Lab 5: Sampling Distributions
STAT 630, Fall 2021

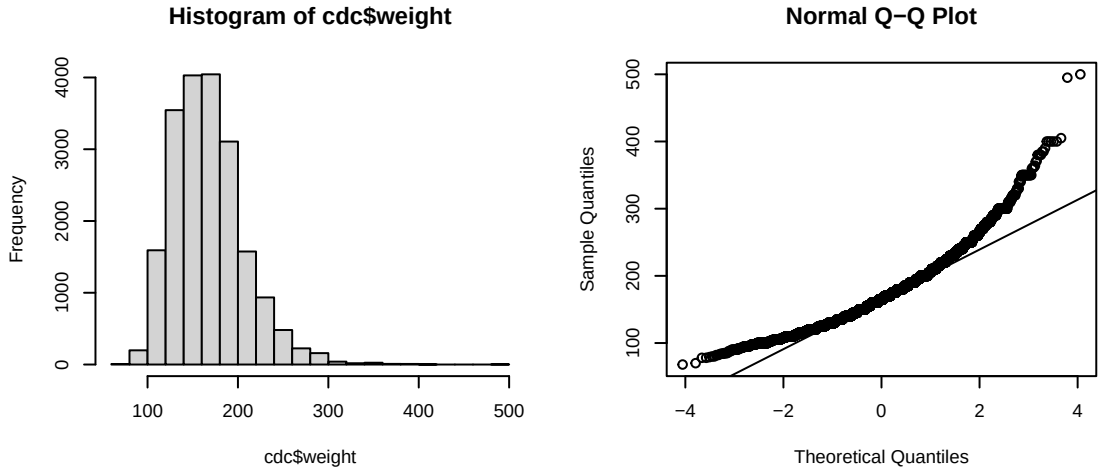
Sampling Distributions using CDC Data

As a demonstration of sampling distributions, we again work with the Behavioral Risk Factor Surveillance System (BRFSS) data from the CDC. We can think of the 20000 individuals in this data set as our hypothetical "population". The R function `sample()` can be used to take samples of individuals from this population.

```
# read data set into R
cdc <- readRDS(url("https://ericwfox.github.io/data/cdc.rds"))
```

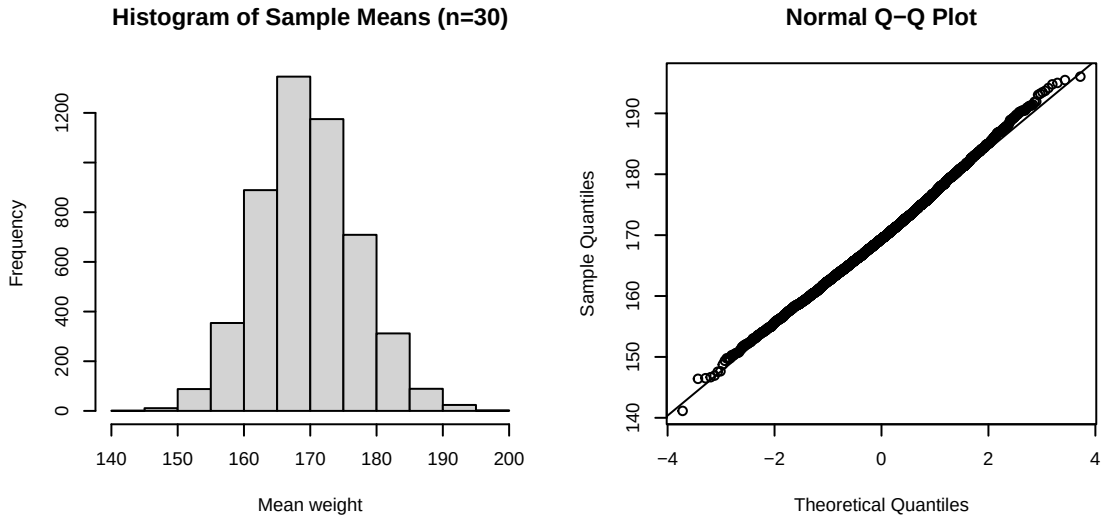
Let's start by computing summary statistics and plotting the histogram and QQ plot for the weights of the 20000 individuals in the data set.

```
summary(cdc$weight)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    68.0   140.0   165.0   169.7   190.0   500.0

par(mfrow=c(1,2), cex=0.6)
hist(cdc$weight)

qqnorm(cdc$weight)
qqline(cdc$weight)
```



