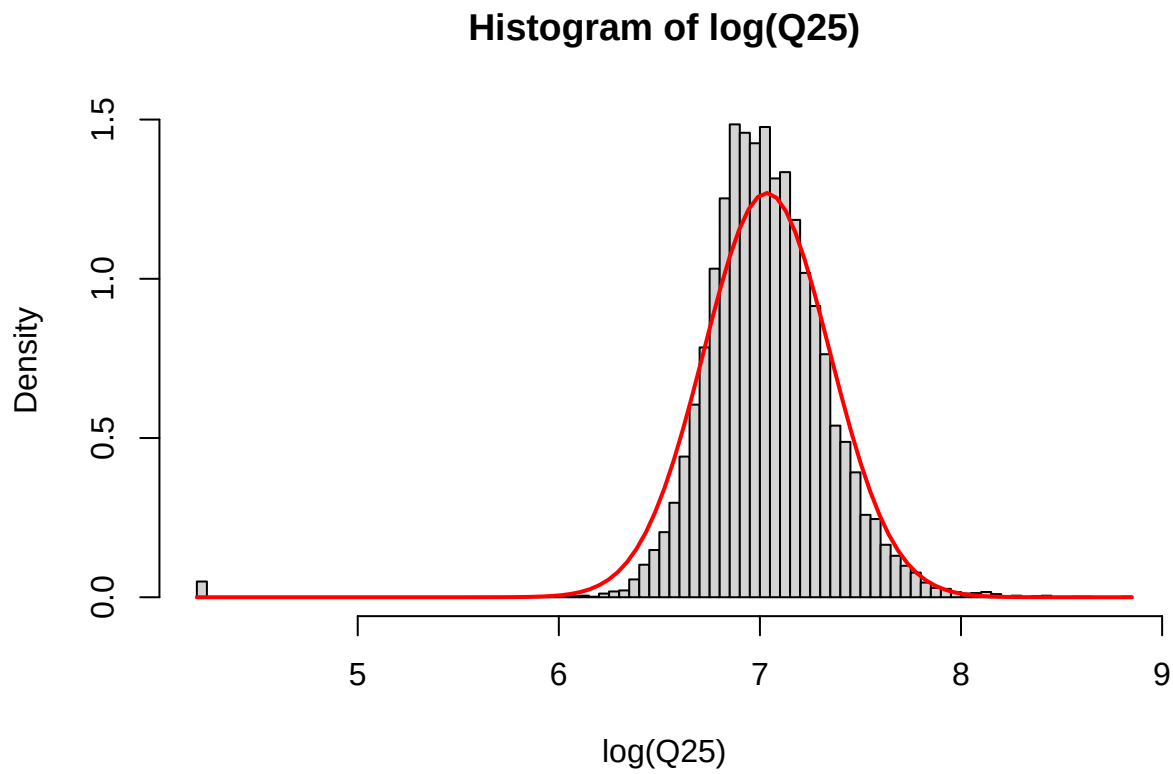
log-transformation

After multiple attempts to transform the `Q25` column, we found that the log-transformation with $f(x) = \log(x - \min(x) + 70)$ is the best transformation for this column.

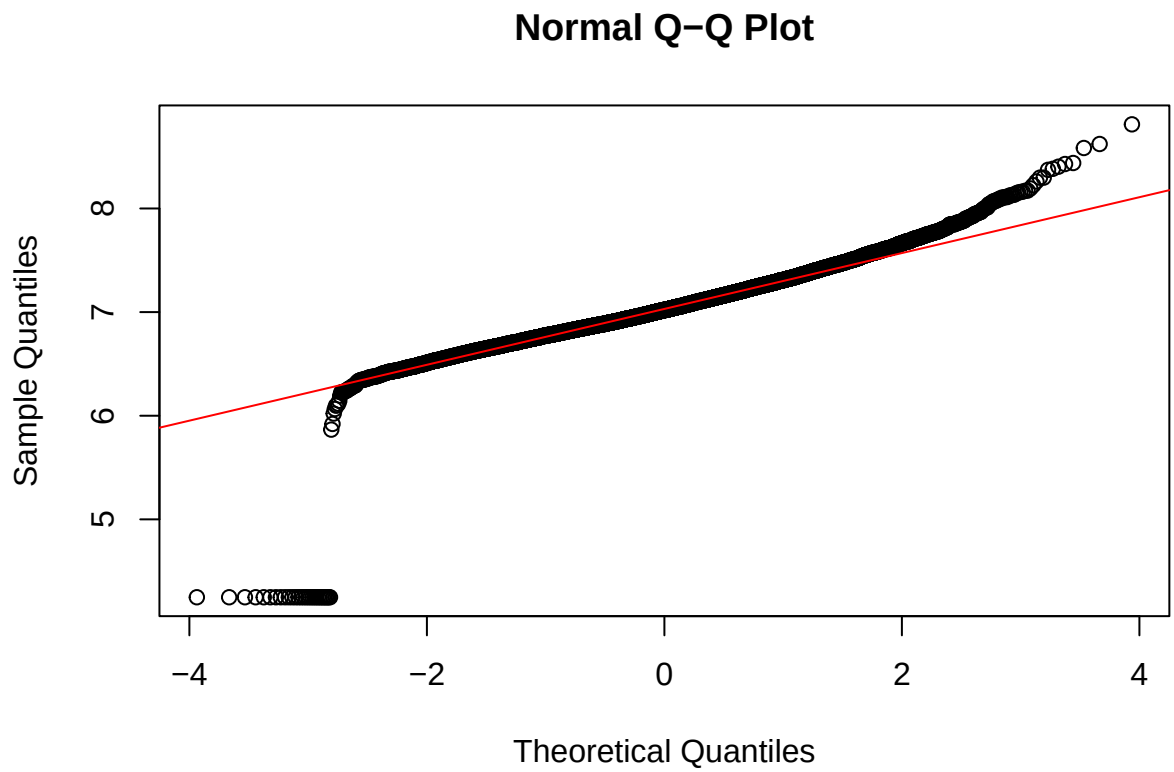
```
Q25_trans <- log(Q25 - min(Q25) + 70)
hist(Q25_trans, breaks = 80, main = "Histogram of log(Q25)", xlab = "log(Q25)")
```




```r
hist(Q25_trans, breaks = 80, main = "Histogram of log(Q25)", xlab = "log(Q25)", probability = TRUE)
curve(dnorm(x, mean = mean(Q25_trans), sd = sd(Q25_trans)), col = "red", lwd = 2, add = TRUE)
```




```
qqnorm(Q25_trans)
qqline(Q25_trans, col = "red")
```

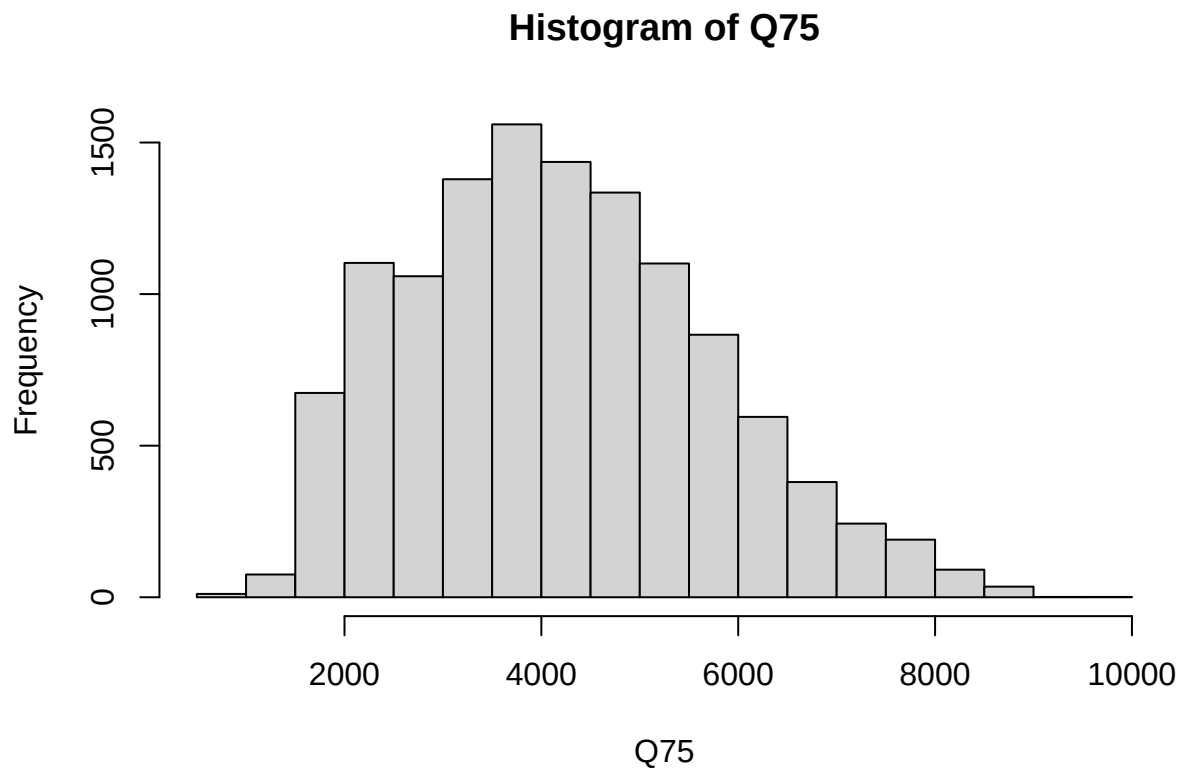


Q75

Basic Analysis

We firstly load the data and draw a histogram of the `Q75` column.

```
Q75 <- voice$Q75
hist(Q75)
```



Then, we calculate the mean and standard deviation of the `Q75` column.

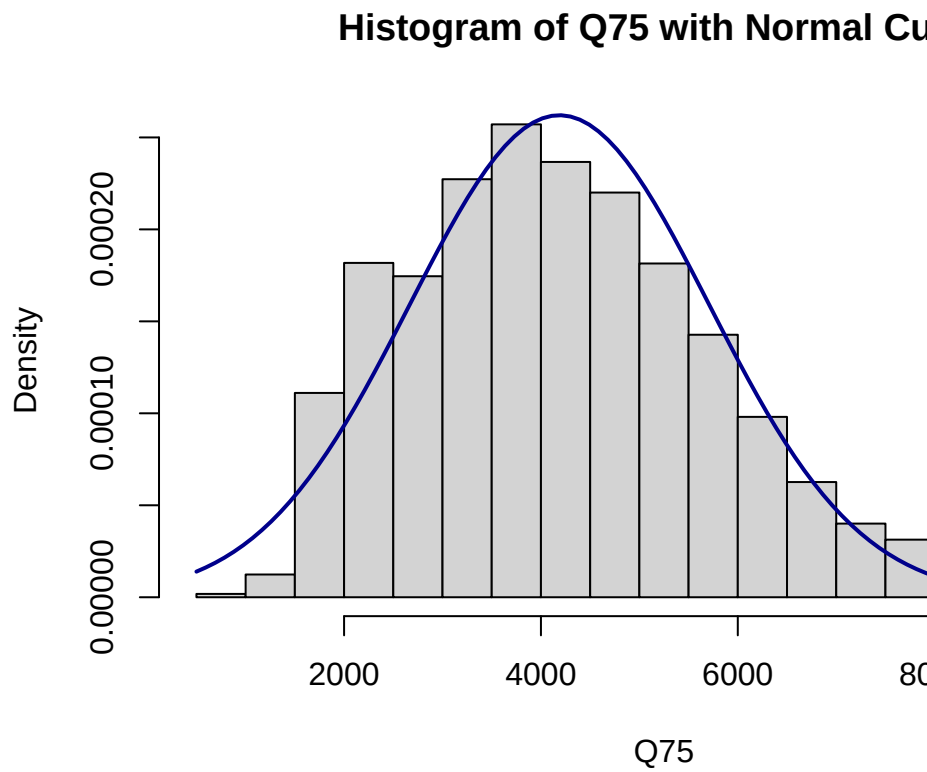
```
Q75_mean <- mean(Q75)
Q75_sd <- sd(Q75)
print(paste("Mean of Q75:", Q75_mean))
```

```
## [1] "Mean of Q75: 4187.87067406675"
```

```
print(paste("Standard deviation of Q75:", Q75_sd))
```

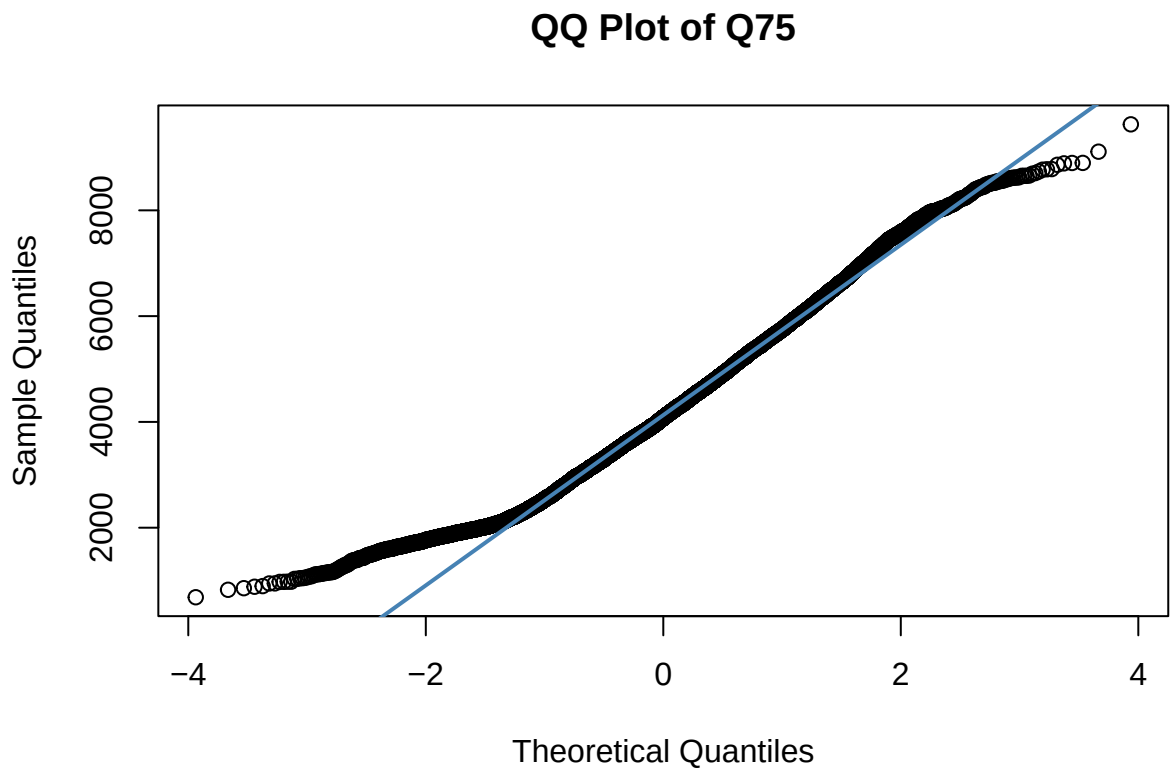
```
## [1] "Standard deviation of Q75: 1522.24666544973"
```

```
hist(Q75, probability = TRUE, main = "Histogram of Q75 with Normal Curve", xlab = "Q75")
curve(dnorm(x, mean = Q75_mean, sd = Q75_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```



Histogram with overlaid normal curve

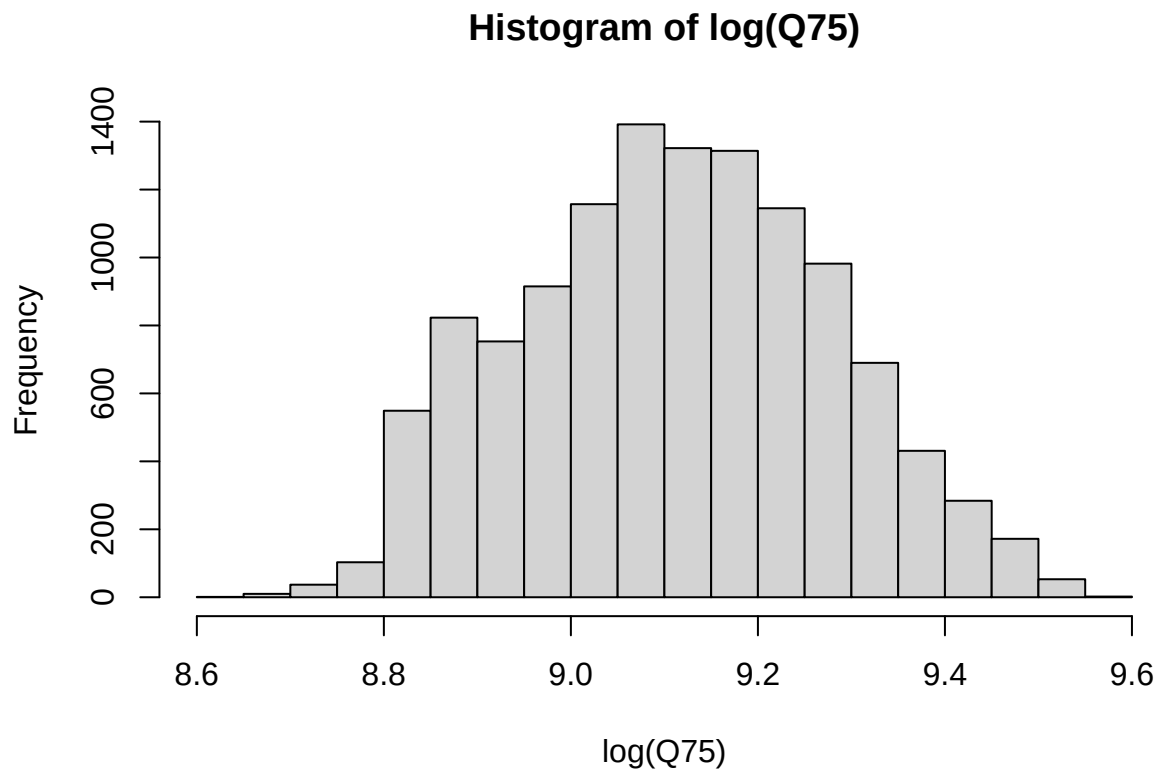
```
qqnorm(Q75, main = "QQ Plot of Q75")
qqline(Q75, col = "steelblue", lwd = 2)
```



QQ plot

Log-transformed `Q75`

```
Q75_trans <- log(Q75 + 5000)
hist(Q75_trans, breaks = 20, main = "Histogram of log(Q75)", xlab = "log(Q75)")
```



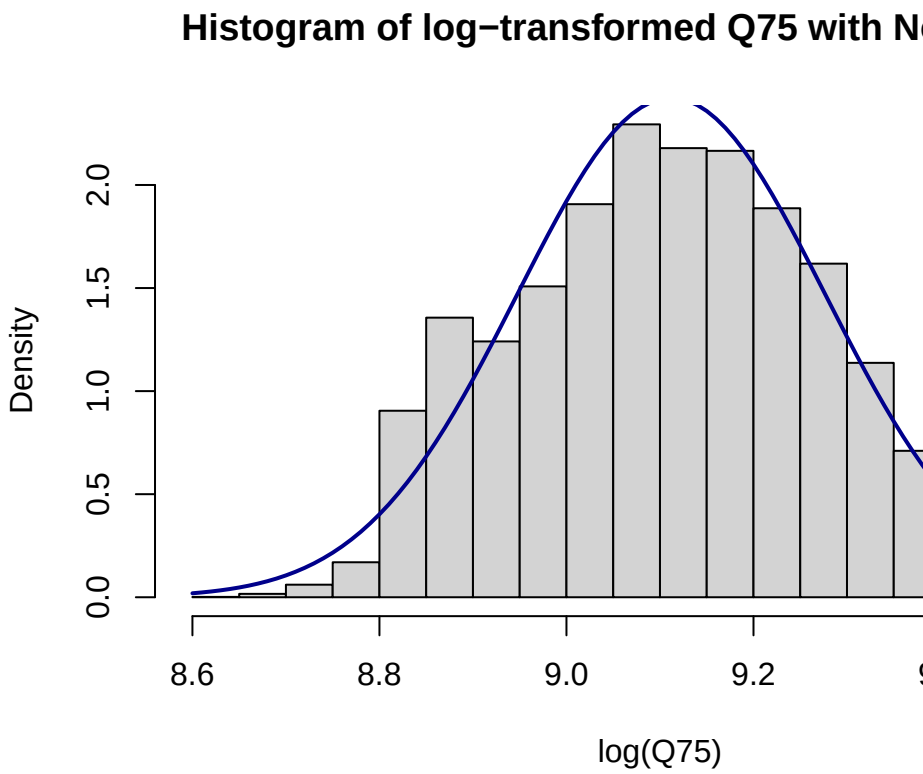

```
Q75_trans_mean <- mean(Q75_trans)
Q75_trans_sd <- sd(Q75_trans)
print(paste("Mean of log-transformed Q75:", Q75_trans_mean))
```

```
## [1] "Mean of log-transformed Q75: 9.11206165864669"
```

```
print(paste("Standard deviation of log-transformed Q75:", Q75_trans_sd))
```

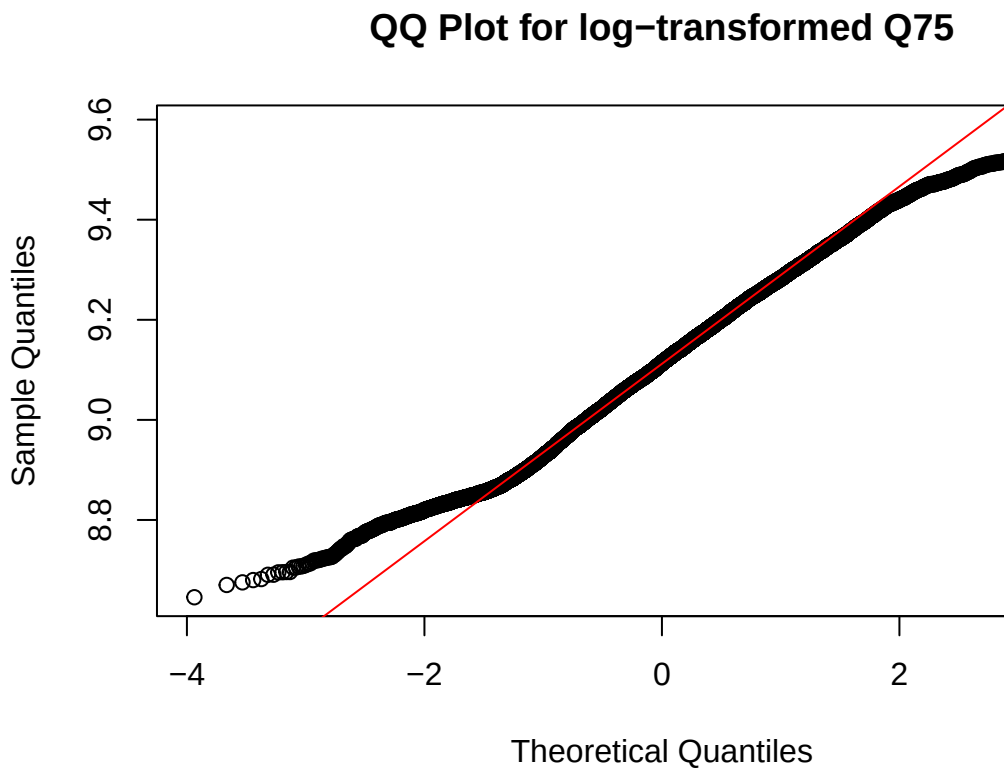
```
## [1] "Standard deviation of log-transformed Q75: 0.164697259619145"
```

```
hist(Q75_trans, probability = TRUE, main = "Histogram of log-transformed Q75 with Normal Curve", xlab =
curve(dnorm(x, mean = Q75_trans_mean, sd = Q75_trans_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt =
```



Histogram with overlaid normal curve

```
qqnorm(Q75_trans, main = "QQ Plot for log-transformed Q75")
qqline(Q75_trans, col = "red")
```

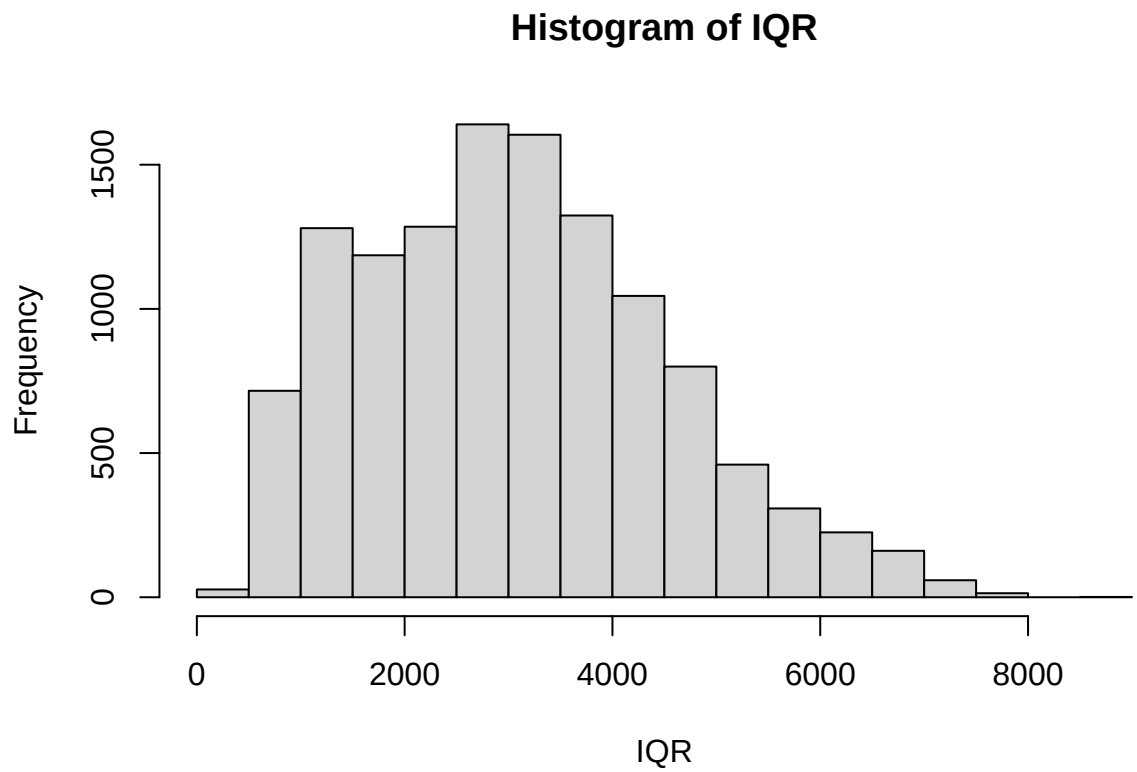


QQ plot for log-transformed `Q75`

IQR

Basic Analysis

```
IQR <- voice$IQR
hist(IQR)
```

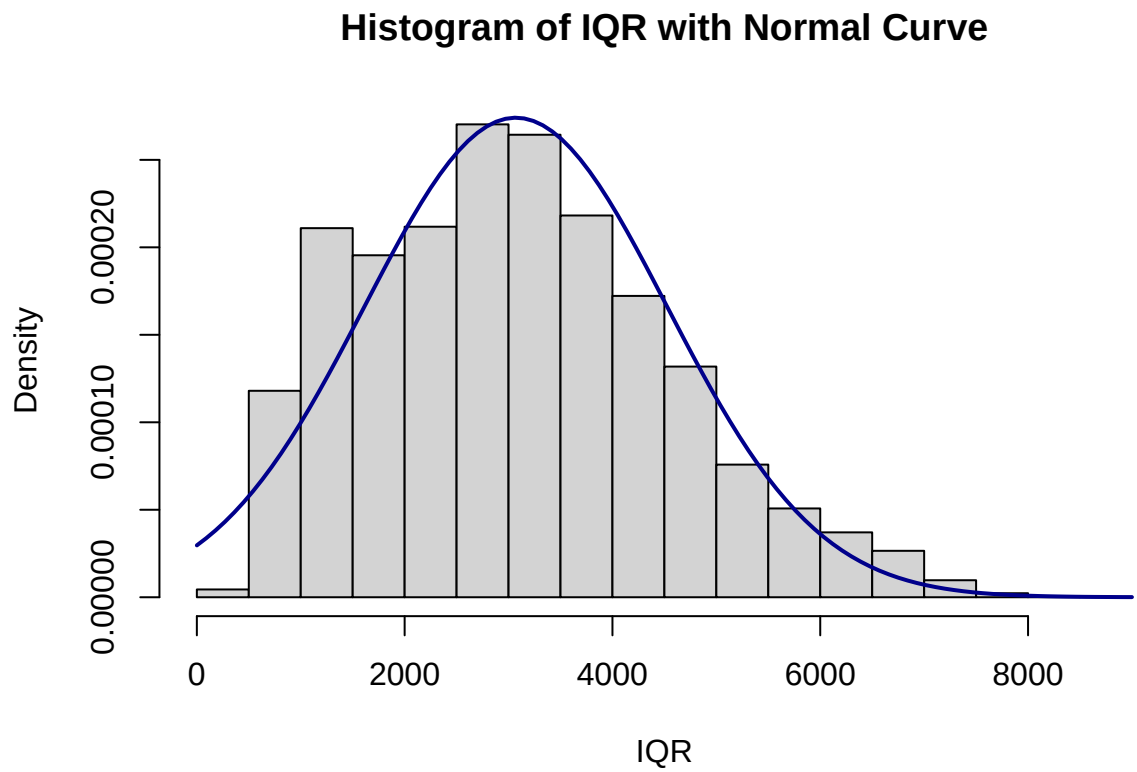


Then, we calculate the mean and standard deviation of the `IQR` column.

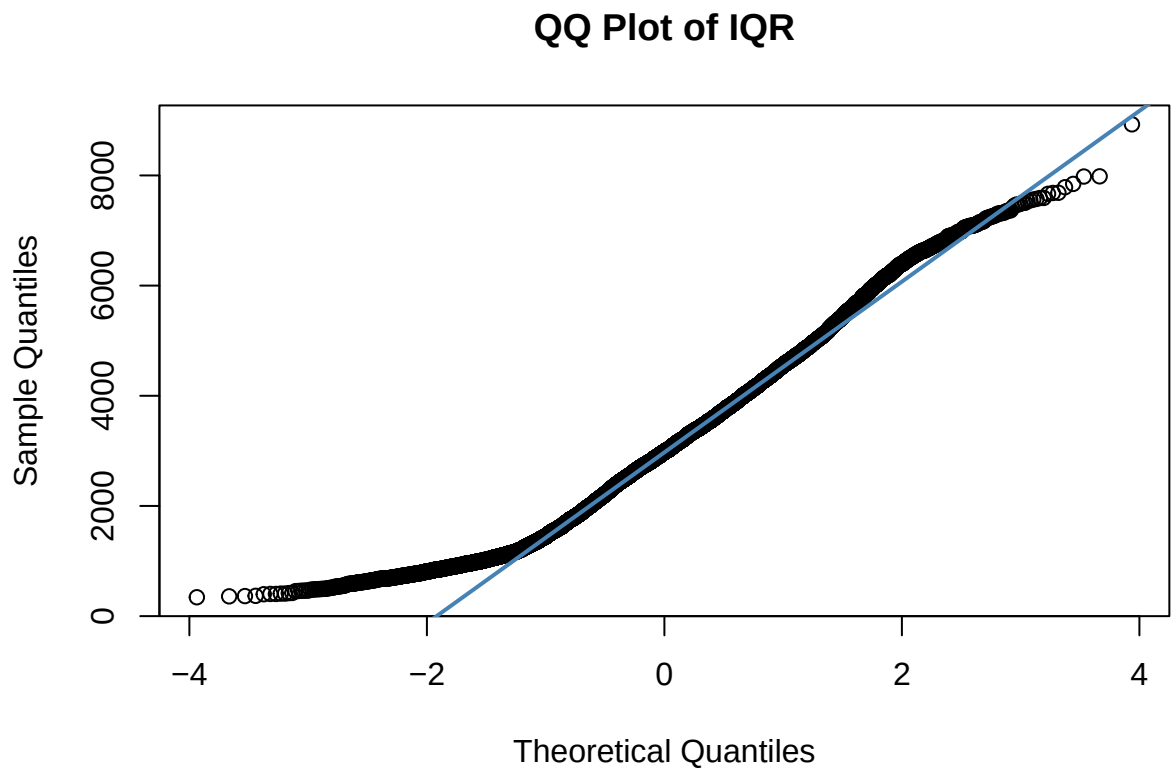
```
IQR_mean <- mean(IQR)
IQR_sd <- sd(IQR)
```

The `IQR` column is almost normally distributed, we can see that from the histogram and QQ plot below.

