Biostatistics-Lecture3: Hypothesis testing

Ruibin Xi

2/16/2020

We first load a few libraries.

```
library(ggplot2)
library(ggridges)
library(tidyverse)
```

加载包

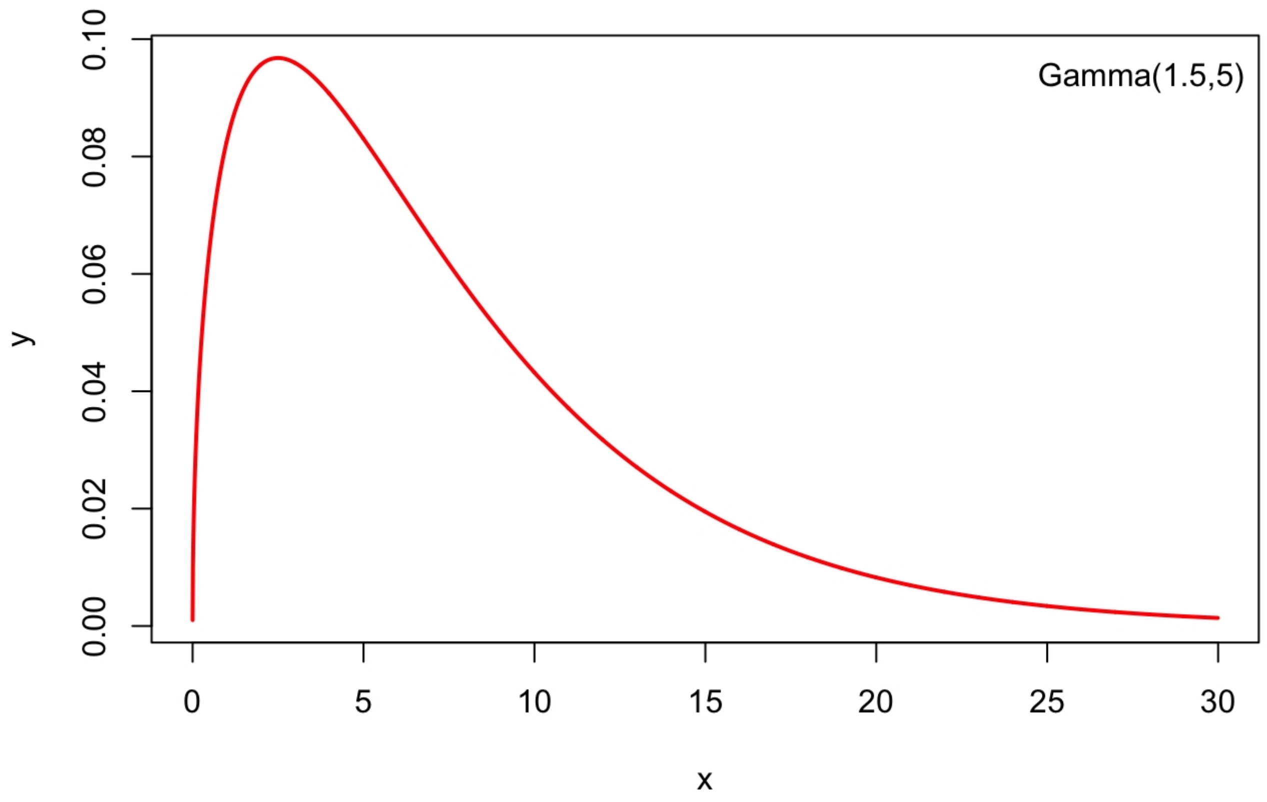
Central Limit Theorem

中心极限定理

We look at the effect of the central limit theorem using simulation. The following gamma distribution is far away from the normal distribution.

