Lab 1

Stat 135

Nicholas Lai

Part 1

1a

First, Housekeeping:

```
load("KaiserBabies.rda")
#Setting the sample size
n <- 10
set.seed(7)
```

Population size calculations:

```
#Extracting the population size, removing na weight vals
total_obs <- nrow(infants)-sum(is.na(infants$wt))
total_obs
```

```
## [1] 1200
```

The sample mean is an unbiased estimator of the population mean, so we will use it to estimate:

```
#take a sample of size 10 from the pop.
mysample <- sample(na.omit(infants$wt), n)

#mean of the sample
sample_avg<- mean(mysample)
sample_avg
```

```
## [1] 134.7
```

The estimator of the population standard deviation is as follows:

```
#estimator of pop. variance
st_error <- (var(mysample)/n )*(1-n/total_obs)
#sqrt to find estimator of st. dev
st_error <- sqrt(st_error)
st_error
```

```
## [1] 4.89217
```

At the 95% confidence level, the critical values are plus and minus 1.96. Therefore, the boundaries of the interval are:

```
sample_avg+1.96*st_error
```

```
## [1] 144.2887
```

```
sample_avg-1.96*st_error
```

```
## [1] 125.1113
```

1b

We expect 950/1000 of 1000 intervals calculated this way to contain the population mean, as that is the definition of 95% confident in this context.

```
#initialize list of sample means
sample_means <- c()
#calculate population average weight of mother
pop_avg <- mean(na.omit(infants$wt))
pop_avg
```

```
## [1] 128.6258
```

```
i <- 1
count <- 0
for (i in 1:1000){
  #calculate a confidence interval like above
  sample <- sample(na.omit(infants$wt), n)
  sample_avg <- mean(sample)
  sample_means[i] <- sample_avg
  st_error <- sqrt((var(sample)/n )*(1-n/total_obs))
  lower <- sample_avg-1.96*st_error
  upper <- sample_avg+1.96*st_error
  #if pop_avg is between the two numbers, count++
  if (pop_avg<=upper & pop_avg>=lower){
    count <- count+1
  }
}
# observations in emprical confidence intervals
count
```

```
## [1] 885
```

As we can observe, less than 950. So there may be problems with our method.

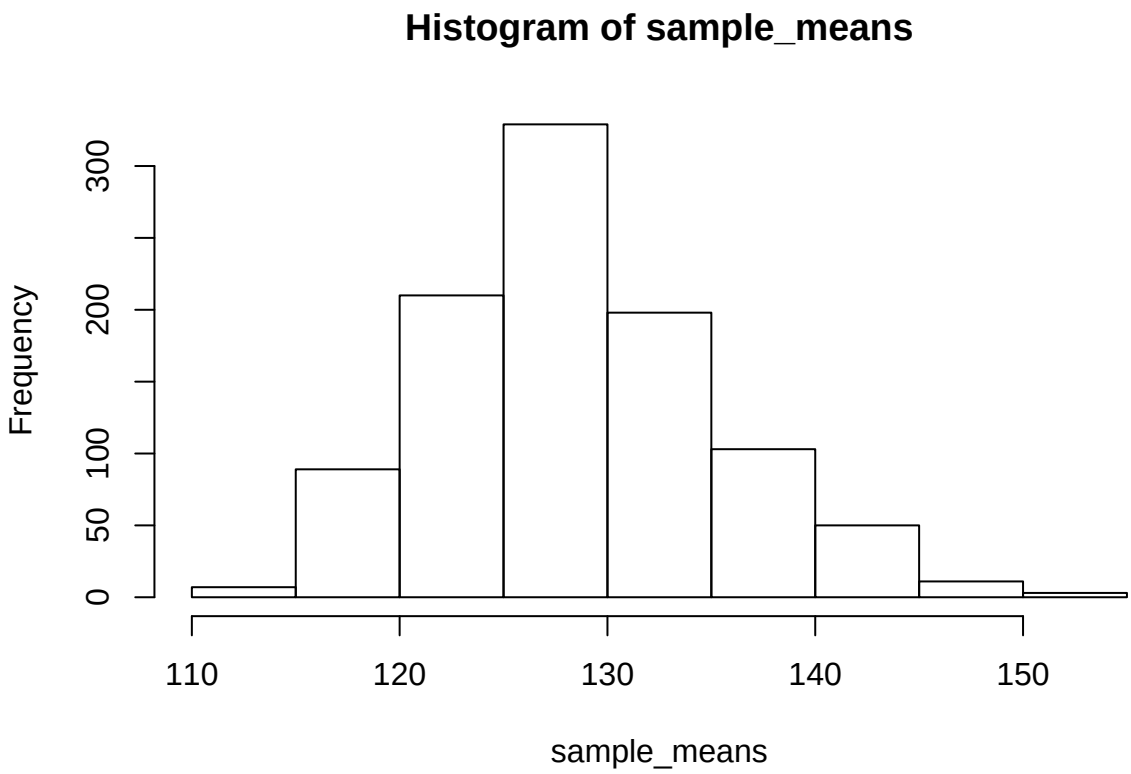
SD of sample averages

```
sd(sample_means)
```