```
hist(IQR, probability = TRUE, main = "Histogram of IQR with Normal Curve", xlab = "IQR", breaks = 20)
curve(dnorm(x, mean = IQR_mean, sd = IQR_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```



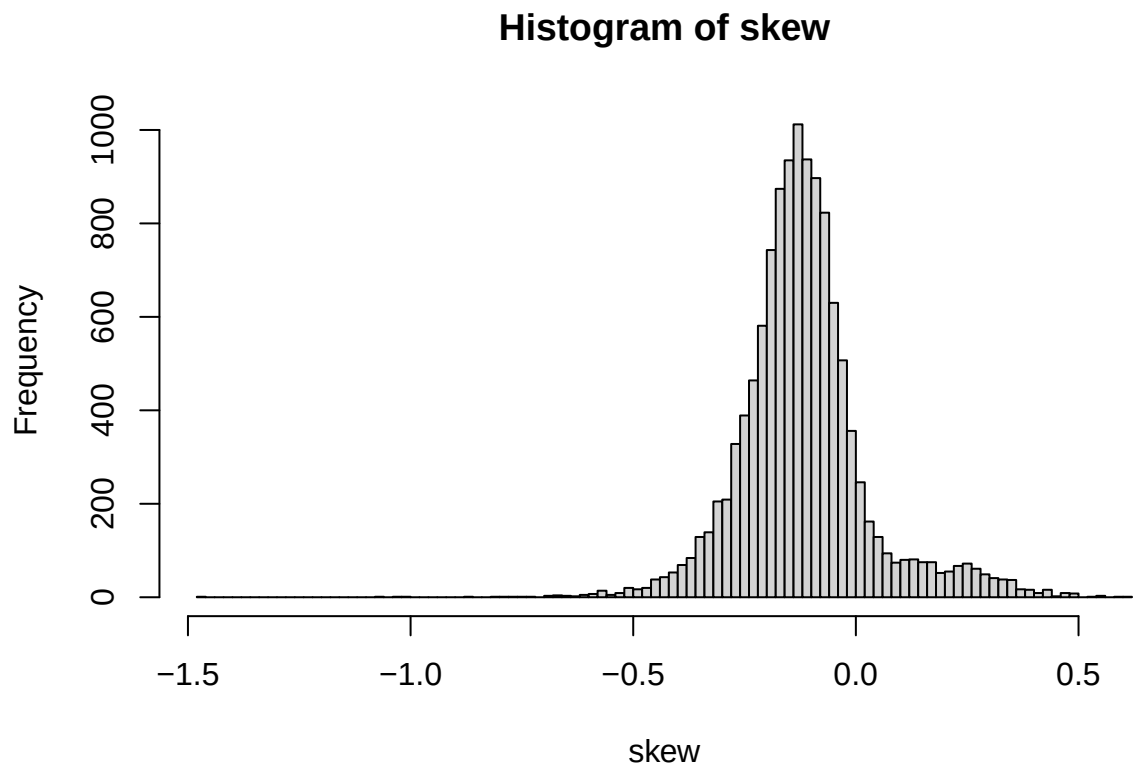

```r
qqnorm(IQR, main = "QQ Plot of IQR")
qqline(IQR, col = "steelblue", lwd = 2)
```



skew

Basic Analysis

```
skew <- voice$skew
hist(skew, breaks = 80, main = "Histogram of skew", xlab = "skew")
```

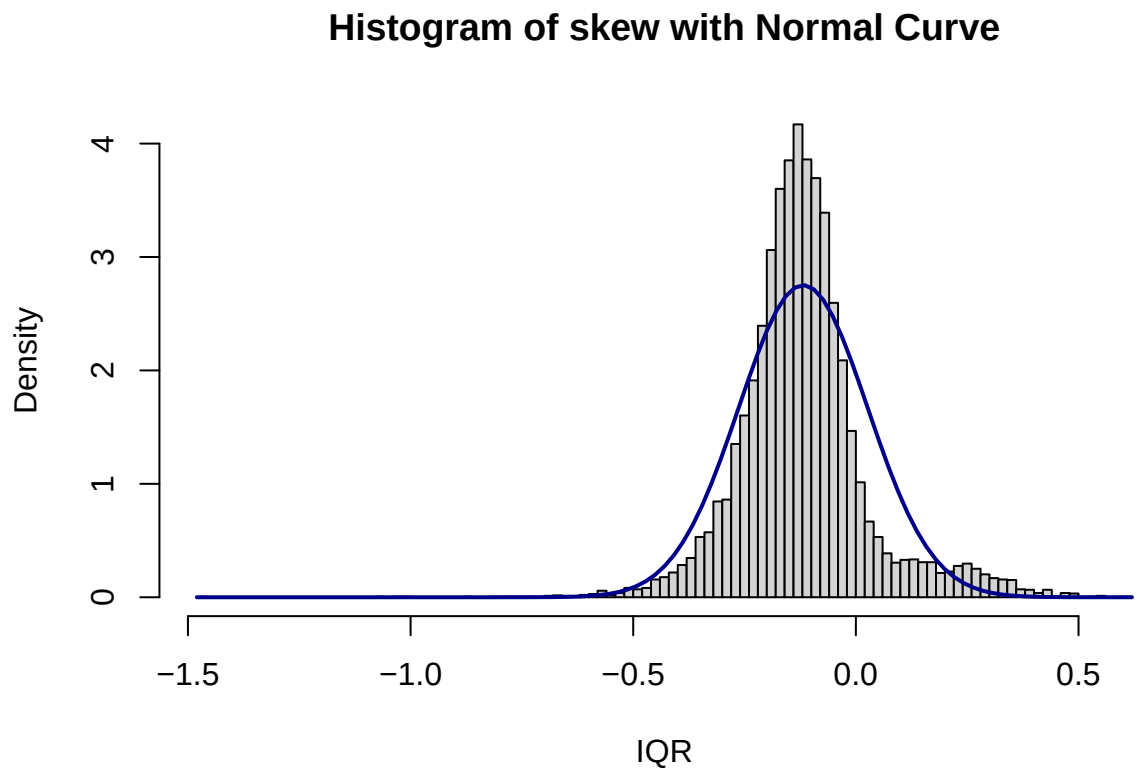


Then, we calculate the mean and standard deviation of the `skew` column.

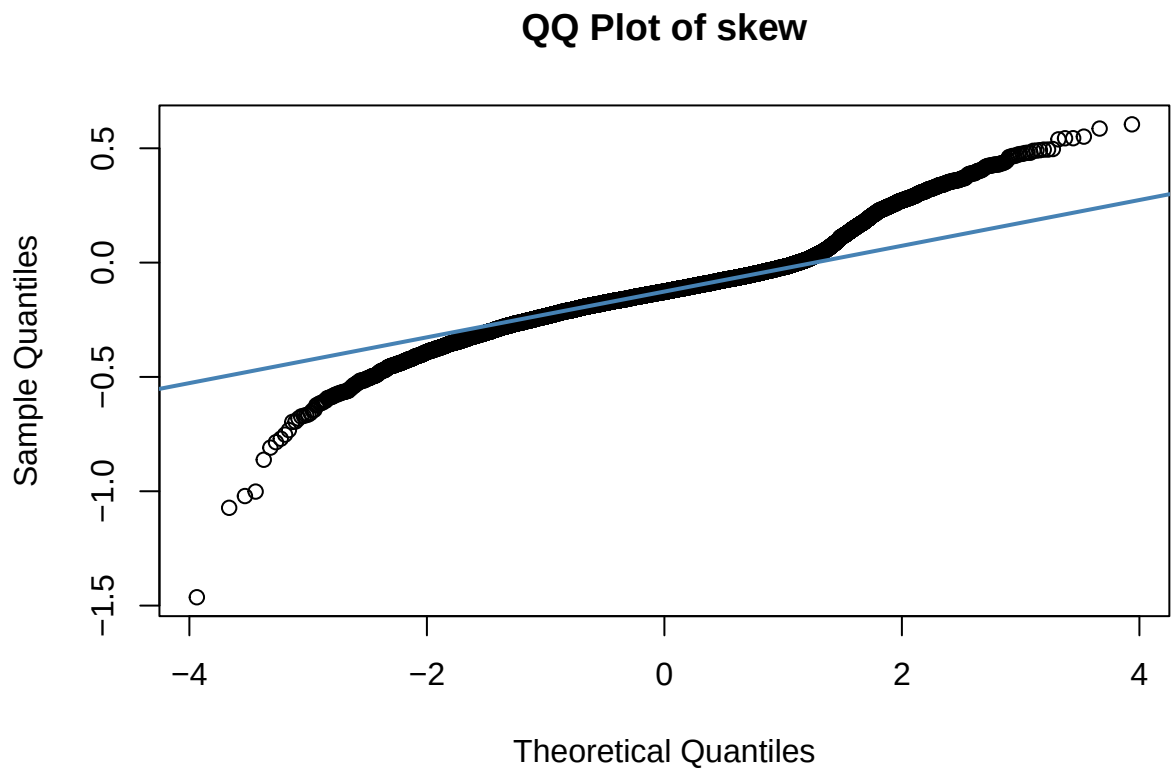
```
skew_mean <- mean(skew)
skew_sd <- sd(skew)
```

The `skew` column is almost normally distributed, we can see that from the histogram and QQ plot below.

```
hist(skew, probability = TRUE, main = "Histogram of skew with Normal Curve", xlab = "IQR", breaks = 80)
curve(dnorm(x, mean = skew_mean, sd = skew_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

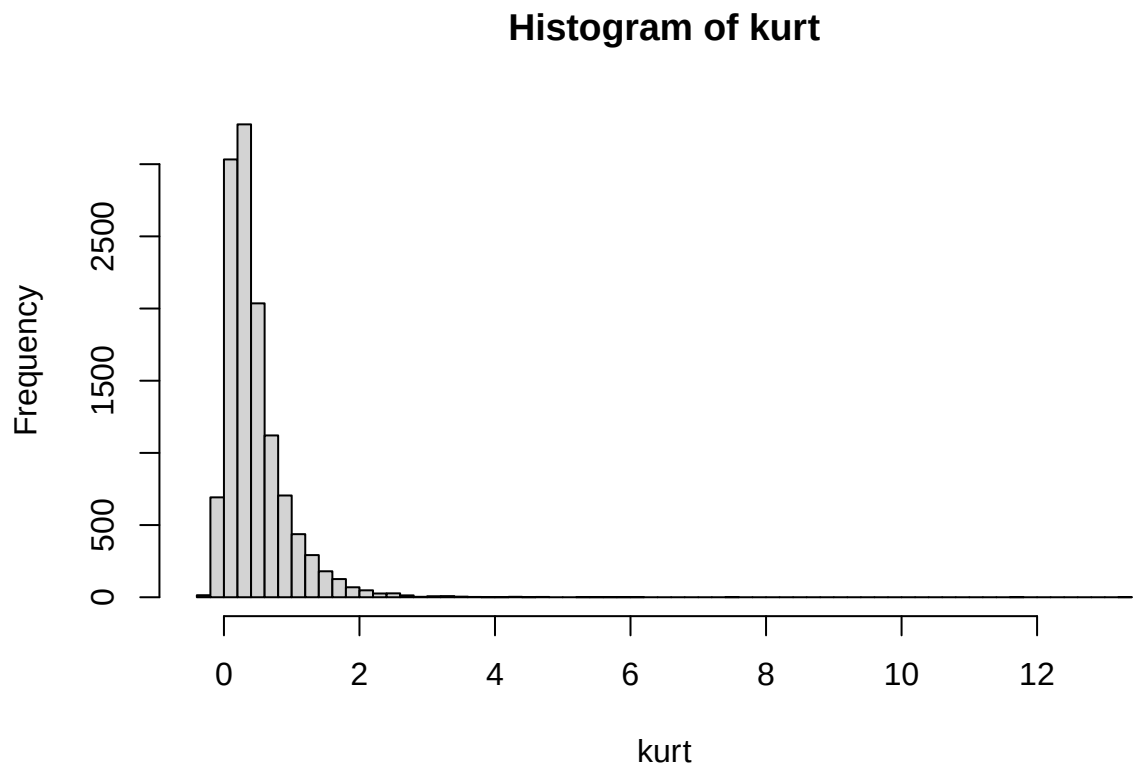



```
qqnorm(skew, main = "QQ Plot of skew")
qqline(skew, col = "steelblue", lwd = 2)
```



kurt

```
kurt <- voice$kurt
hist(kurt, breaks = 80, main = "Histogram of kurt", xlab = "kurt")
```

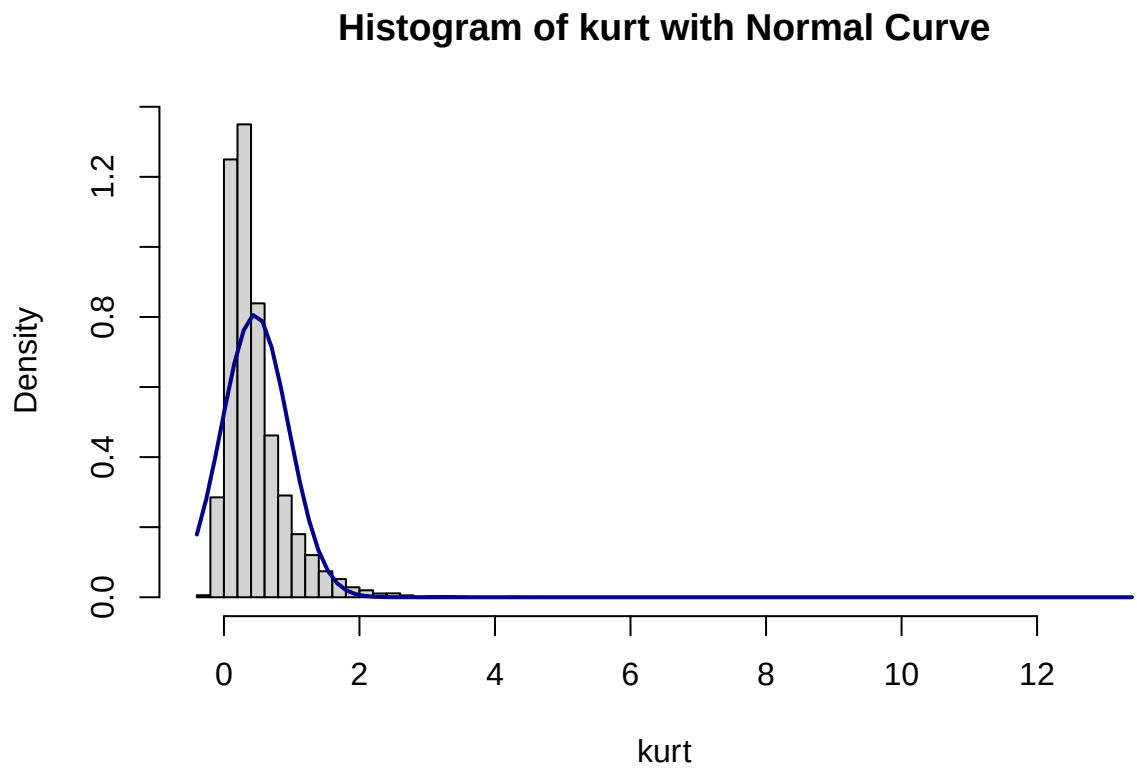


Then, we calculate the mean and standard deviation of the `kurt` column.

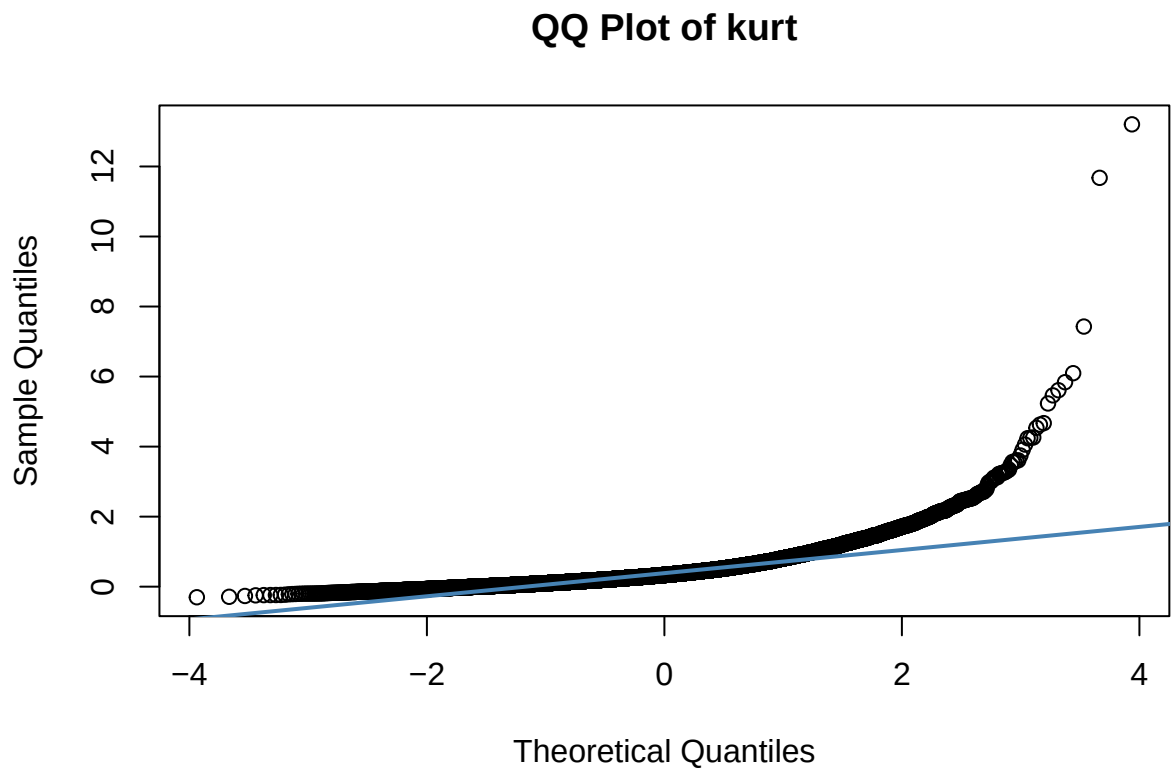
```
kurt_mean <- mean(kurt)
kurt_sd <- sd(kurt)
```

The `kurt` column is almost normally distributed, we can see that from the histogram and QQ plot below.

```
hist(kurt, probability = TRUE, main = "Histogram of kurt with Normal Curve", xlab = "kurt", breaks = 80
curve(dnorm(x, mean = kurt_mean, sd = kurt_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

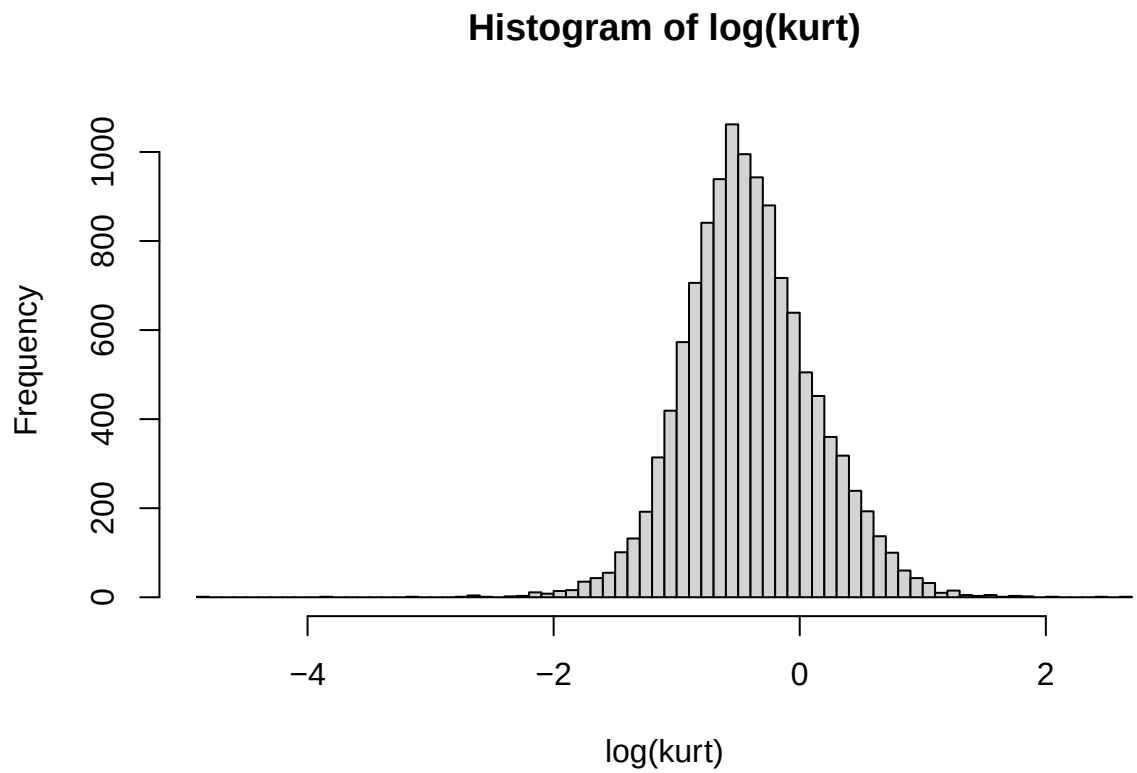



```r
qqnorm(kurt, main = "QQ Plot of kurt")
qqline(kurt, col = "steelblue", lwd = 2)
```



Log-transformed `kurt`

```
kurt_trans <- log(kurt + 0.31)
hist(kurt_trans, breaks = 80, main = "Histogram of log(kurt)", xlab = "log(kurt)")
```



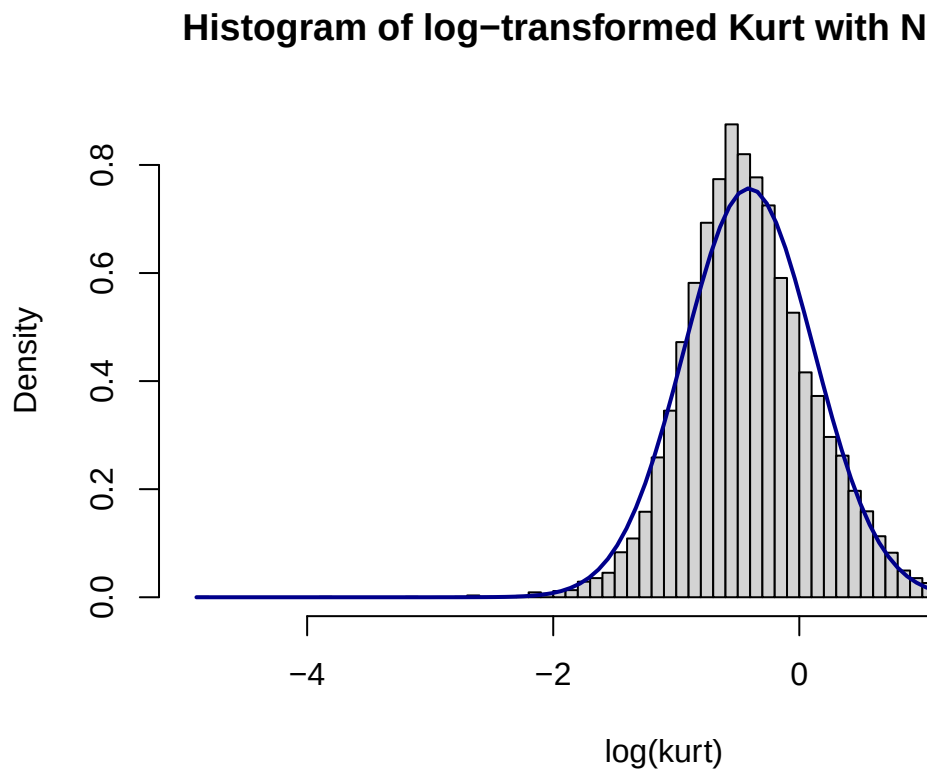

```
kurt_trans_mean <- mean(kurt_trans)
kurt_trans_sd <- sd(kurt_trans)
print(paste("Mean of log-transformed kurt:", kurt_trans_mean))
```

```
## [1] "Mean of log-transformed kurt: -0.409042098034566"
```

```
print(paste("Standard deviation of log-transformed kurt:", kurt_trans_sd))
```

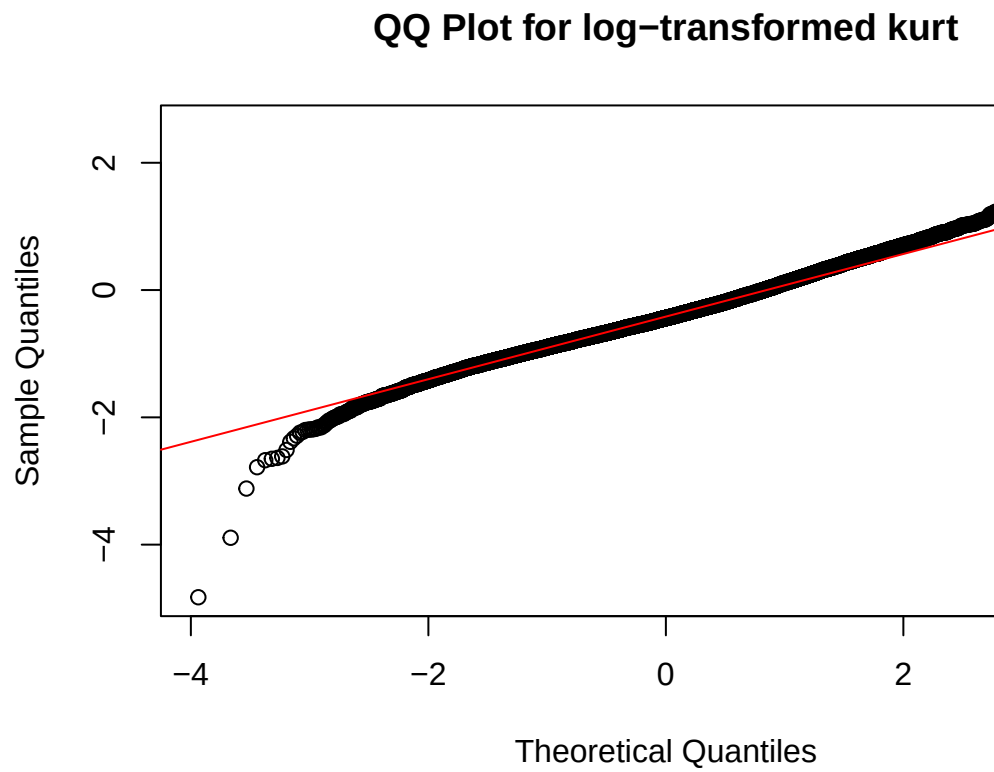
```
## [1] "Standard deviation of log-transformed kurt: 0.527162928424476"
```

```
hist(kurt_trans, probability = TRUE, main = "Histogram of log-transformed Kurt with Normal Curve", xlab
curve(dnorm(x, mean = kurt_trans_mean, sd = kurt_trans_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt
```



Histogram with overlaid normal curve

```
qqnorm(kurt_trans, main = "QQ Plot for log-transformed kurt")
qqline(kurt_trans, col = "red")
```

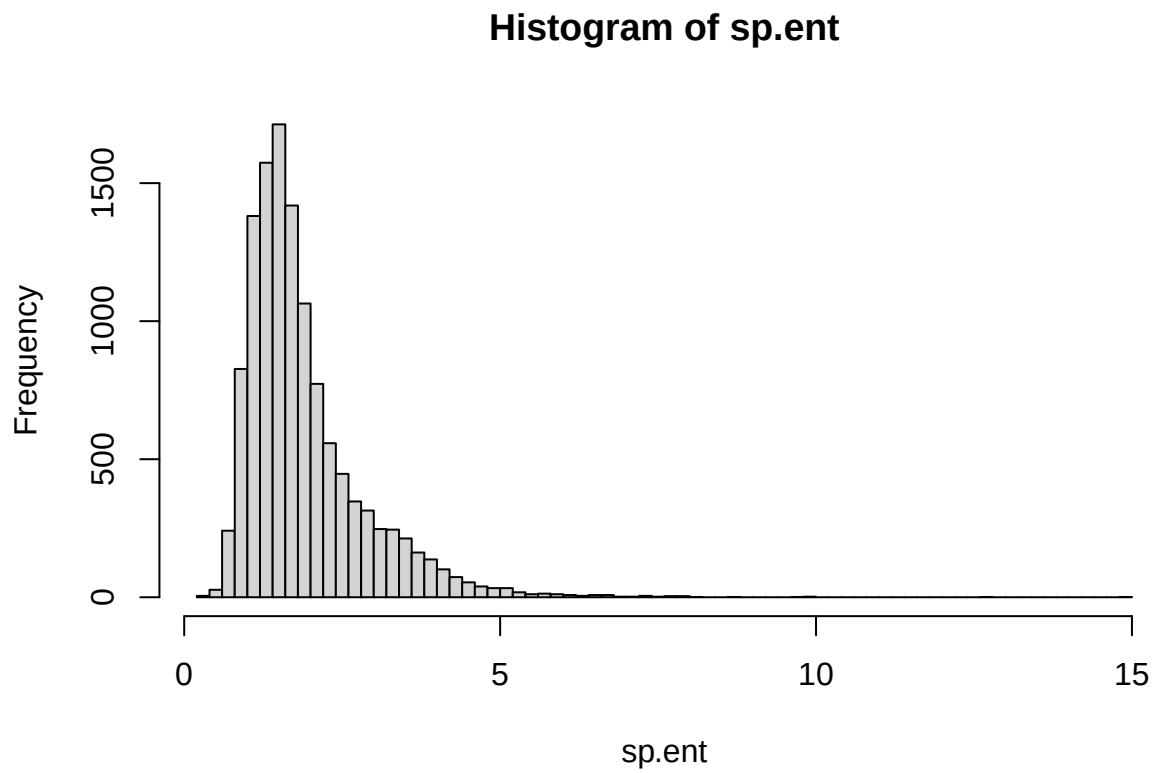


QQ plot for log-transformed `kurt`

sp.ent

Basic Analysis

```
sp_ent <- voice$sp.ent
hist(sp_ent, breaks = 80, main = "Histogram of sp.ent", xlab = "sp.ent")
```



Then, we calculate the mean and standard deviation of the `sp.ent` column.

