Lab 4

Grace Randall

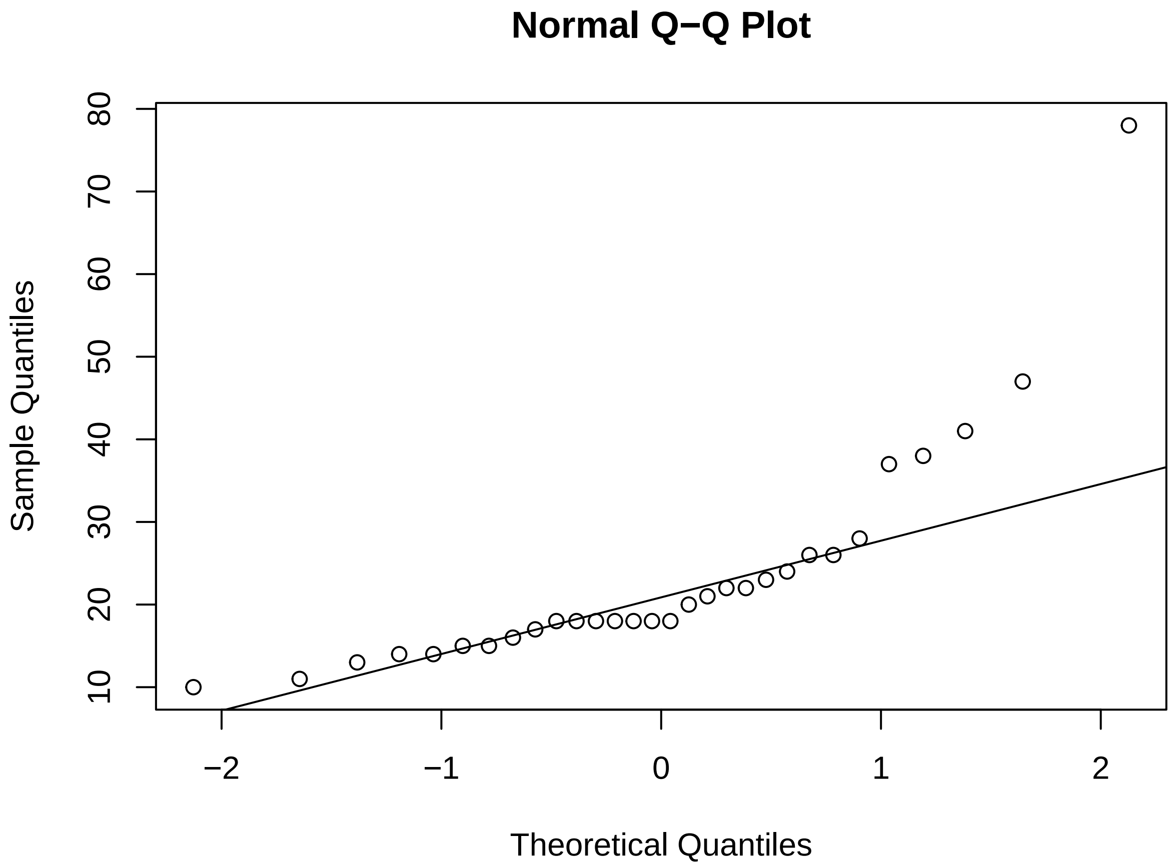
2024-02-14

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
setwd('~/applied_stats/Applied_stats/Lab_4')
library(tidyverse)
```

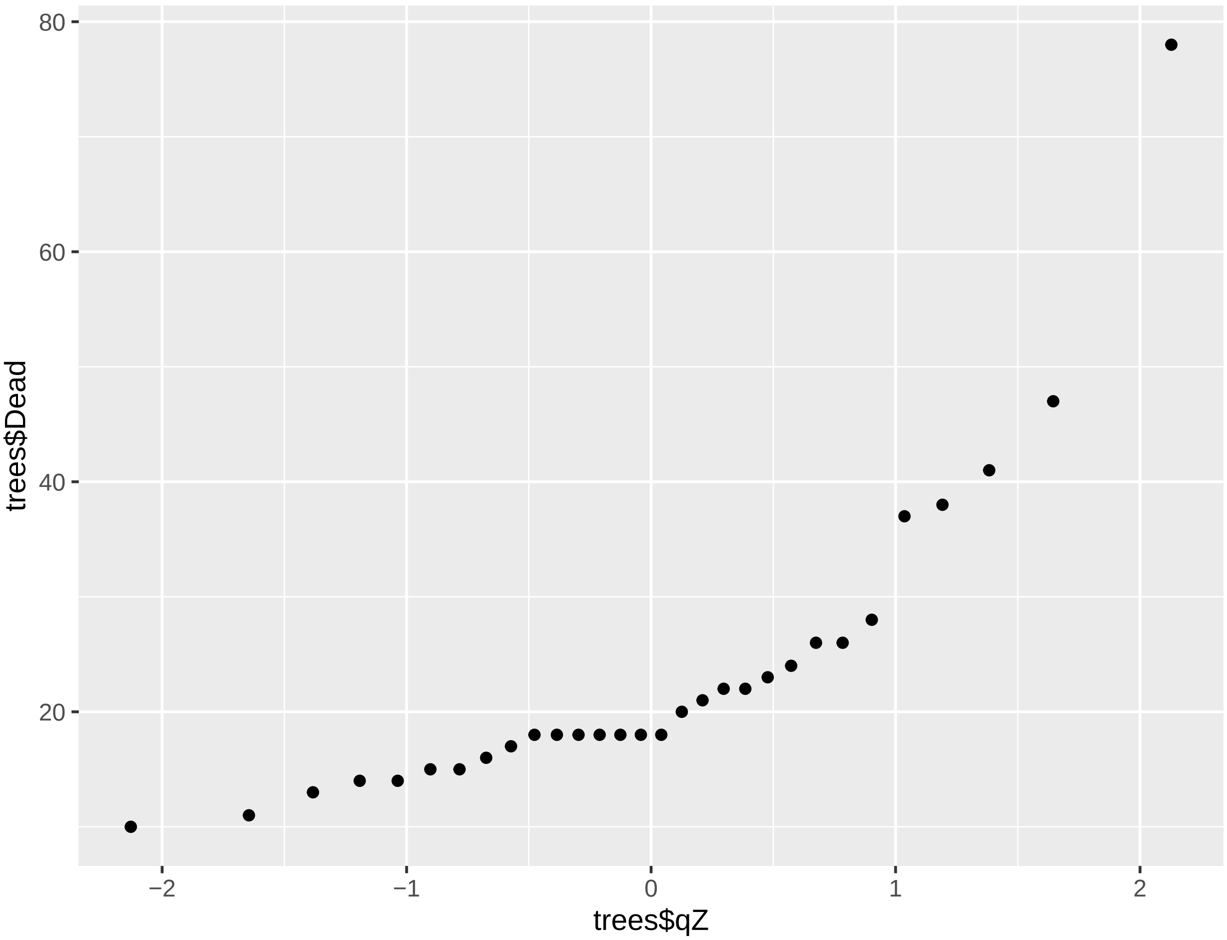
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
#install.packages("moments")
library(moments)