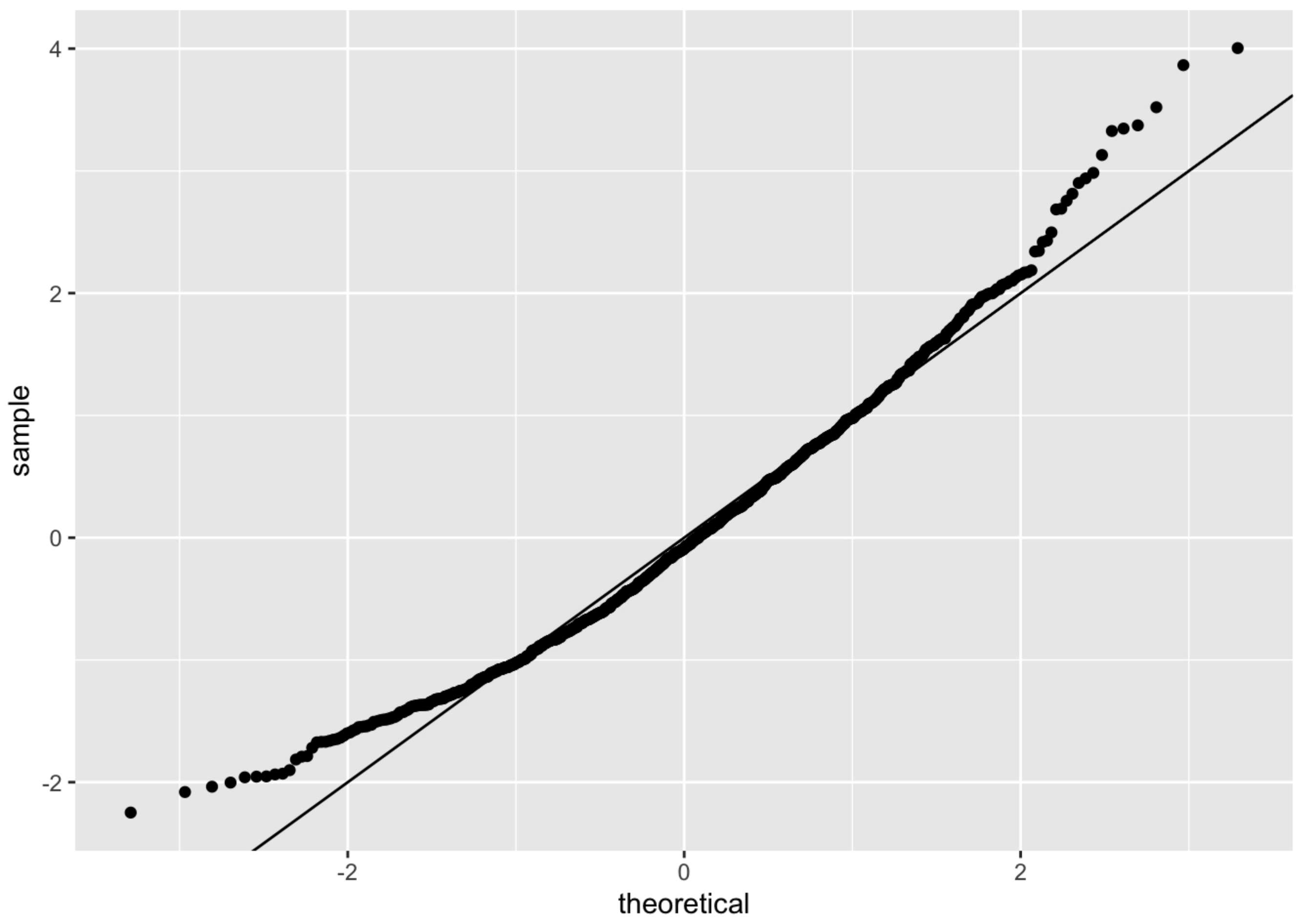
```
x = seq(0.0001,30,by=0.01)
y = dgamma(x,shape=1.5,scale=5)
plot(x,y,type="l",col="red",lwd=2)
legend("topright",legend="Gamma(1.5,5)",bty="n")
```