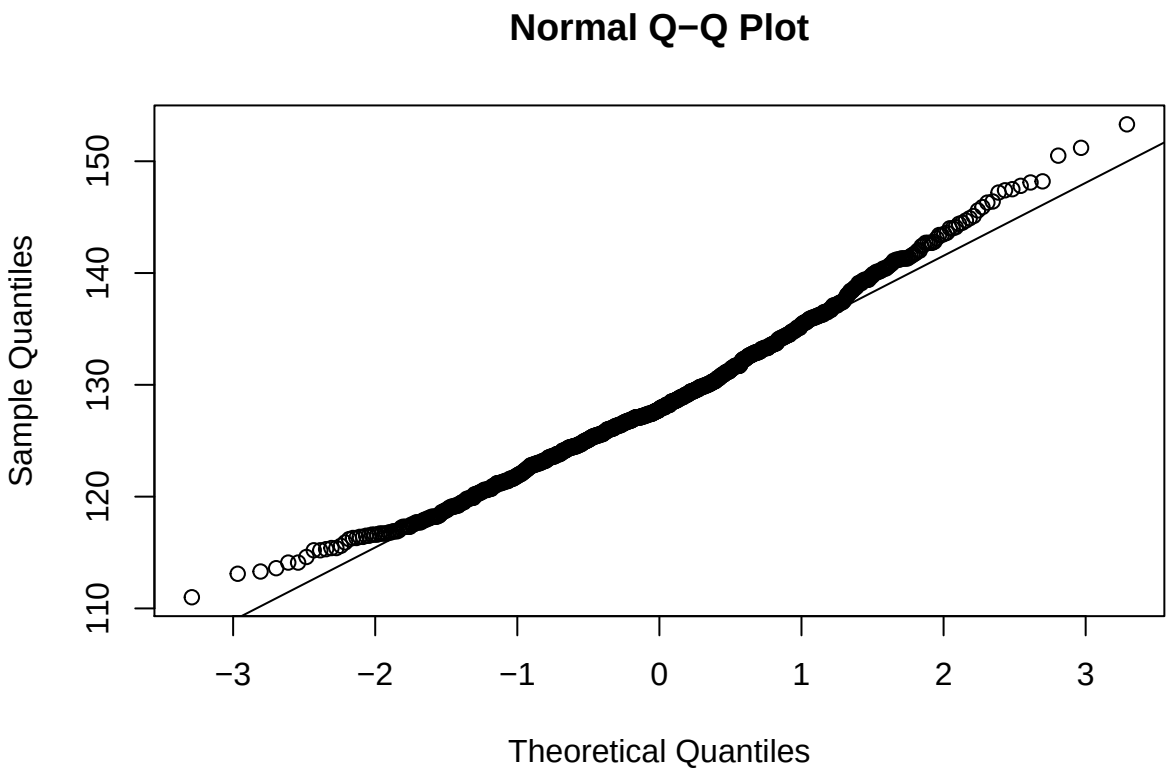
```
## [1] 6.740372
```

St. Dev is higher than the estimate in part 1a. Not good...

```
hist(sample_means)
```




```r
qqnorm(sample_means)
qqline(sample_means)
```