Next, use R to draw 5000 samples of size $n = 30$ from this population of 20000 weights of individuals. Compute the mean and standard deviation of the weights in each sample. Then make a histogram and QQ plot of the 5000 sample means.

```r
set.seed(999)
xbars <- rep(0, 5000)
for(i in 1:5000) {
  samp <- sample(cdc$weight, 30)
  xbars[i] <- mean(samp)
}

par(mfrow=c(1,2), cex=0.6)
hist(xbars, xlab="Mean weight", main="Histogram of Sample Means (n=30)")
qqnorm(xbars)
qqline(xbars)
```



By the central limit theorem, for large n (usually $n \geq 30$) the sample mean approximately follows a normal distribution centered around the mean of the population distribution μ and with standard deviation $\sigma/\sqrt{n}$, where σ is the standard deviation of the population distribution and n is the sample size.

Does the CLT seem to hold true for the sampling distribution generated from the CDC data set? Yes – both the QQ plot and histogram indicate that the sample means approximately follow a normal distribution. We also have the following results for the mean and standard deviation (also called the standard error) of the sampling distribution:

```r
mean(xbars)
```

```
## [1] 169.648
```

```r
mean(cdc$weight) # population mean mu
```

```
## [1] 169.683
```

```r
sd(xbars)
```

```
## [1] 7.368845
```

```r
sd(cdc$weight) / sqrt(30) # sigma / sqrt(n), where sigma is population st dev
```

```
## [1] 7.31775
```

The `for` loop

The `for` loop allows us to execute code as many times as we want without having to type out every iteration. It is one of the most important constructs in computer science; every programming language has it's own version of the `for` loop. While you may occasionally find a need for other types of loops (e.g., `while` loops), in my experience, I've found very few situations where a `for` loop wasn't sufficient.

```r
# it's obviously tedious to generate a sampling distribution without using a loop:
set.seed(999)
samp1 <- sample(cdc$weight, 30)
mean(samp1)
```

```
## [1] 180.8333
```

```r
samp2 <- sample(cdc$weight, 30)
mean(samp2)
```

```
## [1] 172
```

```r
samp3 <- sample(cdc$weight, 30)
mean(samp3)
```

```
## [1] 170.4
```

```r
samp4 <- sample(cdc$weight, 30)
mean(samp4)
```

```
## [1] 162.2667
```

```r
set.seed(999)
xbars <- rep(0, 4)
for(i in 1:4) {
  samp <- sample(cdc$weight, 30)
  xbars[i] <- mean(samp)
}
xbars
```

```
## [1] 180.8333 172.0000 170.4000 162.2667
```

```r
# this loop prints out the variable i at each iteration
for(i in 1:10) {
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```r
# some other examples demonstrating the behavior of loops in R:
x <- c("a", "b", "c", "d")
for(i in 1:4) {
  print(x[i])
}
```

```
## [1] "a"
## [1] "b"
## [1] "c"
## [1] "d"
```