gama分布情况



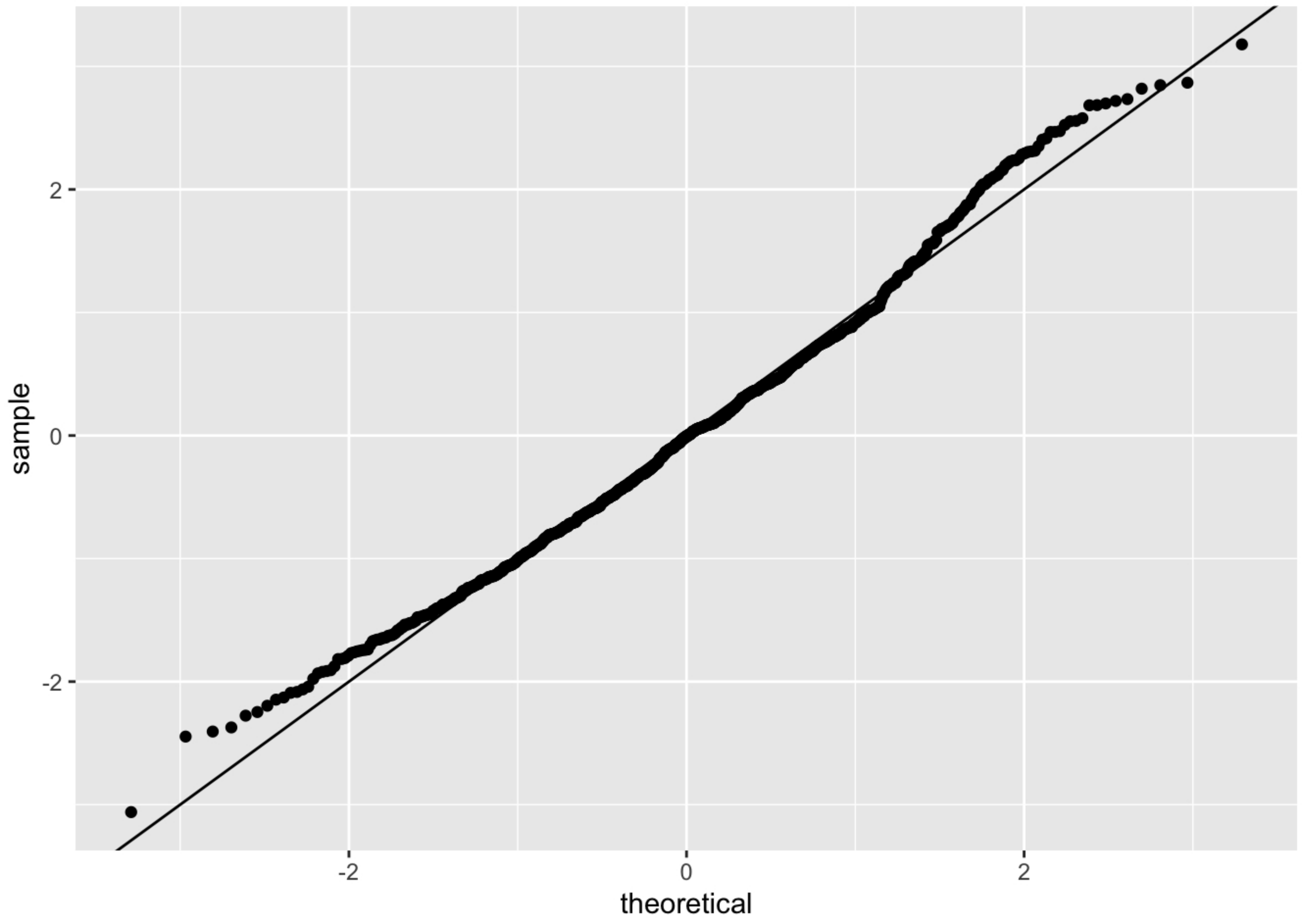
When the sample size is small, the sample mean is still far away from the normal distribution.

```
n = 5          样本量 5
x = c()
for(i in 1:1000){
    x[i] = mean(rgamma(n,shape=1.5,scale=5))      重复1000次，从gama分布里取出5个数，算出均值并存到x里面
}
x = data.frame(x)
x %>% ggplot(aes(sample=scale(x))) + geom_qq() + geom_abline()      看这些均值是否符合正态分布
```



