Course Content and Assignment 2 for Biom612 (Week 2)

Deanne Taylor, Pichai Raman, Joe Dybas, with some edits by Yuanchao Zhang

Jan 22nd, 2018

Contents

1 Packages to install this week

- ggplot2 (CRAN)
- nortest (CRAN)

```
tcga_luad<-read.csv("TCGA_LUAD_clinical.csv",  na.string="NA", stringsAsFactors=FALSE, row.names=1)
```

##The Normal Distribution

R provides several functions to allow you to generate normal distributions, for testing and for statistical use. You can use the help facility and type "?rnorm" to start reading about the Normal methods in the stats package.

Let's generate a normal distribution using rnorm().

```
opar<-par() #A little trick to save your original graphing configuration as a variable "opar" so you ca

par(mfrow=c(2,2)) #split your graphing window into a 2x2 (row,column). You may need to open up your plo

#let's build a random distribution with 1000 points total.

#The default mean for rnorm is 0 and the and the standard deviation is 1.

normdata<-rnorm(10)
hist(normdata)
mean(normdata)
```

```
## [1] -0.1997472
```

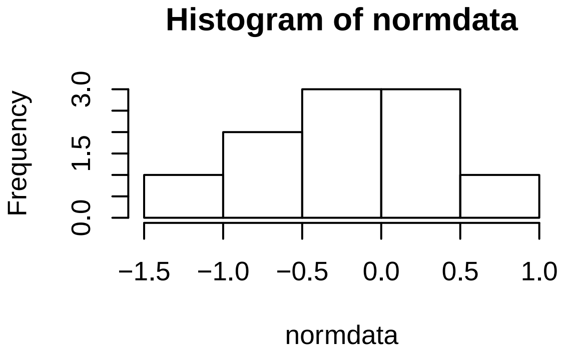
```
sd(normdata)
```

```
## [1] 0.6294271
```

```
#re-run the lines above three more times to fill up your 2x2.  Something should be obvious to the eye i

#At the end of your run, reload the naive parameters  you saved, to reset your graphics window, if you
```

```r
par(opar)
```

```
## Warning in par(opar): graphical parameter "cin" cannot be set

## Warning in par(opar): graphical parameter "cra" cannot be set

## Warning in par(opar): graphical parameter "csi" cannot be set

## Warning in par(opar): graphical parameter "cxy" cannot be set

## Warning in par(opar): graphical parameter "din" cannot be set

## Warning in par(opar): graphical parameter "page" cannot be set
```



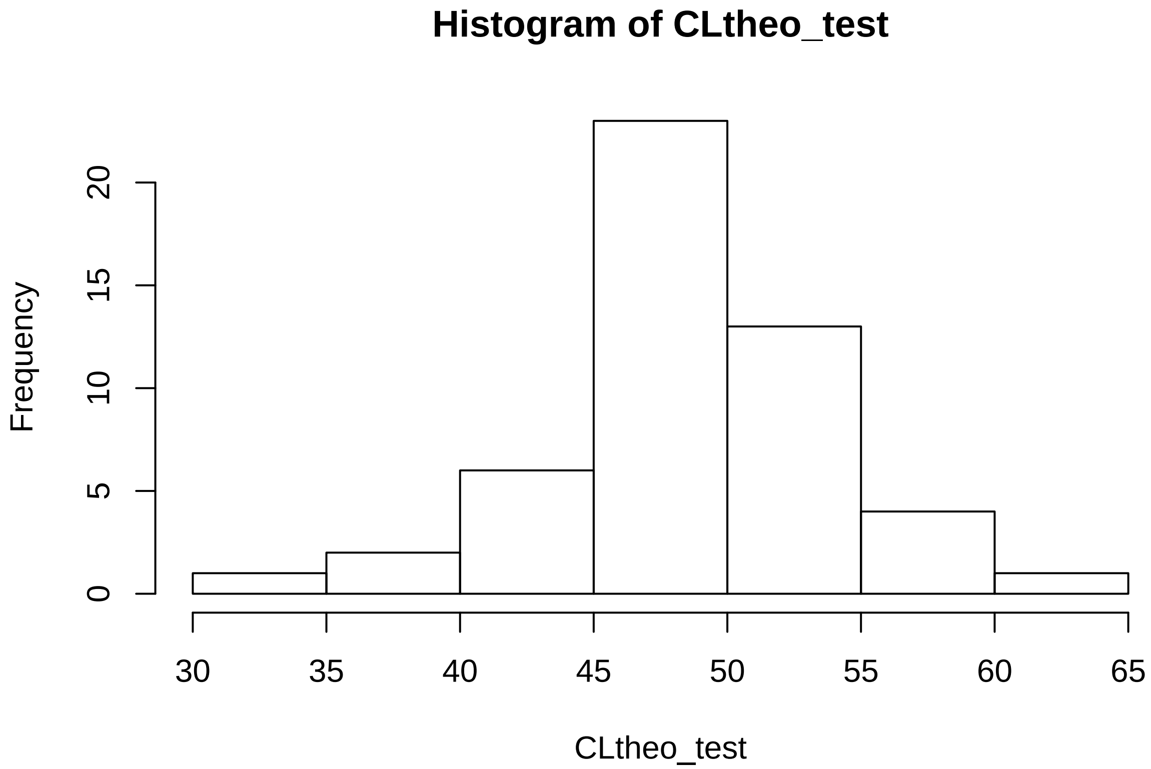
Now, in the chunk above, increase the number of random variables in the normal distribution to 100, then 1000. Run each four times.