trees <-read.csv("~/applied_stats/Applied_stats/Lab_2/AfrPlots.csv", header = T)
trees<- filter(trees,CensusNo == 2)

gas <- read.csv("epa2012.csv", header = T)
```

```
#1
trees <- arrange(trees, Dead)

#2
n=nrow(trees)
trees <- mutate(trees, p = ( 1:n - 0.5)/n)

#3
trees <- mutate(trees, qZ = qnorm(p, mean = 0, sd = 1))

#4
qqnorm(trees$Dead)
qqline(trees$Dead)
```

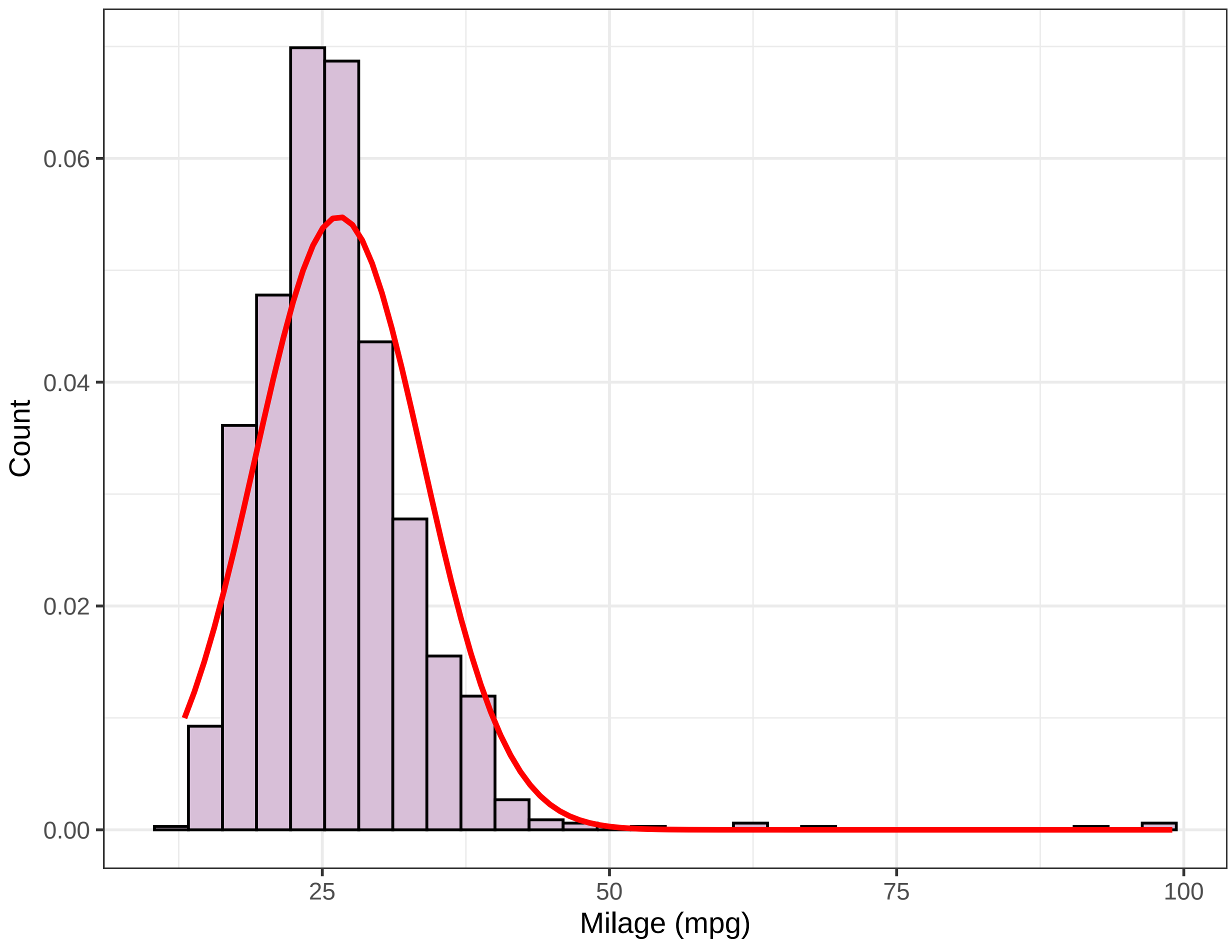


```
ggplot()+
  geom_point(aes(x=trees$qZ,y=trees$Dead))
```

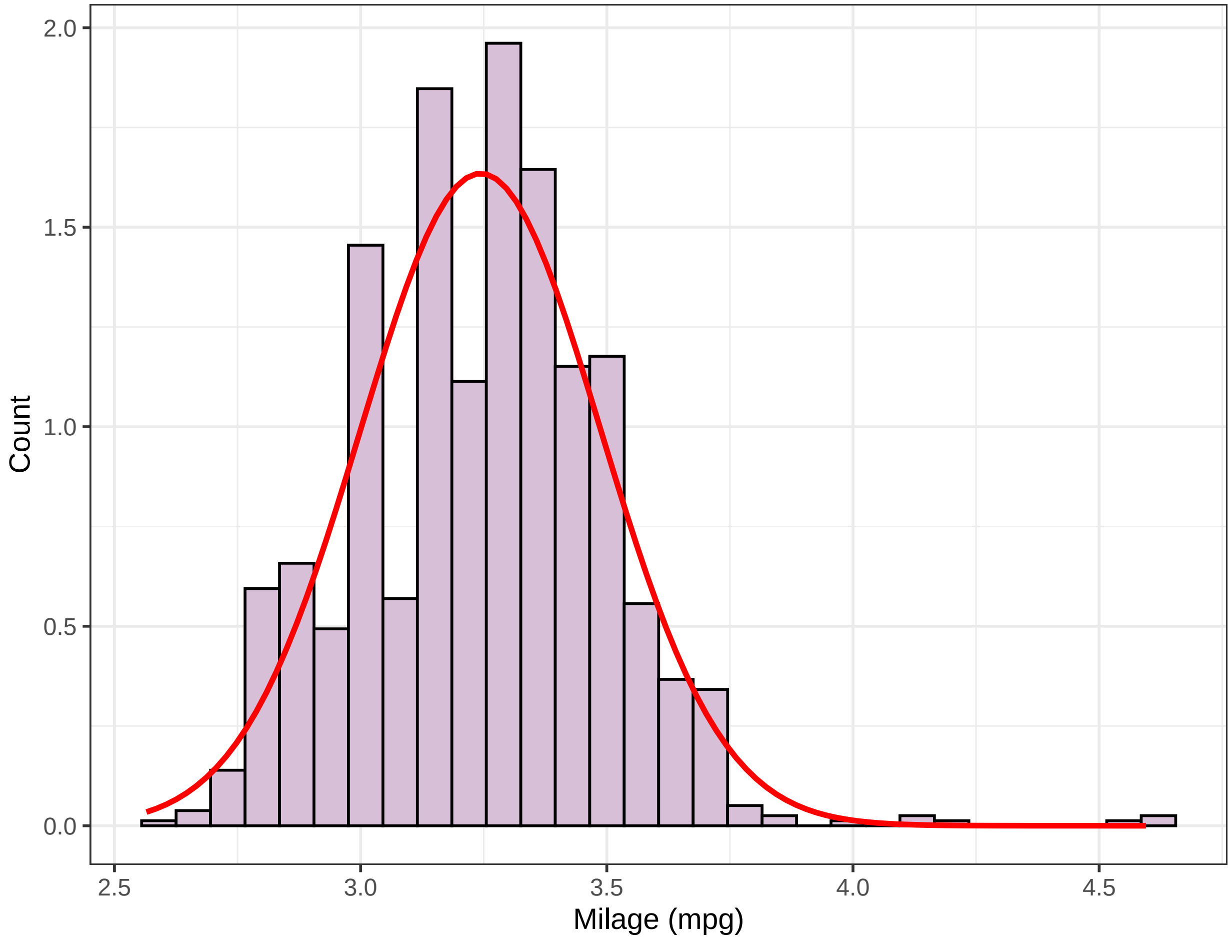


In order to make a qq-plot the data values are first ordered from smallest to largest. They are then assigned percentiles based on where they show up in the data. Those percentiles are then converted into z scores based on where that percentile would be expected to appear on a normal distribution. Then the values are plotted against the expected z score for that place in the data. If the data is normally distributed then the points should make a straight line from bottom left to top right.