By the Histogram and the qqplot, we can see that our sample means are left skewed. This breaks the normality assumption in our method (CLT not working well).

Part 2

2a

Bootstrap function (thanks hank)

```
bootStrap = function(mySample, popSize = NULL, B = 1000, repl = FALSE){
  if (repl) {
    # Bootstrap should be done the same way as original sample, usually without rep
    return(replicate(B, mean(sample(mySample, length(mySample), TRUE))))
  } else {
    vals = sort(unique(mySample))
    counts = table(mySample)
    # makes the bootstrap pop as rounded version of sample, not quite right
    bootPop = rep(vals, round(counts * popSize / length(mySample)))
    return(list(bootPop,
                bootSamps = replicate(B,mean(sample(bootPop, length(mySample), FALSE))))
           )
  }
}
```

The function calculates 1000 sample averages by default, so we can just use that:

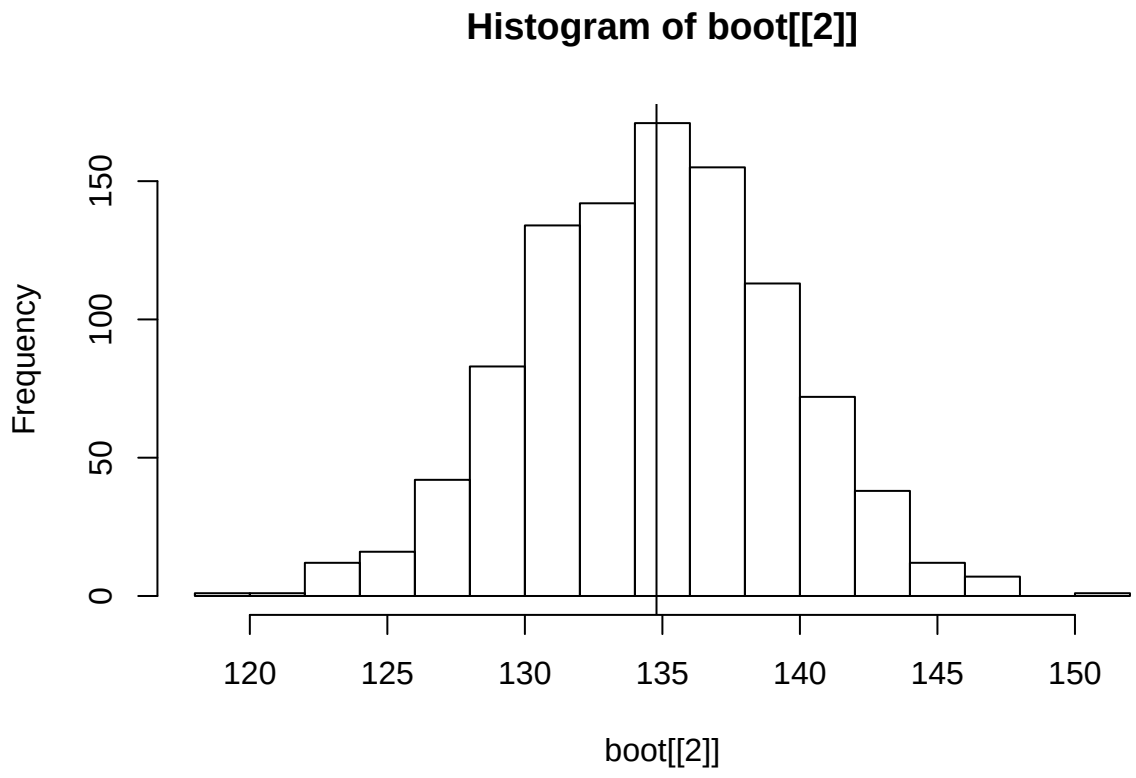
```
boot <- bootStrap(mysample, 1200)
length(boot[[1]])
```

```
## [1] 1200
```

```
#sample averages
length(boot[[2]])
```

```
## [1] 1000
```