How does this change the mean and the standard deviation from what you expect?

1.1 Central Limit Theorem example

Let’s generate a set of random trials using the sample() function (?sample):

```r
CLtheo_test<-c()
limit<-50
rolls=100

for(n in c(1:limit)){
 CLtheo_test<- append(CLtheo_test, sum(sample(c(1,0), rolls, replace=TRUE)))
}

hist(CLtheo_test)
```



```
#Feel free to change the limit of the simulation,  or the number of "rolls" of the "dice" and see how i
```

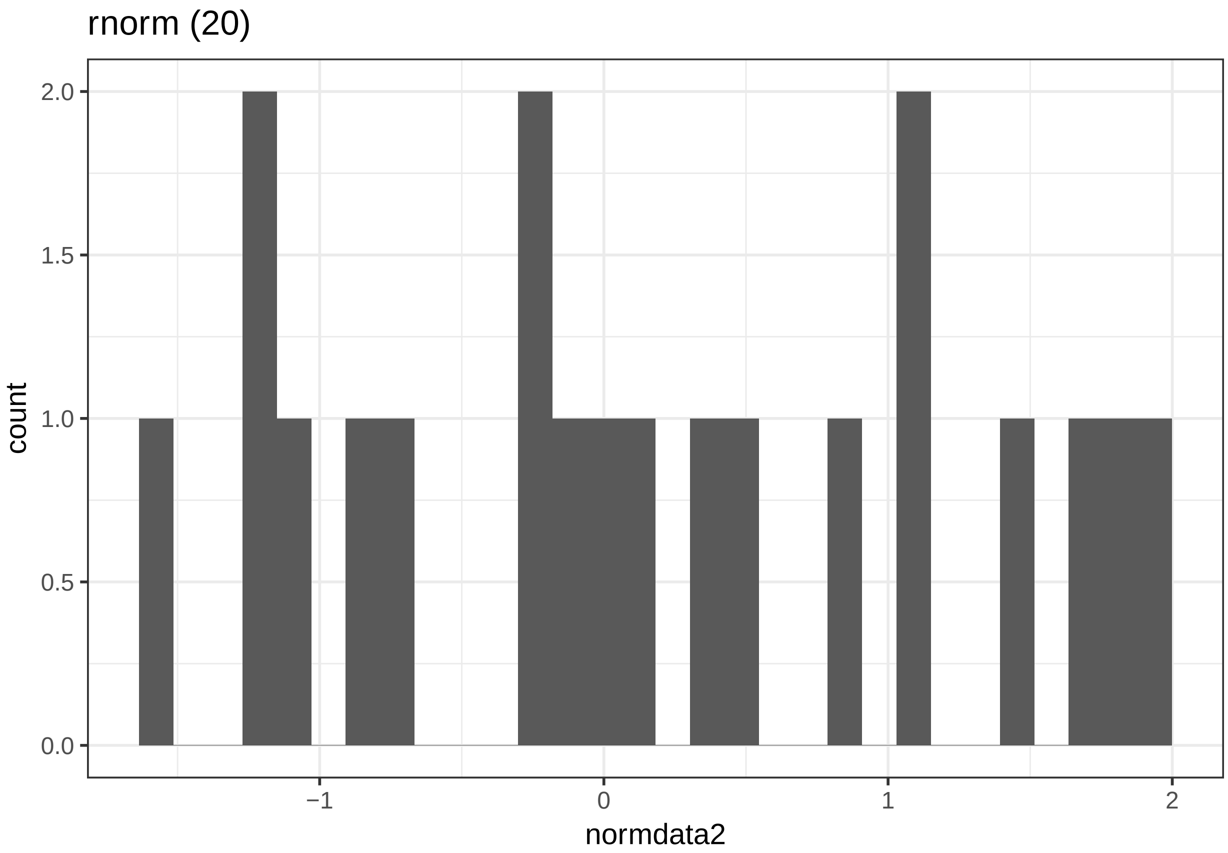
1.2 Testing for normality

Important to do, as many statistical tests require that data being tested is normally distributed.

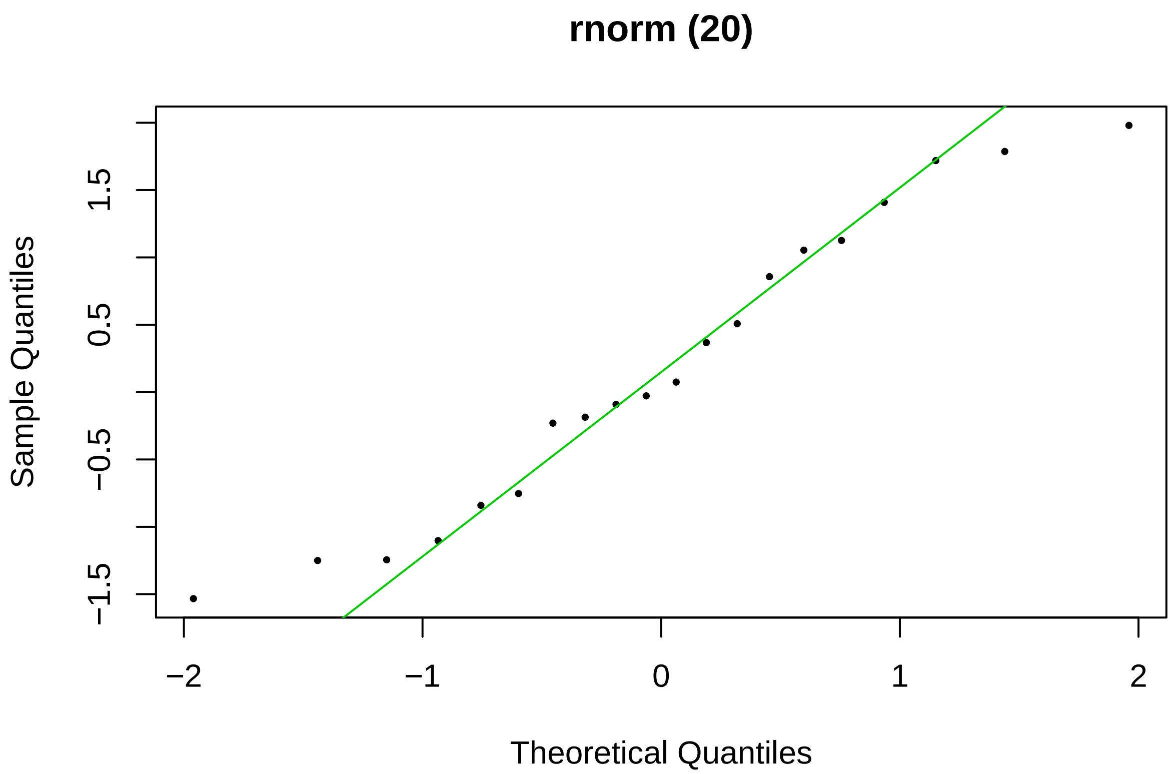
1.2.1 Plotting data, Quantile-quantile plots.

```
