Exploring Week 3 in R: Correlation

Tuesday

Introduction to Today

The goal for today is to put into practice the lecture topics we have gone over thus far.Today we will focus on correlations.

R Tip of the Day: How to quickly add it <- assignment arrow.
Shortcut for Mac: Option -
Shortcut for Windows: Alt -

Try it below!

```
hello <- "hello"
```

Loading Our Libraries

First, we have to load our "libraries". A **library** in R, as a reminder, is an open-source package created by a very kind individual that contains functions (short cuts) to get things done in R.

```
library(car)
library(ggplot2)
library(psych)
library(dplyr)
```

Variance

The **variance** is a numerical measure of how the data values is dispersed around the mean. Let's take our in-class example and work through it in R.

Creating the Data

```
adverts <- c(5,4,4,6,8)
packets <- c(8,9,10,13,15)
```

Calculating Mean Adverts

```
(mean.advert <- mean(adverts))
```

```
## [1] 5.4
```

Calculating Variance (By "Hand") via Formula for Adverts

```r
#subtracting the mean from each value
adverts - mean.advert
```

```
## [1] -0.4 -1.4 -1.4  0.6  2.6
```

```r
#see what happens when we sum the individual variances without squaring
round(sum(adverts - mean.advert))
```

```
## [1] 0
```

```r
#square it
(adverts - mean.advert)^2
```

```
## [1] 0.16 1.96 1.96 0.36 6.76
```

```r
#sum it
sum((adverts - mean.advert)^2)
```

```
## [1] 11.2
```

```r
#the bottom portion of the formula is n - 1
#n represent the sample size aka the number of data points.
#we can use the lenght() function to get this
length(adverts) - 1
```

```
## [1] 4
```

```r
#put it all together
sum((adverts - mean.advert) * (adverts - mean.advert)) / (length(adverts) - 1)
```

```
## [1] 2.8
```

The variance from our by "hand" calculation is 2.8. Let's see if this matches the quick & easy R Function *var()*

```r
#Does it match up with the var( function?)
var(adverts)
```

```
## [1] 2.8
```

It does! Well, now that we know this hand dandy function, let's skip the work & use the function to get the variance of packets

Using the var() function for packets

```r
var(packets)
```

```
## [1] 8.5
```

The variance is 8.5. Now, let's do one more example.

Getting the Variance of Geyser Data

Let's find the variance of the eruption duration in the data set faithful.

```r
head(faithful)
```

```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

We apply the *var()* function to compute the variance of eruptions.

```r
duration = faithful$eruptions     # the eruption durations
var(duration)                     # apply the var function
```

```
## [1] 1.302728
```

The variance is 1.30.

Covariance

The **covariance** of two variables x and y in a data set measures how the two are linearly related. A positive covariance would indicate a positive linear relationship between the variables, and a negative covariance would indicate the opposite.

Calculating Covariance (By "Hand") via Formula for Adverts & Packets

```
#To calculate covariance, for each participant:
# we multiple their advert and packets variance scores together.
(adverts - mean.advert) * (packets - mean(packets))
```

```
## [1]  1.2  2.8  1.4  1.2 10.4
```

```
#Now, we sum at all up
sum((adverts - mean.advert) * (packets - mean(packets)))
```

```
## [1] 17
```

```
#For the bottom of the formula, it is the same as before. n - 1
#The length is the same for both adverts and packets, 5.
length(adverts) == length(packets)
```

```
## [1] TRUE
```

```
length(adverts)
```

```
## [1] 5
```

```
length(packets)
```

```
## [1] 5
```

```
#So, n - 1 = 4 again.
#Let's put it all together now
sum((adverts - mean.advert) * (packets - mean(packets))) / 4
```

```
## [1] 4.25
```

The covariance from our by "hand" calculation is 4.25. Let's see if this matches the quick & easy R Function *cov()*

```
cov(adverts, packets)
```

```
## [1] 4.25
```

It does! It indicates a positive linear relationship between the two variables. Let's do another example with the Geyser data again.

Getting the Covariance of Geyser Data

We apply the *cov()* function to compute the covariance of eruptions and waiting.

