Week 3

Shu Guo

Thursday, February 05, 2015

Population, Samples, and Estimates

Introduction

we are now ready to describe the mathematical theory that permits us to compute p-values in practice. We will also learn about confidence intervals and power calculations.

Population parameters

In the mouse weight example, we have two populations; female mice on control diet and female mice on high fat diet, and the outcome of interest was weight. We consider this population to be fixed, and the randomness comes from the sampling. One reason we have been using this dataset as an example is because we happen to have the weights of all the mice of this type. Read the data:

We can then access the population values and determine, for example, how many we have. Here is the control population:

```
controlPopulation <- dat[dat$Sex == "F" & dat$Diet == "chow", 3]
length(controlPopulation)
```

```
## [1] 225
```

Denote these values as $x_1, \ldots, x_m$. In this case $m = 225$. Now we can do the same with the high fat diet population

```
hfPopulation <- dat[dat$Sex == "F" & dat$Diet == "hf", 3]
length(hfPopulation)
```

```
## [1] 200
```

and denote with $y_1, \ldots, y_n, n = 200$.

Define summaries of interest for these population such as the mean and variance.

the mean:

$$\mu_X = \frac{1}{m}\sum_{i=1}^{m} x_i \text{ and } \mu_Y = \frac{1}{n}\sum_{i=1}^{n} y_i$$

the variance:

$$\sigma_X^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_x)^2 \text{ and } \sigma_Y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^2$$

with the standard deviation being the square root of the variance. We refer to such quantities, that can be obtained from the population, as *population parameters*.

The question we started out asking can now be written mathematically: $\mu_Y - \mu_X = 0$? We take a sample and try to answer the questions with the sample. This is the essence of statistical inference.

Sample estimates

In the previous section, we obtained samples of 12 mice from each population. We represent these with capital letters to indicate that they are random. This is common practice in statistics, although it is not always followed. So the samples are $X_1, \ldots, X_M$ and $Y_1, \ldots, Y_N$ and in this case $N = M = 12$. Since we want to know what $\mu_Y - \mu_X$ is we consider the sample version: $\bar{Y} - \bar{X}$ with

$$\bar{X} = \frac{1}{M}\sum_{i=1}^{M} X_i \text{ and } \bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

Here we described the mathematical theory that mathematically relates $\bar{X}$ to μ_X and $\bar{Y}$ to μ_Y, which will in turn help us understand the relationship between $\bar{Y} - \bar{X}$ and $\mu_Y - \mu_X$.

Central Limit Theorem in practice

We can compute the population parameters of interest using the mean function.

```
mu_hf <- mean(hfPopulation)
mu_control <- mean(controlPopulation)
print(mu_hf - mu_control)
```

```
## [1] 2.375517
```

Compute the population standard deviations as well. Note that we do not use the R function `sd` because this is to compute the population based estimates that divide by the sample size - 1.

```
x <- controlPopulation
N <- length(x)
popvar <- mean((x - mean(x))^2)
identical(var(x), popvar)
```

```
## [1] FALSE
```

```
identical(var(x)*(N-1)/N, popvar)
```

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

Define a function:

```
popvar <- function(x) mean((x - mean(x))^2)
popsd <- function(x) sqrt(popvar(x))
```

Now compute the population SD:

```
sd_hf <- popsd(hfPopulation)
sd_control <- popsd(controlPopulation)
```

In general, we want to estimate these population parameters from samples.

```
N <- 12
hf <- sample(hfPopulation, 12)
control <- sample(controlPopulation, 12)
```

The CLT tells us that, for large N, each of these is approximately normal with average population mean and standard error population variance divided by N. We mentioned that a rule of thumb is that N should be 30 or more. Here we can acually check the approximation and we do that for various values of N.