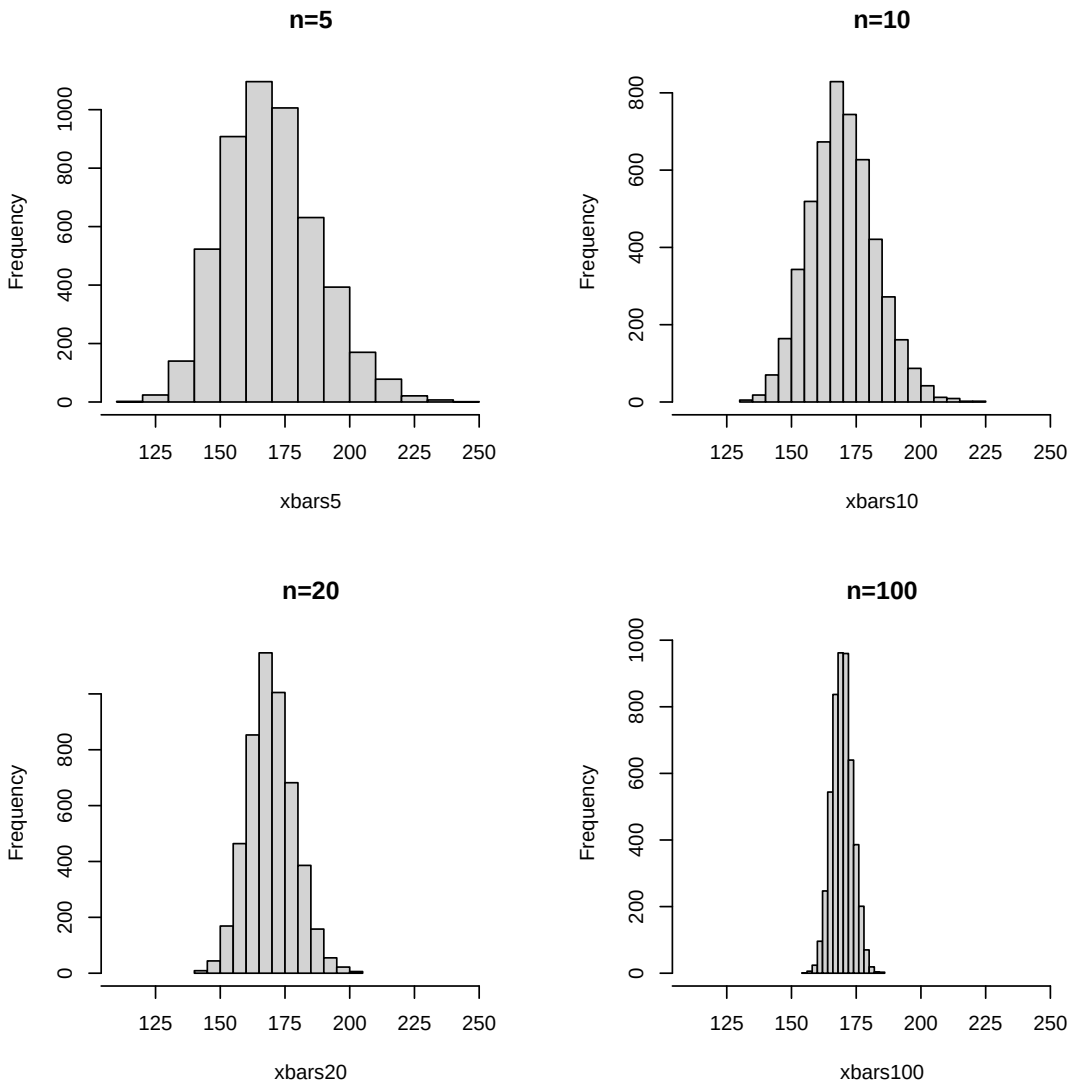
```r
x <- c(5, 12, 13)
for(n in x) {
  print(n^2)
}
```

```
## [1] 25
## [1] 144
## [1] 169
```