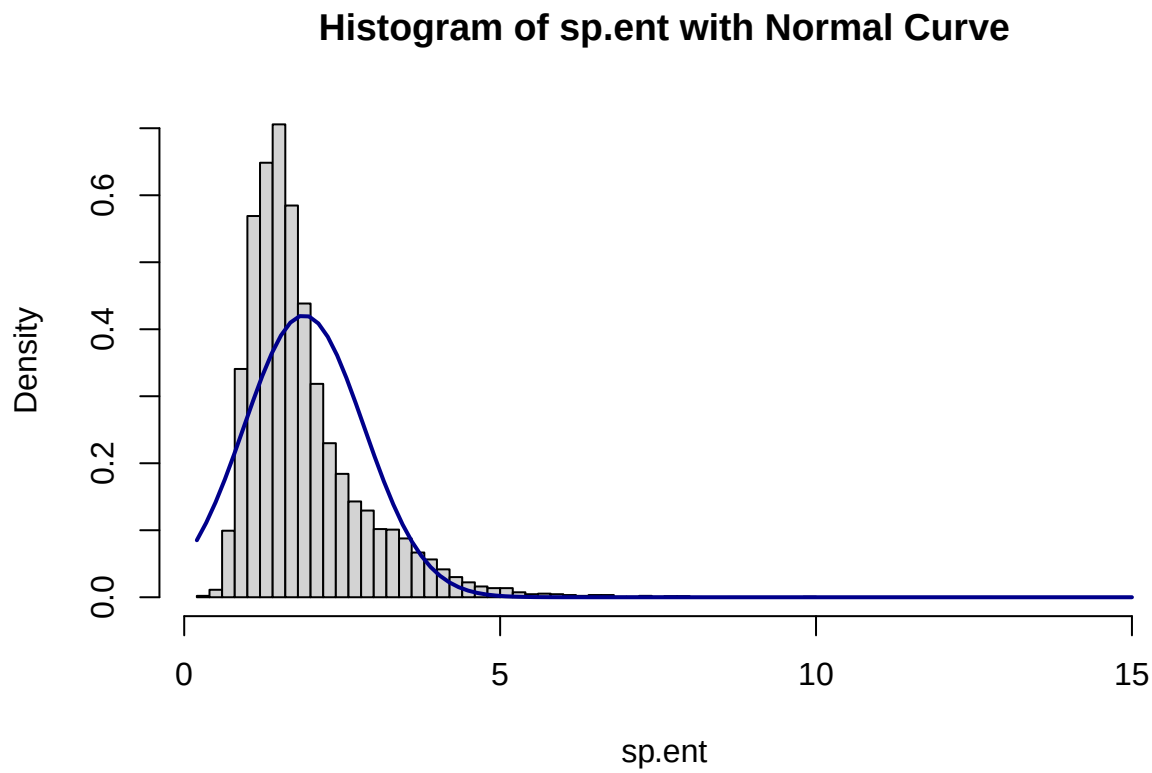
```
sp_ent_mean <- mean(sp_ent)
sp_ent_sd <- sd(sp_ent)
print(paste("Mean of sp.ent:", sp_ent_mean))
```

```
## [1] "Mean of sp.ent: 1.89564130104079"
```

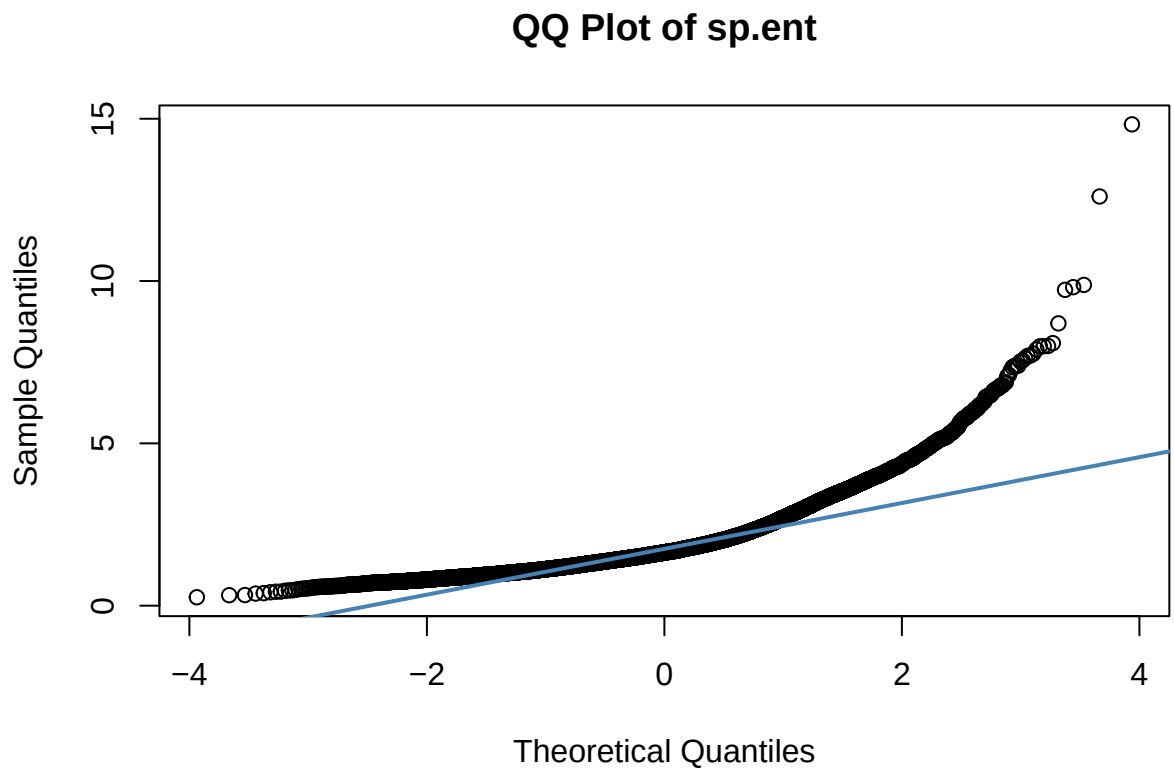
```
print(paste("Standard deviation of sp.ent:", sp_ent_sd))
```

```
## [1] "Standard deviation of sp.ent: 0.948445979428355"
```

```
hist(sp_ent, probability = TRUE, main = "Histogram of sp.ent with Normal Curve", xlab = "sp.ent", breaks
curve(dnorm(x, mean = sp_ent_mean, sd = sp_ent_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

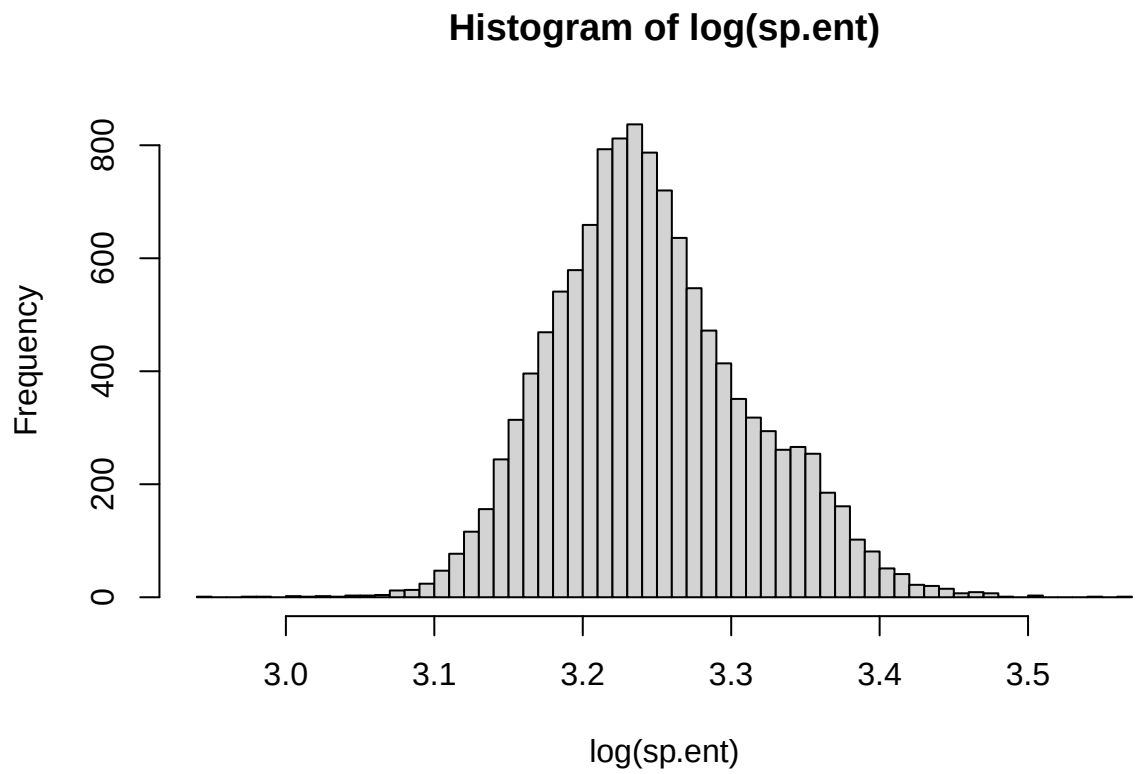



```
qqnorm(sp_ent, main = "QQ Plot of sp.ent")
qqline(sp_ent, col = "steelblue", lwd = 2)
```



Log-transformed `sp.ent`

```
sp_ent_trans <- sqrt(log(sp_ent) + 10)
hist(sp_ent_trans, breaks = 80, main = "Histogram of log(sp.ent)", xlab = "log(sp.ent)")
```



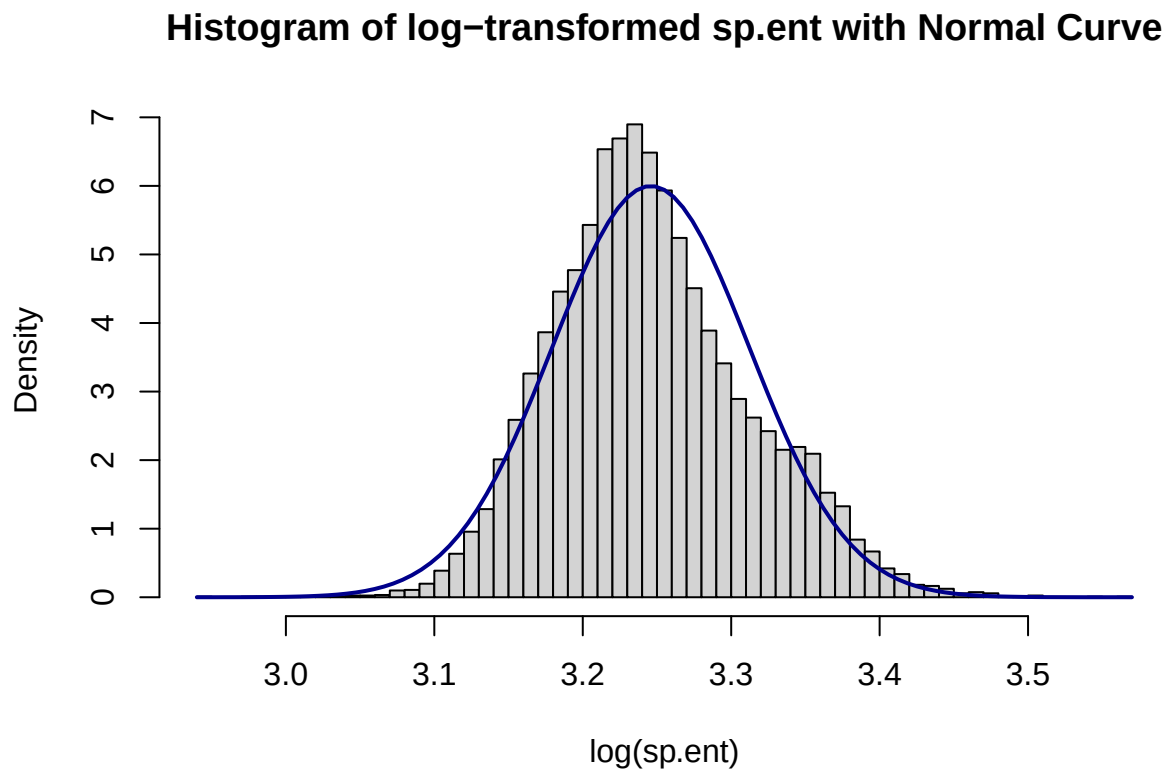

```
sp_ent_trans_mean <- mean(sp_ent_trans)
sp_ent_trans_sd <- sd(sp_ent_trans)
print(paste("Mean of log-transformed sp.ent:", sp_ent_trans_mean))
```

```
## [1] "Mean of log-transformed sp.ent: 3.24581141189549"
```

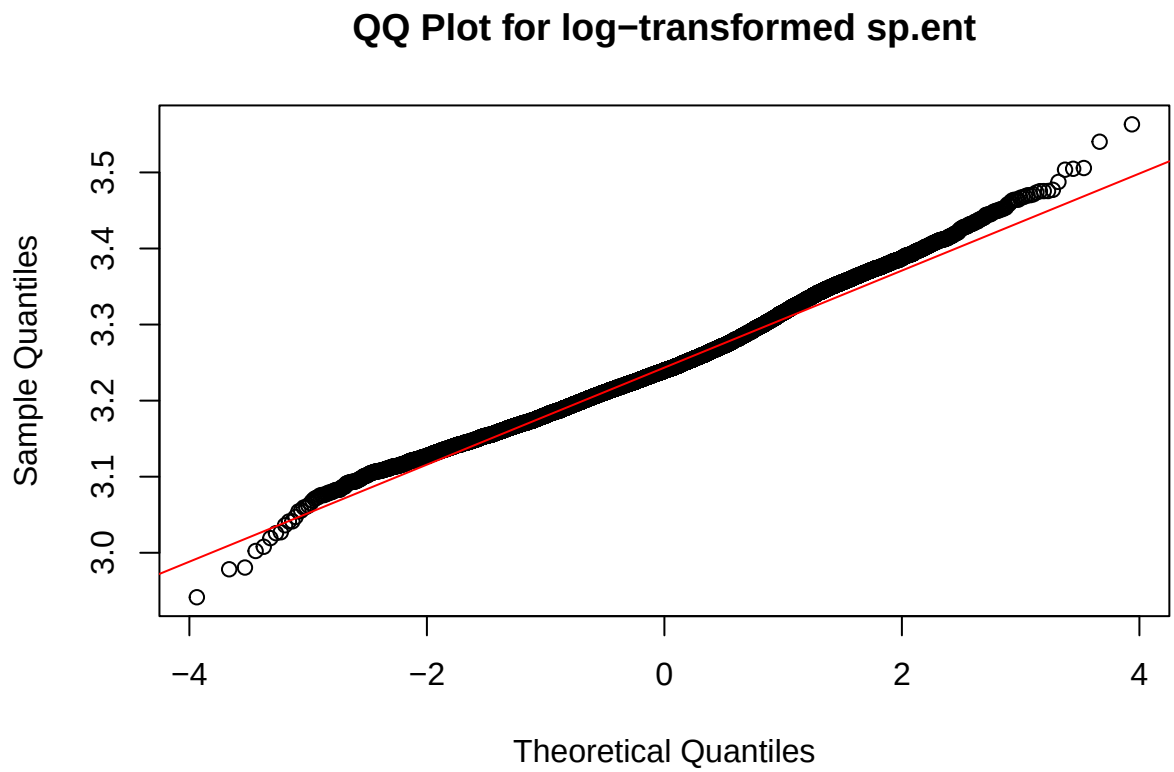
```
print(paste("Standard deviation of log-transformed sp.ent:", sp_ent_trans_sd))
```

```
## [1] "Standard deviation of log-transformed sp.ent: 0.0665096016427285"
```

```
hist(sp_ent_trans, probability = TRUE, main = "Histogram of log-transformed sp.ent with Normal Curve", 
curve(dnorm(x, mean = sp_ent_trans_mean, sd = sp_ent_trans_sd), col = "darkblue", lwd = 2, add = TRUE, 
```



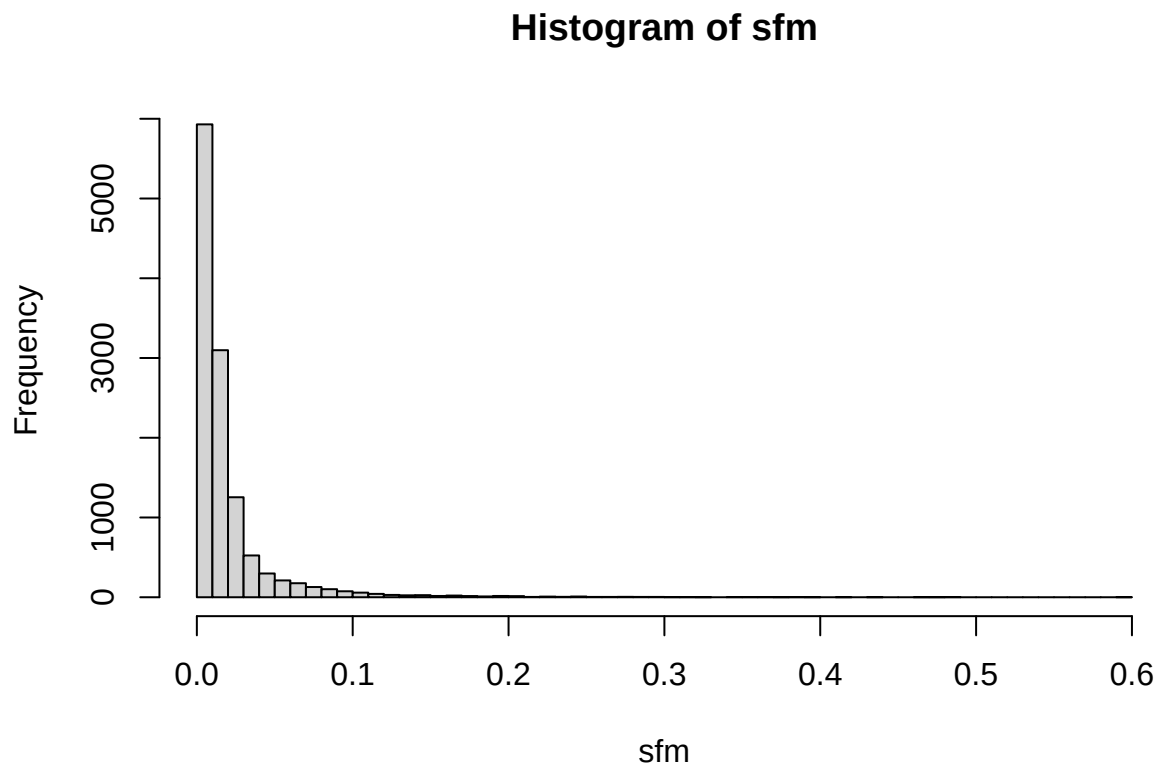

```
qqnorm(sp_ent_trans, main = "QQ Plot for log-transformed sp.ent")
qqline(sp_ent_trans, col = "red")
```



sfm

Basic Analysis

```
sfm <- voice$sfm
hist(sfm, breaks = 80, main = "Histogram of sfm", xlab = "sfm")
```



Then, we calculate the mean and standard deviation of the `sfm` column.

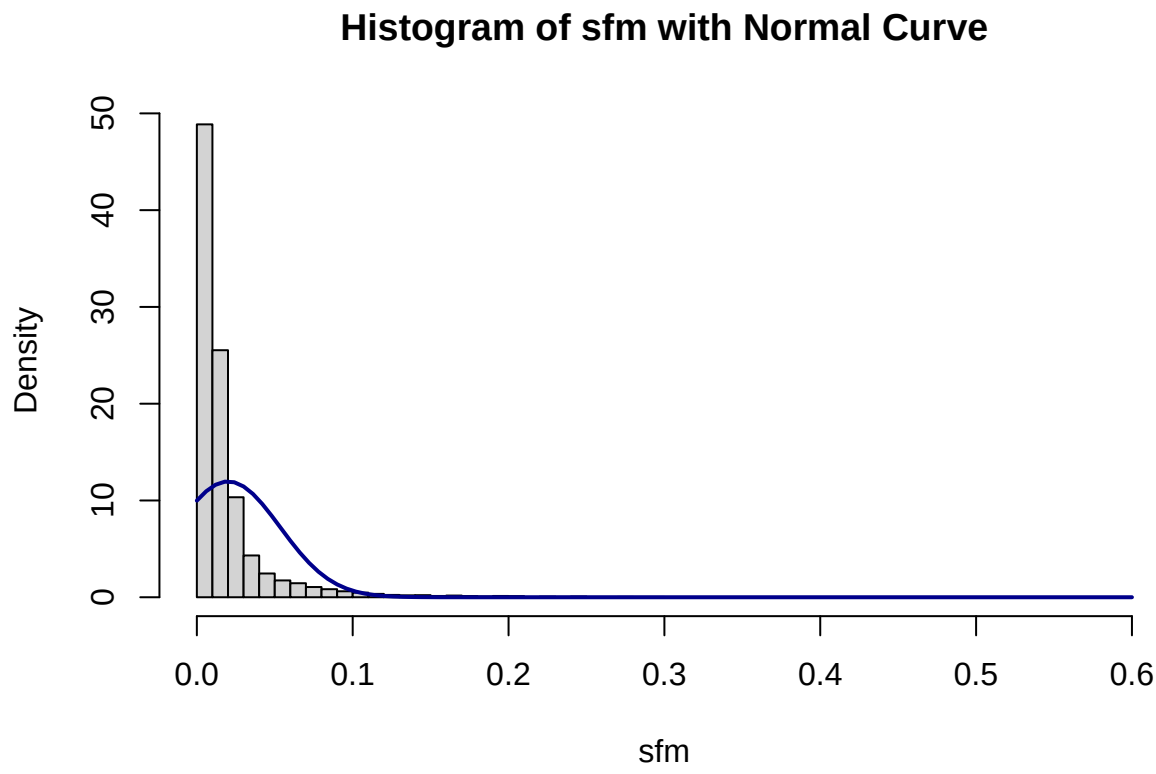
```
sfm_mean <- mean(sfm)
sfm_sd <- sd(sfm)
print(paste("Mean of sfm:", sfm_mean))
```

```
## [1] "Mean of sfm: 0.0200547303977314"
```

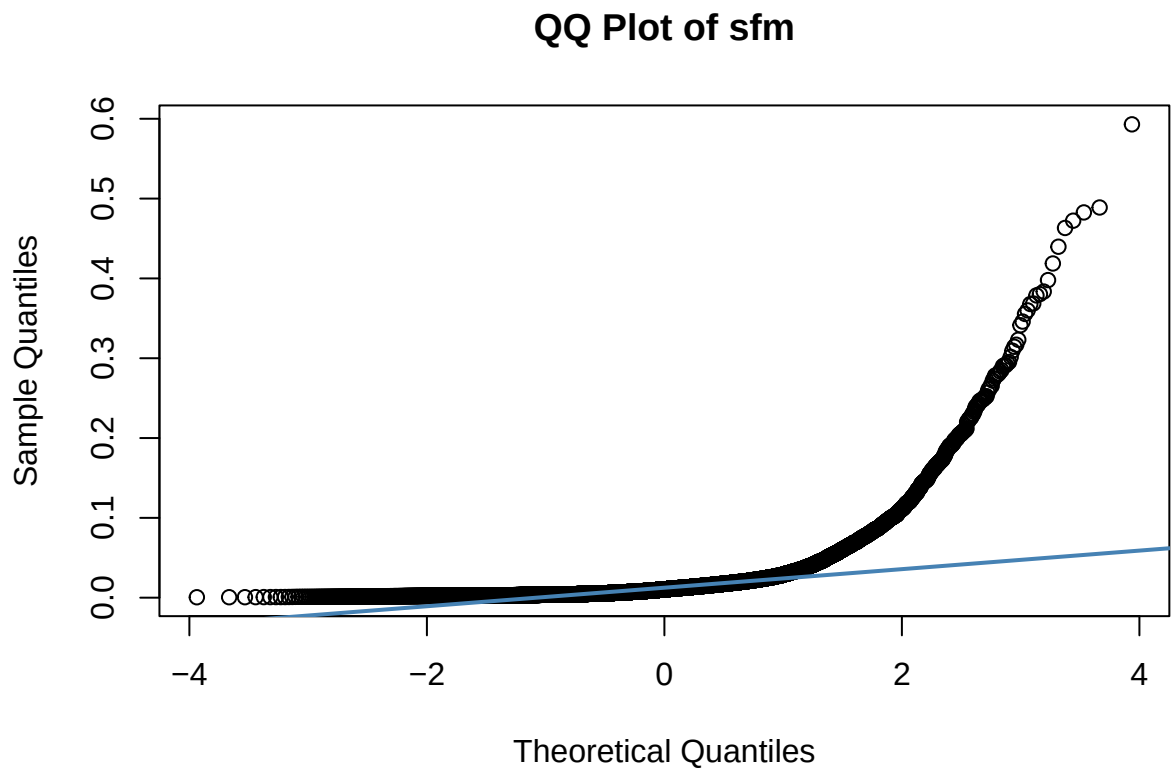
```
print(paste("Standard deviation of sfm:", sfm_sd))
```

```
## [1] "Standard deviation of sfm: 0.0333383291477354"
```

```
hist(sfm, probability = TRUE, main = "Histogram of sfm with Normal Curve", xlab = "sfm", breaks = 80)
curve(dnorm(x, mean = sfm_mean, sd = sfm_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