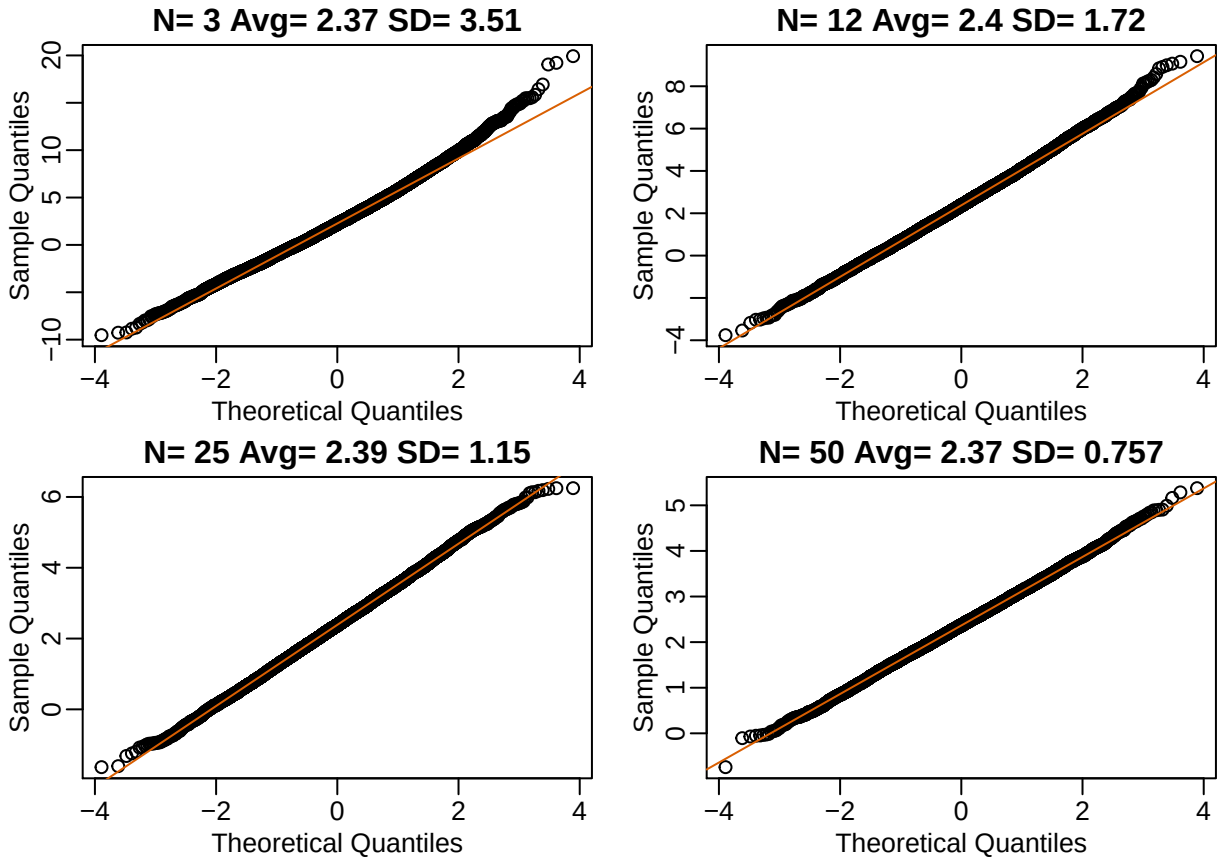
```
Ns <- c(3,12,25,50)
B <- 10000 #number of simulations
res <-  sapply(Ns,function(n){
  replicate(B,mean(sample(hfPopulation,n))-
      mean(sample(controlPopulation,n)))
})
```

Now we can use qq-plots to see how well CLT approximations works for these. If in fact the normal distribution is a good approximation the points should fall on a straight line when compared to normal quantiles. The more it deviates, the worse the approximation. We also show, in the title, the average and SD of the observed distribution showing how the SD decreases with $\sqrt{N}$ as predicted.

```
library(rafalib)
```