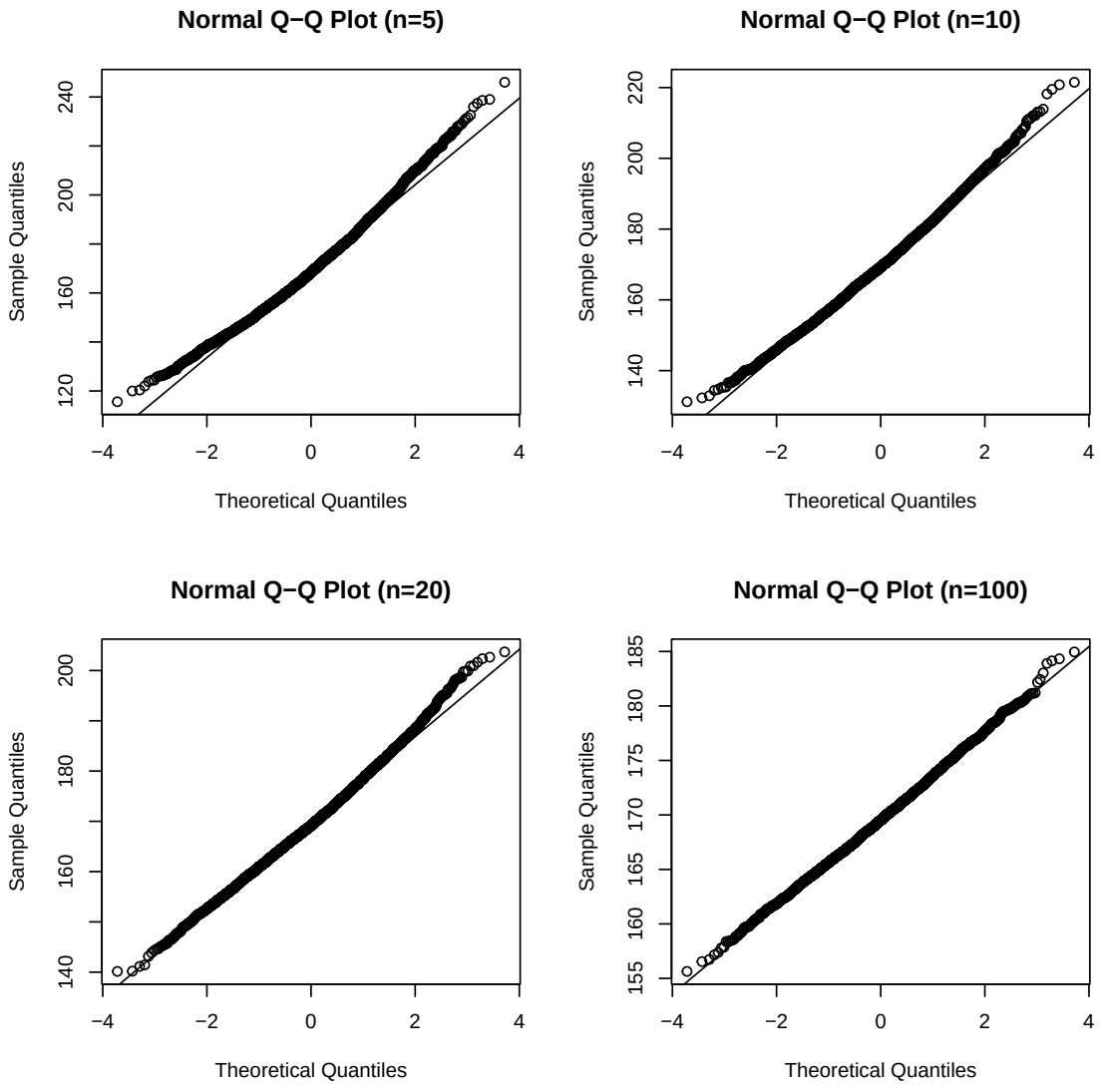
Question: Going back to the CDC data set example, how does varying the sample size affect the sampling distribution for the mean weights?

