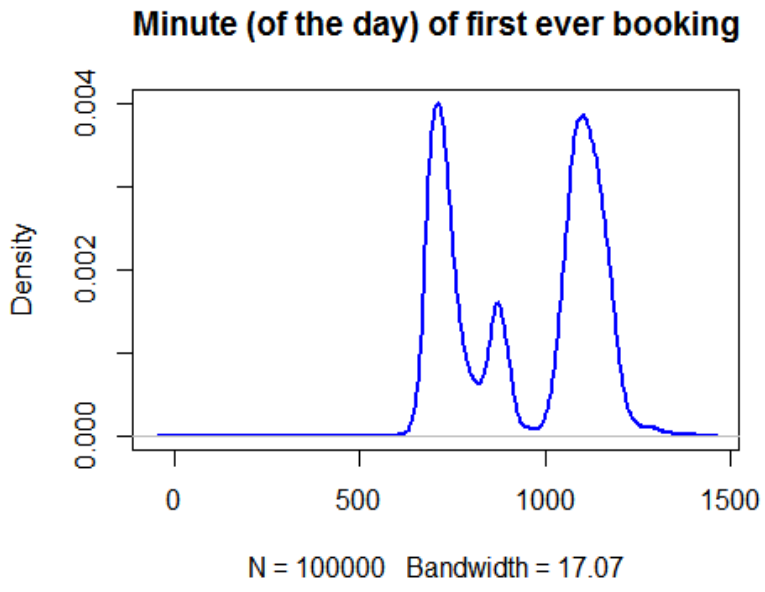
HW3

Black text are answers

Question 1

```
#Load data
bookings <- read.table("C:/Users/wendy-MM/Desktop/first_bookings_datetime_sample.txt", header=TRUE)
bookings$datetime[1:9]

## [1] 4/16/2014 17:30  1/11/2014 20:00  3/24/2013 12:00  8/8/2013 12:00
## [5] 2/16/2013 18:00  5/25/2014 15:00  12/18/2013 19:00 12/23/2012 12:00
## [9] 10/18/2013 20:00
## 18416 Levels: 1/1/2012 17:15 1/1/2012 19:00 1/1/2013 11:00 ... 9/9/2014 19:30

#transfer the time series
hours <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$hour
mins <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$min
minday <- hours*60 + mins
plot(density(minday), main="Minute (of the day) of first ever booking", col="blue", lwd=2)
```



(a) For which times of day do new members typically make their first restaurant booking? (use minday , which is the absolute minute of the day from 0-1440)

i) Use traditional statistical methods to estimat e the population mean of minday , its standard error, and its 95% confidence interval

```
#compute mean, standard error and confidence interval
pop_mean <- mean(minday)
sd_error <- sd(minday)/(length(minday)^0.5)
left <- pop_mean - 1.96 * sd_error
right <- pop_mean + 1.96*sd_error
cat(pop_mean,"\n")

## 942.4964

cat(sd_error,"\n")

## 0.5997673

cat(paste("95% confidence interval:",left,"-",right))

## 95% confidence interval: 941.32080606271 - 943.67189393729
```