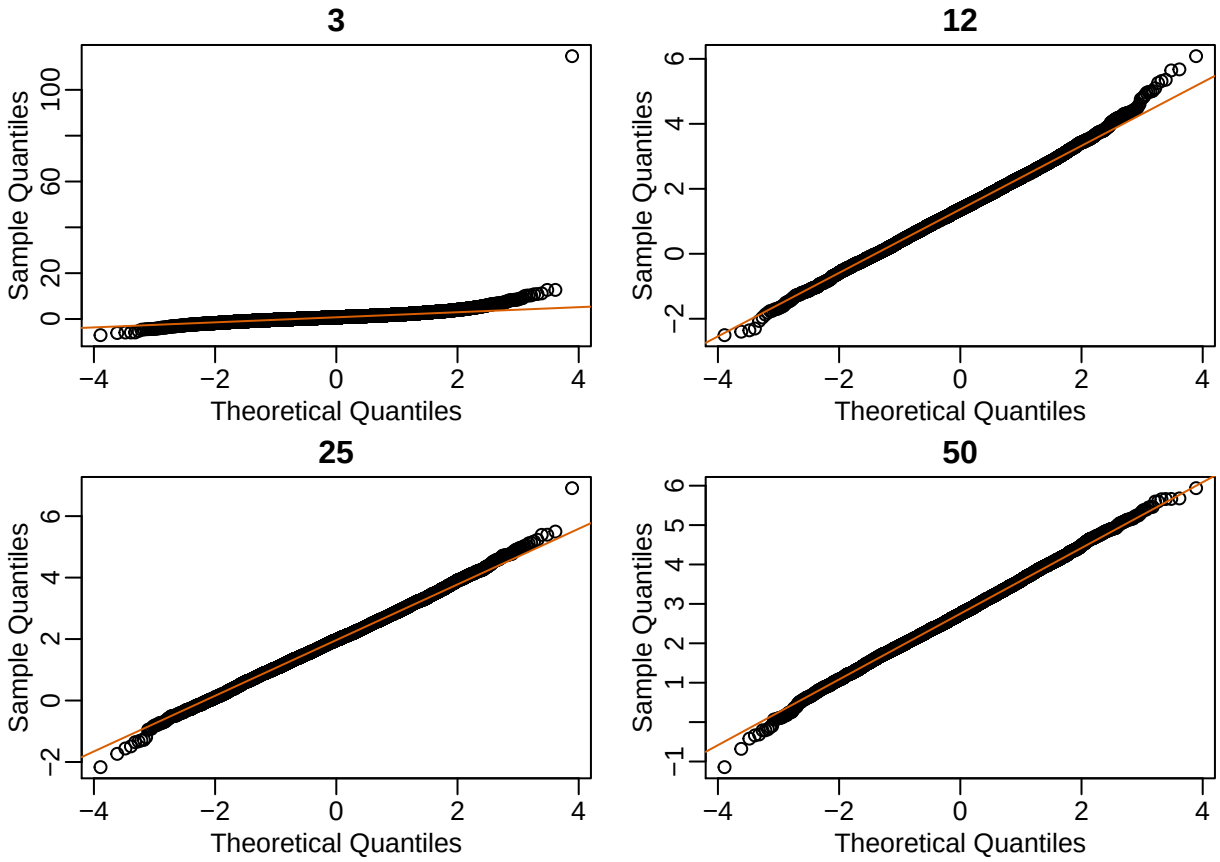
```
## Loading required package: RColorBrewer
```

```
mypar2(2,2)
for(i in seq(along=Ns)){
  title <- paste("N=",Ns[i],"Avg=",signif(mean(res[,i]),3),"SD=",
        signif(popsd(res[,i]),3)) ##popsd defined above
  qqnorm(res[,i],main=title)
  qqline(res[,i],col=2)
}
```



Here we see a pretty good fit even for 3. Why is this? Because the population itself is relatively close to normally distributed, the averages are close to normal as well, (the sum of normals is normals). Now in practice we actually calculate a ratio, we divide by the estimate standard deviation. Here is where the sample size starts to matter more.

```
Ns <- c(3,12,25,50)
B <- 10000 #number of simulations
##function to compute a t-stat
computetstat <- function(n){
  y <-sample(hfPopulation,n)
  x <-sample(controlPopulation,n)
  (mean(y)-mean(x))/sqrt(var(y)/n+var(x)/n)
}
res <-  sapply(Ns,function(n){
  replicate(B,computetstat(n))
})
mypar2(2,2)
for(i in seq(along=Ns)){
  qqnorm(res[,i],main=Ns[i])
  qqline(res[,i],col=2)
}
```



Now we see that for $N = 3$ the CLT does not provide a usable approximation. For $N = 12$ their is a slight deviation at the higher values, although the approximation appears useful. For 25 and 50 the appoximation is spot on. Note that this simulation is not meant as proof that $N = 12$ is large enough, in general. It only applies to this dataset and, as mentioned above, we will not be able to perform this simulation in most situation.