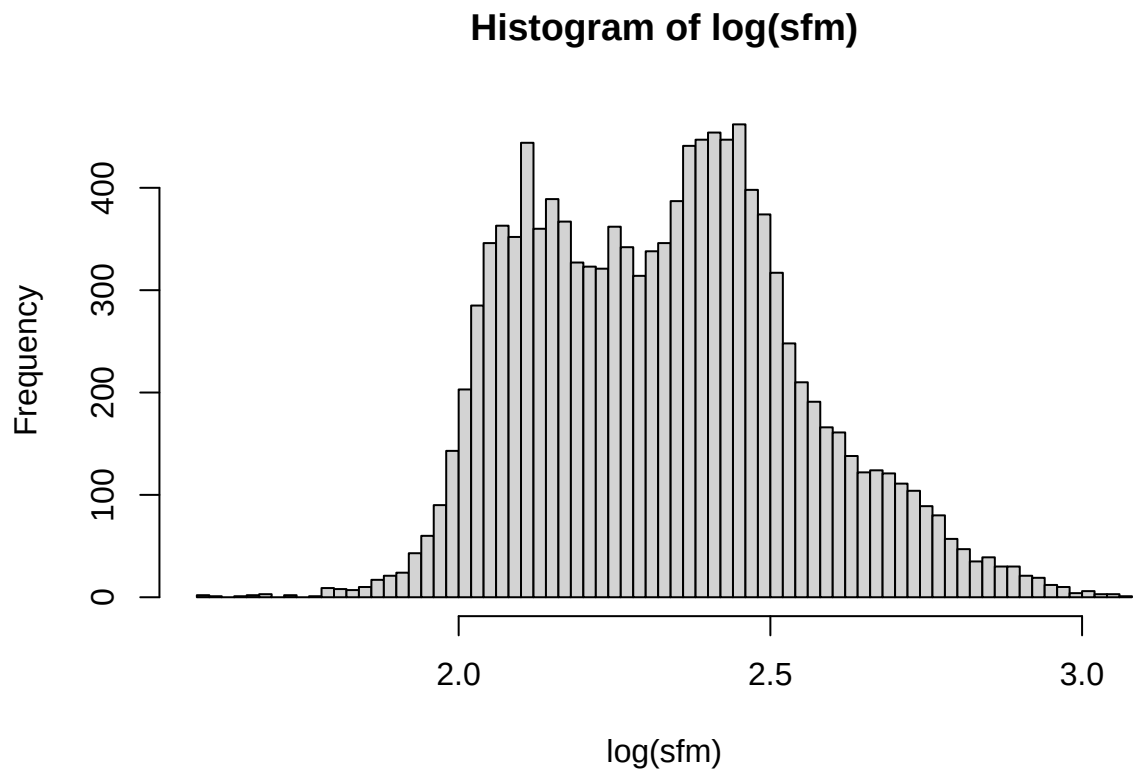



```r
qqnorm(sfm, main = "QQ Plot of sfm")
qqline(sfm, col = "steelblue", lwd = 2)
```



Log-transformed `sfm`

```
sfm_trans <- sqrt(log(sfm) + 10)
hist(sfm_trans, breaks = 80, main = "Histogram of log(sfm)", xlab = "log(sfm)")
```



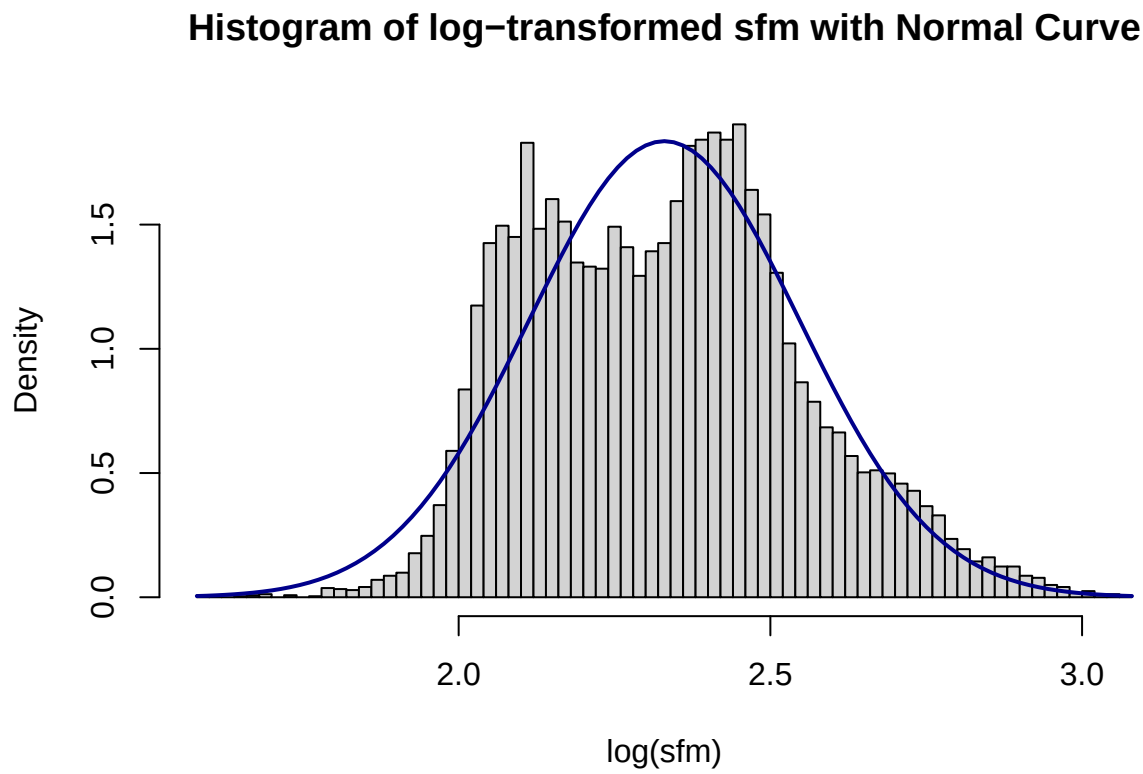

```
sfm_trans_mean <- mean(sfm_trans)
sfm_trans_sd <- sd(sfm_trans)
print(paste("Mean of log-transformed sfm:", sfm_trans_mean))
```

```
## [1] "Mean of log-transformed sfm: 2.32986230002159"
```

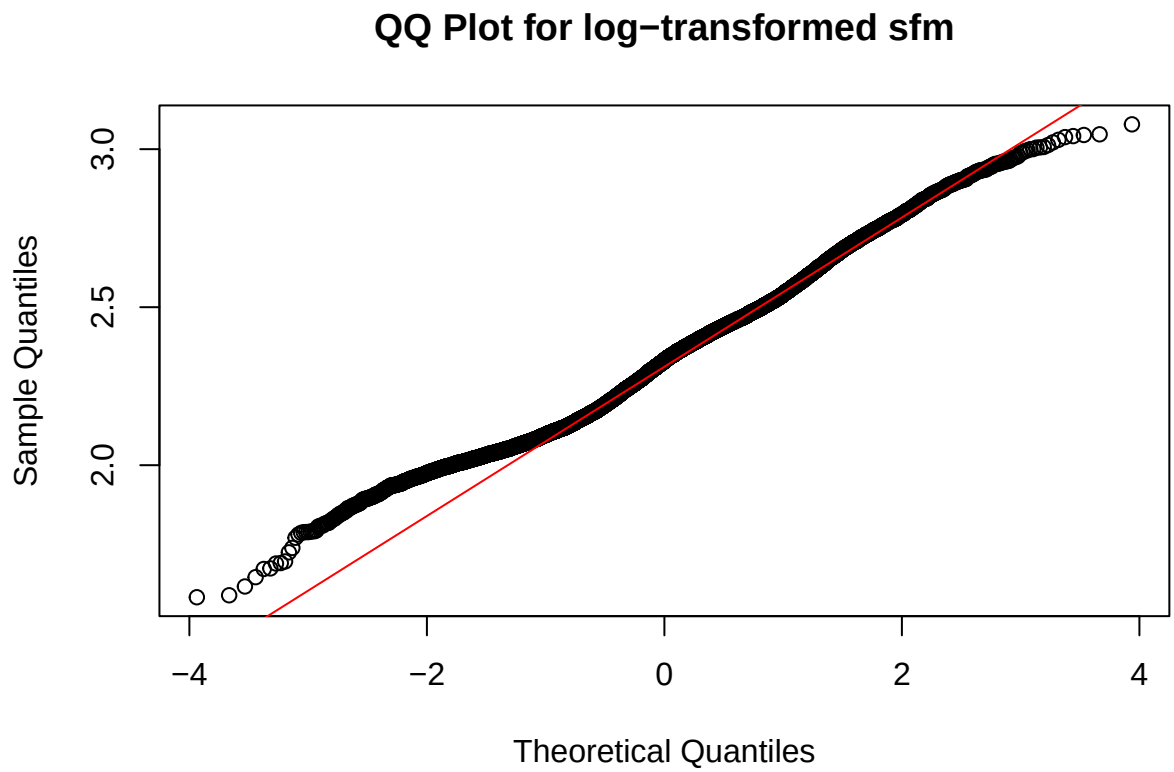
```
print(paste("Standard deviation of log-transformed sfm:", sfm_trans_sd))
```

```
## [1] "Standard deviation of log-transformed sfm: 0.217289569293435"
```

```
hist(sfm_trans, probability = TRUE, main = "Histogram of log-transformed sfm with Normal Curve", xlab =
curve(dnorm(x, mean = sfm_trans_mean, sd = sfm_trans_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt =
```



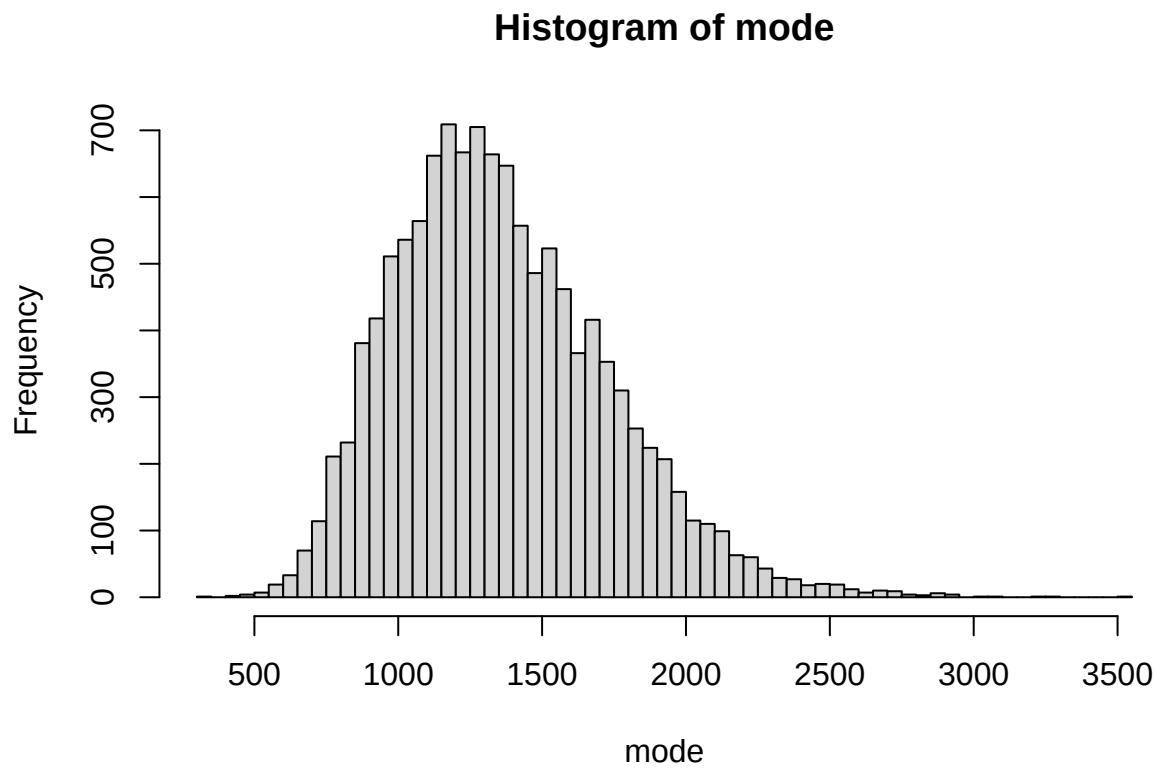

```
qqnorm(sfm_trans, main = "QQ Plot for log-transformed sfm")
qqline(sfm_trans, col = "red")
```



mode

Basic Analysis

```
mode <- voice$mode
hist(mode, breaks = 80, main = "Histogram of mode", xlab = "mode")
```



Then, we calculate the mean and standard deviation of the `mode` column.

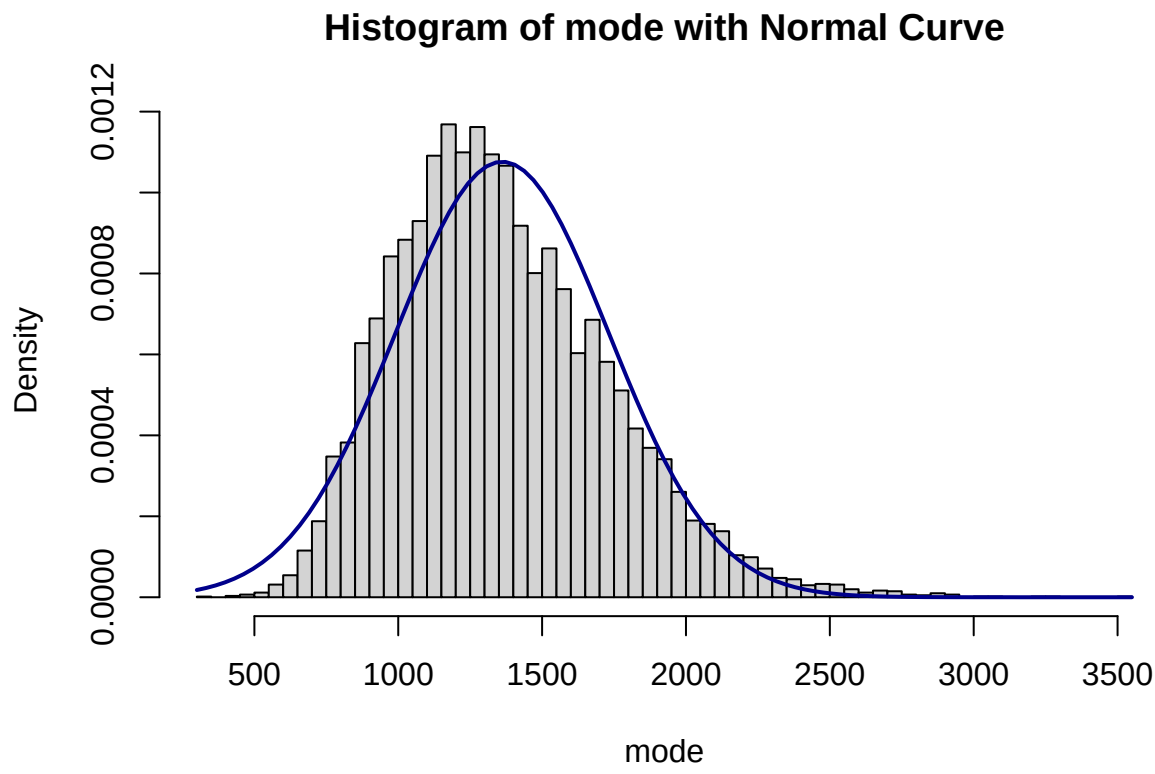
```
mode_mean <- mean(mode)
mode_sd <- sd(mode)
print(paste("Mean of mode:", mode_mean))
```

```
## [1] "Mean of mode: 1361.15949796209"
```

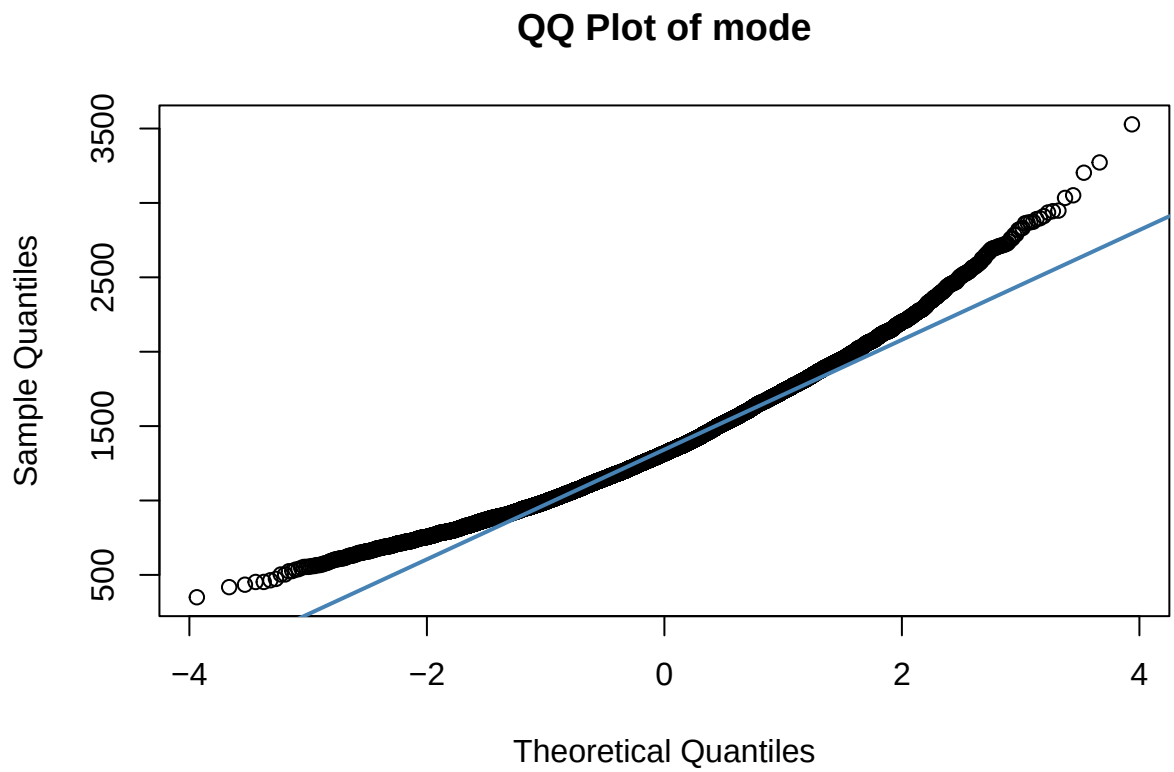
```
print(paste("Standard deviation of mode:", mode_sd))
```

```
## [1] "Standard deviation of mode: 370.71921645733"
```

```
hist(mode, probability = TRUE, main = "Histogram of mode with Normal Curve", xlab = "mode", breaks = 80
curve(dnorm(x, mean = mode_mean, sd = mode_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

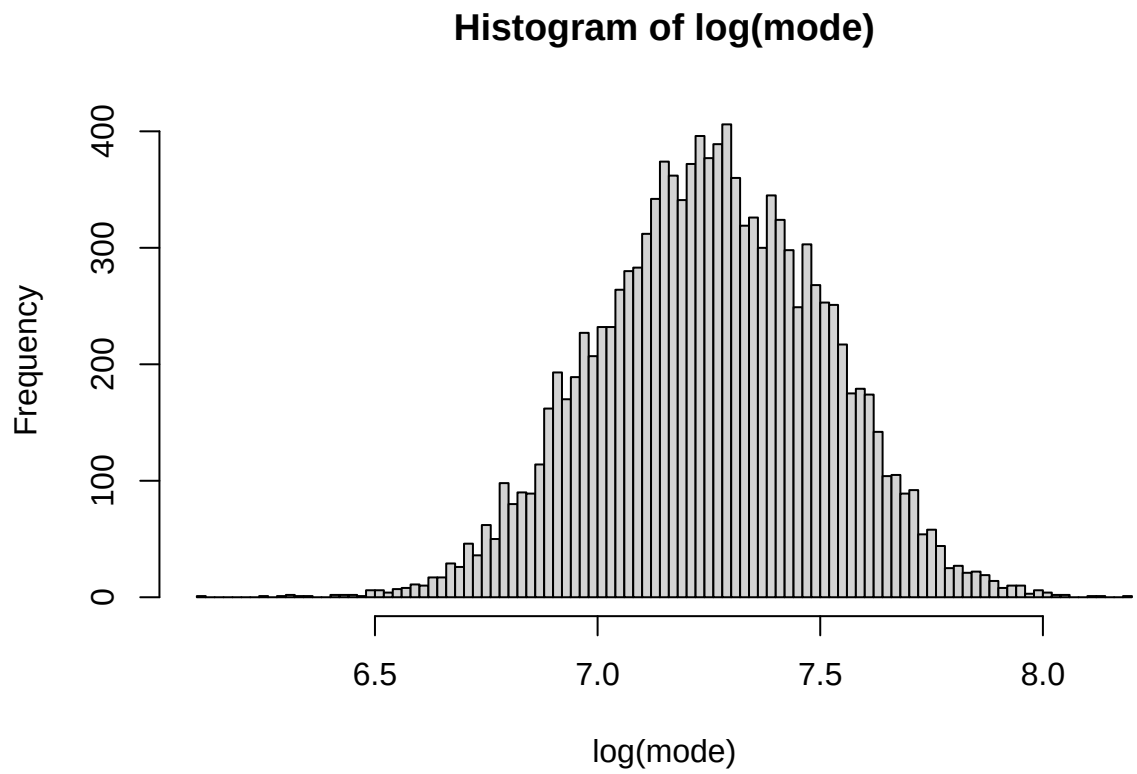



```r
qqnorm(mode, main = "QQ Plot of mode")
qqline(mode, col = "steelblue", lwd = 2)
```



Log-transformed `mode`

```
mode_trans <- log(mode + 100)
hist(mode_trans, breaks = 80, main = "Histogram of log(mode)", xlab = "log(mode)")
```



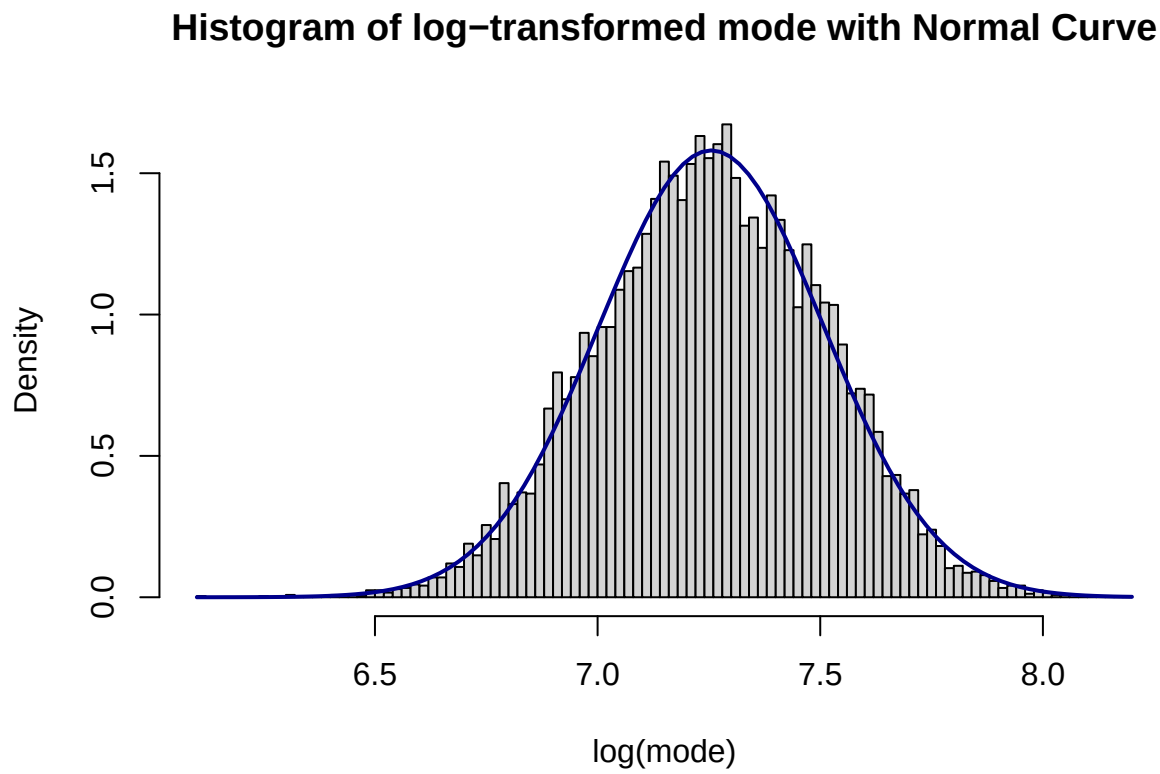

```
mode_trans_mean <- mean(mode_trans)
mode_trans_sd <- sd(mode_trans)
print(paste("Mean of log-transformed mode:", mode_trans_mean))
```

```
## [1] "Mean of log-transformed mode: 7.2553355596648"
```

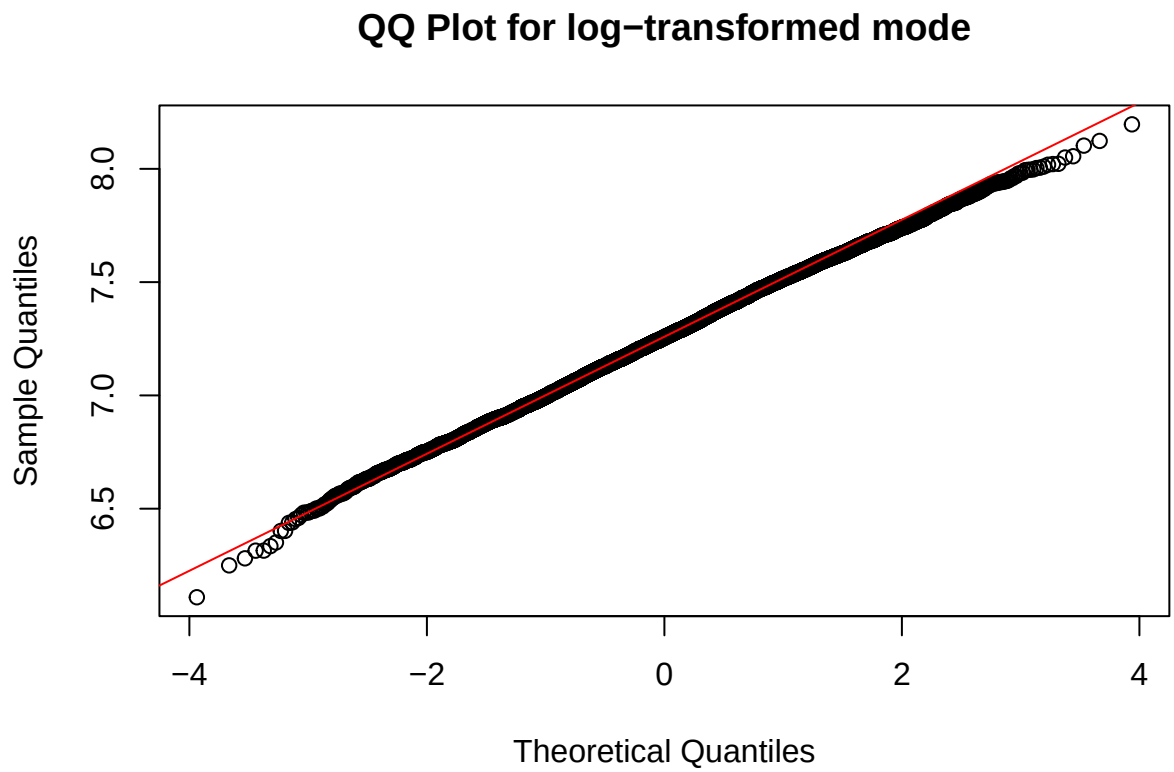
```
print(paste("Standard deviation of log-transformed mode:", mode_trans_sd))
```

```
## [1] "Standard deviation of log-transformed mode: 0.252375165740319"
```

```
hist(mode_trans, probability = TRUE, main = "Histogram of log-transformed mode with Normal Curve", xlab
curve(dnorm(x, mean = mode_trans_mean, sd = mode_trans_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt
```




```r
qqnorm(mode_trans, main = "QQ Plot for log-transformed mode")
qqline(mode_trans, col = "red")
```

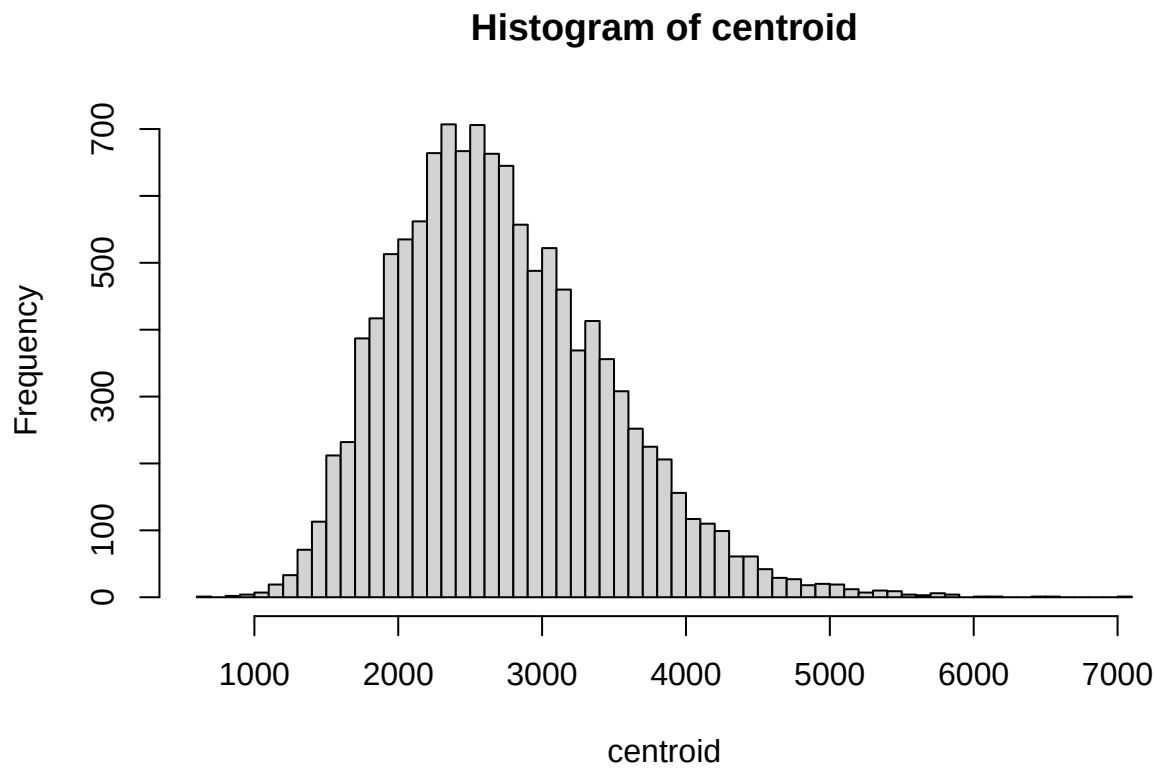


Centroid

Basic Analysis

We firstly load the data and draw a histogram of the `centroid` column.