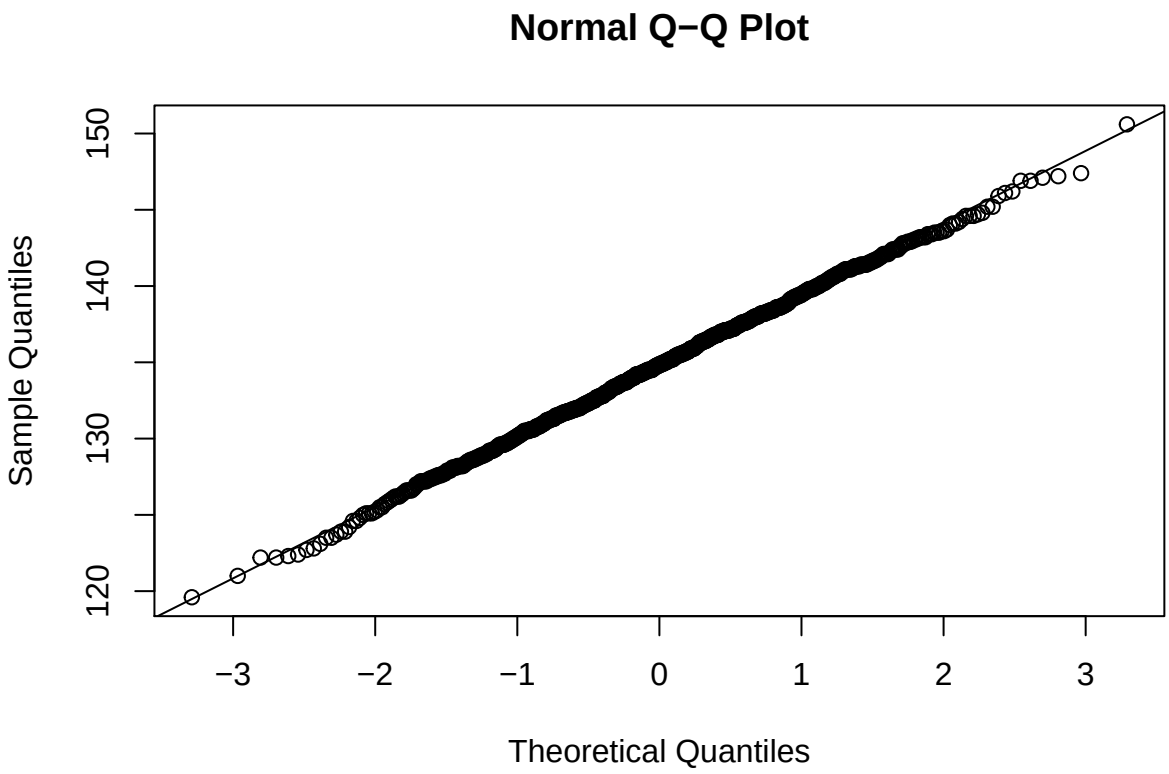
```
hist(boot[[2]], breaks = 15)
abline(v=mean(boot[[2]]))
```




```
sd(boot[[2]])
```

```
## [1] 4.656564
```

```
qqnorm(boot[[2]])
qqline(boot[[2]])
```



The SD is very close to what we had in part 1a. Figures, the bootstrap population is a clone of the original sample, after all. Also, distribution of sample means looks approx normal from hist and qqplot.