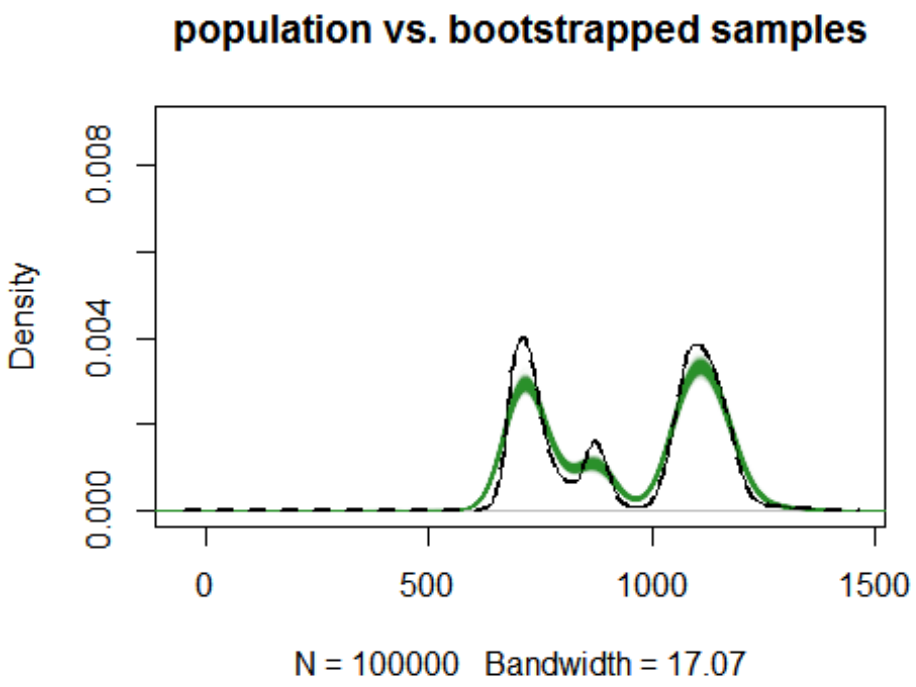
mean : 942.4964

standard error : 0.5997673

95% confidence interval: 941.32080606271 - 943.67189393729

ii) Use 2000 boostrapped samples to estimate the 95% confidence interval of the mean.

```
num_bootstraps <- 2000
set.seed(38)
resamples <-replicate(num_bootstraps, sample(minday, 2000, replace=TRU
E))

#visualize the bootstrapped samples
plot(density(minday), lwd=0, ylim=c(0, 0.009),main="population vs. boot
strapped samples")

plot_resample_density<-function(sample_i)
{lines(density(sample_i), col=rgb(0.0, 0.4, 0.0, 0.01))
  return(mean(sample_i))}
sample_means<-apply(resamples, 2, FUN=plot_resample_density)
lines(density(minday), lwd=2, lty="dashed")
```



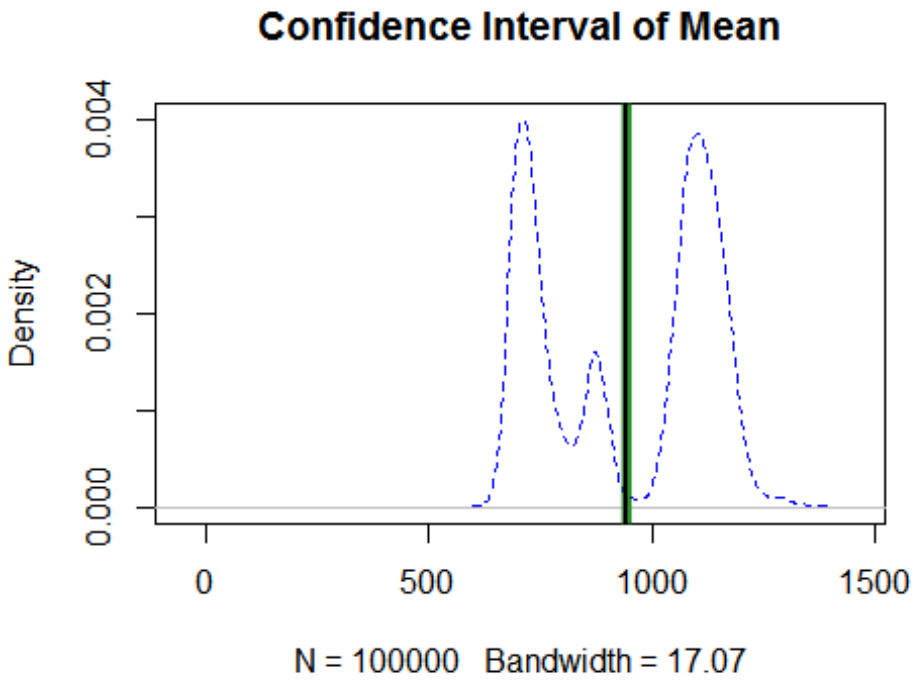

```
sample_means_mean<-mean(sample_means)
cat (sample_means_mean,"\n")

## 942.4224

quantile(sample_means, probs=c(0.025, 0.975))

##     2.5%    97.5%
## 933.9662 950.6551
```