```
duration = faithful$eruptions    # eruption durations
waiting = faithful$waiting       # the waiting period
cov(duration, waiting)           # apply the cov function
```

```
## [1] 13.97781
```

The covariance is 13.98. It indicates a positive linear relationship between the two variables.

Correlation

Covariance can tell us if there is a positive or negative linear relationship. However, magnitude (size) of the effect is hard to interpret as it is specific to the unit of measurement. That is why we will standardize it, the standardization of the covariance is called the correlation coefficient, which ranges from -1 to +1.

We will walk through a correlation example using the mtcars dataset. We will calculate the correlation of mpg and wt.

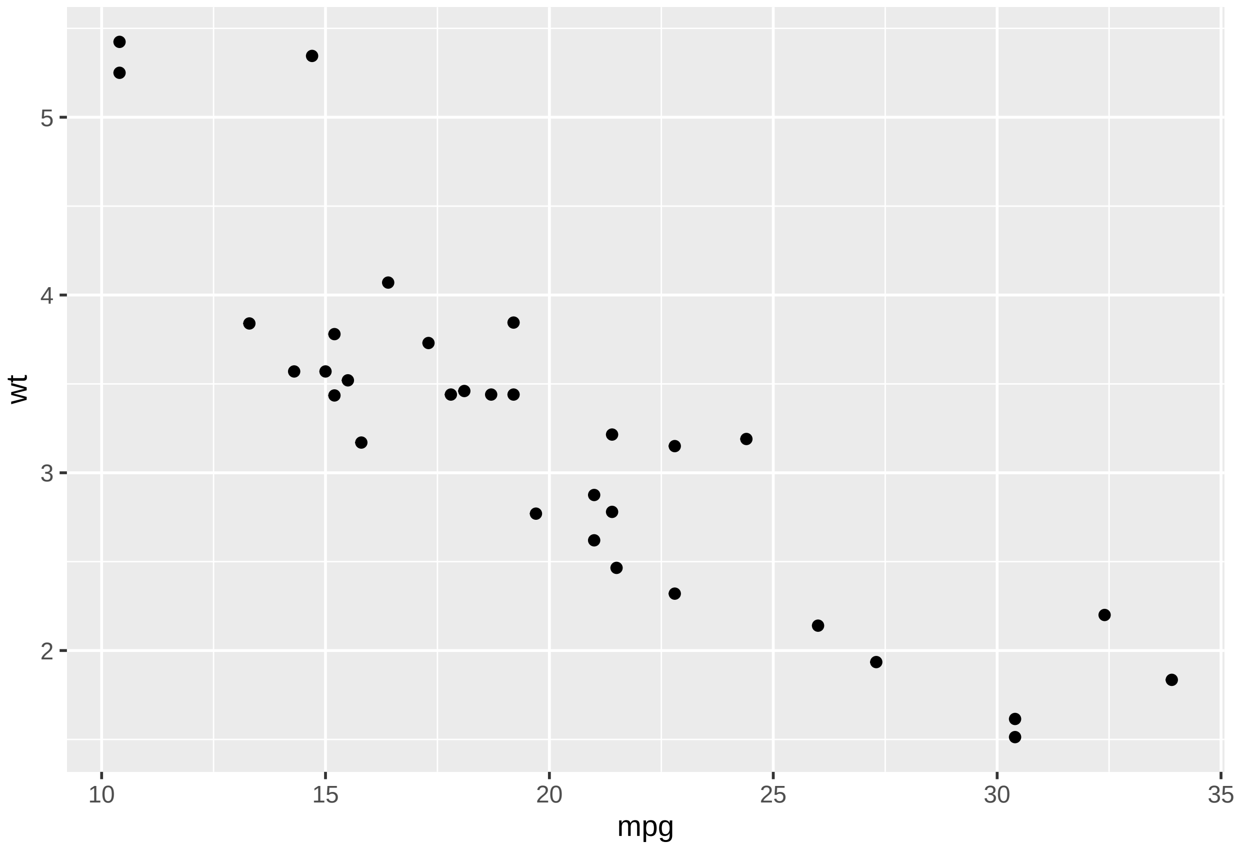
Let's take a peek at it using *head()*

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Visualizing our data with a scatterplot