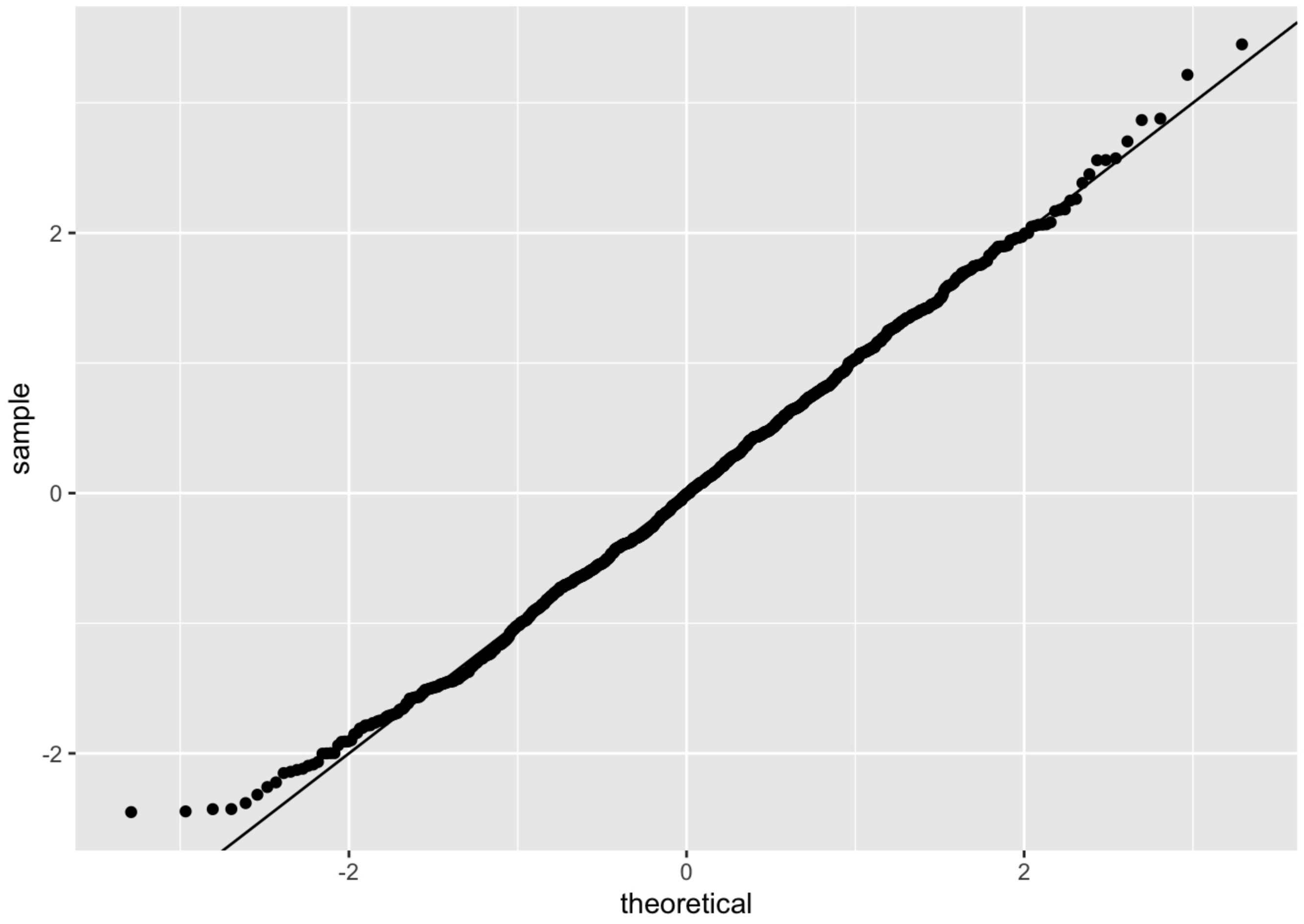
When n = 20, normal approximation is better.

```
n = 20
x = c()
for(i in 1:1000){
    x[i] = mean(rgamma(n,shape=1.5,scale=5))
}
x = data.frame(x)
x %>% ggplot(aes(sample=scale(x))) + geom_qq() + geom_abline()
```



When n is up to 100, the normal approximation is much better.

```
n = 100
x = c()
for(i in 1:1000){
    x[i] = mean(rgamma(n,shape=1.5,scale=5))
}
x = data.frame(x)
x %>% ggplot(aes(sample=scale(x))) + geom_qq() + geom_abline()
```