```
centroid <- voice$centroid
hist(centroid, breaks = 80, main = "Histogram of centroid", xlab = "centroid")
```



Then, we calculate the mean and standard deviation of the `centroid` column.

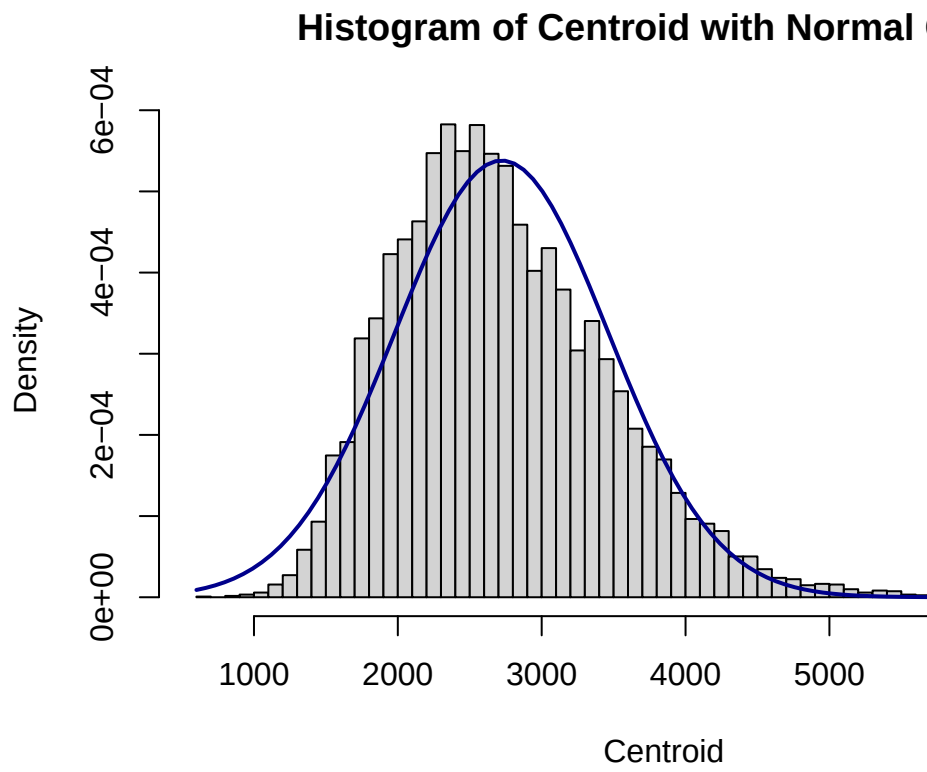
```
centroid_mean <- mean(centroid)
centroid_sd <- sd(centroid)
print(paste("Mean of centroid:", centroid_mean))
```

```
## [1] "Mean of centroid: 2721.31899603461"
```

```
print(paste("Standard deviation of centroid:", centroid_sd))
```

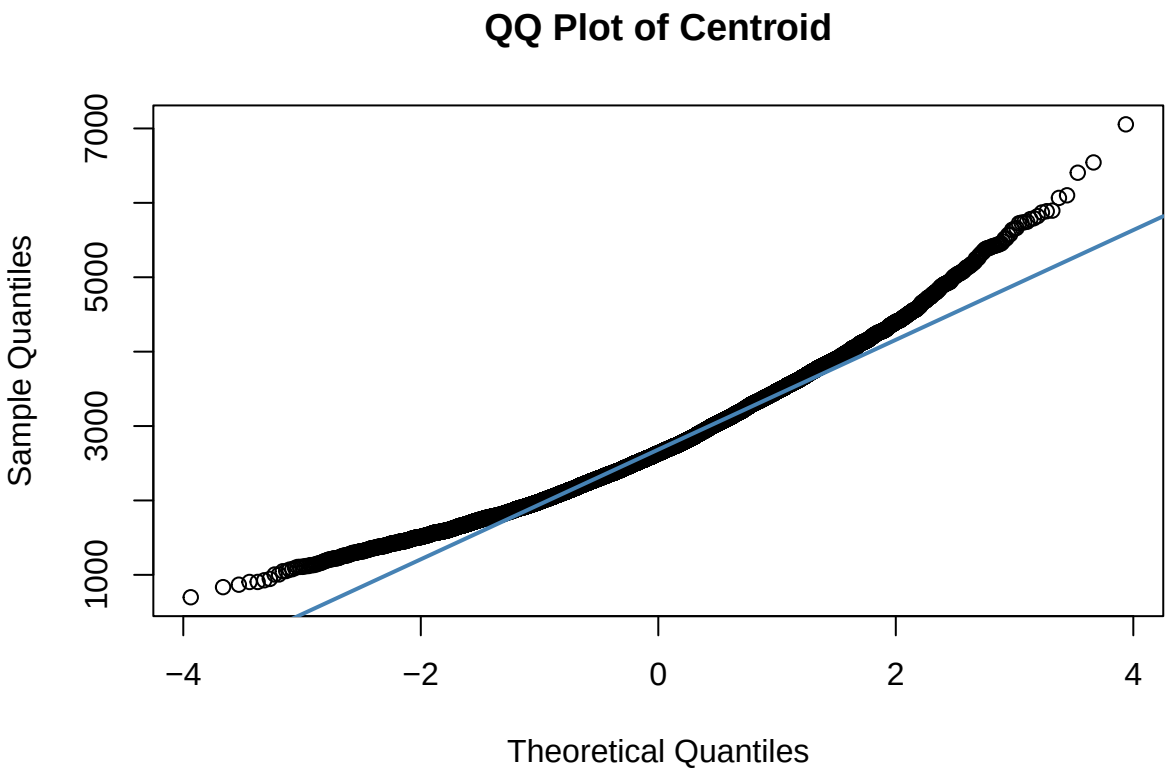
```
## [1] "Standard deviation of centroid: 741.438432305616"
```

```
hist(centroid, probability = TRUE, main = "Histogram of Centroid with Normal Curve", xlab = "Centroid",
curve(dnorm(x, mean = centroid_mean, sd = centroid_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n
```



Histogram with overlaid normal curve

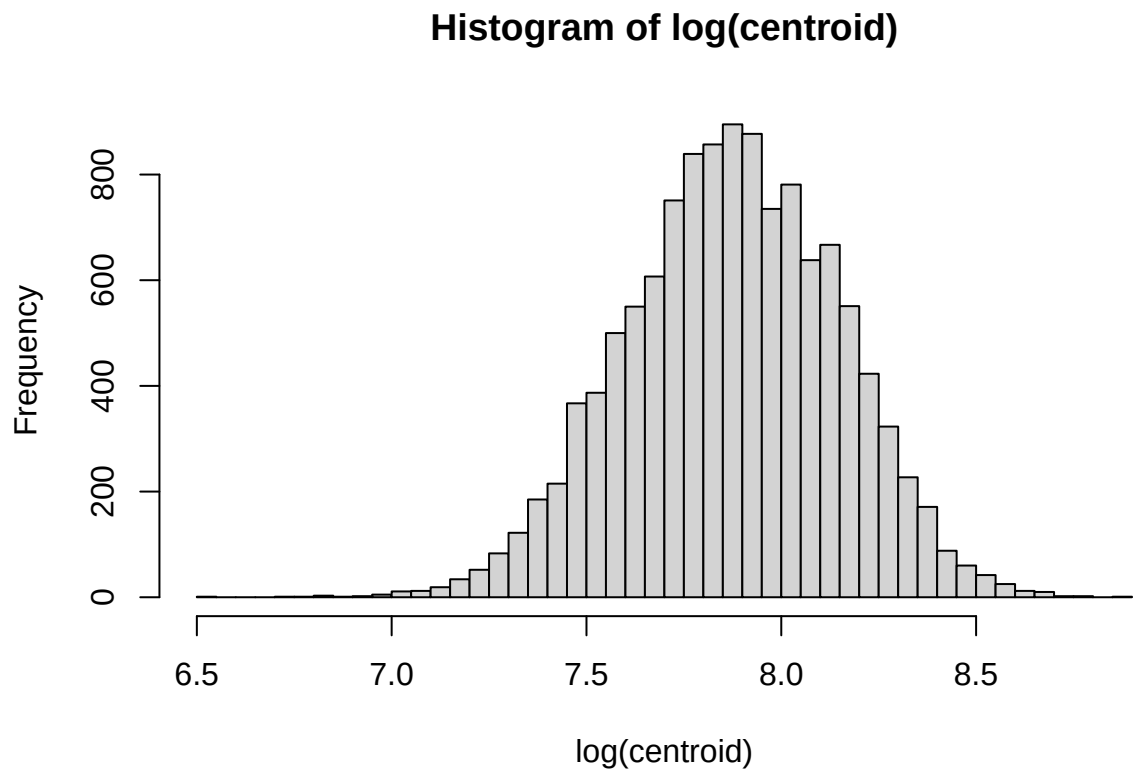
```
qqnorm(centroid, main = "QQ Plot of Centroid")
qqline(centroid, col = "steelblue", lwd = 2)
```



QQ plot

Log-transformed `centroid`

```
centroid_trans <- log(centroid)
hist(centroid_trans, breaks = 80, main = "Histogram of log(centroid)", xlab = "log(centroid)")
```



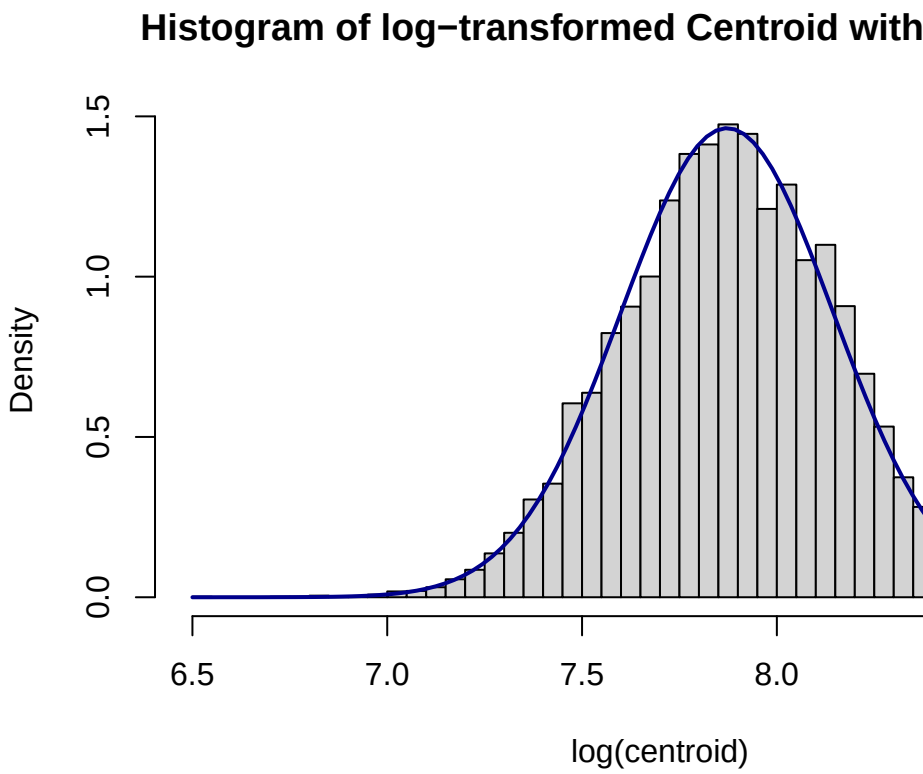

```
centroid_trans_mean <- mean(centroid_trans)
centroid_trans_sd <- sd(centroid_trans)
print(paste("Mean of log-transformed centroid:", centroid_trans_mean))
```

```
## [1] "Mean of log-transformed centroid: 7.87214455103932"
```

```
print(paste("Standard deviation of log-transformed centroid:", centroid_trans_sd))
```

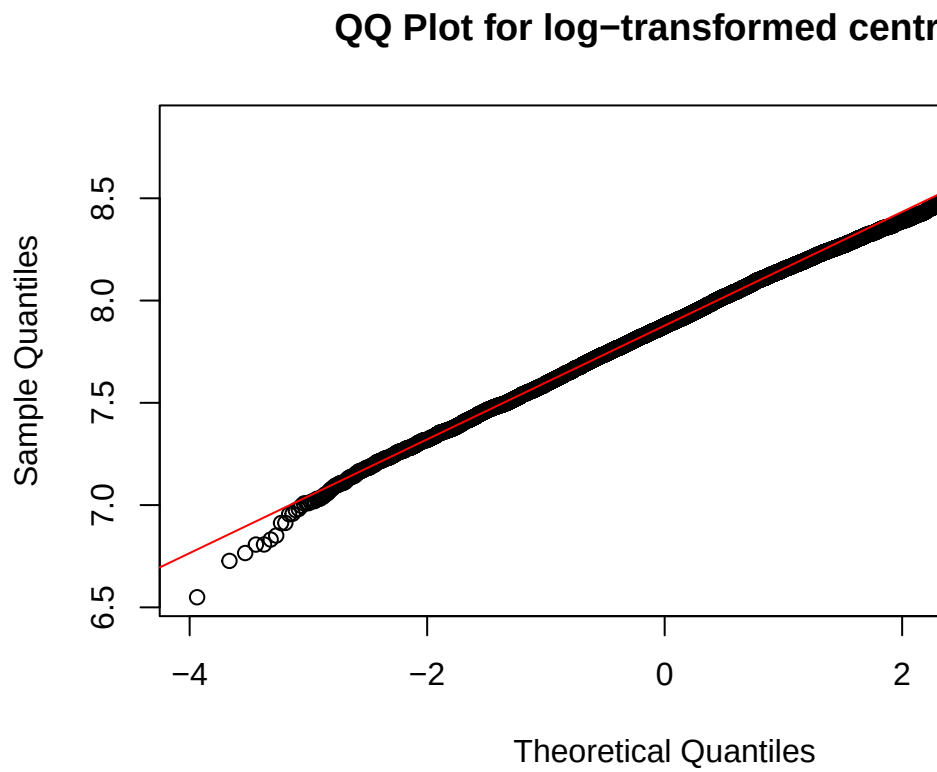
```
## [1] "Standard deviation of log-transformed centroid: 0.272639109655038"
```

```
hist(centroid_trans, probability = TRUE, main = "Histogram of log-transformed Centroid with Normal Curve
curve(dnorm(x, mean = centroid_trans_mean, sd = centroid_trans_sd), col = "darkblue", lwd = 2, add = TRU
```



Histogram with overlaid normal curve

```
qqnorm(centroid_trans, main = "QQ Plot for log-transformed centroid")
qqline(centroid_trans, col = "red")
```

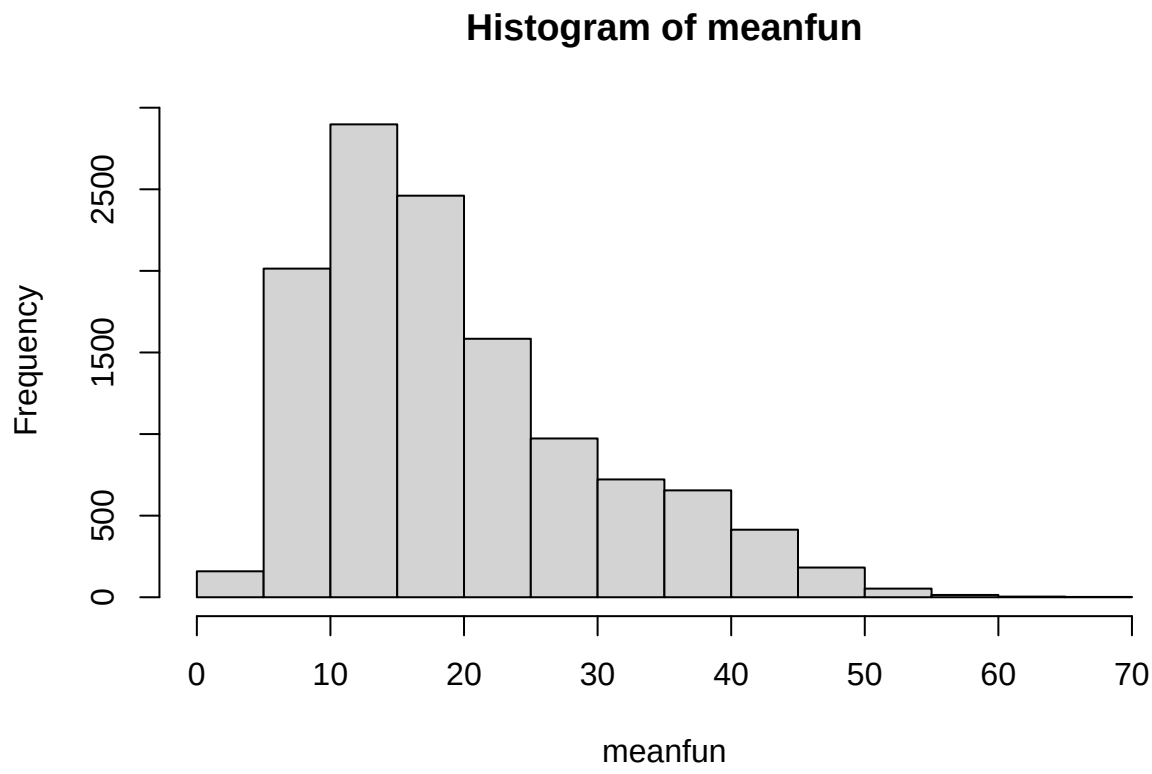


QQ plot for log-transformed `centroid`

MeanFun

Basic Analysis

```
meanfun <- voice$meanfun
hist(meanfun)
```

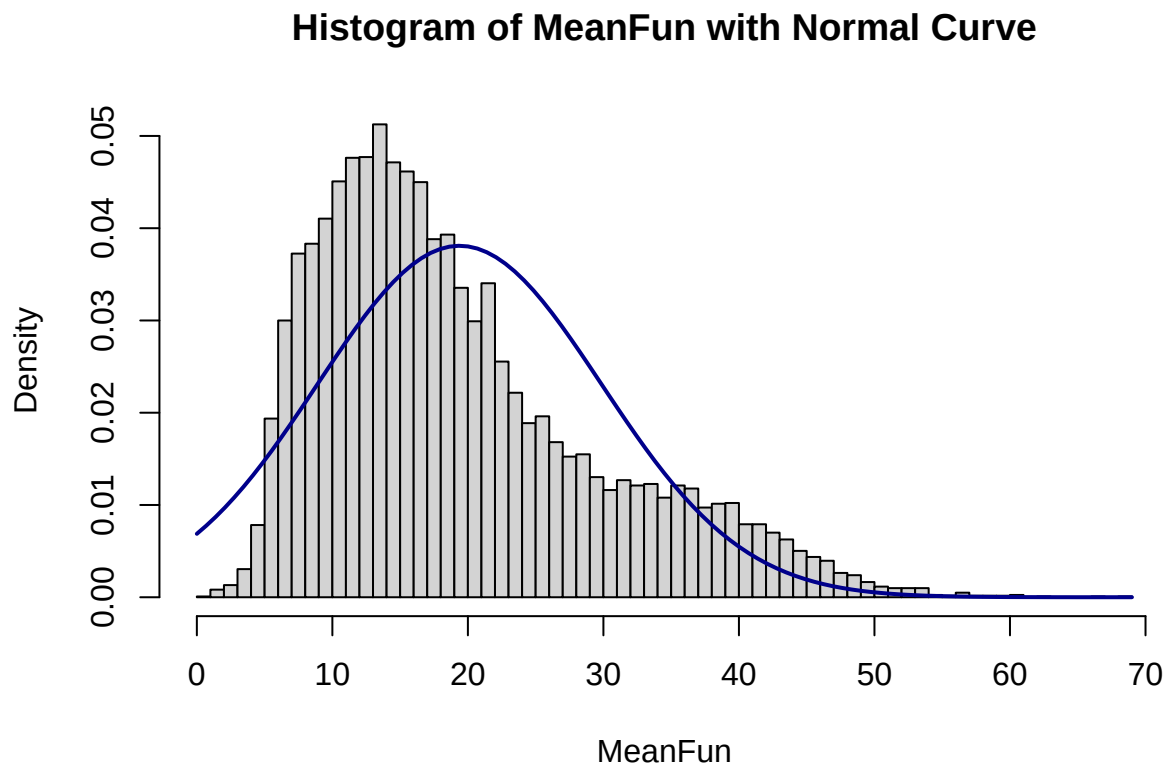


Then, we calculate the mean and standard deviation of the `meanfun` column.

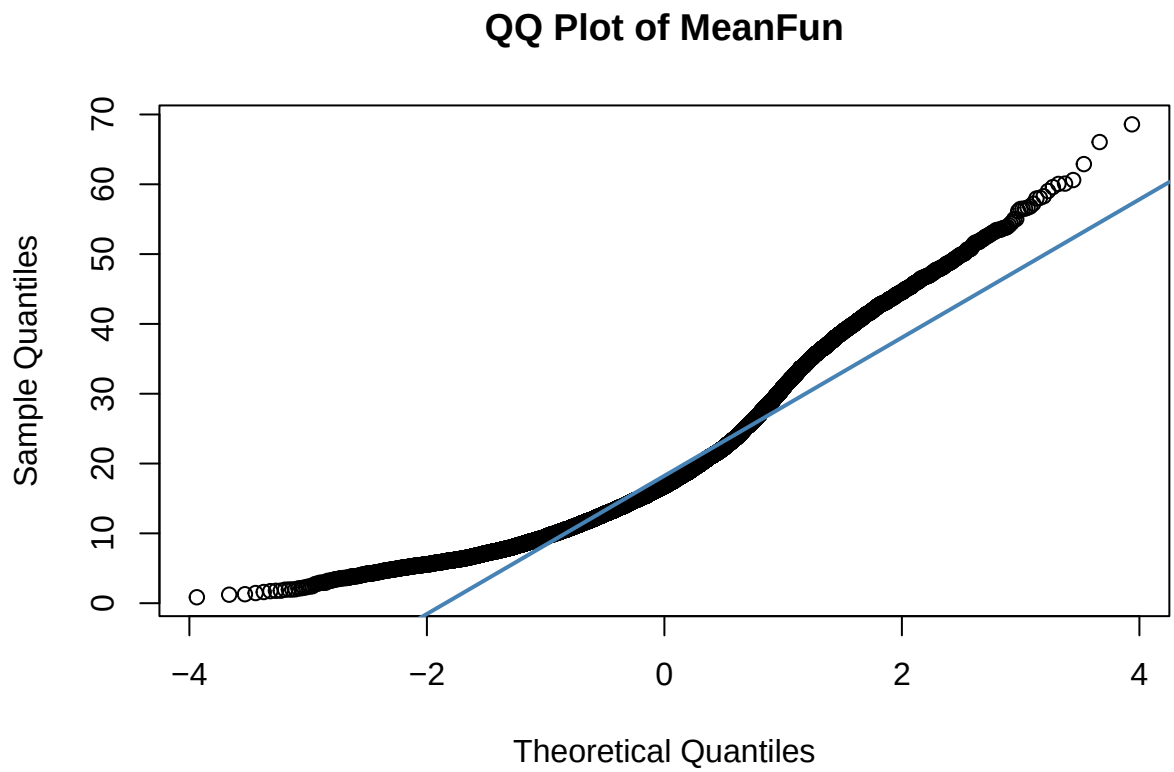
```
meanfun_mean <- mean(meanfun)
meanfun_sd <- sd(meanfun)
```

It is very obvious that the `meanfun` column is not normally distributed, we can see that from the histogram and QQ plot below.

```
hist(meanfun, probability = TRUE, main = "Histogram of MeanFun with Normal Curve", xlab = "MeanFun", bre
curve(dnorm(x, mean = meanfun_mean, sd = meanfun_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

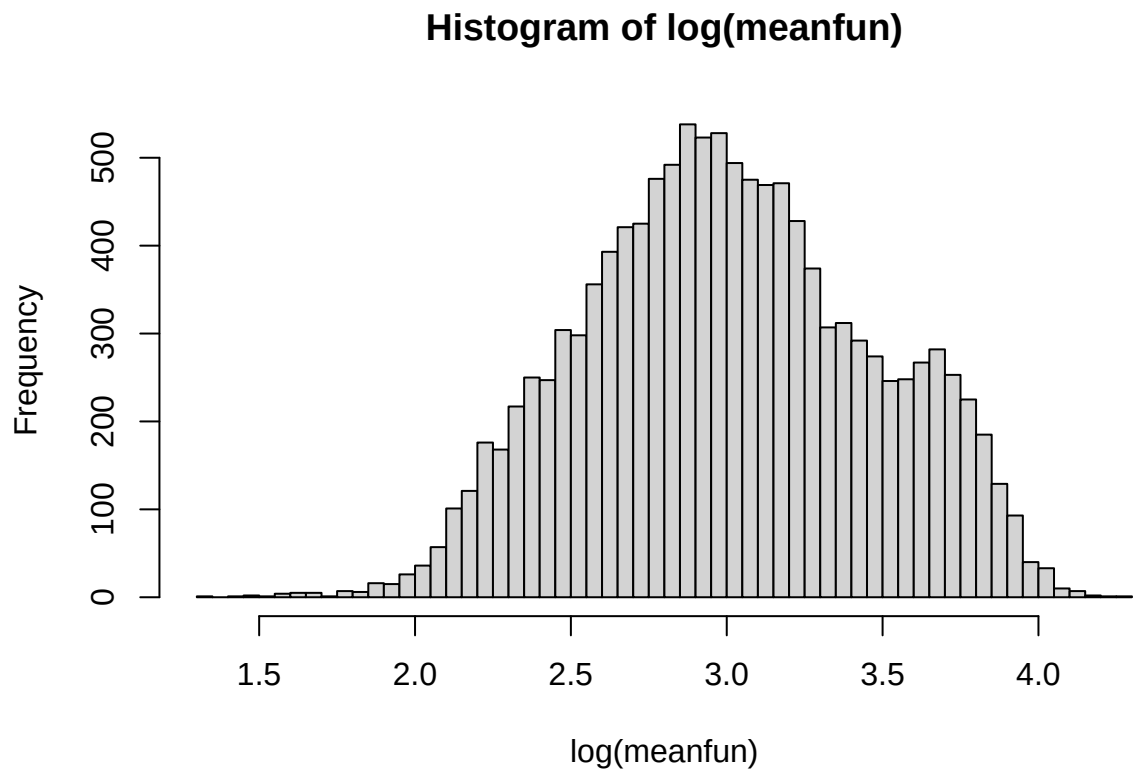



```
qqnorm(meanfun, main = "QQ Plot of MeanFun")
qqline(meanfun, col = "steelblue", lwd = 2)
```



Log-transformed `meanfun`

```
meanfun_trans <- log(meanfun + 3)
hist(meanfun_trans, breaks = 80, main = "Histogram of log(meanfun)", xlab = "log(meanfun)")
```



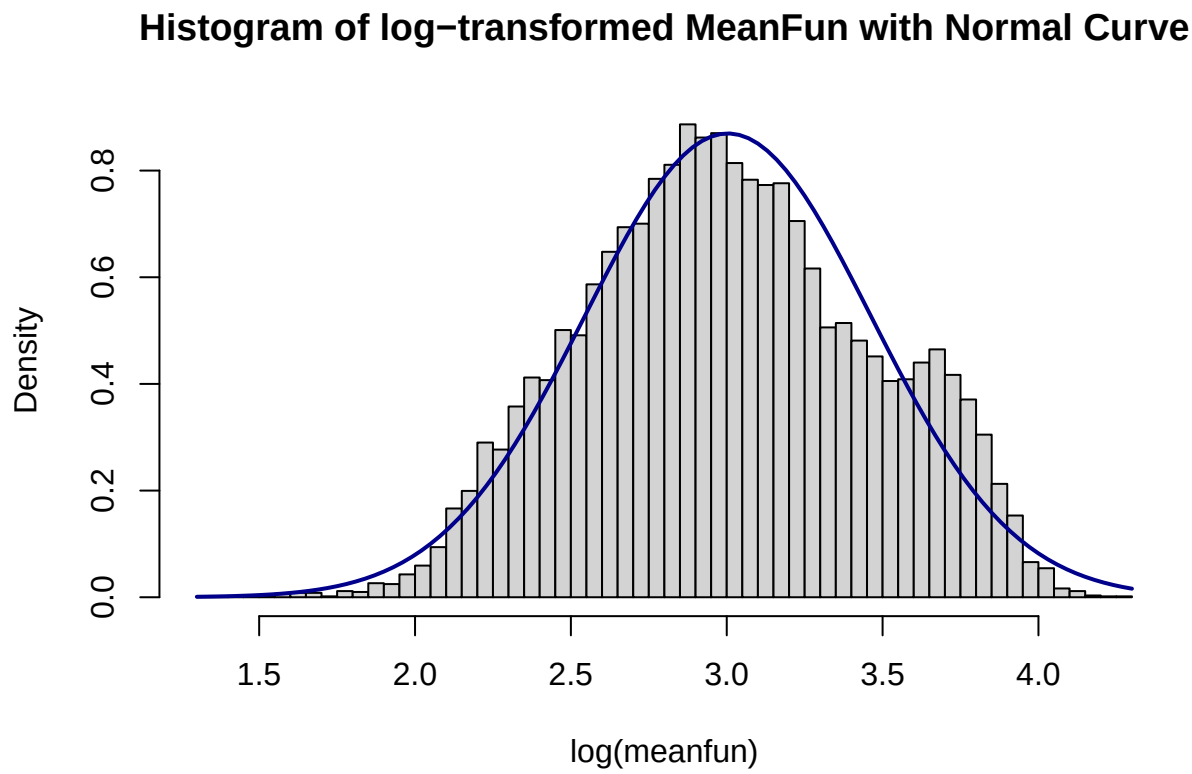

```
meanfun_trans_mean <- mean(meanfun_trans)
meanfun_trans_sd <- sd(meanfun_trans)
print(paste("Mean of log-transformed meanfun:", meanfun_trans_mean))
```

```
## [1] "Mean of log-transformed meanfun: 3.00357669730122"
```