```
ggplot(mtcars, aes(mpg, wt)) + geom_point()
```

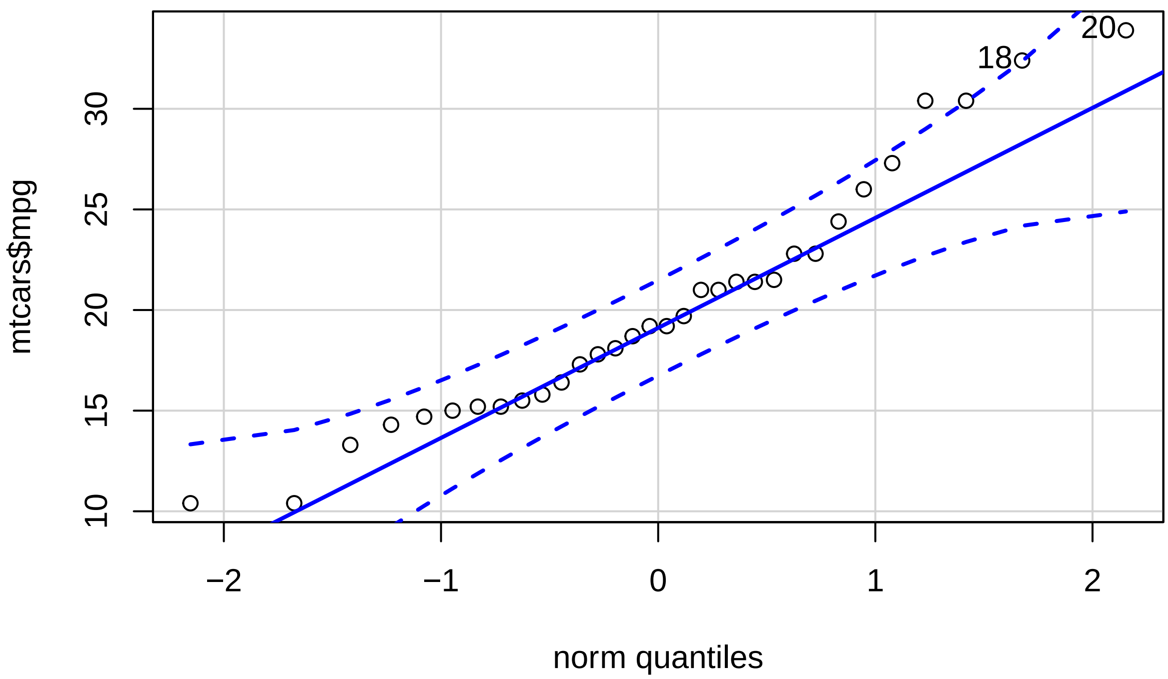


Very cool, *visually* it looks like a negative relationship.

Checking Assumptions: Normality

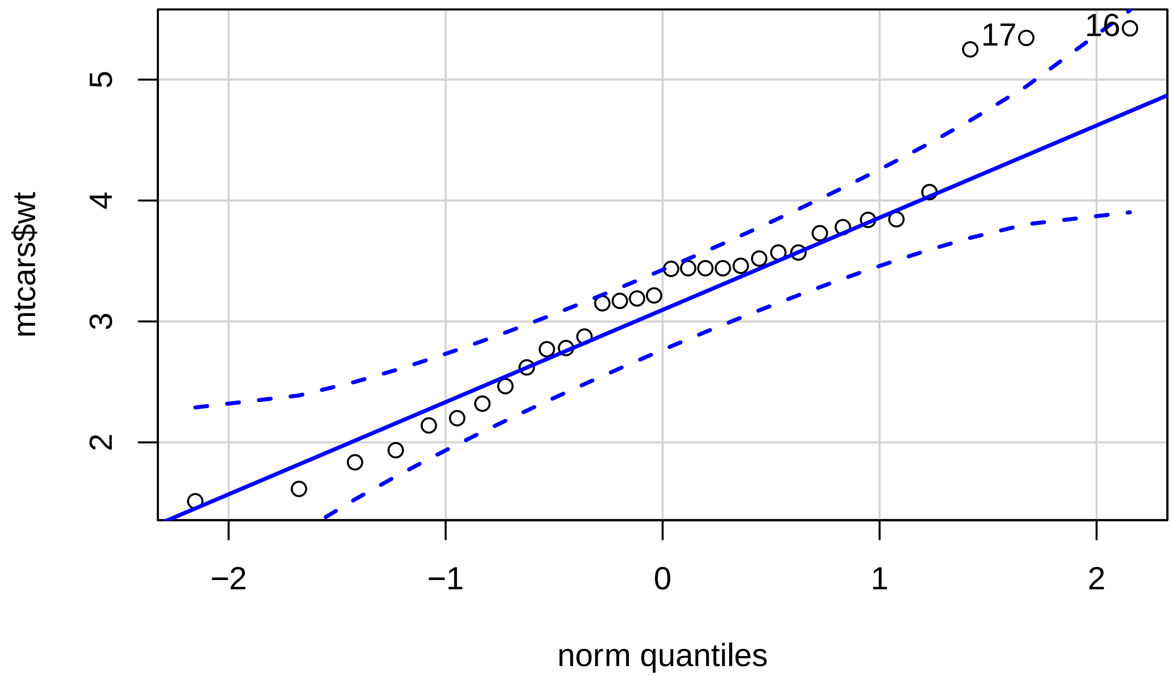
QQ Plot