#First test a qq-plot on a truly normal distribution
num_points<-20
normdata2<-rnorm(num_points)
#let's use qplot to generate a histogram using qqplots.More on qqplots2 next week.

qplot(normdata2) + geom_histogram() + theme_bw() + ggtitle(paste("rnorm (", num_points, ")", sep=""))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

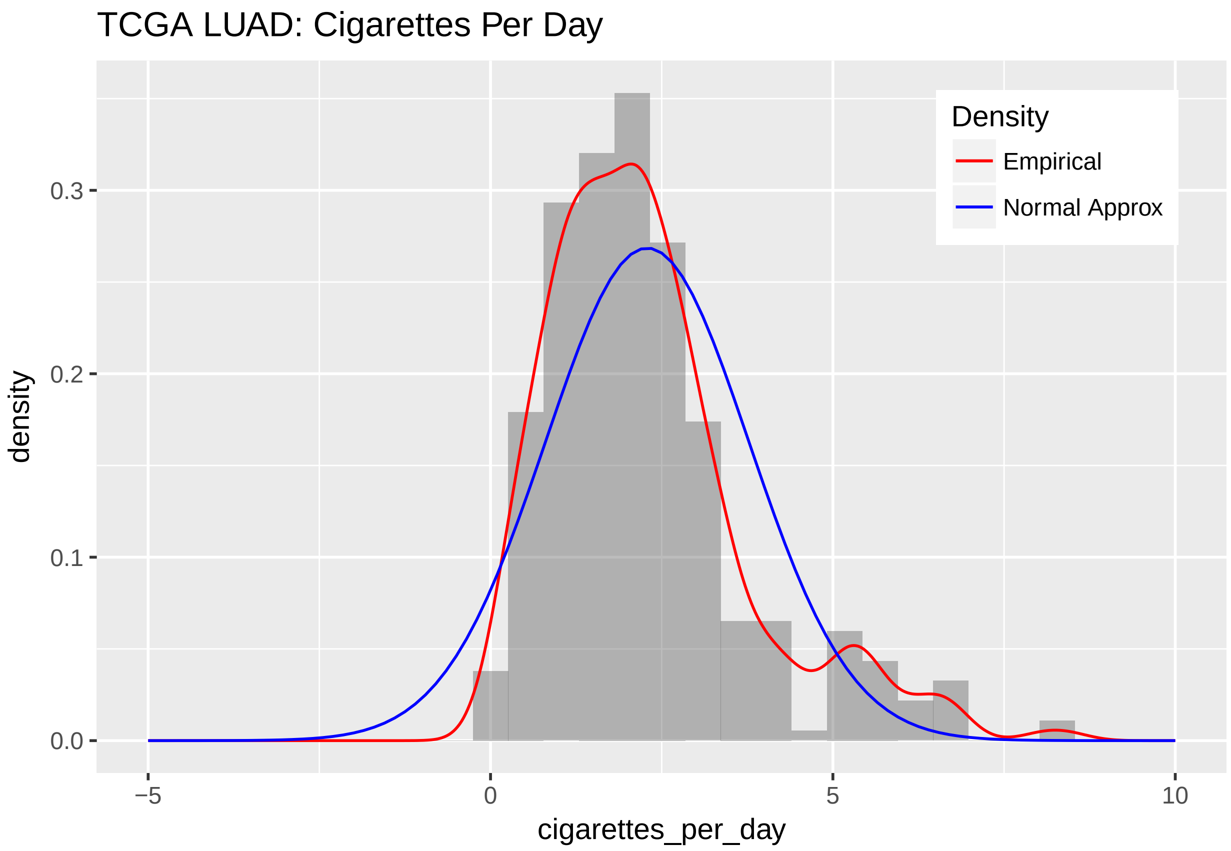