2b

Calculating the 95% interval:

```
quantile(boot[[2]], c(0.025,.975))
```

```
##  2.5% 97.5%
## 125.5 143.5
```

Looks VERY similar to the interval calculated in part 1a.

Part 3

Now, we do everything in part 1, 2 with a sample size of 100.

```
n <- 100
```

The sample mean is an unbiased estimator of the population mean, so we will use it to estimate:

```
#take a sample of size 10 from the pop.
mysample <- sample(na.omit(infants$wt), n)

#mean of the sample
```

```
sample_avg<- mean(mysample)
sample_avg
```

```
## [1] 127.25
```

The estimator of the population standard deviation is as follows:

```
#estimator of pop. variance
st_error <- (var(mysample)/n )*(1-n/total_obs)
#sqrt to find estimator of st. dev
st_error <- sqrt(st_error)
st_error
```

```
## [1] 1.971152
```

At the 95% confidence level, the critical values are plus and minus 1.96. Therefore, the boundaries of the interval are:

```
sample_avg+1.96*st_error
```

```
## [1] 131.1135
```

```
sample_avg-1.96*st_error
```

```
## [1] 123.3865
```

1b

We expect 950/1000 of 1000 intervals calculated this way to contain the population mean, as that is the definition of 95% confident in this context.

```
#initialize list of sample means
sample_means <- c()
#calculate population average weight of mother
pop_avg <- mean(na.omit(infants$wt))
pop_avg
```

```
## [1] 128.6258
```

```
i <- 1
count <- 0
for (i in 1:1000){
  #calculate a confidence interval like above
  sample <- sample(na.omit(infants$wt), n)
  sample_avg <- mean(sample)
  sample_means[i] <- sample_avg
  st_error <- sqrt((var(sample)/n )*(1-n/total_obs))
  lower <- sample_avg-1.96*st_error
  upper <- sample_avg+1.96*st_error
  #if pop_avg is between the two numbers, count++
  if (pop_avg<=upper & pop_avg>=lower){
    count <- count+1
  }
}
# observations in emprical confidence intervals
count
```

```
## [1] 951
```

As we can observe, very close to 950. This means that there is strong evidence that we can be confident in the methodology of our confidence intervals.

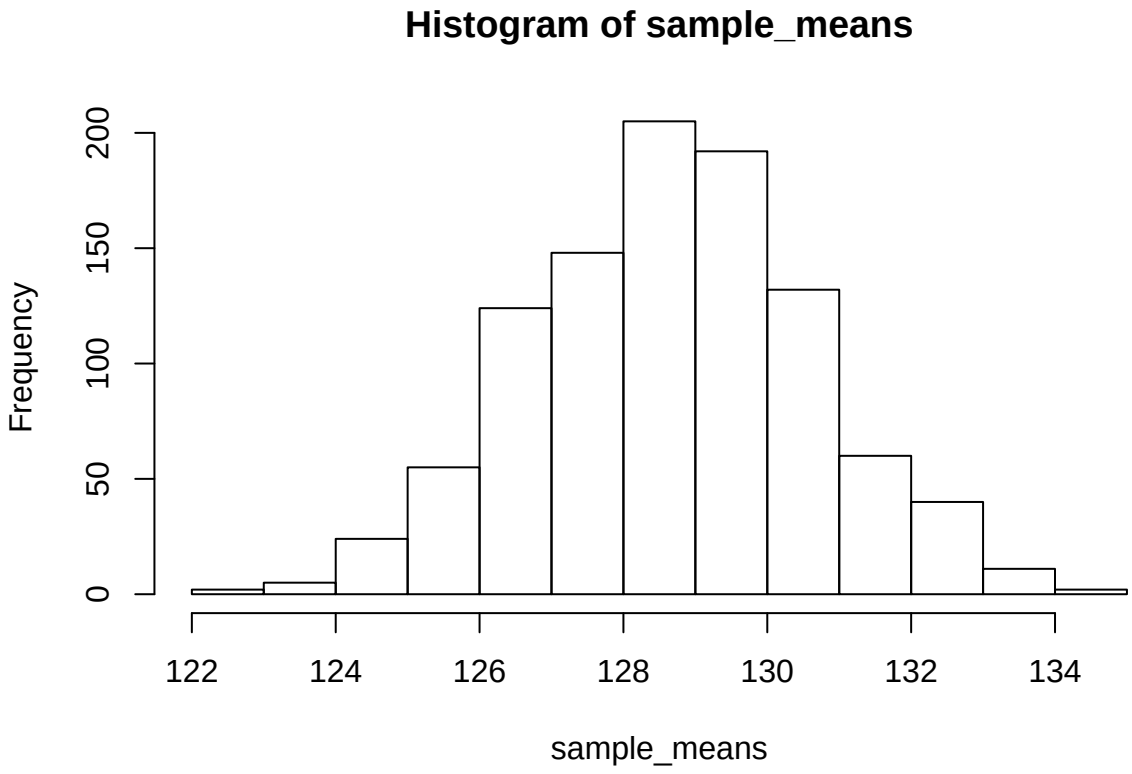
SD of sample averages

```
sd(sample_means)
```