boostrapped samples mean : 942.4224

95% confidence interval: 933.9662 - 950.6551

Note that the 95% confidence interval of the mean is wider here.

```
#visualize boostrapped samples means
plot(density(minday), col="blue", lty="dashed", main="Confidence Interv
al of Mean")

plot_resample_mean<-function(sample_i) {abline(v=mean(sample_i), col=rg
b(0.0, 0.4, 0.0, 0.01))}
sample_means<-apply(resamples, 2, FUN=plot_resample_mean)
abline(v=sample_means_mean, lwd=2)
abline(v=pop_mean, lty="dashed")
```



(b) By what time of day, have half the new members of the day already arrived at their restaurant?

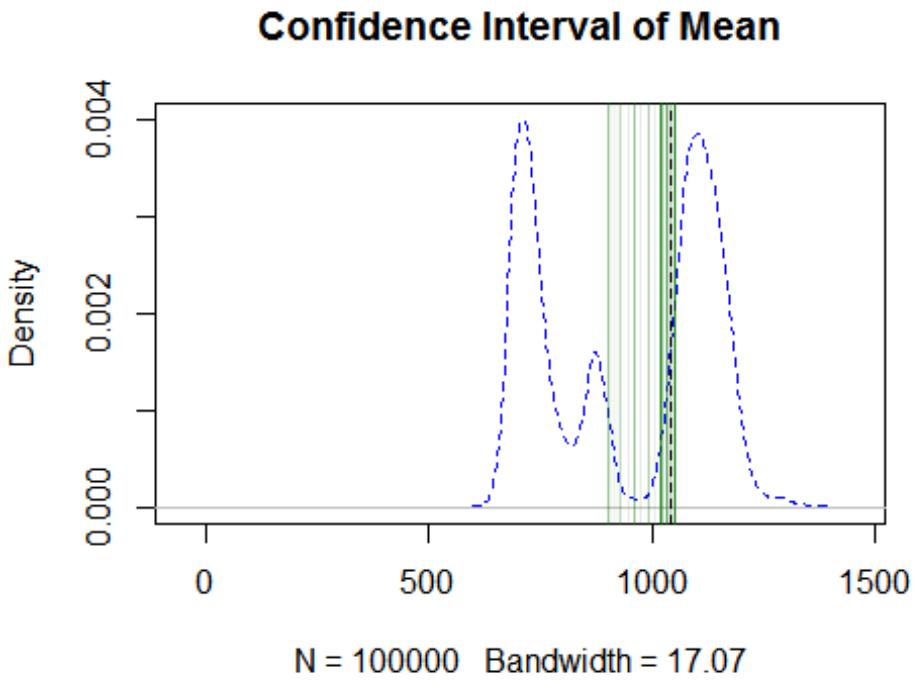
i) Estimate the median of minday

```
median(minday)

## [1] 1040
```

The estimated median of minday is 1040

ii) Use 2000 bootstrapped samples to estimate the 95% confidence interval of the median

```
#visualize the bootstrapped samples medians
plot(density(minday), col="blue", lty="dashed", main="Confidence Interv
al of Mean")

plot_resample_median<-function(sample_i) {
  abline(v=median(sample_i), col=rgb(0.0, 0.4, 0.0, 0.01))
  return(median(sample_i))
  }
sample_medians<-apply(resamples, 2, FUN=plot_resample_median)
abline(v=median(minday), lty="dashed")
```




```
sample_medians_median <- median(sample_medians)
sample_medians_mean <- mean(sample_medians)
cat("use median as estimator:",sample_medians_median,"\n")

## use median as estimator: 1043.75

cat("use mean as estimator:",sample_medians_mean,"\n")

## use mean as estimator: 1028.88

quantile(sample_medians, probs=c(0.025, 0.975))

##      2.5%     97.5%
##  912.4375 1050.0000
```