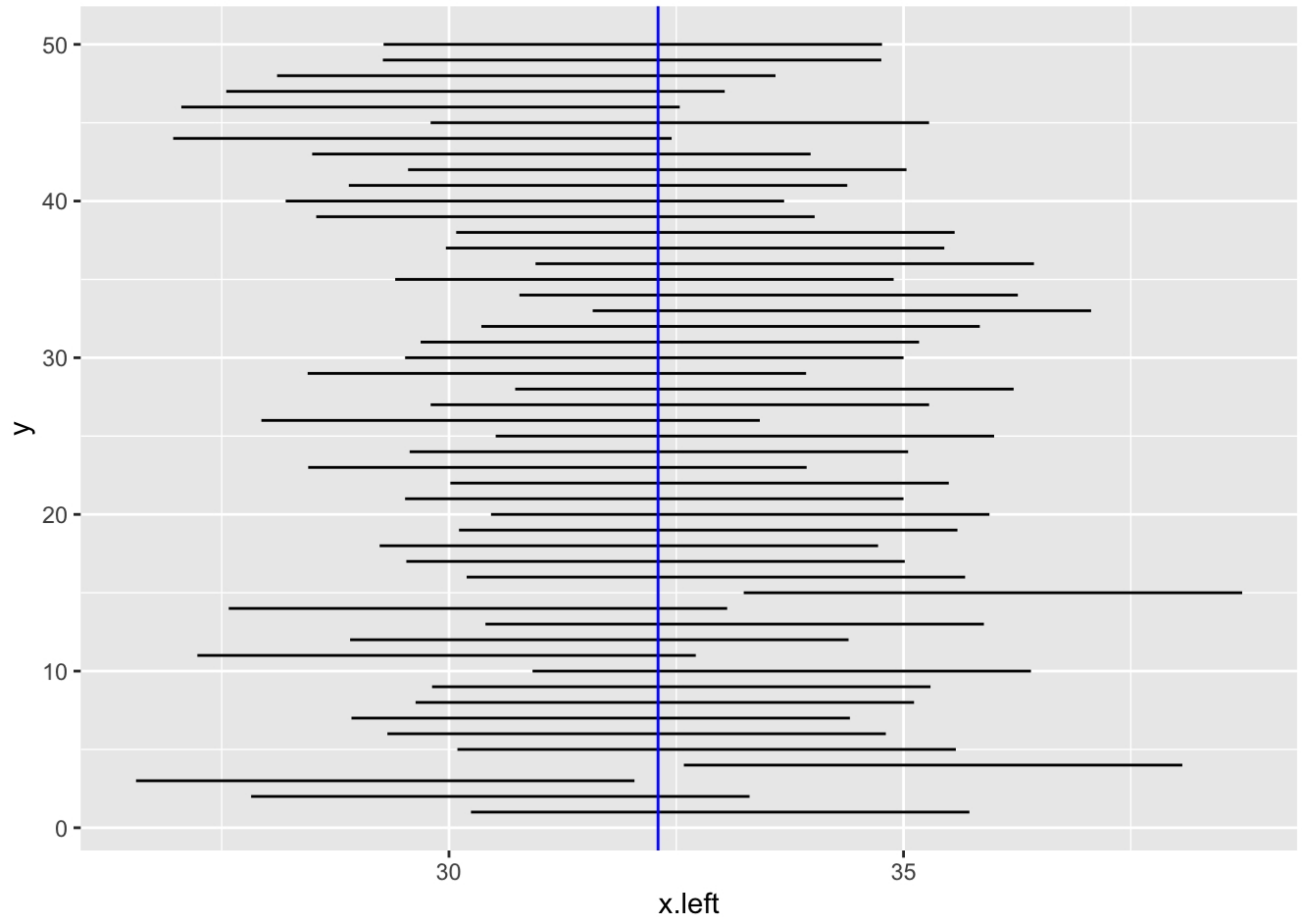
The confidence interval

```
mu = 32.3
sd = 6.13
n = 20
m.max = 50
set.seed(37823)
bar.x = c()
for(i in 1:m.max){
    x = rnorm(n,mean=mu,sd=sd)
    bar.x = c(bar.x,mean(x))
}

dta.conf = data.frame(x.left=bar.x-2*sd/sqrt(n),x.right=bar.x+2*sd/sqrt(n),y=1:m.max)
dta.conf %>% ggplot(aes(x=x.left,xend=x.right,y=y,yend=y)) + geom_segment() + geom_vline(xintercept=mu,col="blue")
```

样本量20，重复实验50次

set.seed 规定起始点，确保每次运算都一样



黑线 计算得到的置信区间
蓝线 真值mu

In general, we should use the following code to calculate the confidence interval based on normal assumption. We use the BMI data in the MASS package as an exmaple.

```
library(MASS)
data(Pima.tr)
n = nrow(Pima.tr)
params = Pima.tr %>% summarise(mean= mean(bmi),sd = sd(bmi))
alpha = 0.05
err = qnorm(1-alpha/2)*params$sd/sqrt(n)
c(params$mean-err,params$mean+err)
```

qnorm 算标准正态分布的分位数

计算置信区间

```
## [1] 31.46041 33.15959
```

The proportion test uses normal approximation. Here we consider the TERT mutation data (http://www.sciencemag.org/content/339/6122/957.full).

```
x = 50
n = 70
p = 50/70
alpha = 0.05
SE = sqrt(p*(1-p)/n)
CI = c(p-qnorm(1-alpha/2)*SE,p+qnorm(1-alpha/2)*SE)
```