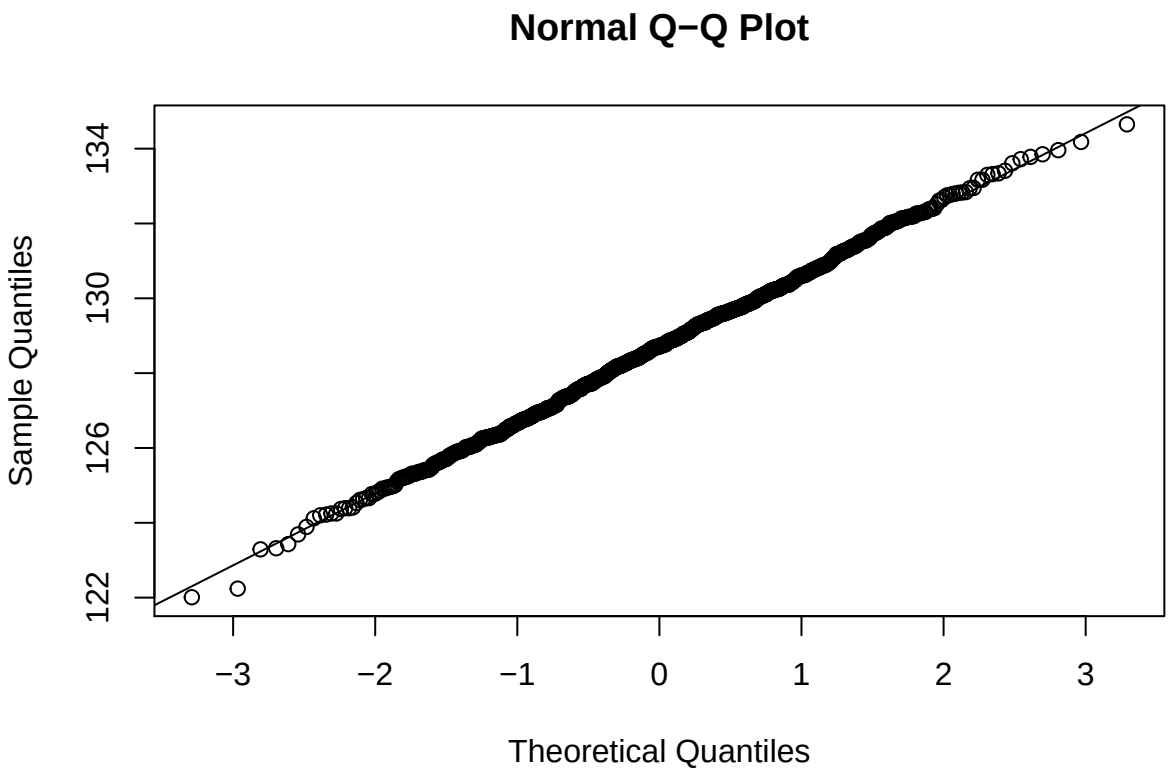
```
## [1] 1.967976
```

St. Dev is very close to the estimate in part 1a. Nice!

```
hist(sample_means)
```




```
qqnorm(sample_means)
qqline(sample_means)
```



By the Histogram and the qqplot, we can see that our sample means are roughly normal, as the data closely follows the theoretical normal quantiles and isn't