```r
qqPlot(mtcars$mpg)
```




```
## [1] 20 18
```

```r
qqPlot(mtcars$wt)
```




```
## [1] 16 17
```

Looks good. Now, let's do Shapiro-Wilks test.

Shapiro-Wilks

```r
# Shapiro-Wilk normality test for mpg
shapiro.test(mtcars$mpg) # => p = 0.1229
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$mpg
## W = 0.94756, p-value = 0.1229
```

```r
# Shapiro-Wilk normality test for wt
shapiro.test(mtcars$wt) # => p = 0.09
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$wt
## W = 0.94326, p-value = 0.09265
```

p is > 0.05 for both, our assumption of normality is good to go.

From the normality plots/tests, we conclude that both populations may come from normal distributions.

Now, we can run our correlation!

Pearson Correlation Test

```r
r.mtcars <- cor.test(mtcars$wt, mtcars$mpg,
                   method = "pearson")
r.mtcars
```

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$wt and mtcars$mpg
## t = -9.559, df = 30, p-value = 1.294e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594
```

The p-value of the test is 1.29410^{-10} aka .00000000013, which is less than the significance level alpha = 0.05. We can conclude that wt and mpg are significantly negatively correlated with a correlation coefficient of -0.87 and p-value of 1.29410^{-10}. As mpg increasesm wt decreases. As wt increases, mpg decreases. For our statistical tests we *want* the p value to be less than .05. This allows us to accept the *alternative hypothesis* that there *is* is a significant relationship between the two variables.

Access to the values returned by cor.test() function

The function *cor.test()* returns a list containing the following components:
p.value: the p-value of the test
estimate: the correlation coefficient