To answer this question, we use R to draw 5000 samples of size $n = 5, 10, 20, 100$ from the population of 20000 weights of individuals in the CDC data set. Again, we compute the mean of the weights in each sample; and then make histograms and QQ plots of the 5000 sample means generated for each sample size.

```
set.seed(999)
xbars5 <- rep(0, 5000)
xbars10 <- rep(0, 5000)
xbars20 <- rep(0, 5000)
xbars100 <- rep(0, 5000)
for(i in 1:5000) {
  samp5 <- sample(cdc$weight, 5)
  xbars5[i] <- mean(samp5)

  samp10 <- sample(cdc$weight, 10)
  xbars10[i] <- mean(samp10)

  samp20 <- sample(cdc$weight, 20)
  xbars20[i] <- mean(samp20)

  samp100 <- sample(cdc$weight, 100)
  xbars100[i] <- mean(samp100)
}

par(mfrow=c(2,2), cex=0.75)
hist(xbars5, xlim=c(110, 260), xaxt="n", main = "n=5")
axis(side=1, at=seq(100, 250, 25), labels=seq(100, 250, 25))
hist(xbars10, xlim=c(110, 260), xaxt="n", main = "n=10")
axis(side=1, at=seq(100, 250, 25), labels=seq(100, 250, 25))
hist(xbars20, xlim=c(110, 260), xaxt="n", main = "n=20")
axis(side=1, at=seq(100, 250, 25), labels=seq(100, 250, 25))
hist(xbars100, xlim=c(110, 260), xaxt="n", main = "n=100")
axis(side=1, at=seq(100, 250, 25), labels=seq(100, 250, 25))
```


n=5
Frequency
0
200
400
600
800
1000
125
150
175
200
225
250
xbars5
n=10
Frequency
0
200
400
600
800
125
150
175
200
225
250
xbars10
n=20
Frequency
0
200
400
600
800
125
150
175
200
225
250
xbars20
n=100
Frequency
0
200
400
600
800
1000
125
150
175
200
225
250
xbars100

```r
par(mfrow=c(2,2), cex=0.75)
qqnorm(xbars5, main="Normal Q-Q Plot (n=5)")
qqline(xbars5)
qqnorm(xbars10, main="Normal Q-Q Plot (n=10)")
qqline(xbars10)
qqnorm(xbars20, main="Normal Q-Q Plot (n=20)")
qqline(xbars20)
qqnorm(xbars100, main="Normal Q-Q Plot (n=100)")
qqline(xbars100)
```



```r
mean(cdc$weight) # population mean
```

```
## [1] 169.683
```

```r
# means of each sampling distribution
mean(xbars5)
```

```
## [1] 169.6472
```

```r
mean(xbars10)
```

```
## [1] 169.8427
```

```r
mean(xbars20)
```

```
## [1] 169.6234
```

```r
mean(xbars100)
```

```
## [1] 169.5839
```

```r
# standard deviations of each sampling distribution
sd(xbars5)
```

```
## [1] 18.00604
```

```r
sd(cdc$weight) / sqrt(5)
```

```
## [1] 17.92475
```

```r
sd(xbars10)
```

```
## [1] 12.71319
```

```r
sd(cdc$weight) / sqrt(10)
```

```
## [1] 12.67472
```

```r
sd(xbars20)
```

```
## [1] 8.934549
```

```r
sd(cdc$weight) / sqrt(20)
```

```
## [1] 8.962377
```

```r
sd(xbars100)
```

```
## [1] 3.973856
```

```r
sd(cdc$weight) / sqrt(100)
```

```
## [1] 4.008097
```

Remark: We can also generate a sampling distribution for other statistics such as the median.

```
set.seed(999)
meds <- rep(0, 5000)
for(i in 1:5000) {
  samp <- sample(cdc$weight, 30)
  meds[i] <- median(samp)
}

par(mfrow=c(1,2), cex=0.6)
hist(meds, main="Histogram of Sample Medians (n=30)")
qqnorm(meds)
qqline(meds)
```