left or right skewed. This is good evidence that CLT holds for n=100.

Part 2

2a

The bootstrap function calculates 1000 sample averages by default, so we can just use that:

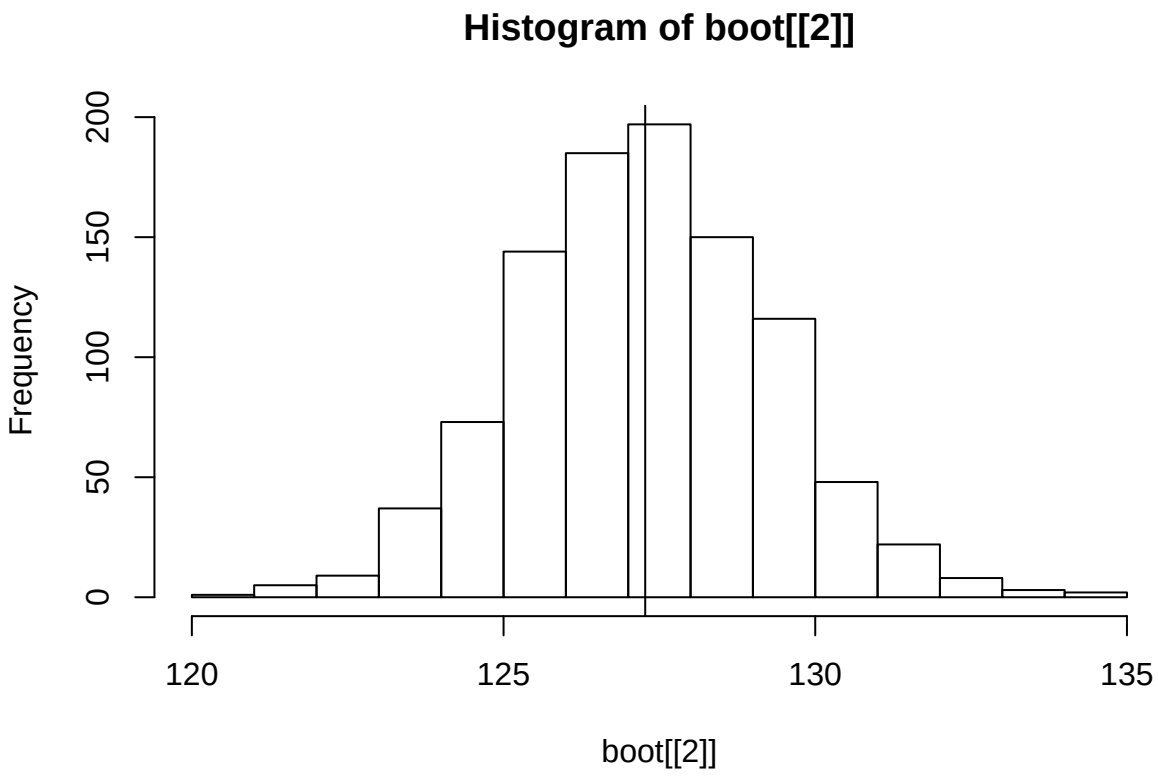
```
boot <- bootStrap(mysample, 1200)
length(boot[[1]])
```

```
## [1] 1200
```

```
#sample averages
length(boot[[2]])
```

```
## [1] 1000
```