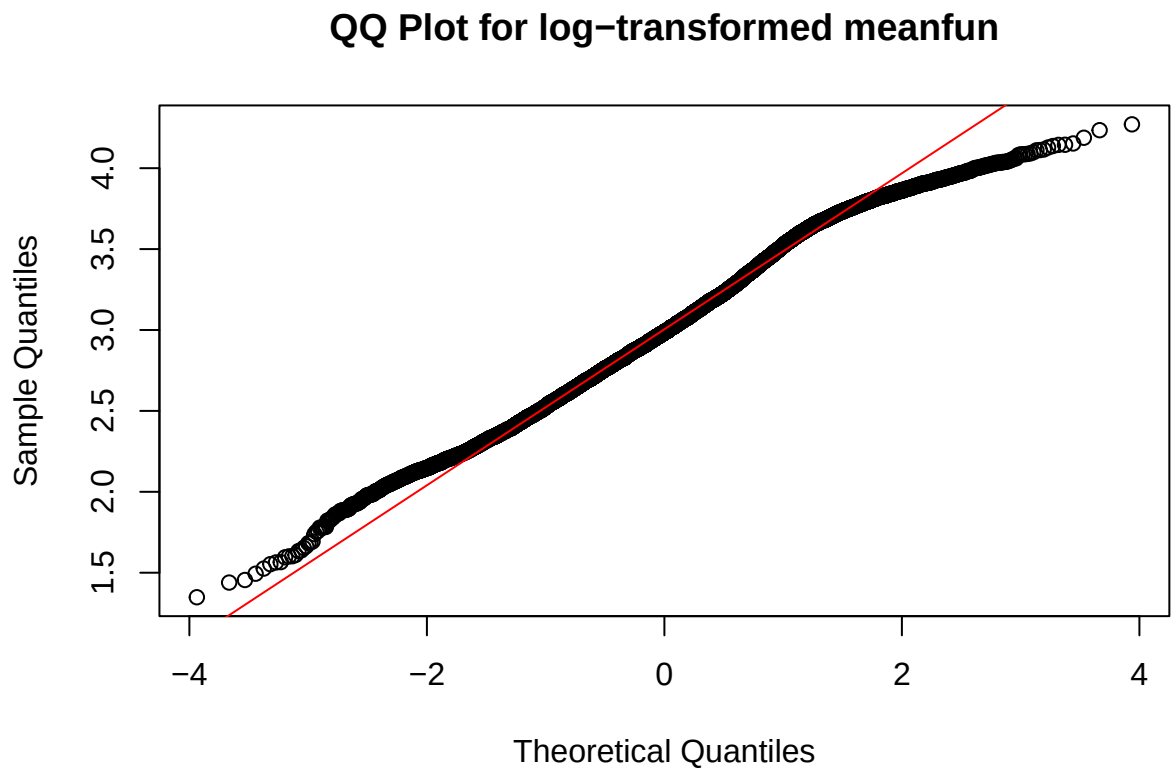
```
print(paste("Standard deviation of log-transformed meanfun:", meanfun_trans_sd))
```

```
## [1] "Standard deviation of log-transformed meanfun: 0.458730809979718"
```

```
hist(meanfun_trans, probability = TRUE, main = "Histogram of log-transformed MeanFun with Normal Curve"
curve(dnorm(x, mean = meanfun_trans_mean, sd = meanfun_trans_sd), col = "darkblue", lwd = 2, add = TRUE
```



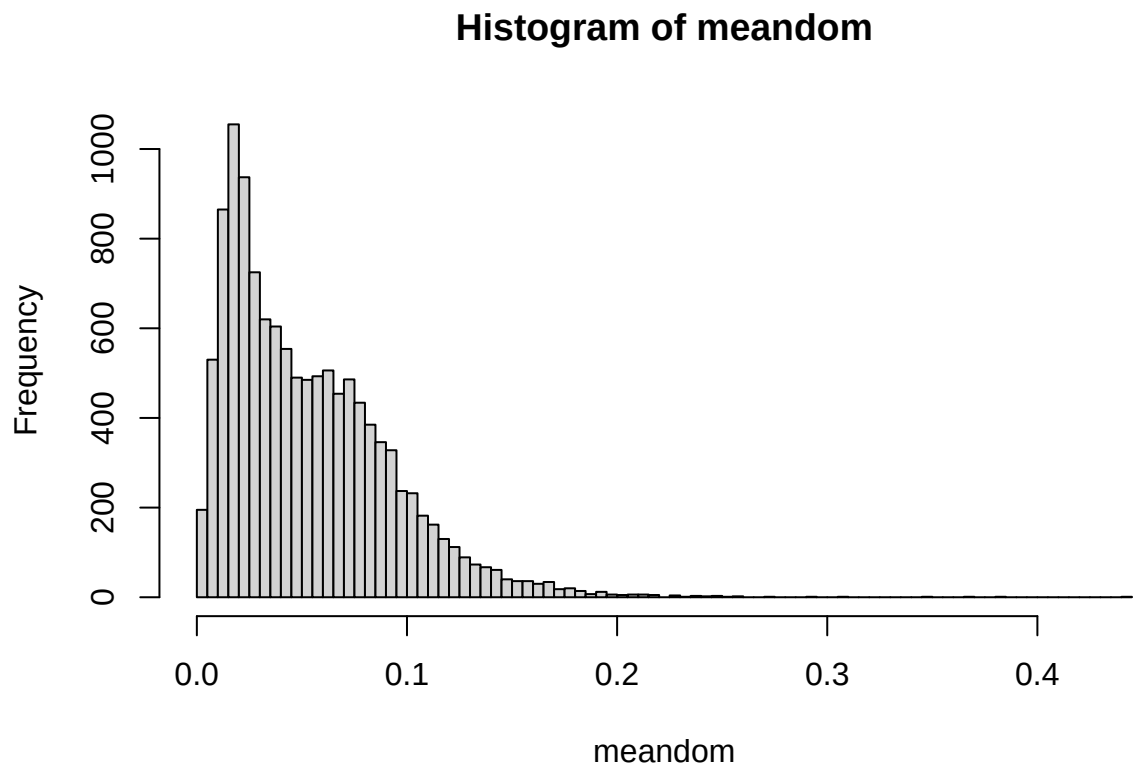

```
qqnorm(meanfun_trans, main = "QQ Plot for log-transformed meanfun")
qqline(meanfun_trans, col = "red")
```



MeanDom

Basic Analysis

```
meandom <- voice$meandom
hist(meandom, breaks = 80, main = "Histogram of meandom", xlab = "meandom")
```

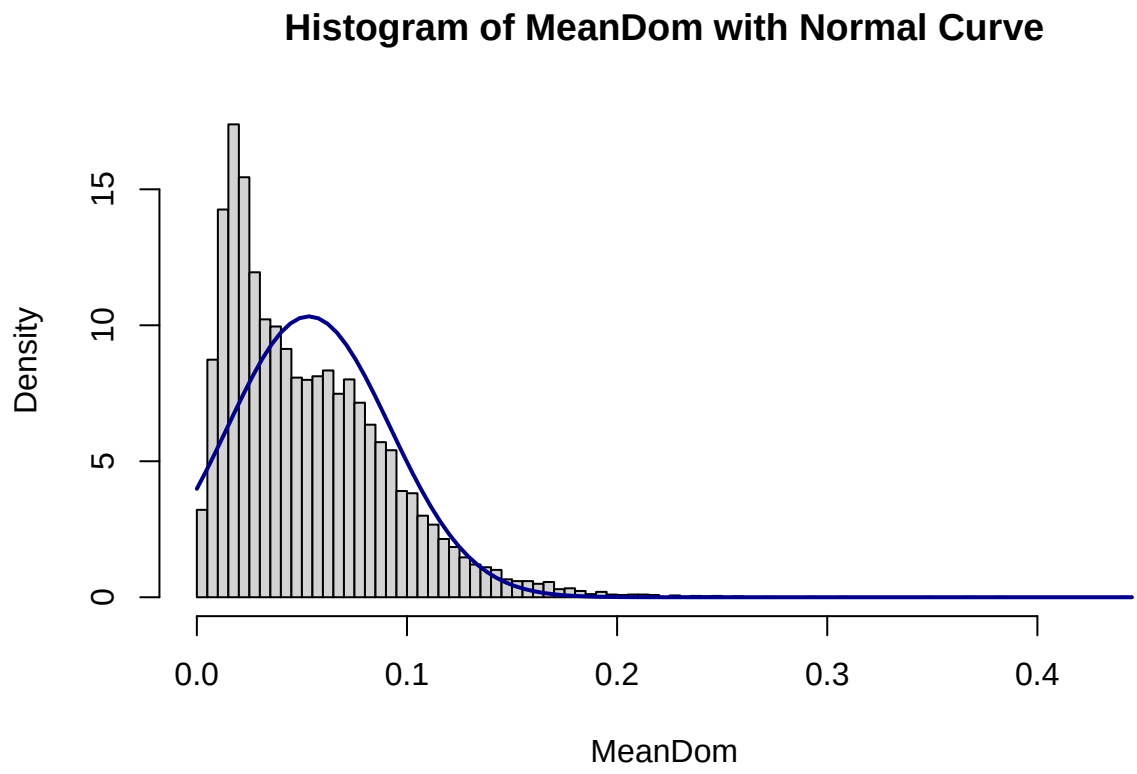


Then, we calculate the mean and standard deviation of the `meandom` column.

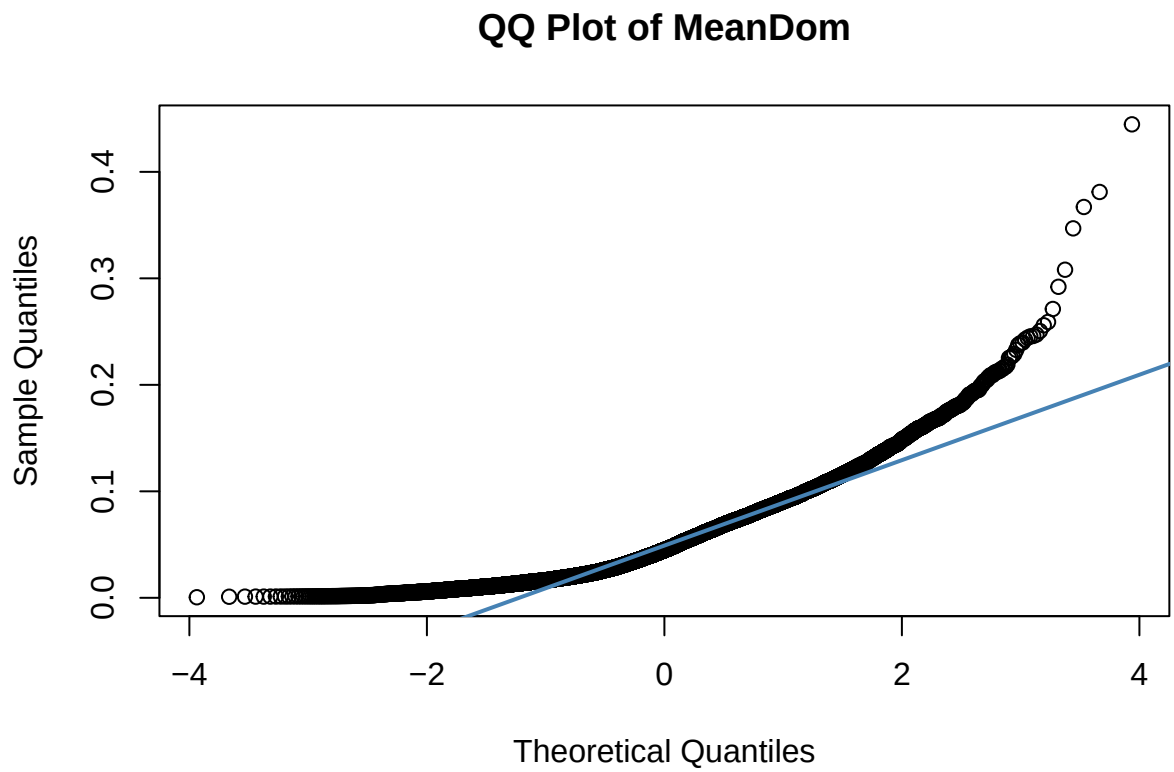
```
meandom_mean <- mean(meandom)
meandom_sd <- sd(meandom)
```

We then draw the histogram and QQ plot for the `meandom` column to know more about its distribution.

```
hist(meandom, probability = TRUE, main = "Histogram of MeanDom with Normal Curve", xlab = "MeanDom", br
curve(dnorm(x, mean = meandom_mean, sd = meandom_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n"
```



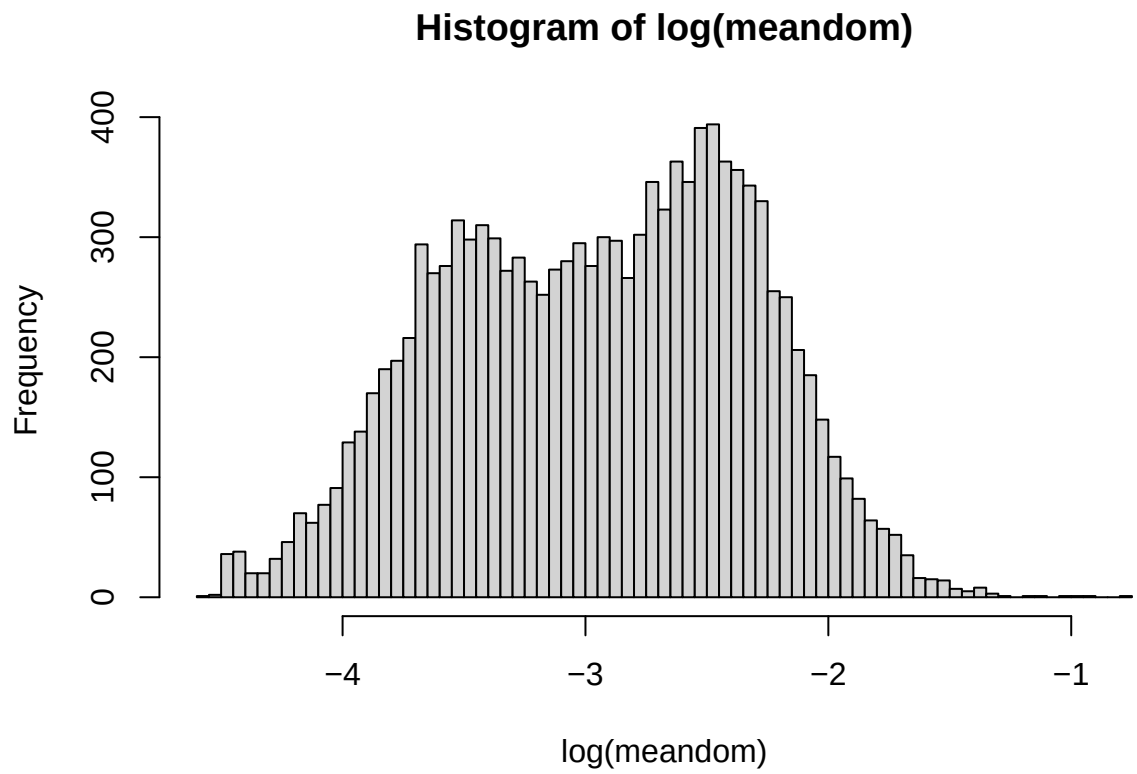

```
qqnorm(meandom, main = "QQ Plot of MeanDom")
qqline(meandom, col = "steelblue", lwd = 2)
```



log-transformed `meandom`

After multiple attempts to transform the `meandom` column, we found that the log-transformation with $f(x) = \log(x + 0.01)$ is the best transformation for this column.

```
meandom_trans <- log(meandom + 0.01)
hist(meandom_trans, breaks = as.integer(sqrt(length(meandom_trans))), main = "Histogram of log(meandom)
```



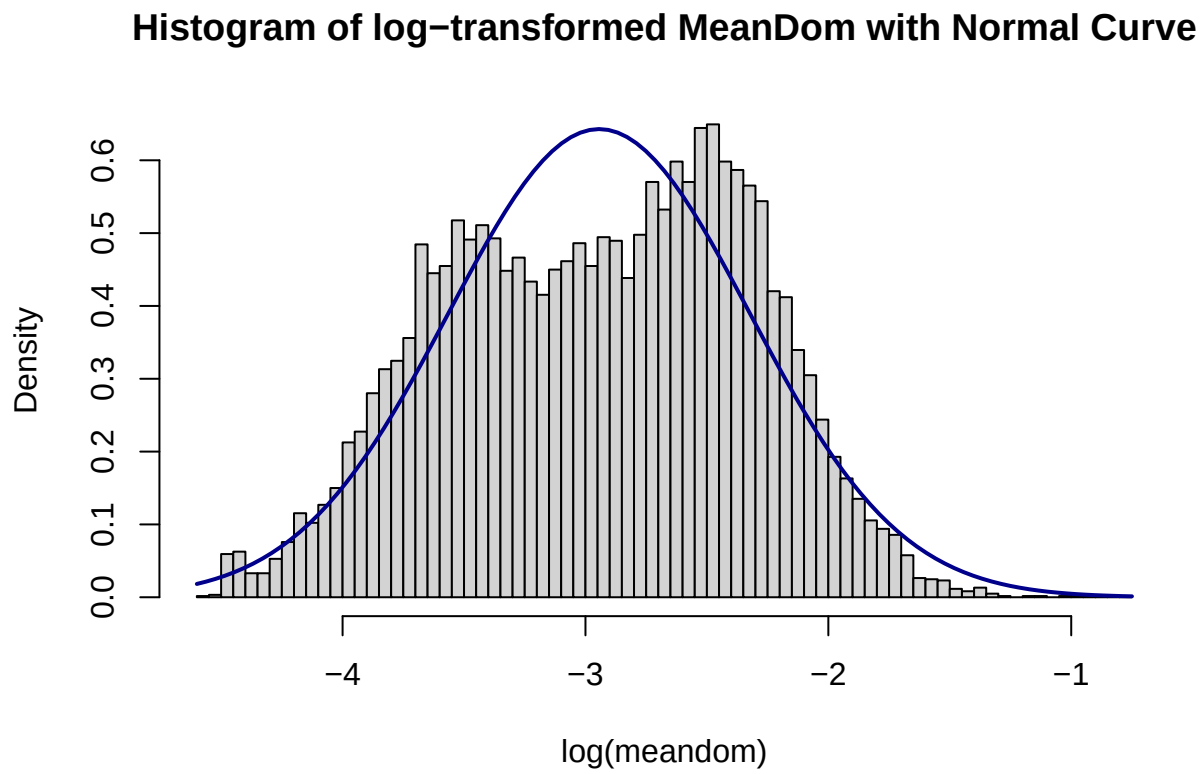

```r
meandom_trans_mean <- mean(meandom_trans)
meandom_trans_sd <- sd(meandom_trans)
print(paste("Mean of log-transformed meandom:", meandom_trans_mean))
```

```
## [1] "Mean of log-transformed meandom: -2.94383343200092"
```

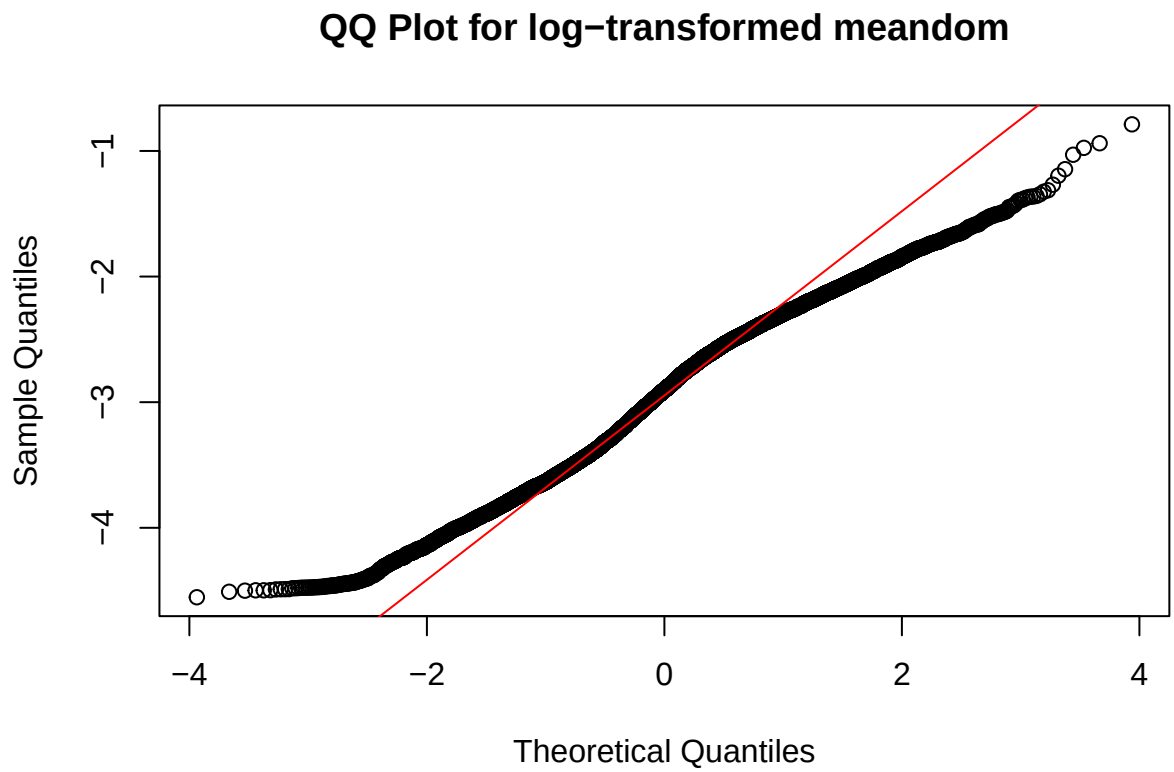
```r
print(paste("Standard deviation of log-transformed meandom:", meandom_trans_sd))
```

```
## [1] "Standard deviation of log-transformed meandom: 0.620535805851869"
```

```r
hist(meandom_trans, probability = TRUE, main = "Histogram of log-transformed MeanDom with Normal Curve"
curve(dnorm(x, mean = meandom_trans_mean, sd = meandom_trans_sd), col = "darkblue", lwd = 2, add = TRUE
```



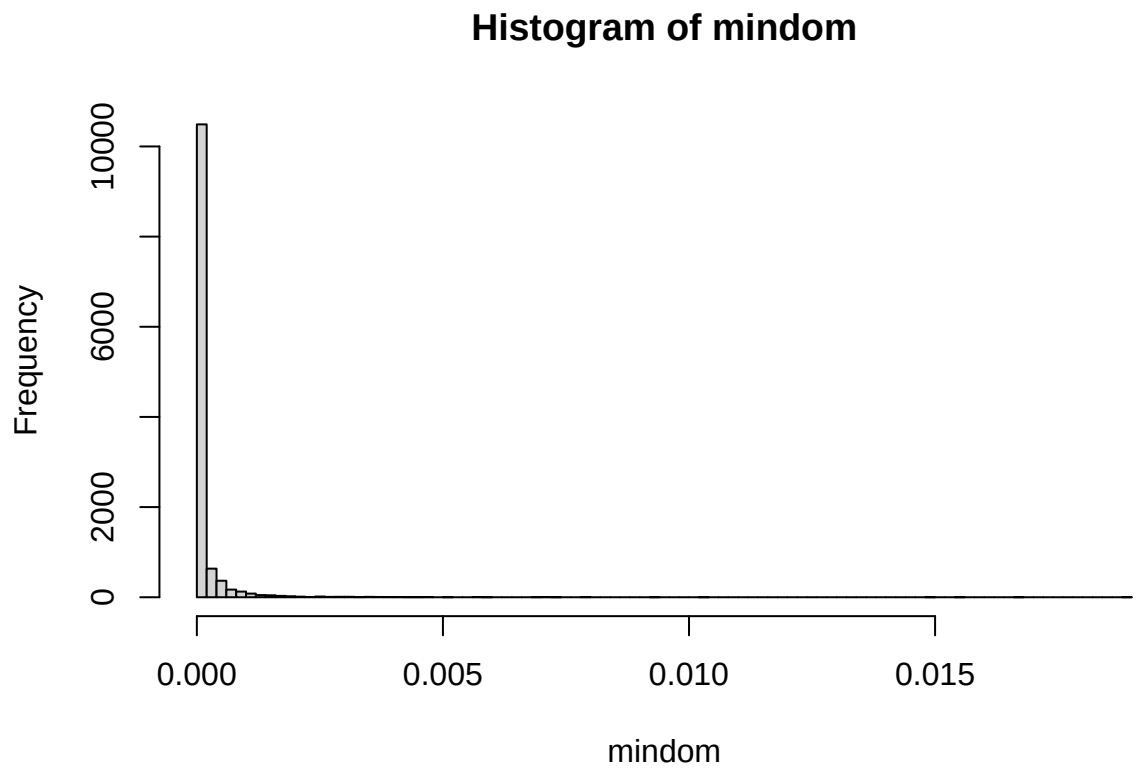

```
qqnorm(meandom_trans, main = "QQ Plot for log-transformed meandom")
qqline(meandom_trans, col = "red")
```



MinDom

Basic Analysis

```
mindom <- voice$mindom
hist(mindom, breaks = 80, main = "Histogram of mindom", xlab = "mindom")
```

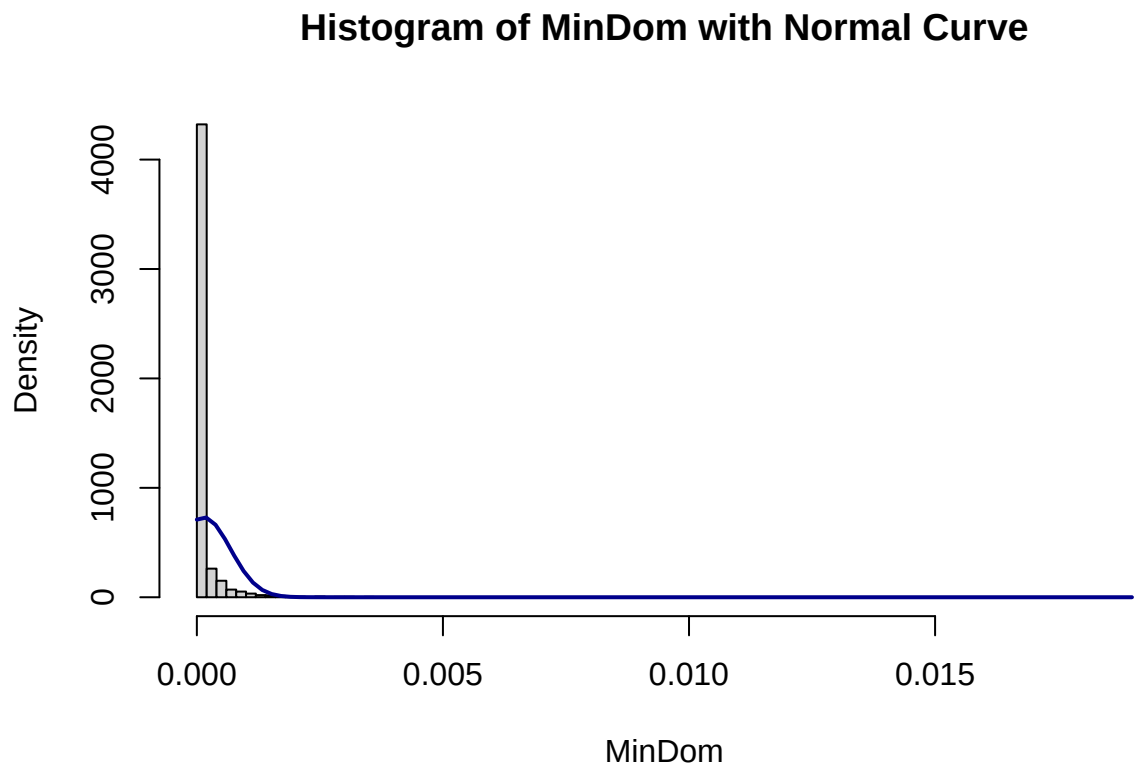


Then, we calculate the mean and standard deviation of the `mindom` column.