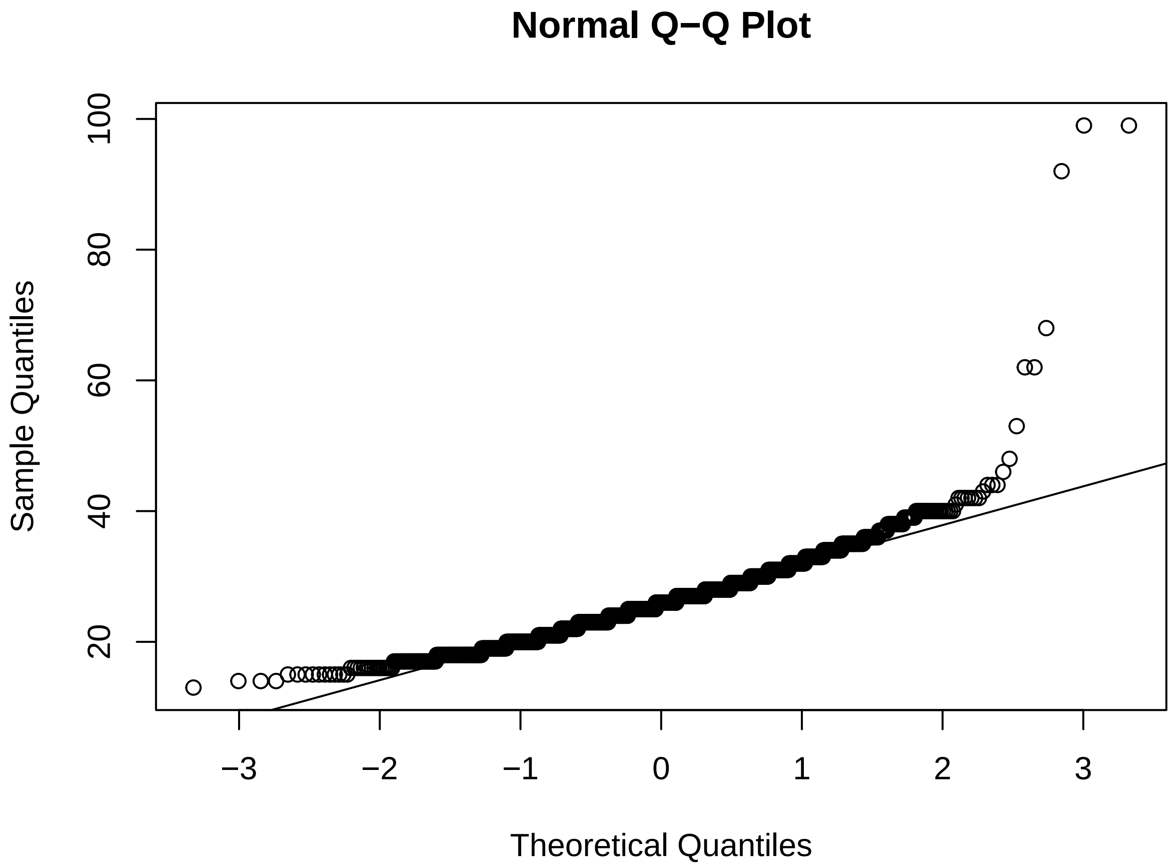
```
#a

plot_a1 <- ggplot(gas, aes(x = gas$hwy_mpg)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "thistle") +
  stat_function(fun = dnorm,
                args = list(mean = mean(gas$hwy_mpg),
                            sd = sd(gas$hwy_mpg)),
                            lwd = 1,
                            col = 'red') +
  xlab("Milage (mpg)") + ylab("Count") +
  theme_bw()

plot_a1
```



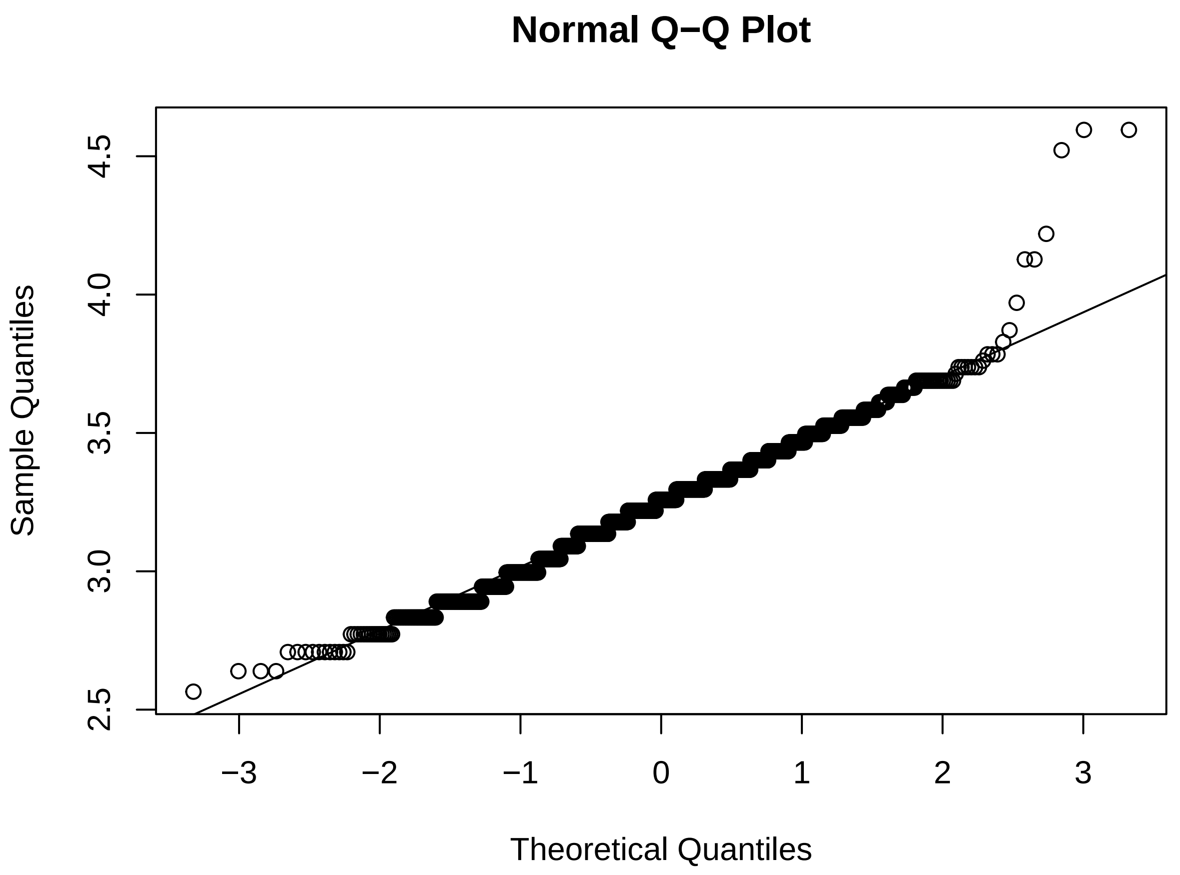
```r
#loggas=log(gas$hwy_mpg)
plot_a2 <- ggplot(gas, aes(x = log(gas$hwy_mpg))) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "thistle") +