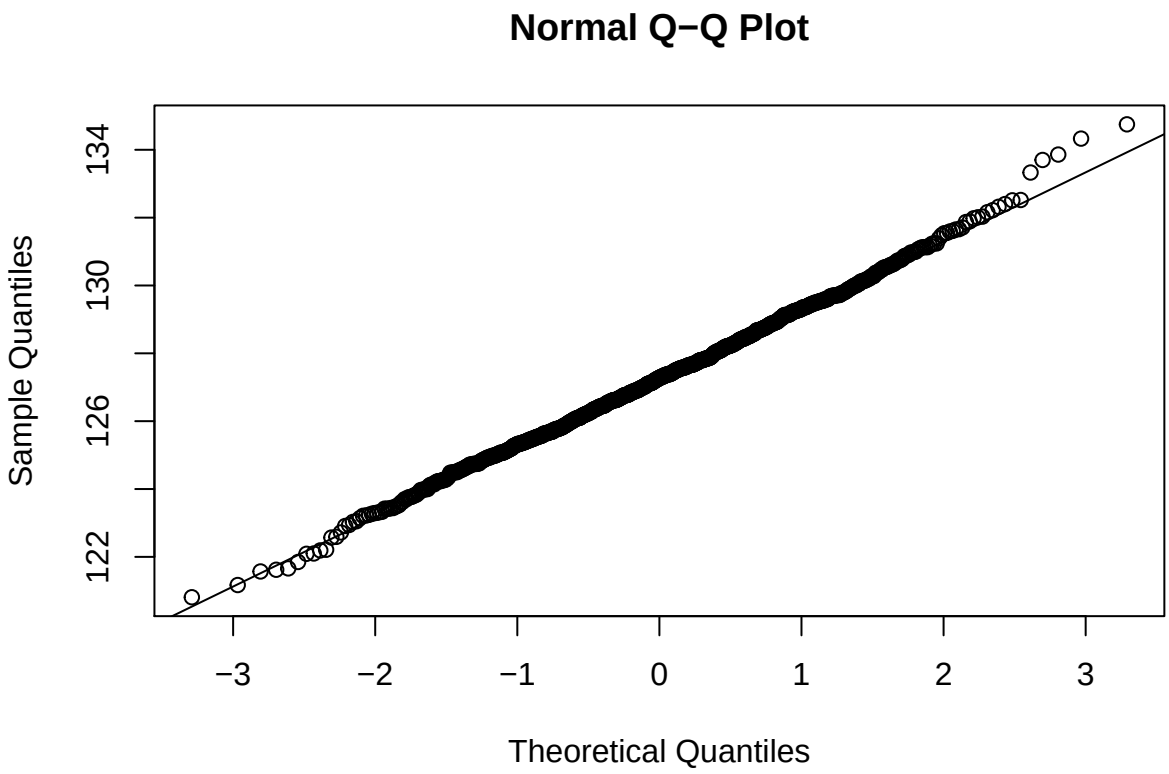
```
hist(boot[[2]], breaks = 15)
abline(v=mean(boot[[2]]))
```




```
sd(boot[[2]])
```

```
## [1] 2.03331
```

```
qqnorm(boot[[2]])
qqline(boot[[2]])
```



The SD is very close to what we had in part 1a. Figures, the bootstrap population is a clone of the original sample, after all. Also, distribution of sample means looks approx normal from hist and qqplot. qqplot shows slight evidence of more extreme values on both tails, if anything to gripe at.

2b

Calculating the 95% interval:

```
quantile(boot[[2]], c(0.025,.975))
```

```
##     2.5%    97.5%
## 123.3298 131.2438
```

Looks similar to the interval calculated in part 1a.