```
#ignore any binwidth errors today.

#Let's look at this distribution in normdata2 versus an idealized normal distribution (theoretical quan

qqnorm(normdata2,pch=16,cex=0.5, main=paste("rnorm (", num_points, ")", sep=""))
qqline(normdata2, col=3)
```



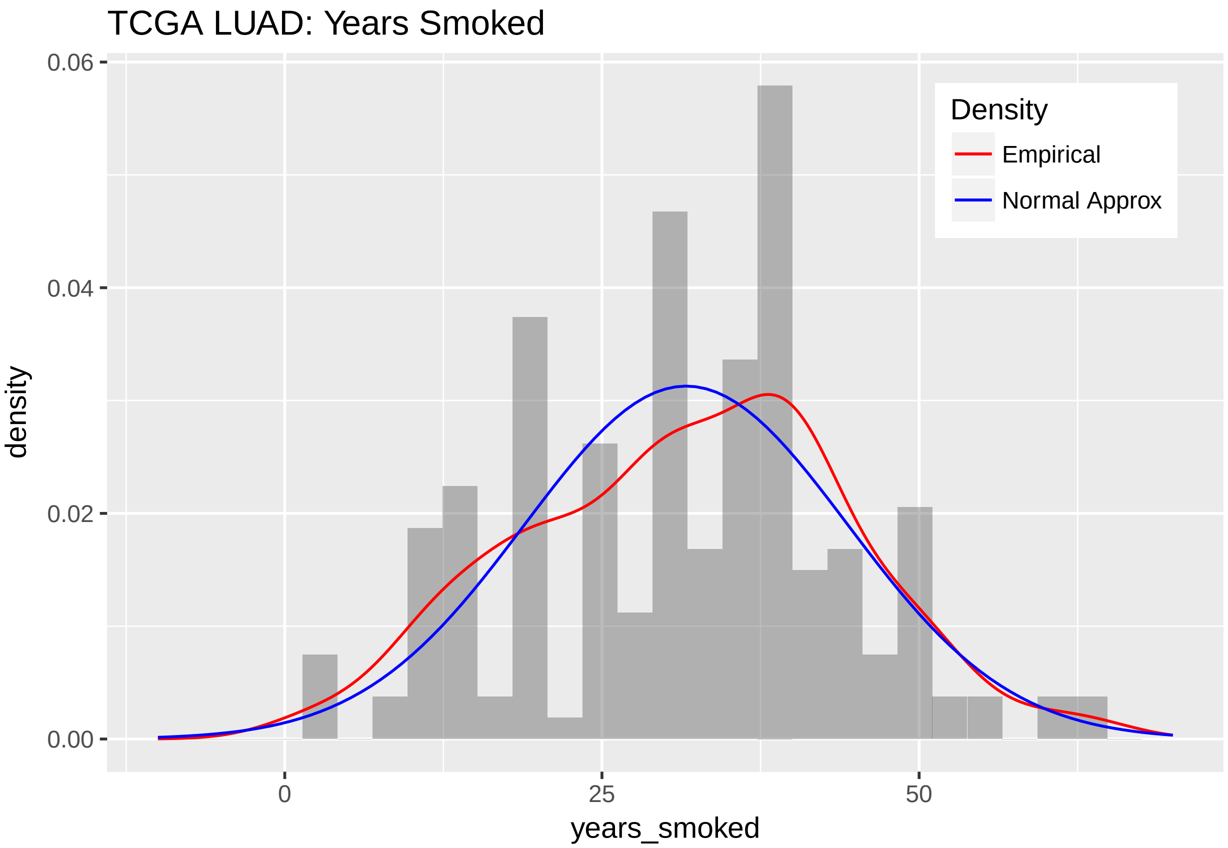
```
#Increase the number of num_points to a much higher number. What do you observe?

#Let's use real data: Plotting the distributions from TCGA-LUAD from Week 1.

#Let's first plot Cigarettes per day.
#mean and sd:

luad_cpd_sd<-sd(tcga_luad$cigarettes_per_day, na.rm=TRUE)
luad_cpd_mean<-mean(tcga_luad$cigarettes_per_day, na.rm=TRUE)

#Using ggplot2, graphing the cigarettes per day -- first with the histogram, then overlay a blue normal

qplot(x = cigarettes_per_day, xlim=c(-5,10), data = tcga_luad, geom = "blank") +
geom_histogram(aes(y = ..density..), alpha = 0.4) +
geom_line(aes(y = ..density.., colour = 'Empirical'),stat = 'density') +
stat_function(fun = dnorm, args = list(mean=luad_cpd_mean, sd=luad_cpd_sd), aes(colour = 'Normal Approx
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 166 rows containing non-finite values (stat_bin).