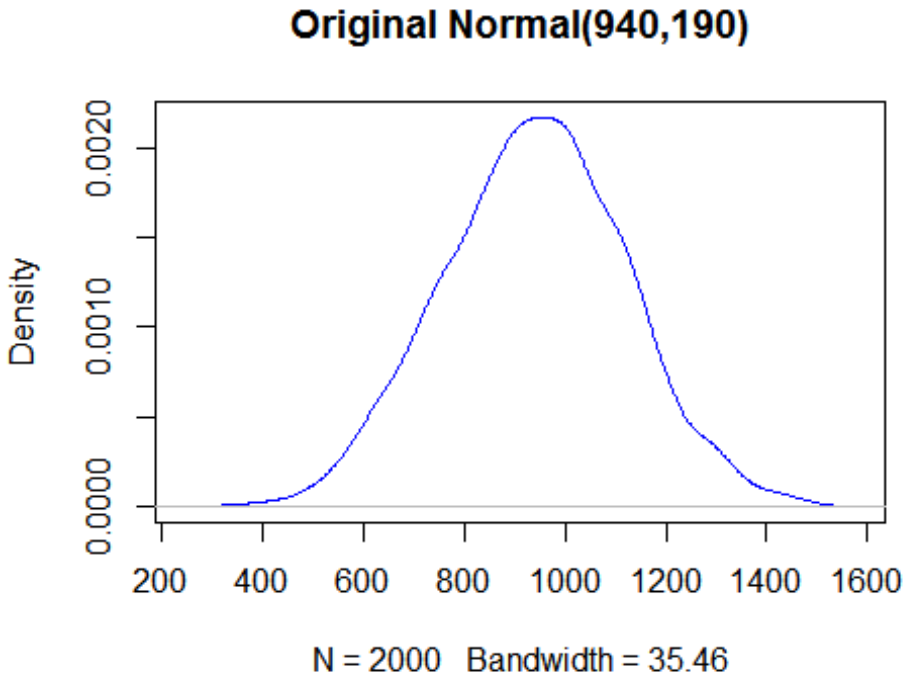
The 95% confidence interval of the bootstrapped samples median is 912.4375 to 1050.0000. Here I use median and mean to estimate population median.

Question 2

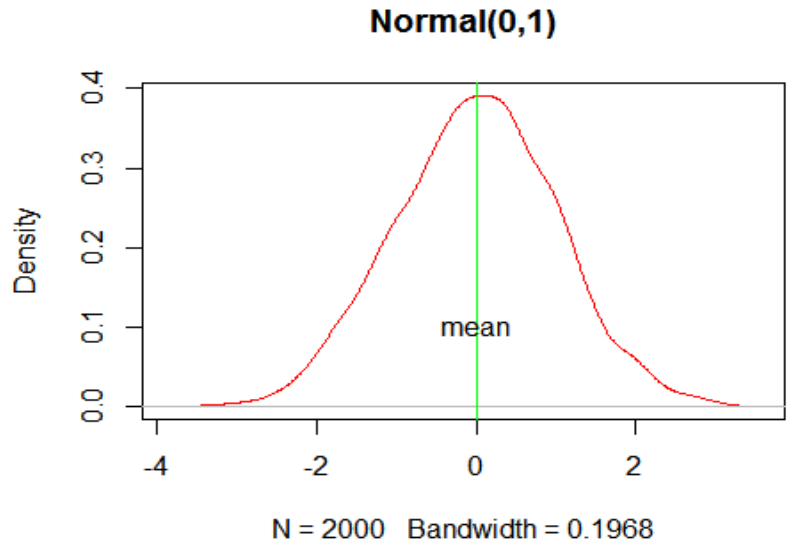
(a) Create a normal distribution (mean=940, sd=190) and standardize it (let's call it rnorm_std)

```
# Create a normal distribution
set.seed(38)
```

```r
normal_dist <- rnorm(2000,mean = 940,sd=190)
plot(density(normal_dist),main="Original Normal(940,190)",col="blue")
```




```r
# standardize the original normal distribution
rnorm_std <- (normal_dist-mean(normal_dist))/sd(normal_dist)
plot(density(rnorm_std),main="Normal(0,1)",col="Red")
abline(v=mean(rnorm_std),col = "green")
text(mean(rnorm_std),0.1,"mean")
```