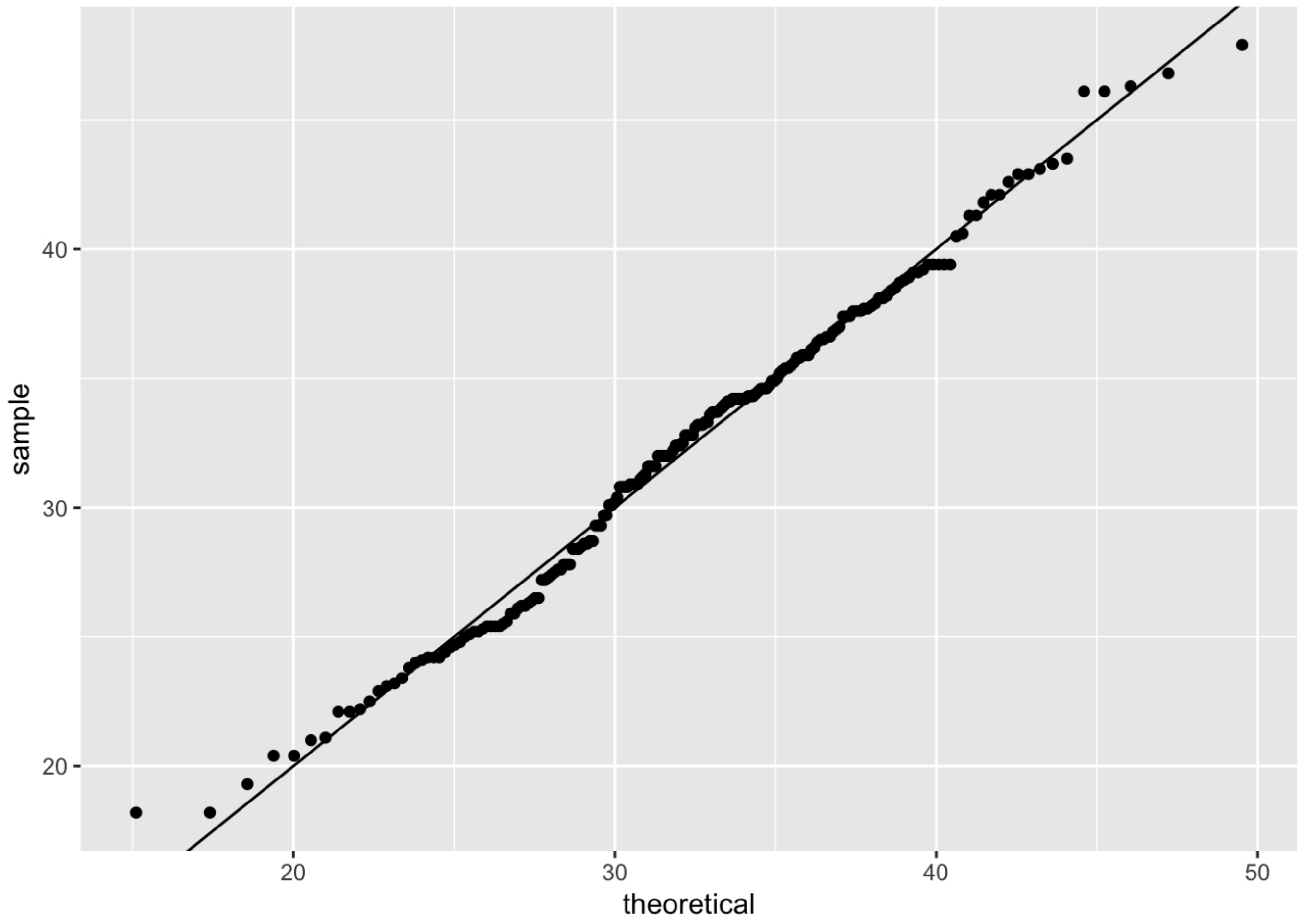
If using t-statistics, we could use the following code.

```
err <- qt(1-alpha/2,df=n-1)*params$sd/sqrt(n)
c(params$mean-err,params$mean+err)
```

```
## [1] 30.8483 33.7717
```

We need to check if the BMI data largely follows a normal distribution.

```
library(gridExtra)
xp1 <- Pima.tr %>% ggplot(aes(sample=bmi)) + geom_qq(dparams=params) + geom_abline()
xp1
```



```
#xp2 <- Pima.tr %>% ggplot(aes(bmi)) + geom_density()
#m1 <- grid.arrange(xp1, xp2, ncol = 1, nrow = 2)
```

We could also use the so called Shapiro’s test to test the normality.

```
shapiro.test(Pima.tr$bmi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Pima.tr$bmi
## W = 0.99104, p-value = 0.2523
```

Hypothesis testing

I use Aspirin data to show the chi-squre test.

```
x = matrix(c(104,10933,189,10845),nrow=2)
colnames(x) = c("aspirin","placebo")
rownames(x) = c("HeartAttack","NoHeartAttack")
chisq.test(x)
```

默认双边检验，alternative 可以
改设定为单边检验

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 24.429, df = 1, p-value = 7.71e-07
```

We use the BMI data to show the t-test.