## Warning: Removed 166 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing missing values (geom_bar).
```

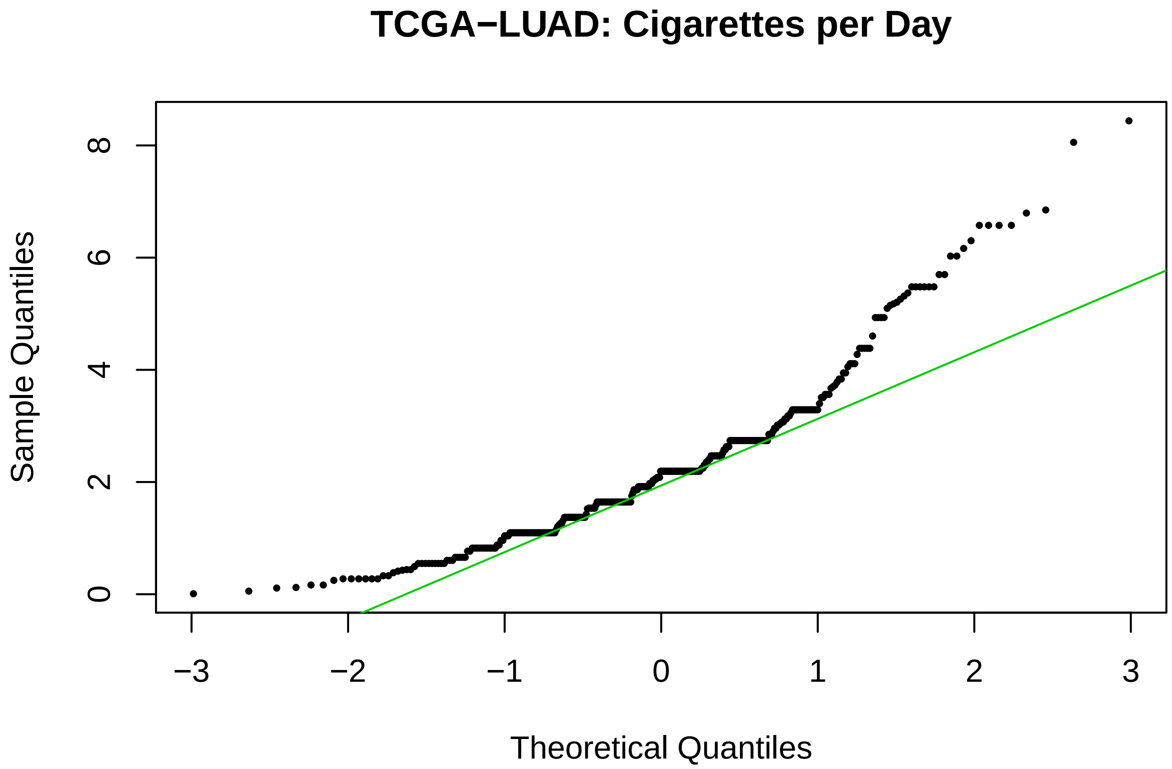


```
#Repeat this for years_smoked below:
luad_ys_sd<-sd(tcga_luad$years_smoked, na.rm=TRUE)
luad_ys_mean<-mean(tcga_luad$years_smoked, na.rm=TRUE)