Extract the p.value

```
r.mtcars$p.value
```

```
## [1] 1.293959e-10
```

Extract the correlation coefficient

```
r.mtcars$estimate
```

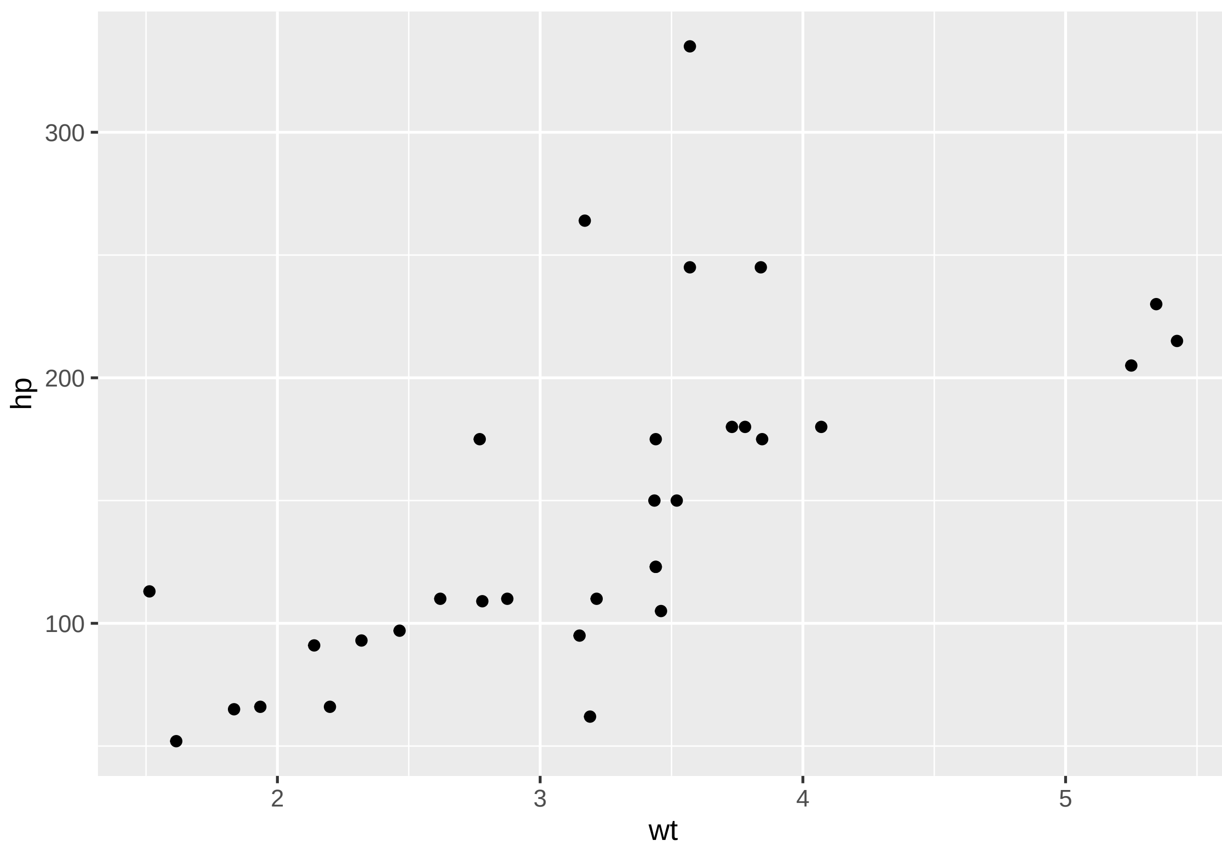
```
##        cor
## -0.8676594
```

Let's do one more correlation example.

Correlation of wt and hp

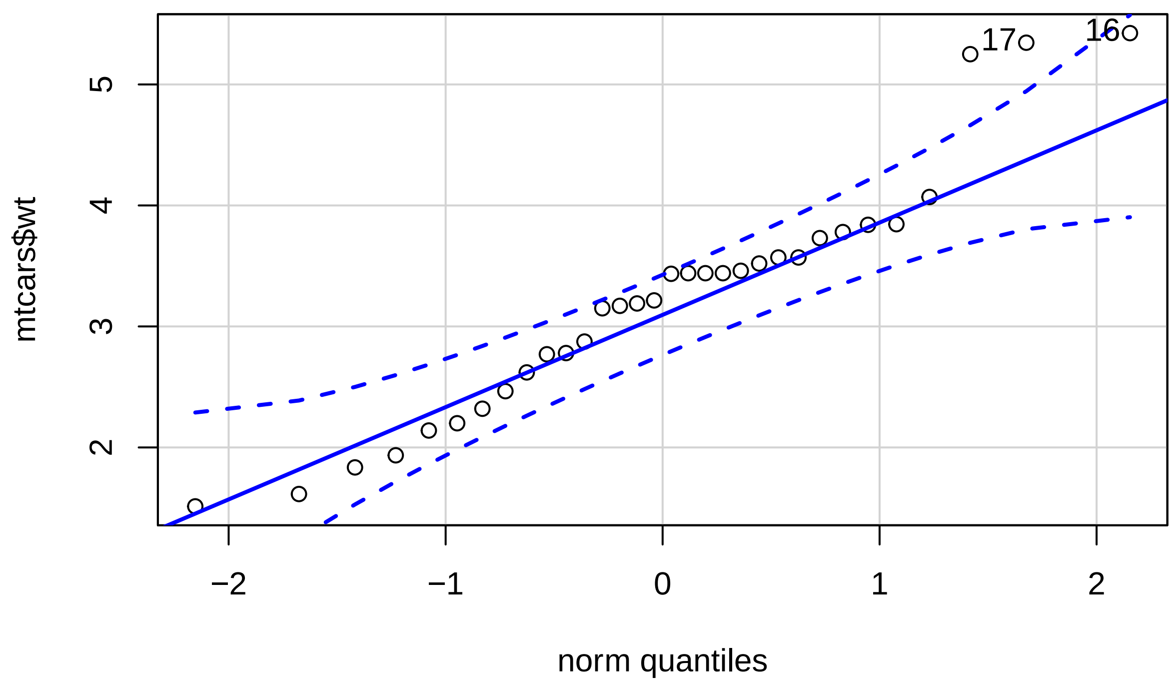
First, let's make a scatterplot

```
ggplot(mtcars, aes(wt, hp)) + geom_point()
```