```
women.bmi = 26.5
n = nrow(Pima.tr)
alpha = 0.05
c.cri = qt(1-alpha/2,df=n-1)
t.value = (mean(Pima.tr$bmi)-women.bmi)/(sd(Pima.tr$bmi)/sqrt(n))
t.value
```

单样本 t- test

```
## [1] 13.40342
```

t-test 统计量

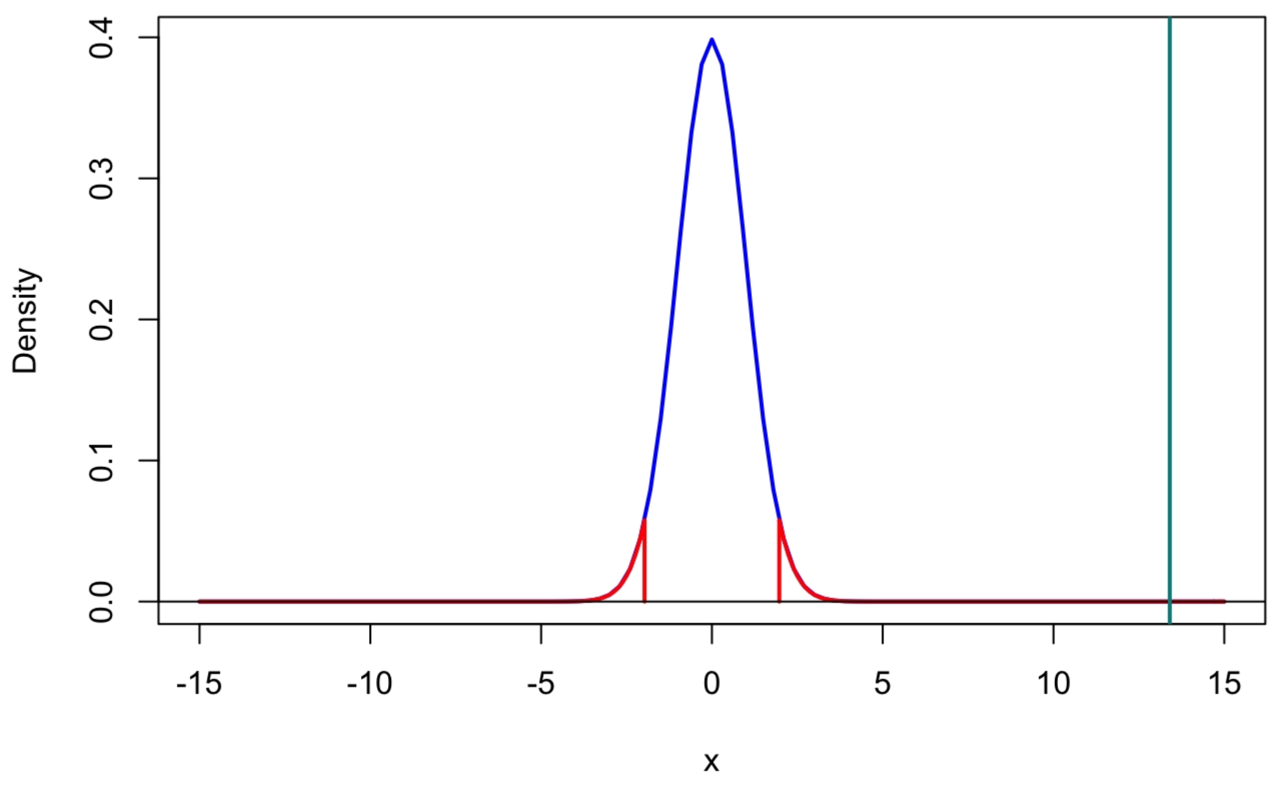
```
pvalue = 2*pt(t.value,lower.tail = FALSE,df=n-1)
pvalue
```

pt：possibility of t-test 算双边检验的p-value

```
## [1] 1.295754e-29
```

Show the rejection region and the t-statistic.

```
curve(dt(x,df=n-1),xlim=c(-15,15),col="blue",lwd=2,ylab="Density")
curve(dt(x,df=n-1),xlim=c(-15,-c.cri),col="red",lwd=2,add=T)
curve(dt(x,df=n-1),xlim=c(c.cri,15),col="red",lwd=2,add=T)
lines(rep(-c.cri,2),c(0,dt(-c.cri,df=n-1)),col="red",lwd=2)
lines(rep(c.cri,2),c(0,dt(c.cri,df=n-1)),col="red",lwd=2)
abline(h=0)
abline(v=t.value,col="darkcyan",lwd=2)
```



We now use the mice2 data in the datarium package to show paired t-test.