qplot(x = years_smoked, data = tcga_luad, xlim=c(-10,70), geom = "blank") +  geom_histogram(aes(y = ..d
geom_line(aes(y = ..density.., colour = 'Empirical'),stat = 'density') +
stat_function(fun = dnorm, args = list(mean=luad_ys_mean, sd=luad_ys_sd), aes(colour = 'Normal Approx')
```

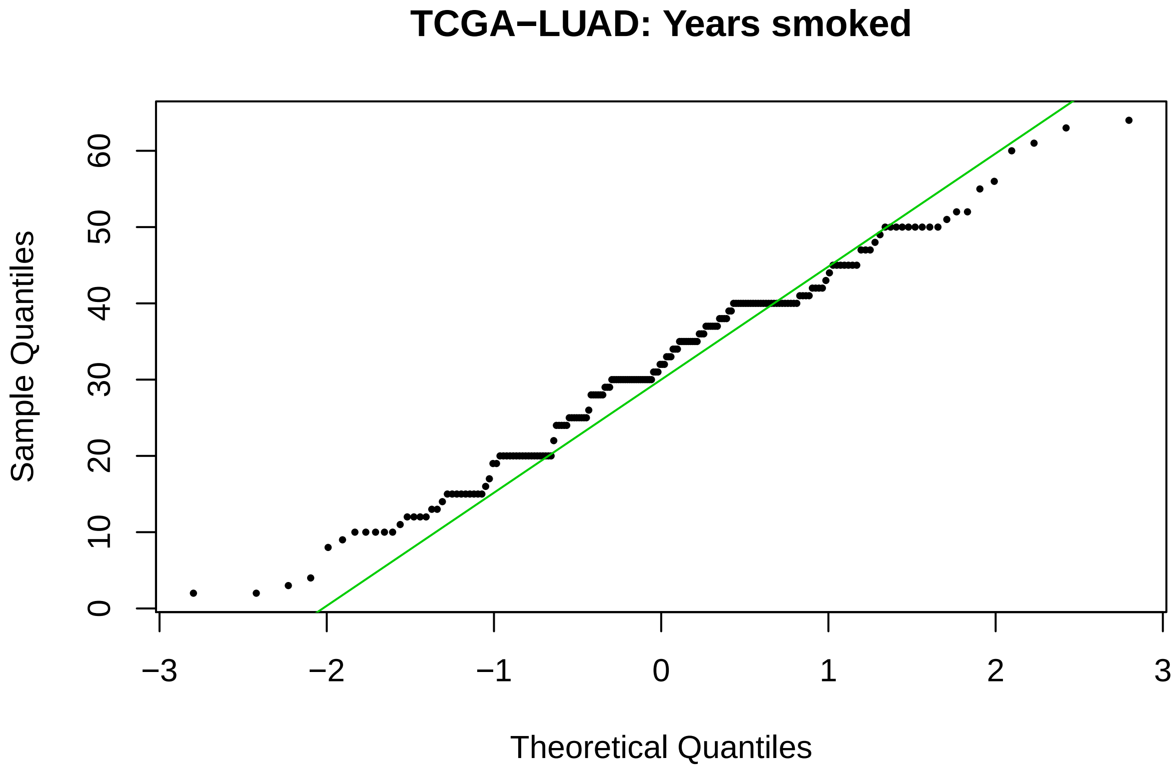
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 328 rows containing non-finite values (stat_bin).

## Warning: Removed 328 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing missing values (geom_bar).
```



```
#So basically both of them look like a little "like" they might be normal based on just the general dis

qqnorm(tcga_luad$cigarettes_per_day,pch=16,cex=0.5, main="TCGA-LUAD: Cigarettes per Day")

#now add a line approximating a linear "fit" to the majority of the data:

qqline(tcga_luad$cigarettes_per_day, col=3)
```



```
#Interpret the interesting shape of the points above the line -- check out the cigarettes_per_day histo

qqnorm(tcga_luad$years_smoked, pch=16,cex=0.5, main="TCGA-LUAD: Years smoked")
qqline(tcga_luad$years_smoked, col=3) #adds a trendline
```



In the qqplots (qqnorm), you can now clearly note that the cigarettes per day distribution deviates quite a bit from normal whereas the “years smoked”" distribution is probably more normally distributed. You can tell this by how far the sample and theoretical quantiles deviate from one-another (as evidenced by how far off the line they are). However, even if you see some points far off the line they are NOT outliers. A qqplot does not determine outliers! There are other methods for that, which we will discuss later in the semester.

Plots are all well and good, but what is the statistical evidence for deviations from normality?

The next thing you may do is perform a statistical test to measure the deviation from normal.

At this point you will download the package ‘nortest’ so you can run some of tests in this package. Then it is quite straightforward to run the test

```
#Testing the theoretical normal distribution

nortest_points<-100
normdata_nortest<-rnorm(nortest_points)

#Sharpio-Wilk test:
shapiro.test(normdata_nortest)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  normdata_nortest
## W = 0.9922, p-value = 0.8357
```

```
#Let's run the Anderson-darling test on normal data:
ad.test(normdata_nortest)
```

```
##
##  Anderson-Darling normality test
##
## data:  normdata_nortest
## A = 0.17531, p-value = 0.922
```

```
shapiro.test(tcga_luad$cigarettes_per_day)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tcga_luad$cigarettes_per_day
## W = 0.90998, p-value = 9.873e-14
```

```
shapiro.test(tcga_luad$years_smoked)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tcga_luad$years_smoked
## W = 0.98572, p-value = 0.04686
```

This test verifies what was gleaned from the QQ-plot. More information about the anderson-darling test can be found here (http://www.itl.nist.gov/div898/handbook/eda/section3/eda35e.htm)

Assignment 2

Create a new R Markdown file for submitting this assignment. You should submit both your knit file in HTML and the original Rmd file to the Week 1 homework submission link in CANVAS. This homework is due 1/29/18 at 11:59 PM (evening).

Question 1: Reading assignment question

Question 2

Question 3

Question 4