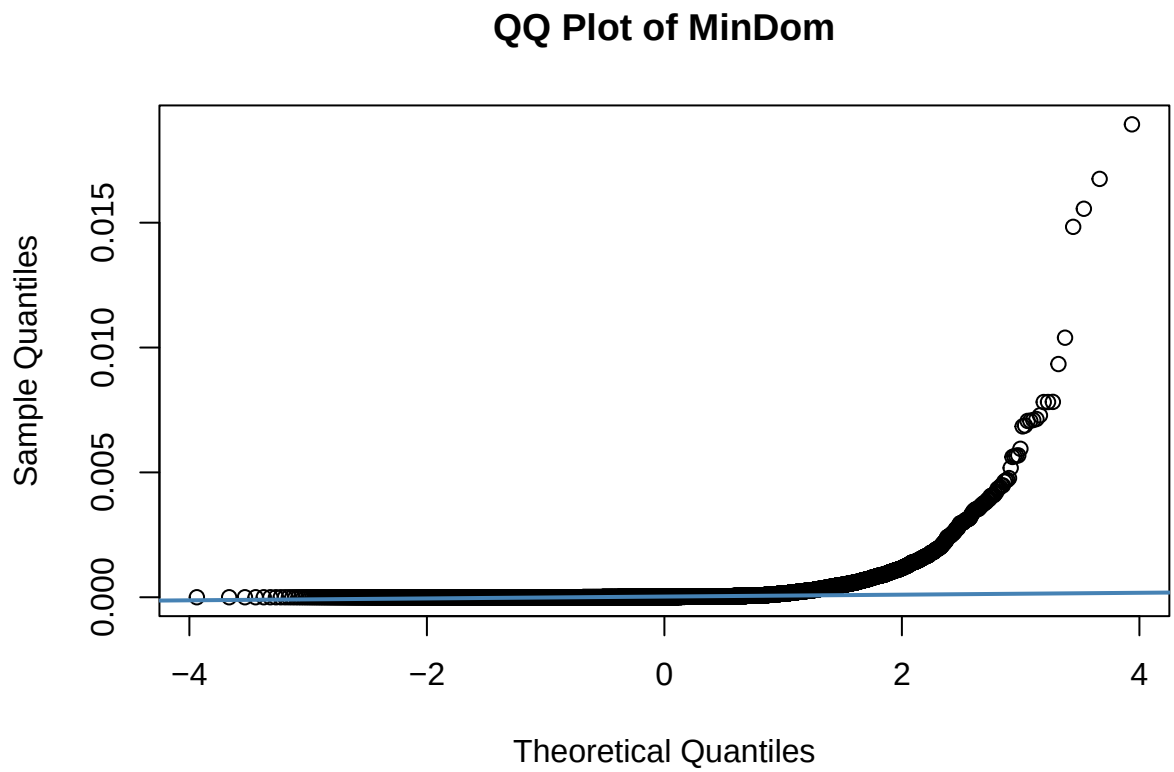
```
mindom_mean <- mean(mindom)
mindom_sd <- sd(mindom)
```

We then draw the histogram and QQ plot for the `mindom` column to know more about its distribution.

```
hist(mindom, probability = TRUE, main = "Histogram of MinDom with Normal Curve", xlab = "MinDom", breaks
curve(dnorm(x, mean = mindom_mean, sd = mindom_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

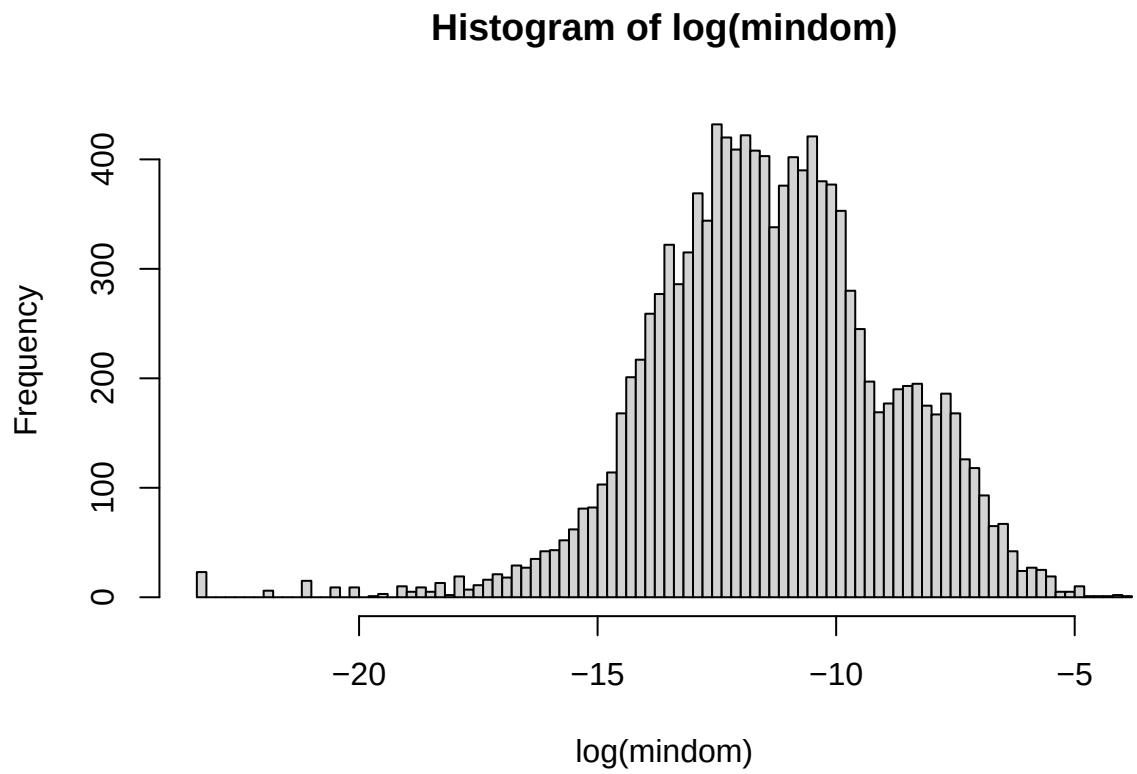



```
qqnorm(mindom, main = "QQ Plot of MinDom")
qqline(mindom, col = "steelblue", lwd = 2)
```



log-transformed `mindom`

```
mindom_trans <- log(mindom)
hist(mindom_trans, breaks = as.integer(sqrt(length(mindom_trans))), main = "Histogram of log(mindom)", i
```



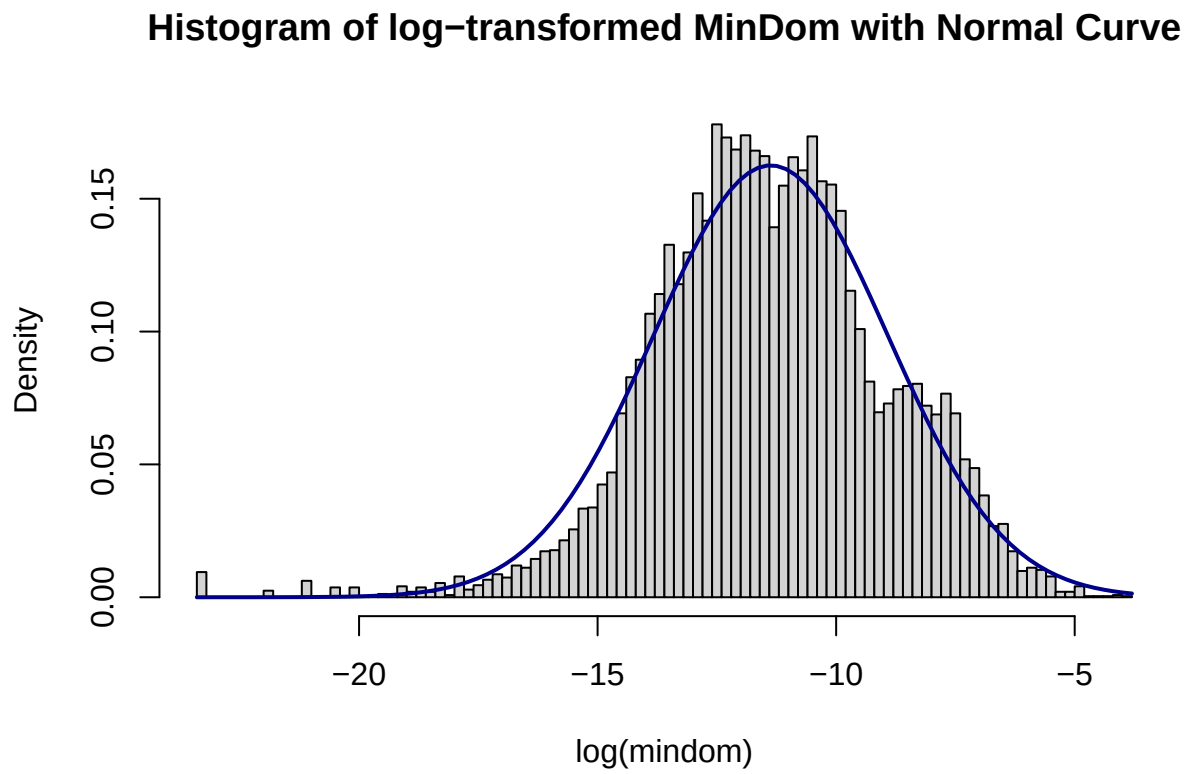

```
mindom_trans_mean <- mean(mindom_trans)
mindom_trans_sd <- sd(mindom_trans)
print(paste("Mean of log-transformed mindom:", mindom_trans_mean))
```

```
## [1] "Mean of log-transformed mindom: -11.3730692247243"
```

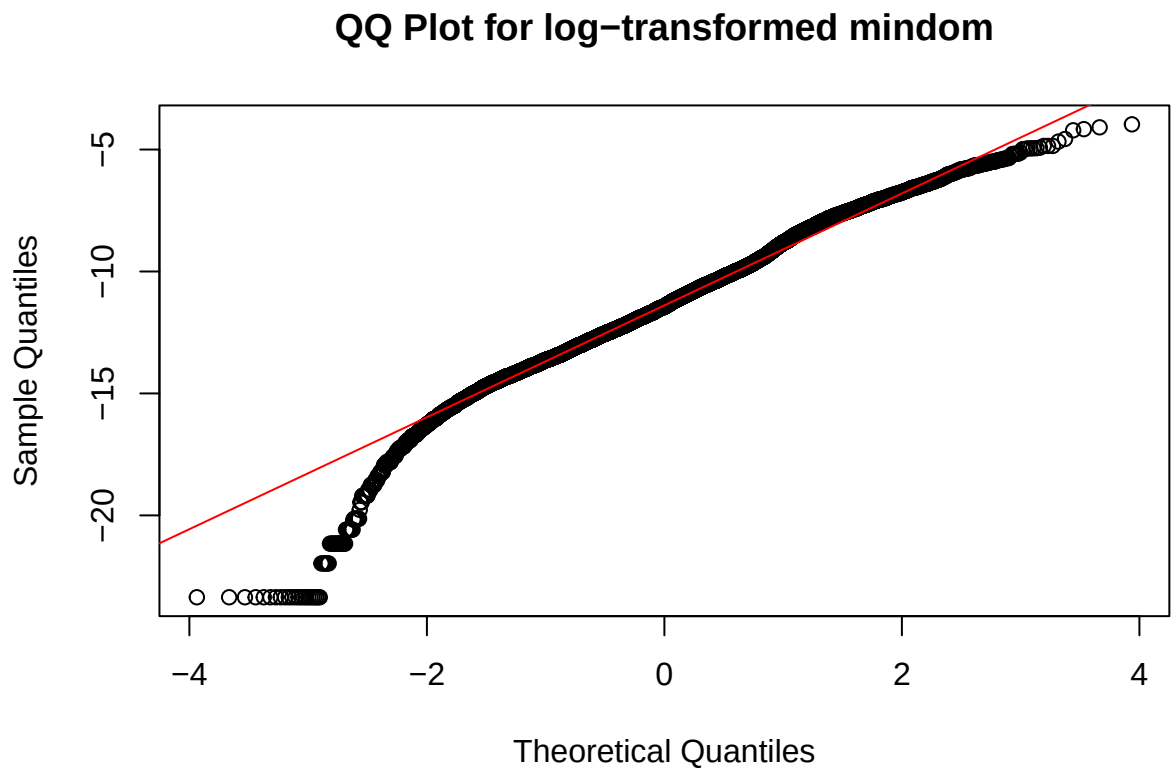
```
print(paste("Standard deviation of log-transformed mindom:", mindom_trans_sd))
```

```
## [1] "Standard deviation of log-transformed mindom: 2.45400913865441"
```

```
hist(mindom_trans, probability = TRUE, main = "Histogram of log-transformed MinDom with Normal Curve", 
curve(dnorm(x, mean = mindom_trans_mean, sd = mindom_trans_sd), col = "darkblue", lwd = 2, add = TRUE, 
```



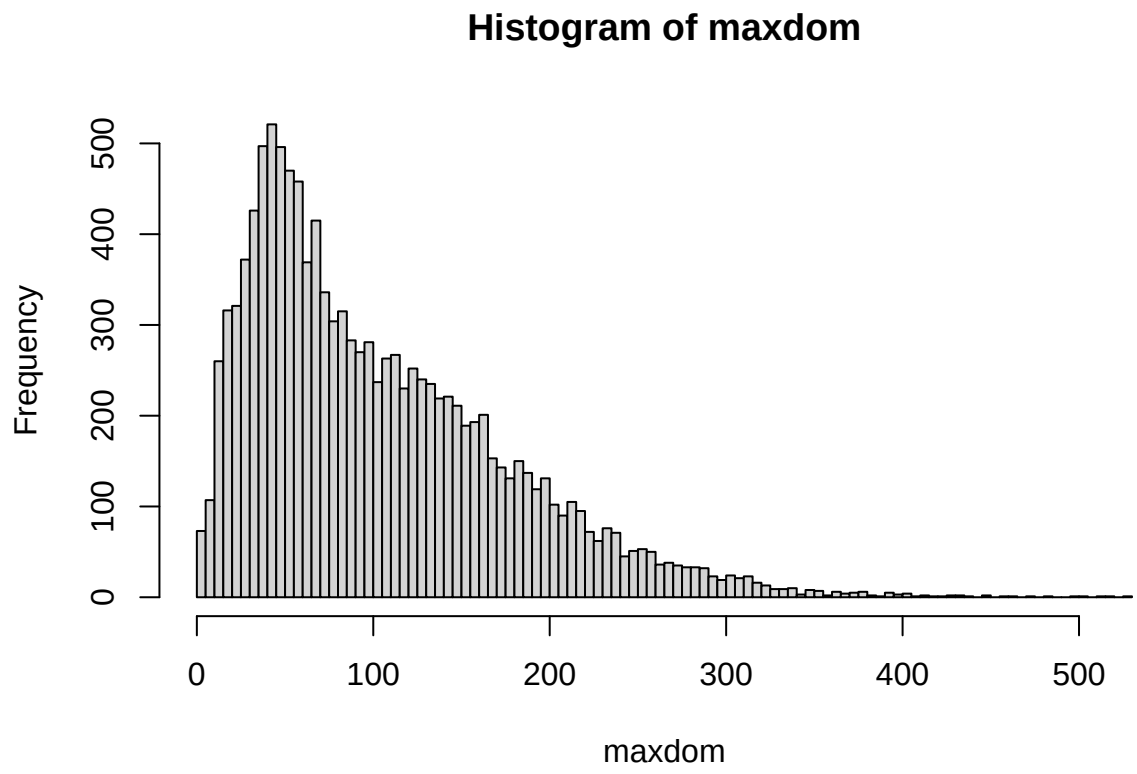

```
qqnorm(mindom_trans, main = "QQ Plot for log-transformed mindom")
qqline(mindom_trans, col = "red")
```



MaxDom

Basic Analysis

```
maxdom <- voice$maxdom
hist(maxdom, breaks = 80, main = "Histogram of maxdom", xlab = "maxdom")
```

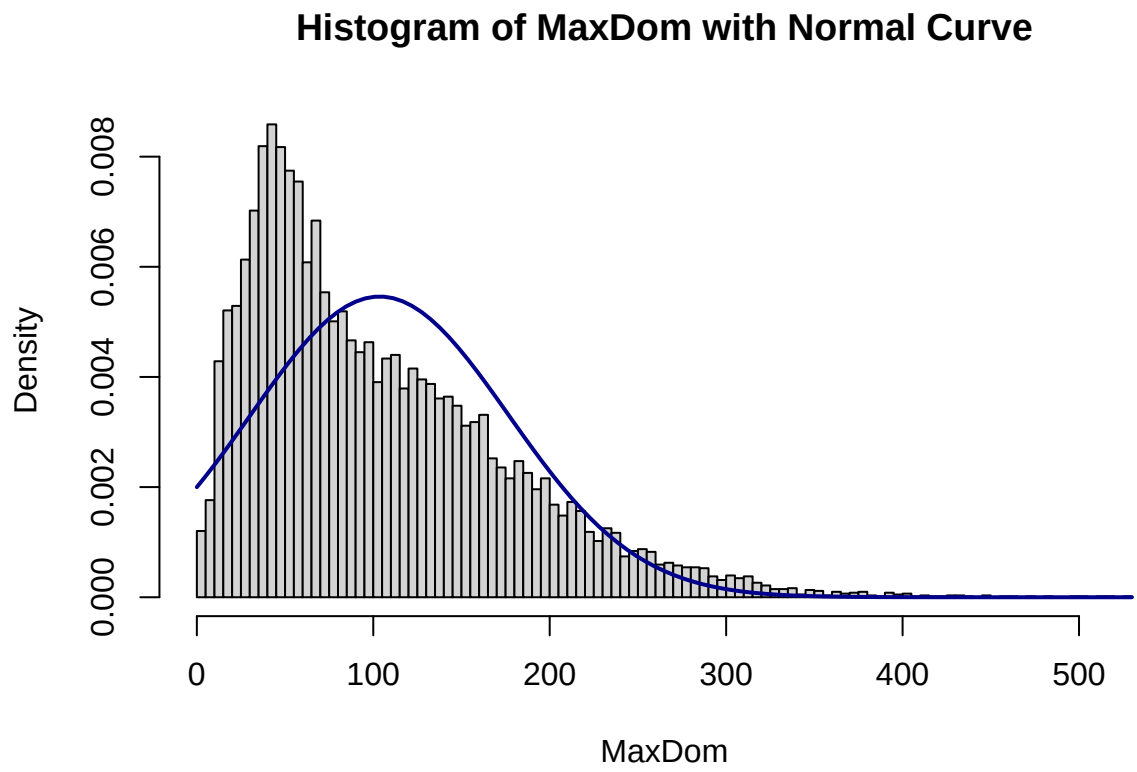


Then, we calculate the mean and standard deviation of the `maxdom` column.

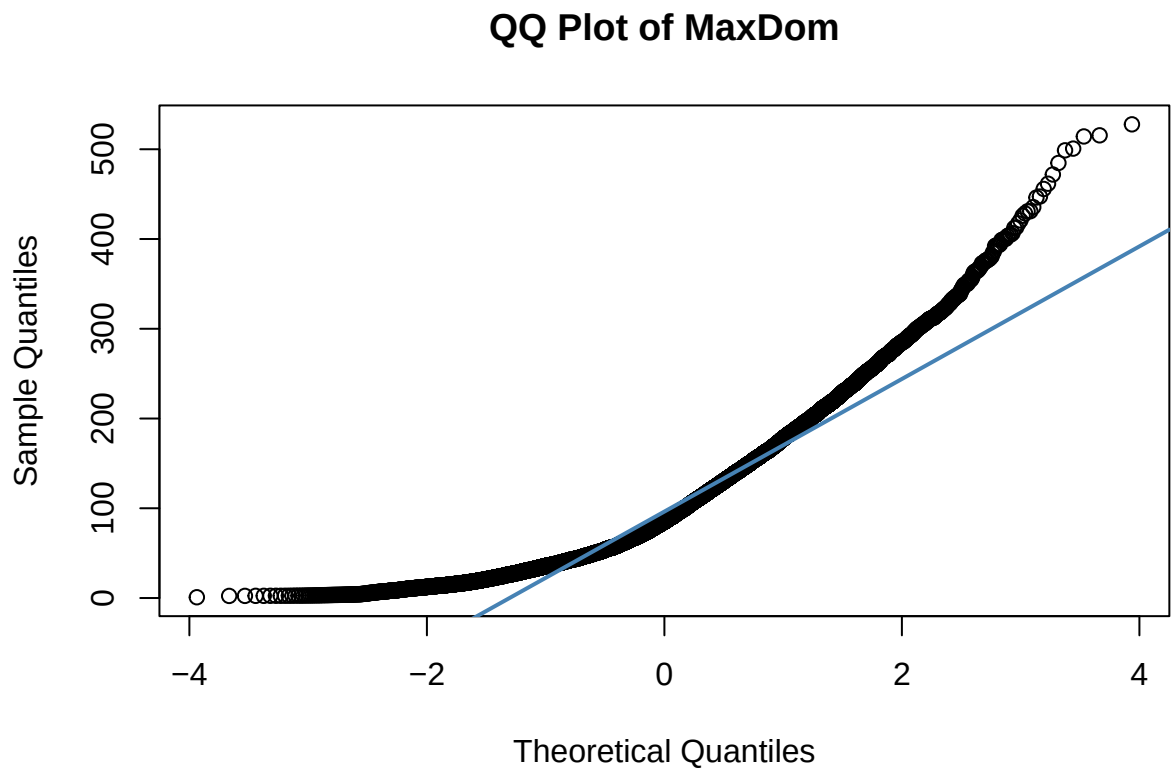
```
maxdom_mean <- mean(maxdom)
maxdom_sd <- sd(maxdom)
```

We then draw the histogram and QQ plot for the `maxdom` column to know more about its distribution.

```
hist(maxdom, probability = TRUE, main = "Histogram of MaxDom with Normal Curve", xlab = "MaxDom", breaks
curve(dnorm(x, mean = maxdom_mean, sd = maxdom_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n")
```

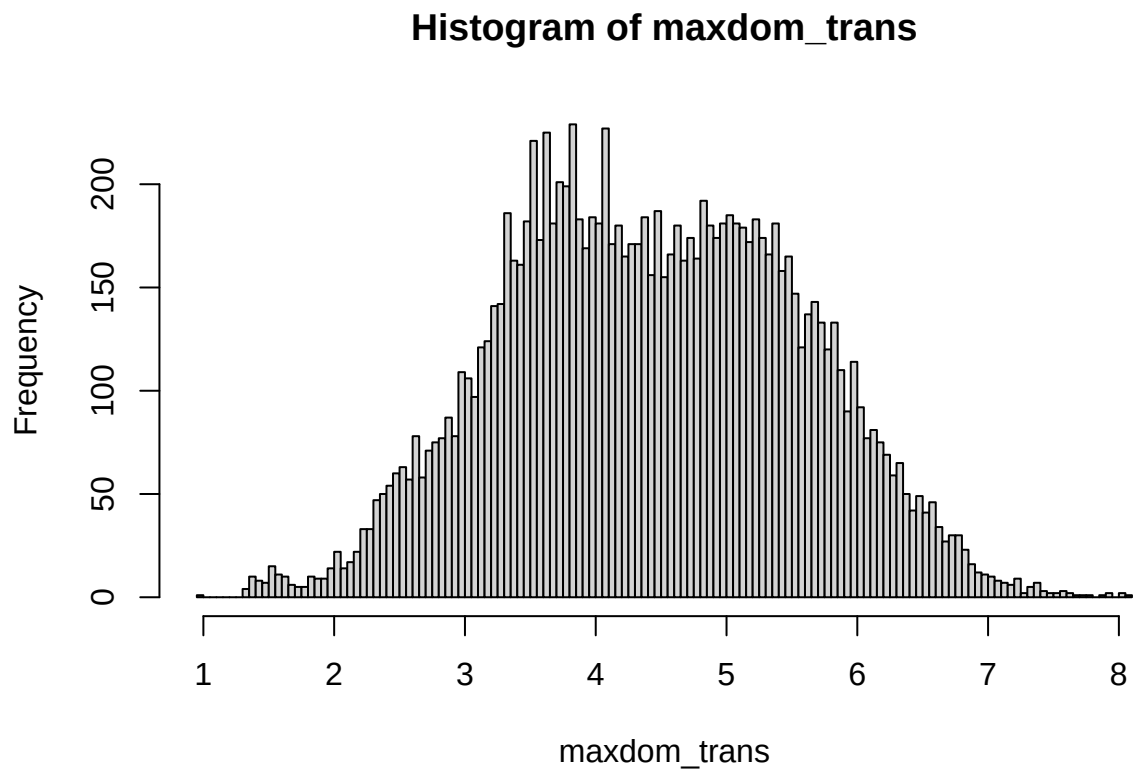



```
qqnorm(maxdom, main = "QQ Plot of MaxDom")
qqline(maxdom, col = "steelblue", lwd = 2)
```

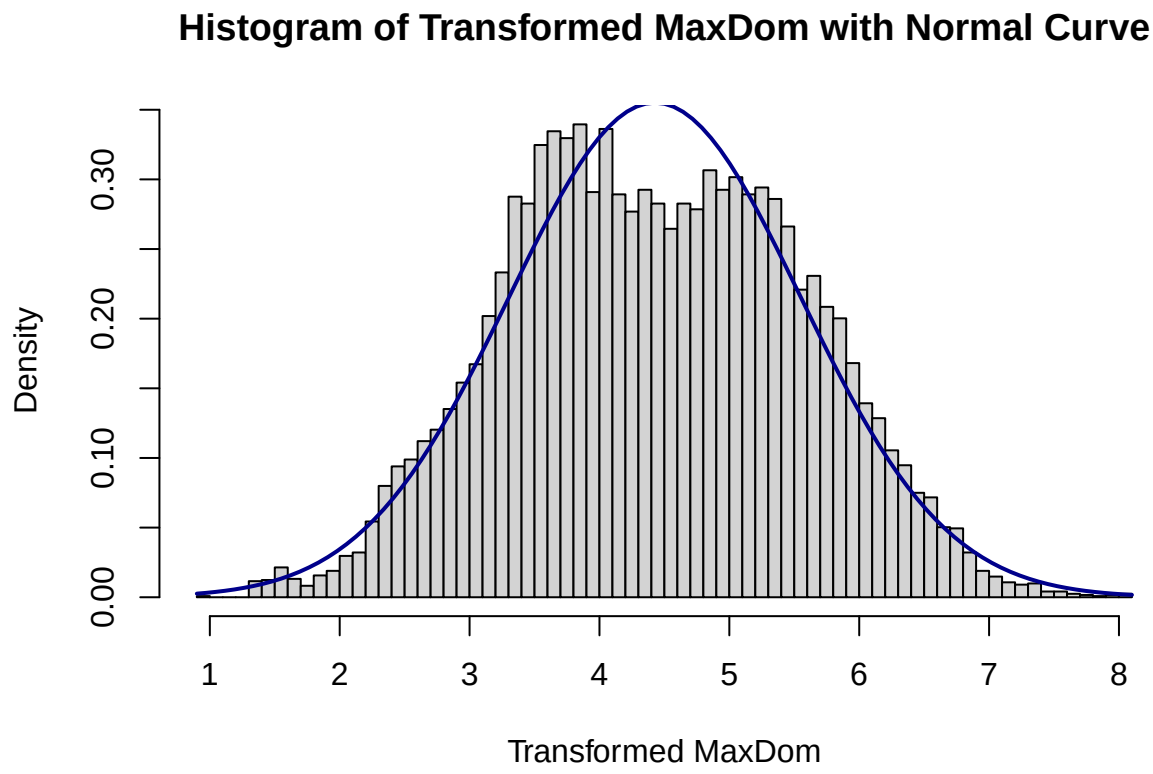


Transformed `maxdom`

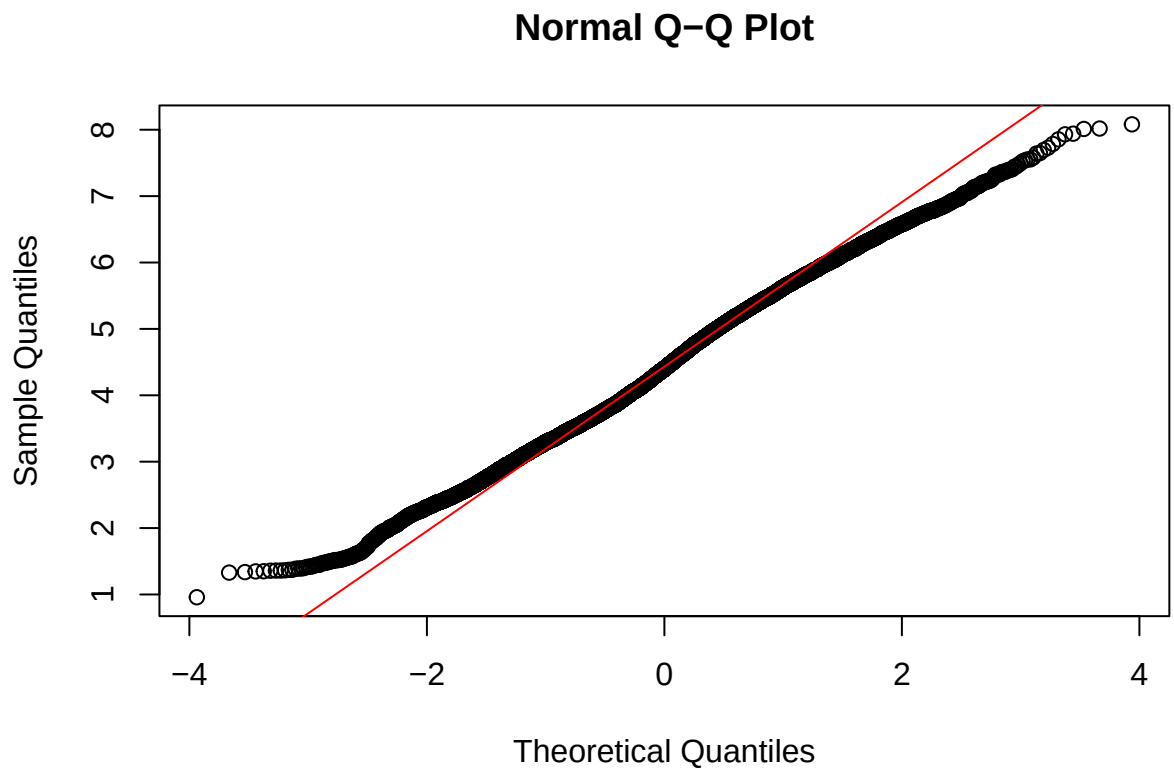
```
maxdom_trans <- maxdom^(1 / 3)
hist(maxdom_trans, breaks = as.integer(sqrt(length(maxdom_trans))), main = "Histogram of maxdom_trans",
```