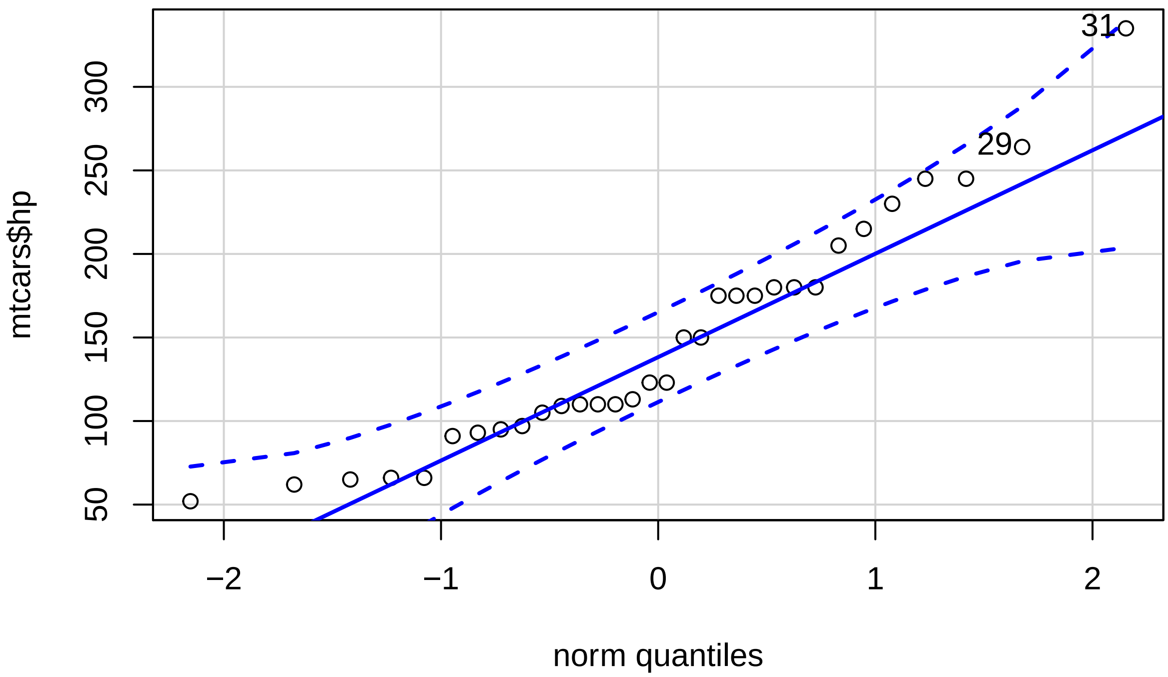
Now, Q-Q-Plots

```r
qqPlot(mtcars$wt)
```




```
## [1] 16 17
```

```r
qqPlot(mtcars$hp)
```




```
## [1] 31 29
```

Finally, Shapiro-Wilks

```r
shapiro.test(mtcars$wt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$wt
## W = 0.94326, p-value = 0.09265
```

```r
shapiro.test(mtcars$hp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$hp
## W = 0.93342, p-value = 0.04881
```

hp is slightly under .05, but it will work for today.

Correlation time!

```r
wt.hp.mtcars <- cor.test(mtcars$wt, mtcars$hp,
                    method = "pearson")
wt.hp.mtcars
```

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$wt and mtcars$hp
## t = 4.7957, df = 30, p-value = 4.146e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4025113 0.8192573
## sample estimates:
##       cor
## 0.6587479
```

The p-value of the test is 4.146^{-05} aka .00004, which is less than the significance level alpha = 0.05. We can conclude that wt and hp are significantly positively correlated with a correlation coefficient of 0.66 and p-value of 4.146^{-05}. As wt increases, so does hp. As hp increases,so does wt.