```
data("mice2", package = "datarium")
head(mice2, 3)
```

paired-sample 成对样本 t-test

```
##   id before after
## 1  1  187.2 429.5
## 2  2  194.2 404.4
## 3  3  231.7 405.6
```

目的
1. 按组去看均值
2. 按样本ID去看均值
看处理前和处理后的区别：after-before

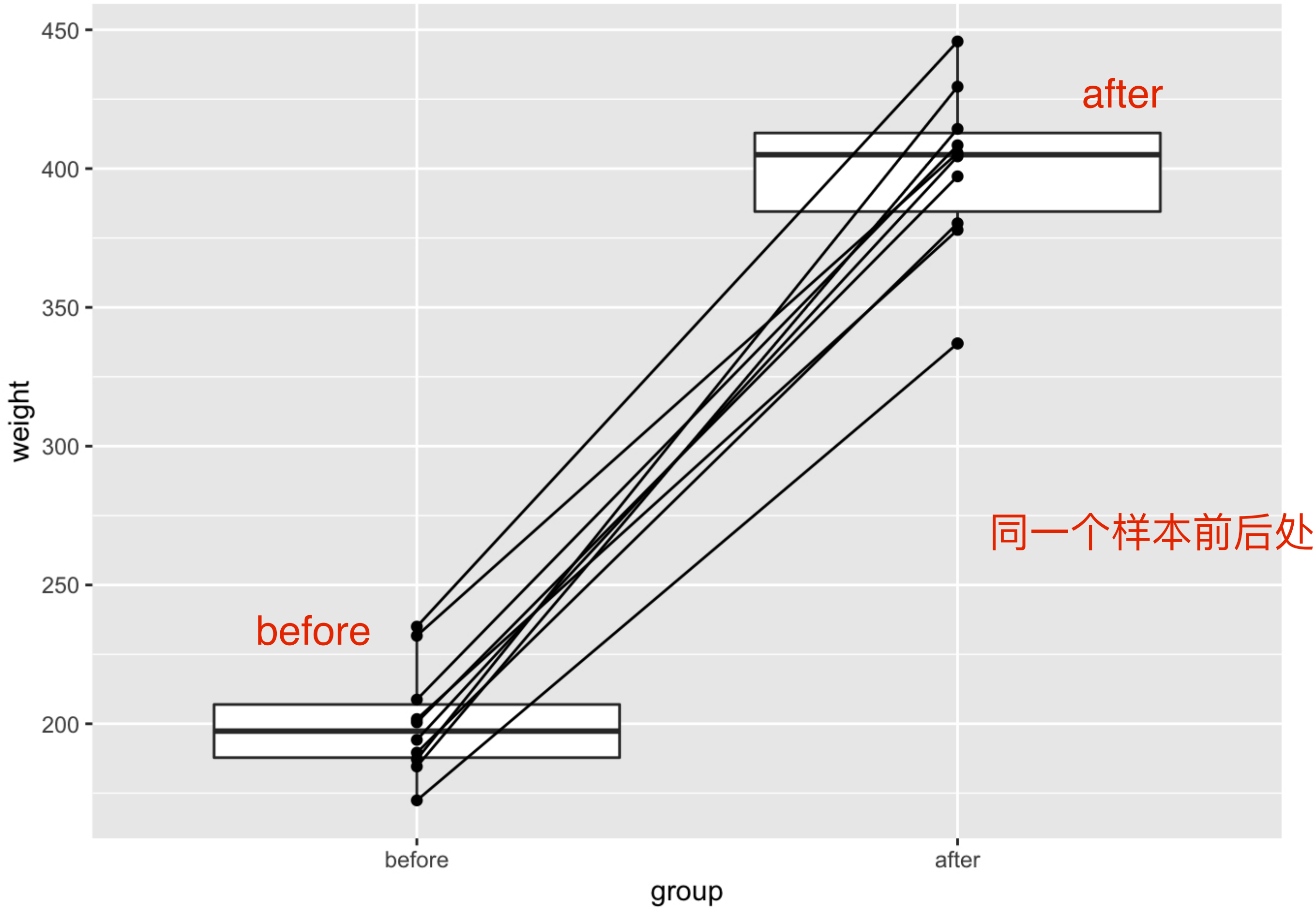
The data is in wide format. We transform it to long data. Gather the before and after values in the same column.

```
mice2.long <- mice2 %>%
  gather(key = "group", value = "weight", before, after)
head(mice2.long, 3)
```

```
##   id  group weight
## 1  1 before  187.2
## 2  2 before  194.2
## 3   3 before  231.7
```

Visualization.

```
bp <- mice2.long %>% as.tibble() %>% mutate(group.new=factor(group,levels=c("before","after"))) %>% ggplot(aes(group.new,weight))+geom_boxplot()+xlab("group"
bp1 <- bp +   geom_segment(data=mice2,mapping=aes(x=1,y=before,xend=2,yend=after)) + geom_point(data=mice2,aes(x=1,y=before)) + geom_point(data=mice2,aes(x=2
bp1
```



after

before

同一个样本前后处理组连线

The paired t-test

```
t.test(mice2$after,mice2$before,alternative="greater",paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  mice2$after and mice2$before
## t = 25.546, df = 9, p-value = 5.195e-10
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  185.166     Inf
## sample estimates:
## mean of the differences
##                  199.48
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.