i) What should we expect the mean and standard deviation of rnorm_std to be, and why?

Since we just subtracted the mean of original normal vector from all its values, and then dividing them by the standard deviation, the mean might become 0, and the standard deviation might become 1. That is so-called standardization.

```
cat ("mean of rnorm_std: ",mean(rnorm_std),"\n")

## mean of rnorm_std:  -1.96547e-16

cat ("standard deviation: ",sd(rnorm_std))

## standard deviation:  1
```

As expectated, the mean is extremely close to zero and the standard deviation is one.

ii) What should the distribution (shape) of rnorm_std look like, and why?

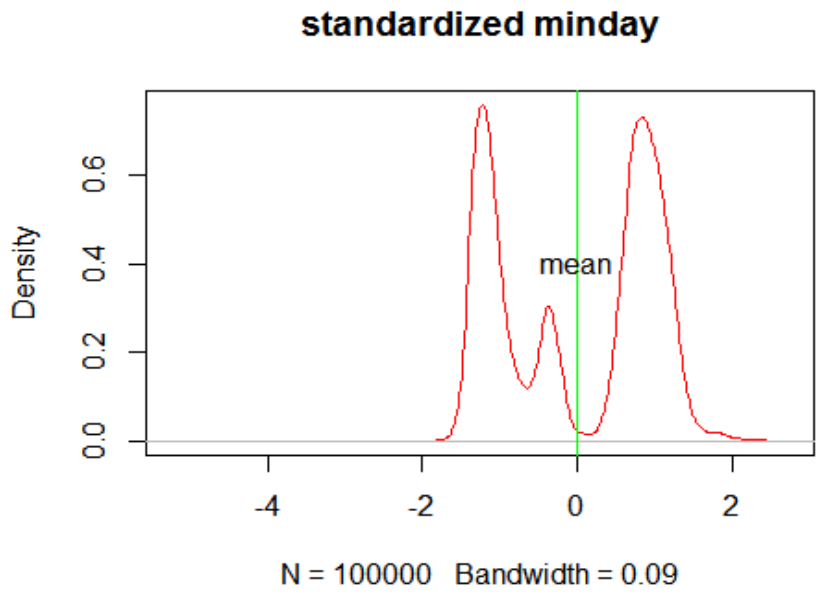
From the above figure, it's clearly a bell shape same as the original normal distribution.

iii) What do we generally call distributions that are normal and standardized?

Standard normal distribution,a normal distribution with zero mean ($mu = 0$) and unit variance ($sigma^2$=1$).

(b) Create a standardized version of minday from above (let's call it minday_std)

```
minday_std <- (minday-mean(minday))/sd(minday)
plot(density(minday_std),main="standardized minday",col="Red")
abline(v=mean(minday_std),col = "green")
text(mean(minday_std),0.4,"mean")
```



i) What should we expect the mean and standard deviation of minday_std to be, and why?

Here we do standardization again, so the mean would be zero and standard deviation would be one.

```
cat ("mean of rnorm_std: ",mean(minday_std),"\n")

## mean of rnorm_std:  -4.25589e-17

cat ("standard deviation: ",sd(minday_std))

## standard deviation:  1
```

Above outputs show the answer! The mean closes to 0 and standard deviation is 1. However, this time the distribution is not normal.