  stat_function(fun = dnorm,
                args = list(mean = mean(log(gas$hwy_mpg)),
                            sd = sd(log(gas$hwy_mpg))),
                lwd = 1,
                col = 'red') +
  xlab("Milage (mpg)") + ylab("Count") +
  theme_bw()

plot_a2
```



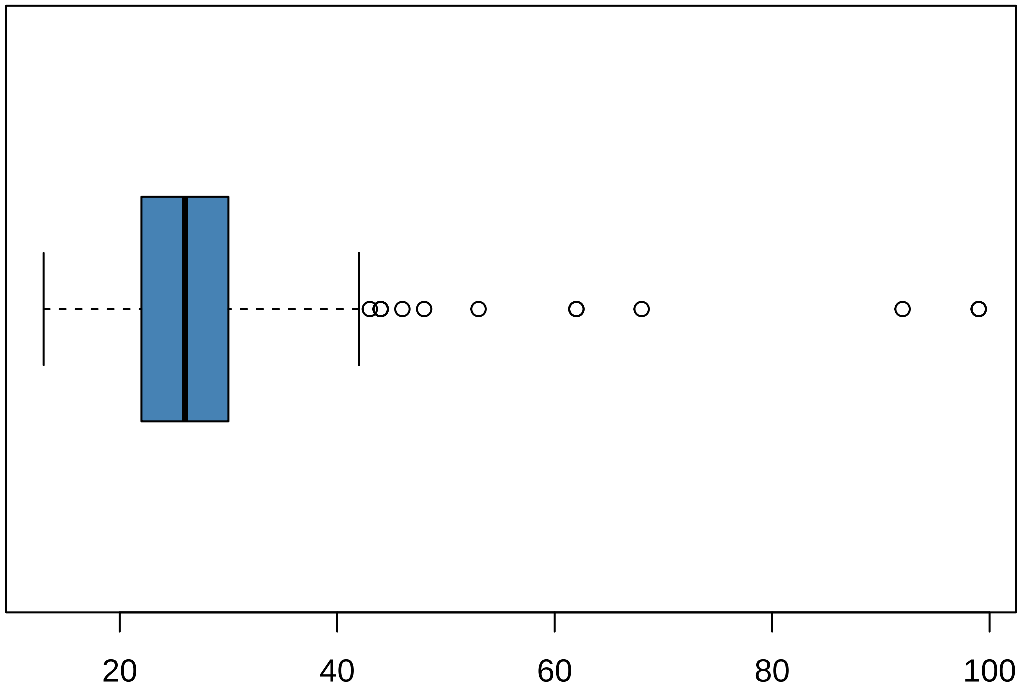
```
#b
#original data
qqnorm(gas$hwy_mpg)
qqline(gas$hwy_mpg)
```



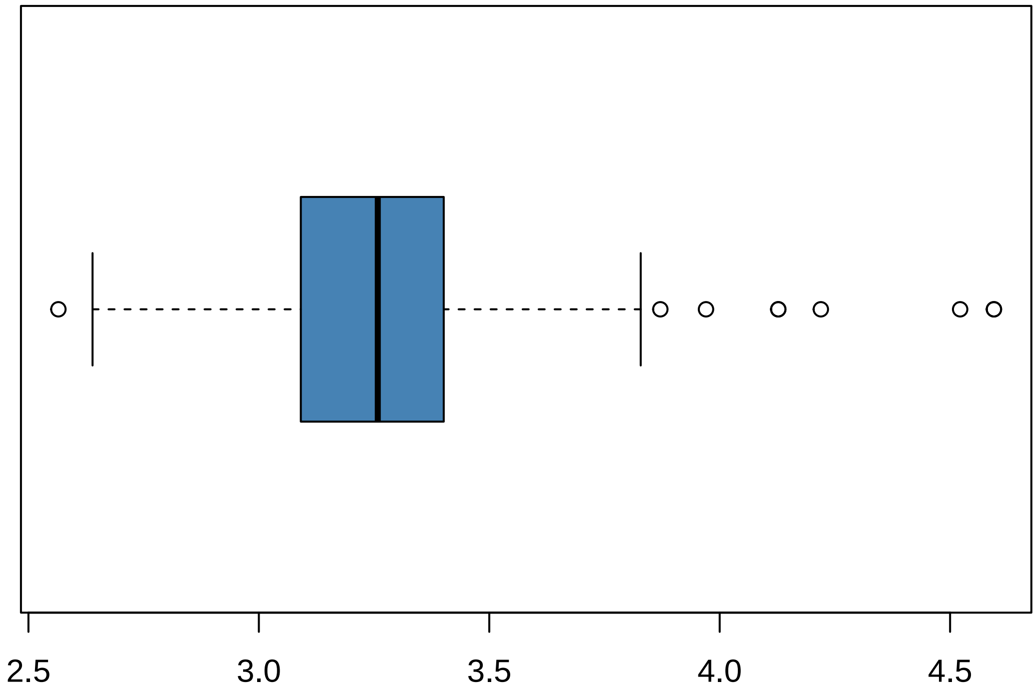
```r
#log transformed
qqnorm(log(gas$hwy_mpg))
qqline(log(gas$hwy_mpg))
```



```
#c
#original data
boxplot(gas$hwy_mpg, horizontal=TRUE, col='steelblue')
```



```r
#log transformed
boxplot(log(gas$hwy_mpg), horizontal=TRUE, col='steelblue')
```



```
#d
#original data
skewness(gas$hwy_mpg)
```

```
## [1] 2.991233
```

```
kurtosis(gas$hwy_mpg)
```

```
## [1] 26.73724
```

```
#log transformed
skewness(log(gas$hwy_mpg))
```

```
## [1] 0.4469149
```

```
kurtosis((gas$hwy_mpg))
```

```
## [1] 26.73724
```

the log transformed version of the data has a distribution that is closer to a normal distribution than the original data. the log transformed data has a lower skewness and its qq plot is more linear.