```
hist(maxdom_trans, probability = TRUE, main = "Histogram of Transformed MaxDom with Normal Curve", xlab
curve(dnorm(x, mean = mean(maxdom_trans), sd = sd(maxdom_trans)), col = "darkblue", lwd = 2, add = TRUE
```



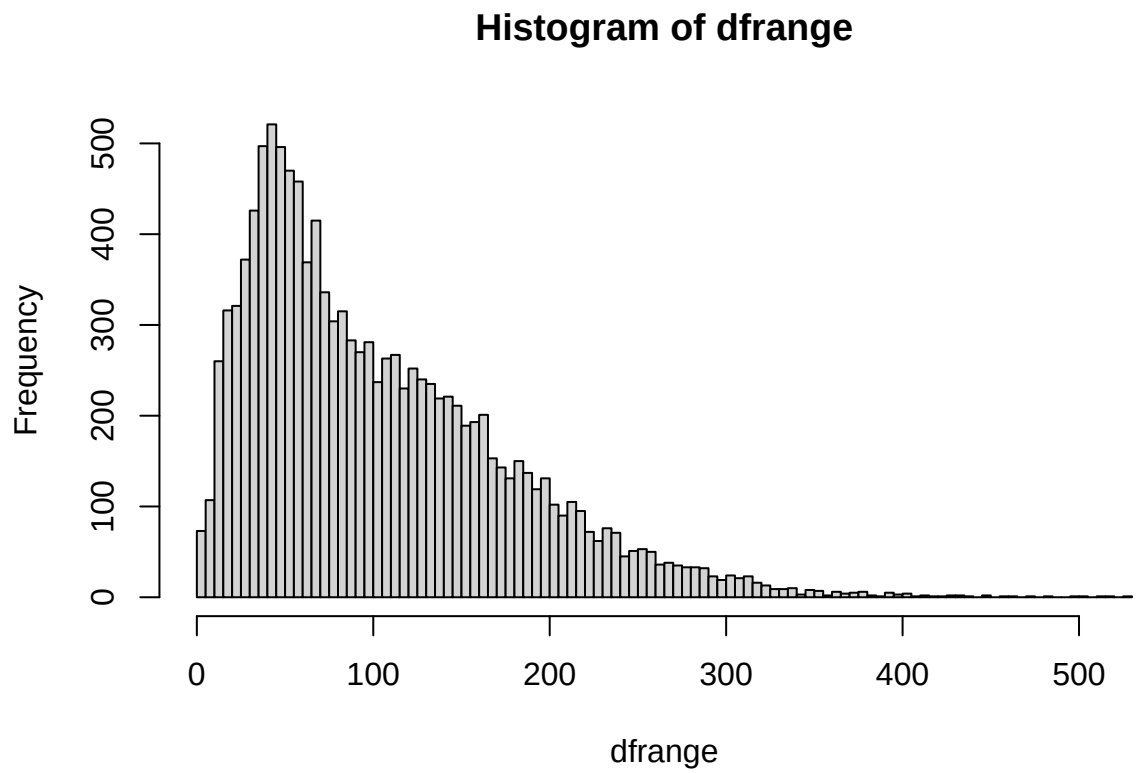

```
qqnorm(maxdom_trans)
qqline(maxdom_trans, col = "red")
```



dfrange

Basic Analysis

```
dfrange <- voice$dfrange
hist(dfrange, breaks = 80, main = "Histogram of dfrange", xlab = "dfrange")
```

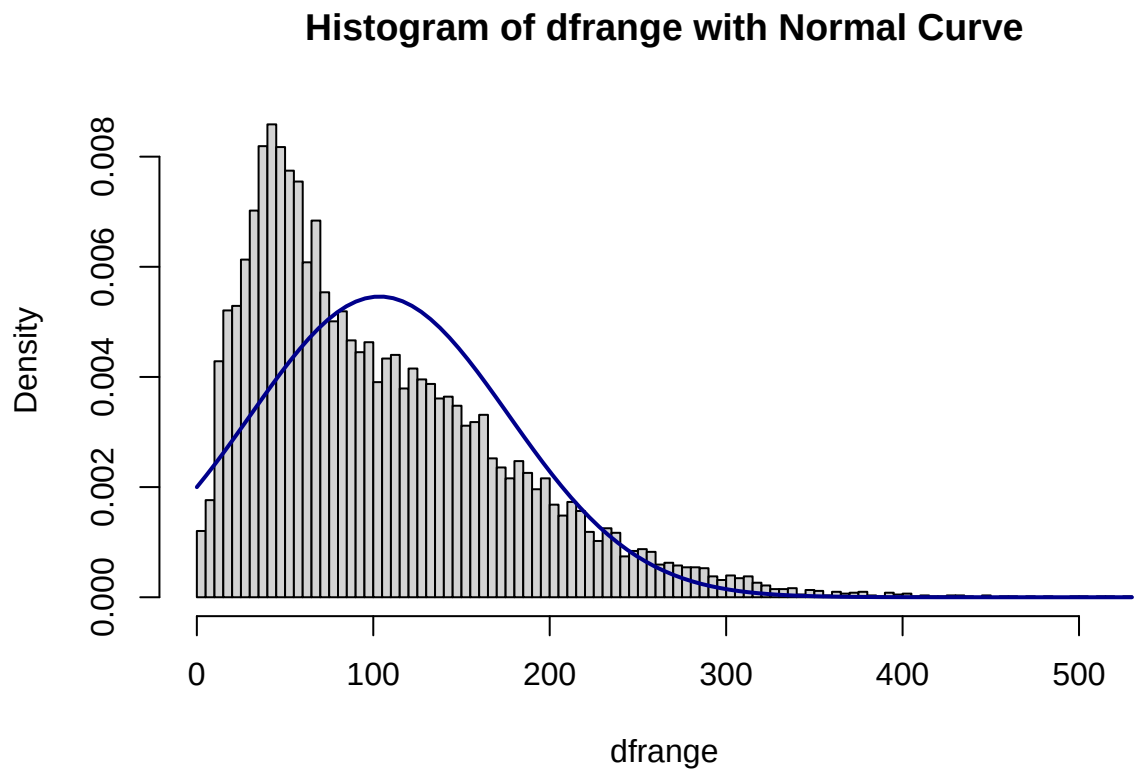


Then, we calculate the mean and standard deviation of the `dfrange` column.

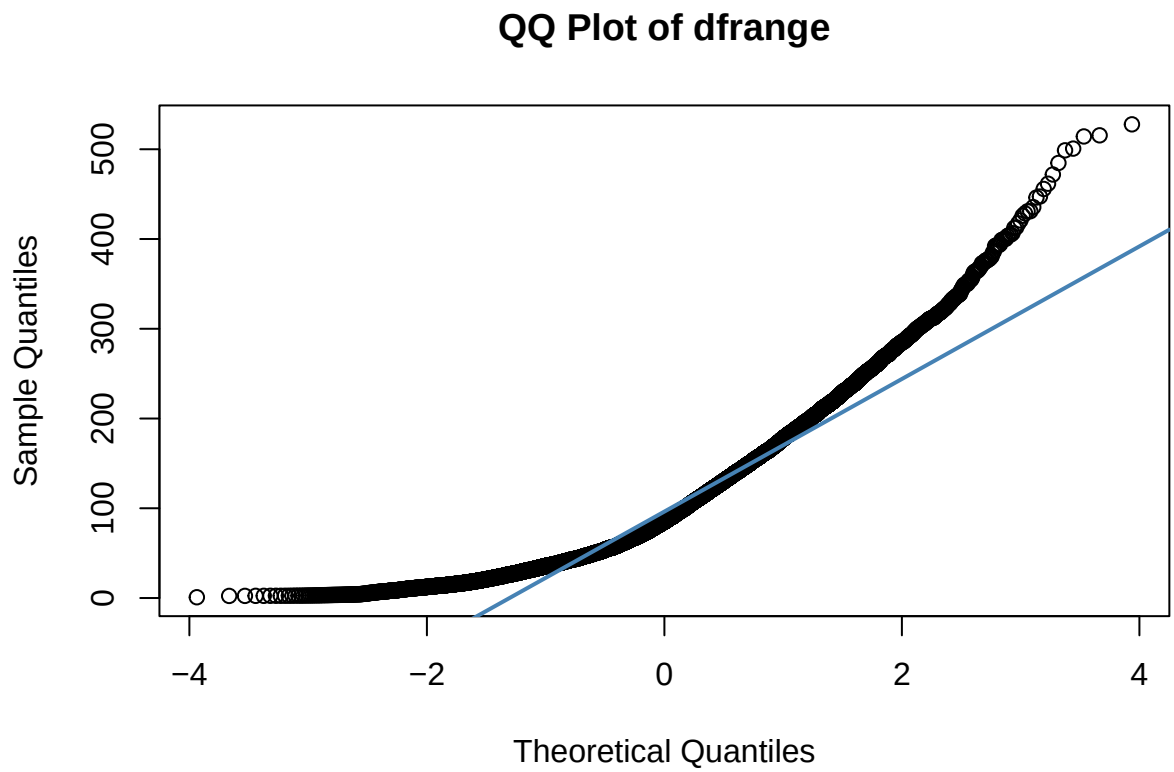
```
dfrange_mean <- mean(dfrange)
dfrange_sd <- sd(dfrange)
```

We then draw the histogram and QQ plot for the `dfrange` column to know more about its distribution.

```
hist(dfrange, probability = TRUE, main = "Histogram of dfrange with Normal Curve", xlab = "dfrange", br
curve(dnorm(x, mean = dfrange_mean, sd = dfrange_sd), col = "darkblue", lwd = 2, add = TRUE, yaxt = "n"
```

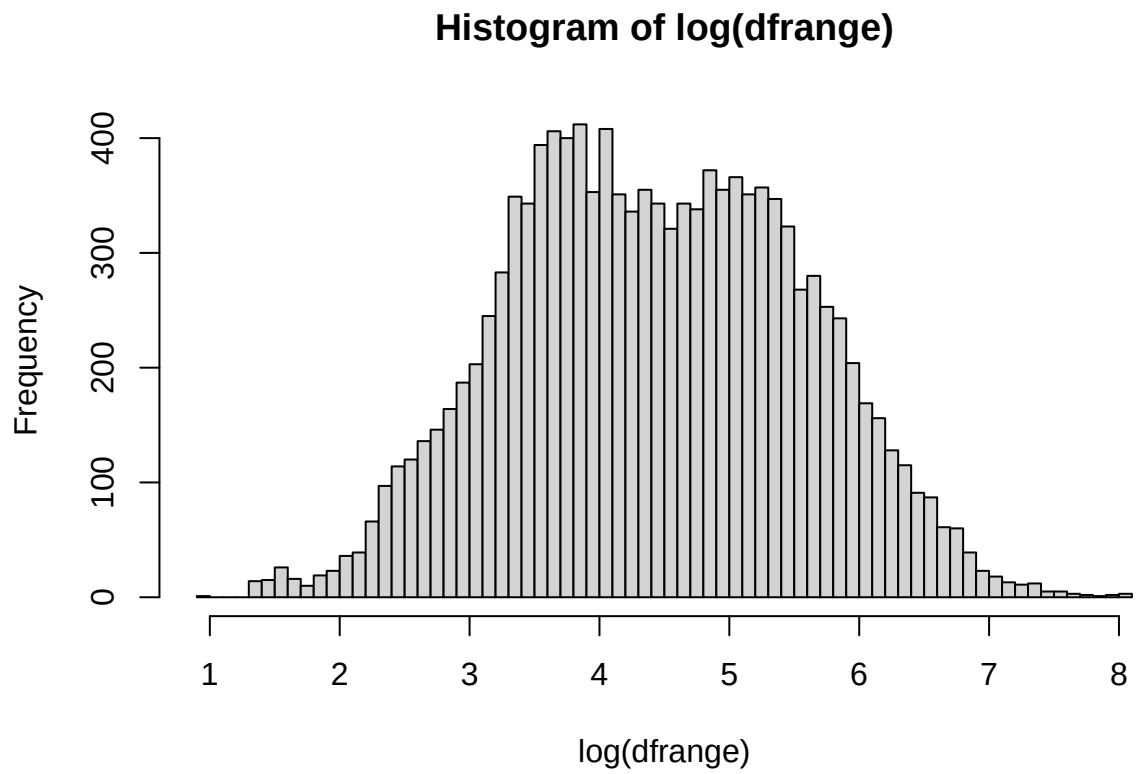



```r
qqnorm(dfrange, main = "QQ Plot of dfrange")
qqline(dfrange, col = "steelblue", lwd = 2)
```



log-transformed `dfrange`

```
dfrange_trans <- dfrange^(1 / 3)
hist(dfrange_trans, breaks = 80, main = "Histogram of log(dfrange)", xlab = "log(dfrange)")
```



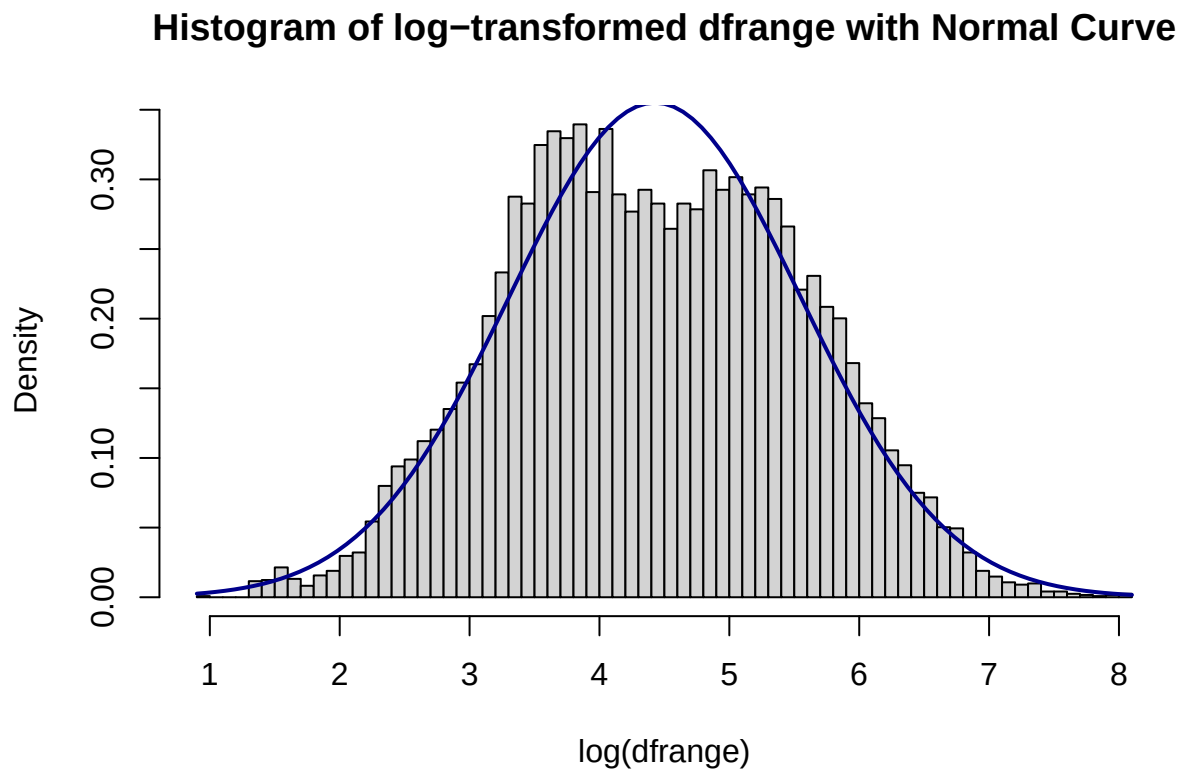

```
dfrange_trans_mean <- mean(dfrange_trans)
dfrange_trans_sd <- sd(dfrange_trans)
print(paste("Mean of log-transformed dfrange:", dfrange_trans_mean))
```

```
## [1] "Mean of log-transformed dfrange: 4.42689633559556"
```

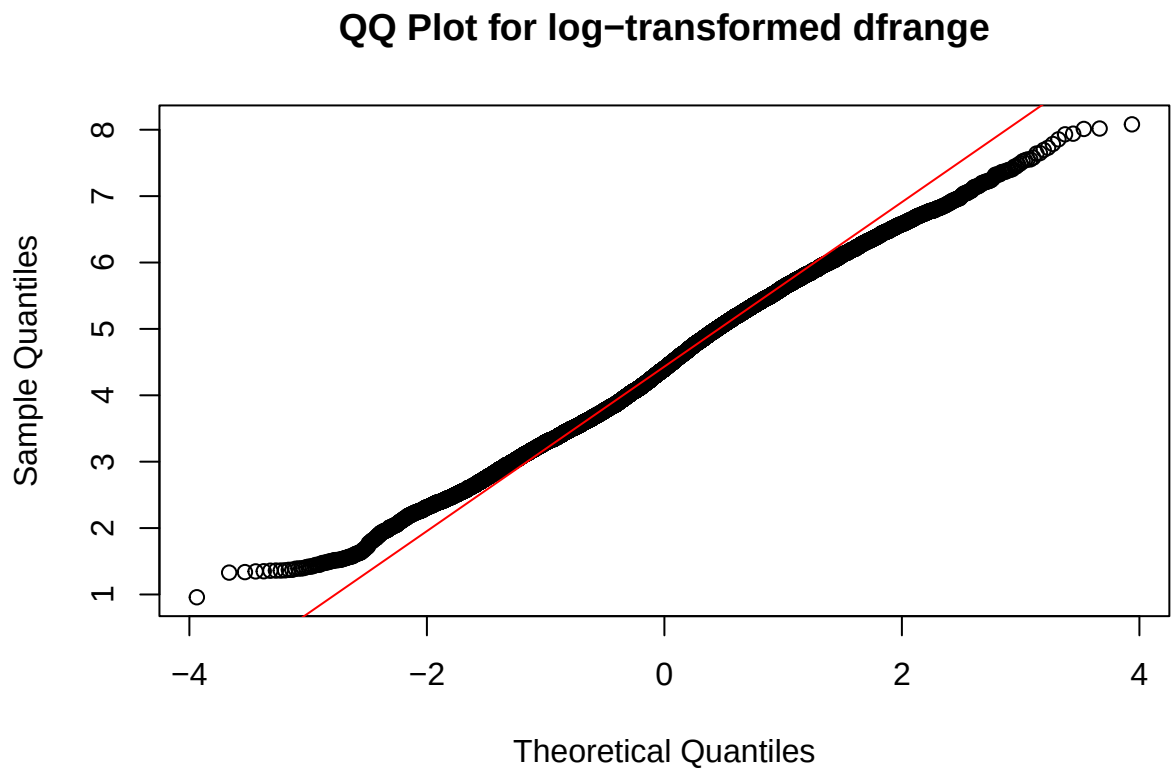
```
print(paste("Standard deviation of log-transformed dfrange:", dfrange_trans_sd))
```

```
## [1] "Standard deviation of log-transformed dfrange: 1.12357313428108"
```

```
hist(dfrange_trans, probability = TRUE, main = "Histogram of log-transformed dfrange with Normal Curve"
curve(dnorm(x, mean = dfrange_trans_mean, sd = dfrange_trans_sd), col = "darkblue", lwd = 2, add = TRUE
```

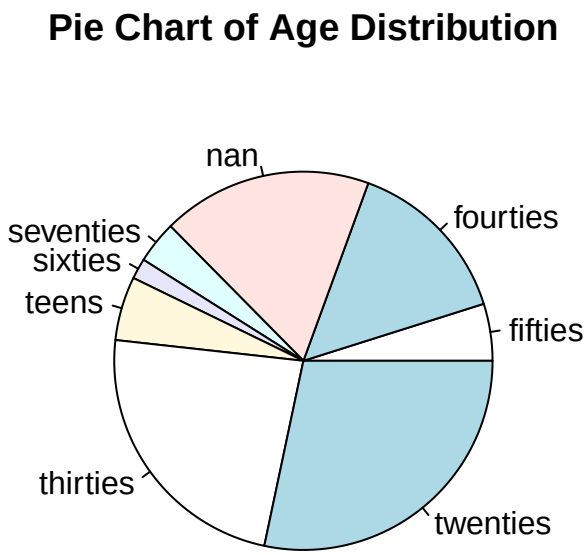



```
qqnorm(dfrange_trans, main = "QQ Plot for log-transformed dfrange")
qqline(dfrange_trans, col = "red")
```

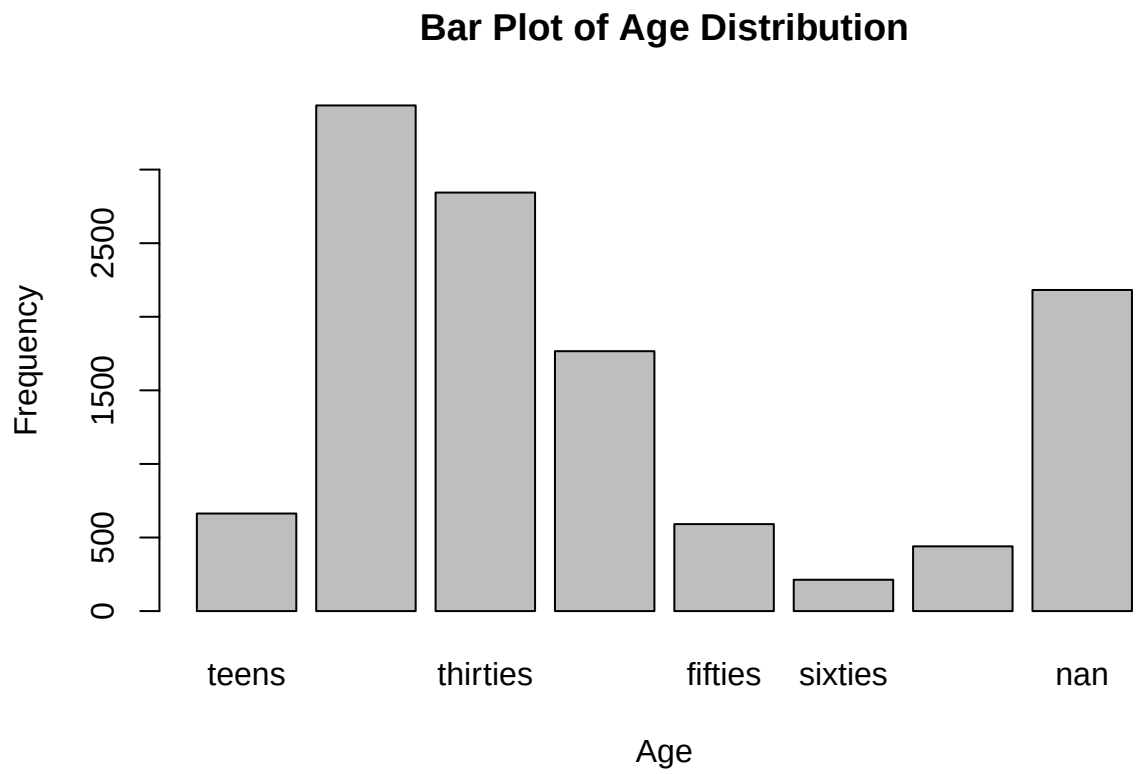


Age

```
age <- voice$age
age_table <- table(age)
pie(age_table, main = "Pie Chart of Age Distribution")
```

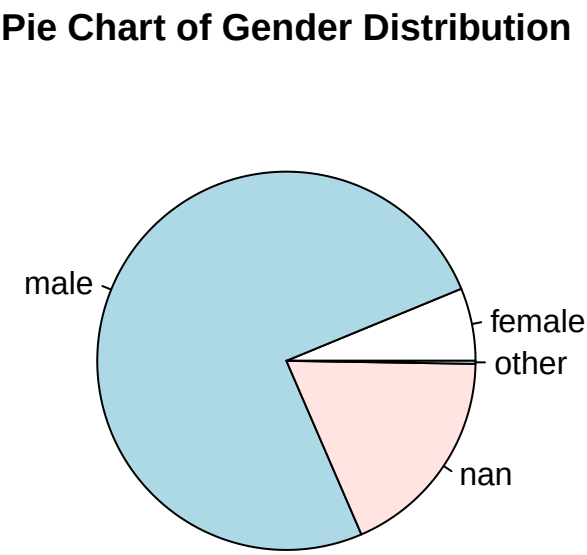



```
barplot(table(factor(age, levels = c("teens", "twenties", "thirties", "fourties", "fifties", "sixties",
```



Gender

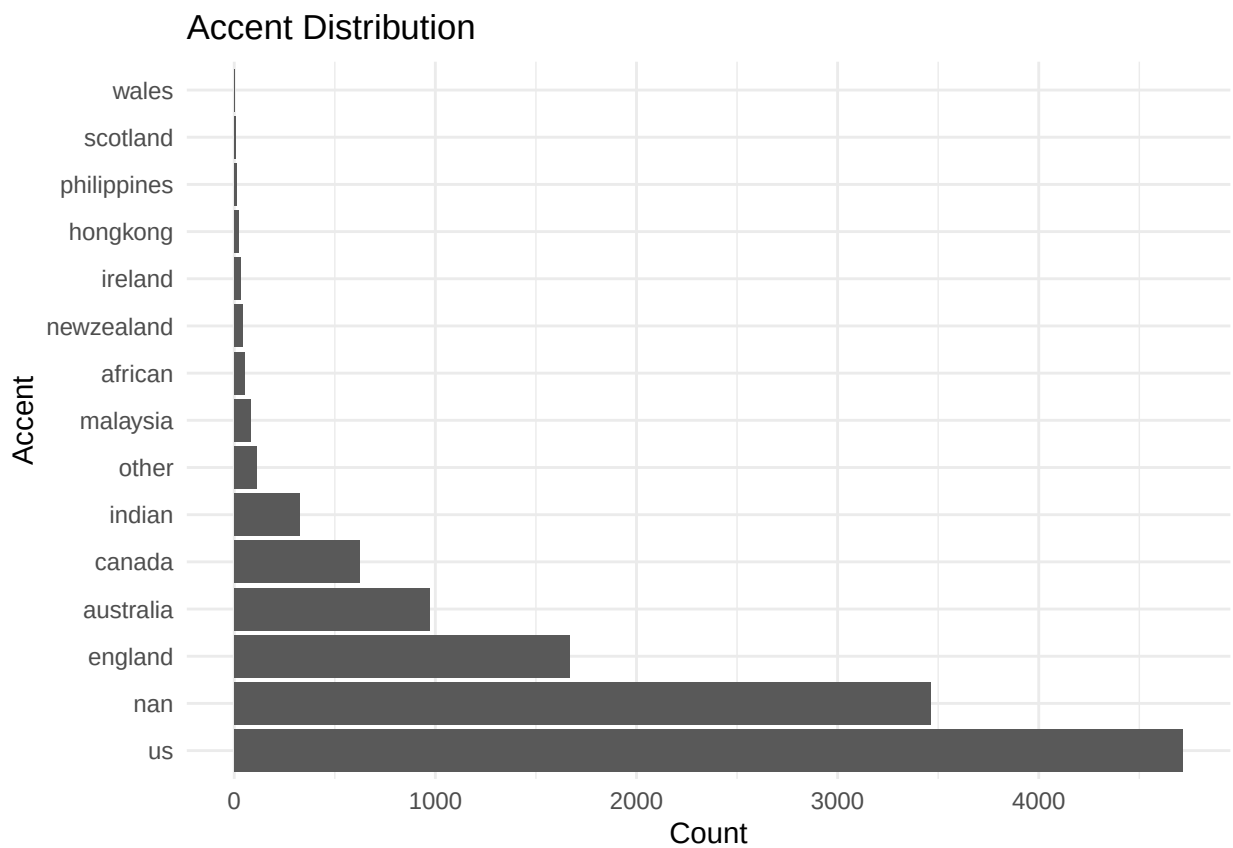
```
gender <- voice$gender
pie(table(gender), main = "Pie Chart of Gender Distribution")
```



Accent

```
library(ggplot2)

ggplot(voice, aes(x = reorder(accent, -table(accent)[accent]), y = after_stat(count))) +
  geom_bar(stat = "count") +
  coord_flip() +
  labs(x = "Accent", y = "Count", title = "Accent Distribution") +
  theme_minimal() +
  theme(axis.text.y = element_text(hjust = 1))
```



Merge Data

```
transformed_voice <- data.frame(
  id = voice$id,
  meanfreq = meanfreq_trans,
  sd = sd_trans,
  median = median_trans,
  Q25 = Q25_trans,
  Q75 = Q75_trans,
  IQR = IQR,
  skew = skew,
  kurt = kurt_trans,
  sp.ent = sp_ent_trans,
  sfm = sfm_trans,
  mode = mode_trans,
  centroid = centroid_trans,
  meanfun = meanfun_trans,
  meandom = meandom_trans,
  mindom = mindom_trans,
  maxdom = maxdom_trans,
  dfrange = dfrange_trans,
  age = age,
  gender = gender,
  accent = voice$accent
```

```r
)
```

```r
head(transformed_voice)
```

```
##   id meanfreq       sd   median      Q25      Q75      IQR        skew
## 1  0 8.166690 8.581521 8.005468 7.456112 9.171794 2961.459  0.11656897
## 2  1 8.340455 8.599874 8.367010 7.550676 9.296616 4069.043  0.04560770
## 3  2 8.056571 8.546849 7.867097 7.333423 9.110954 2593.316 -0.16147499
## 4  3 8.385794 8.704387 8.139291 7.418224 9.415997 5687.242  0.02416762
## 5  4 8.424453 8.725525 8.421371 7.425456 9.478406 6466.170  0.11711588
## 6  5 8.311154 8.696711 7.849907 7.226416 9.389455 5656.297  0.13049391
##          kurt   sp.ent      sfm     mode centroid  meanfun   meandom     mindom
## 1  0.25601554 3.291912 2.286142 7.529048 8.166690 3.564867 -2.647211  -6.923621
## 2  0.20816643 3.355465 2.494345 7.694165 8.340455 3.819150 -2.901194  -7.255858
## 3 -0.35918085 3.272155 2.239860 7.425196 8.056571 3.372699 -2.392059  -8.128257
## 4  0.57881969 3.401972 2.743840 7.737472 8.385794 3.702937 -3.892395 -17.268256
## 5  0.46910942 3.367561 2.776193 7.774467 8.424453 3.476591 -3.558232 -13.891542
## 6  0.07404453 3.363855 2.719465 7.666223 8.311154 3.493280 -3.680297 -14.317547
##     maxdom  dfrange      age gender accent
## 1 5.790664 5.790654 twenties female canada
## 2 4.676579 4.676568 twenties female canada
## 3 5.484731 5.484728 twenties female canada
## 4 3.095723 3.095723      nan    nan    nan
## 5 4.400148 4.400148      nan    nan    nan
## 6 3.193816 3.193816      nan    nan    nan
```

```r
write.csv(transformed_voice, "../data/transformated/train.csv", row.names = FALSE)
```