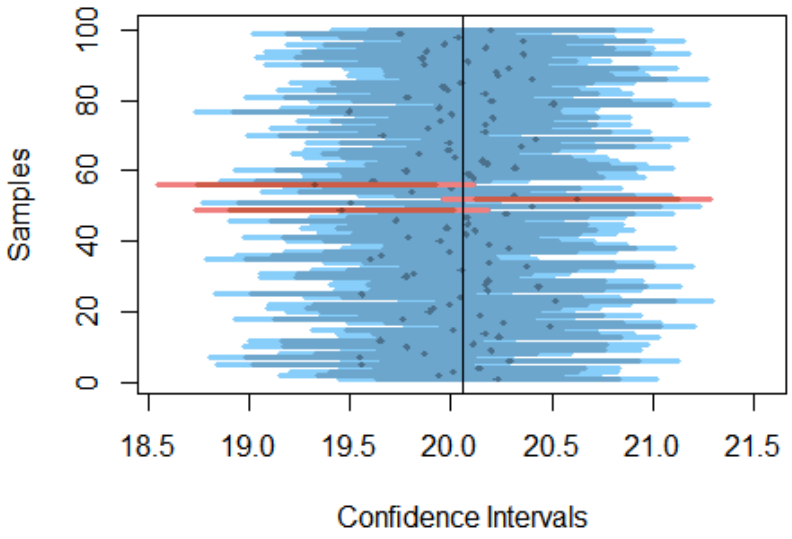
ii) What should the distribution of minday_std look like compared to minday , and why?

Same as the previous answer, we centered the data then linearly scaled it, so the shpe might look similar. Besides, note that the x-axis and y-axis would change.

Question 3

(a) Simulate 100 samples (each of size 100), from a normally distributed population of 10,000:visualize_sample_ci(num_samples=100, sample_size = 100, pop_size=10000, distr_func=rnorm, mean=20, sd=3)

```
#Load the function from teacher's script
source("confidence_intervals.R")
visualize_sample_ci(num_samples = 100, sample_size = 100, pop_size=1000
0, distr_func=rnorm, mean=20, sd=3)
```



i) How many samples do we expect to NOT include the population mean in its 95% CI?

I expect 5 samples NOT include the population mean in its 95% CI. But here we are using simulation, the result might be different.

```
#simulation on how many samples do we expect to NOT include the populat
ion mean in its 95% CI.

#I've made a few modification on teacher's code

bad_vec <- c()
for (i in 1:10000){
  bad = visualize_sample_ci(num_samples = 100, sample_size = 100, pop_s
ize=10000, distr_func=rnorm, mean=20, sd=3,draw = F)
  bad_vec[i] <- length(bad)
}
#Use mean as estimator
cat (mean(bad_vec),"\n")

## 5.2034
```

Here I did a tiny simulation experiment to estimate many samples NOT include the population mean in its 95% CI. The answer is close to 5!

ii) How many samples do we expect to NOT include the population mean in their 99% CI?

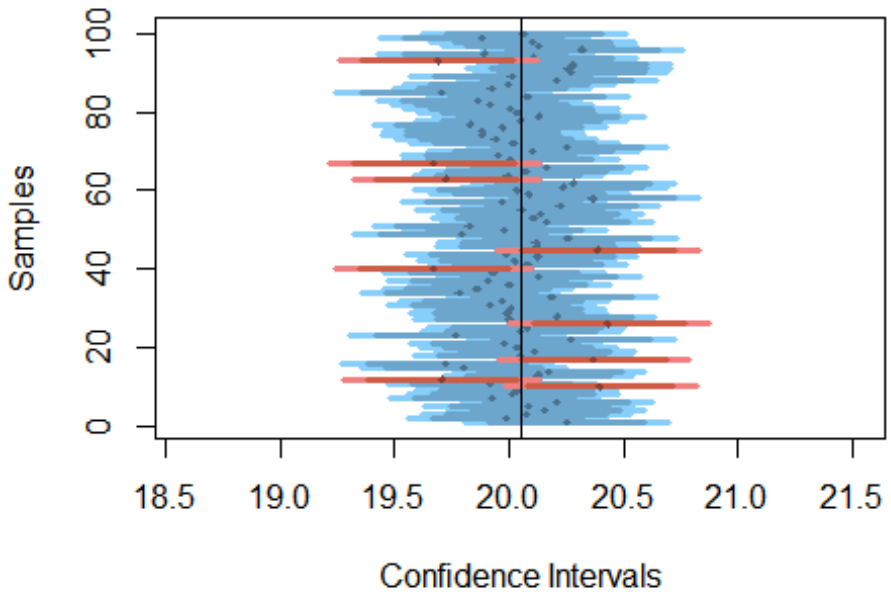
I expect 1 samples NOT include the population mean in its 99% CI.

```r
#simulation on how many samples do we expect to NOT include the populat
ion mean in its 95% CI.
bad_vec <- c()
for (i in 1:10000){
  bad = visualize_sample_ci(num_samples = 100, sample_size = 100, pop_s
ize=10000, distr_func=rnorm, mean=20, sd=3,draw = F,ci=99)
  bad_vec[i] <- length(bad)
}
#Use mean as estimator
cat (mean(bad_vec),"\n")

## 1.0856
```

Also, by simulation, the result is close to 1.

(b) Rerun the previous simulation with larger samples (300 each): visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size=10000, distr_func=rnorm, mean=20, sd=3)

```r
set.seed(38)
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size=1000
0, distr_func=rnorm, mean=20, sd=3)
```



i) Now that the size of each sample has increased, do we expect their 95% and 99% CI to become wider or narrower than before?

Comapre the plot in (a), here the blue lines become shorter; that is, the 95% and 99% CI become narrower than before. According to the formula of confidence interval of population mean, $\overline{x} \pm t(s/\sqrt{n})$, if the sample size n increase, than the interval will become narrower.

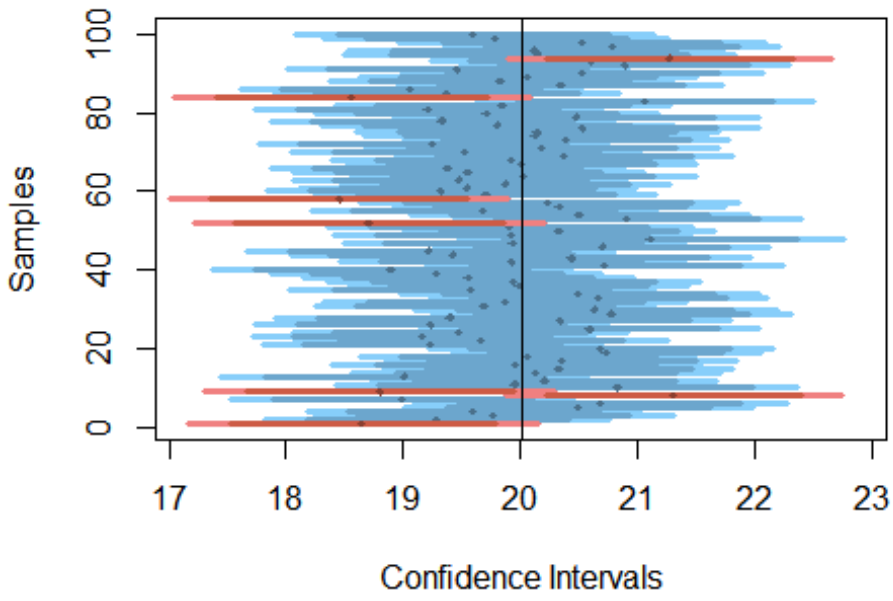
ii) This time, how many samples (out of the 100) would we expect to NOT include the population mean in its 95% CI?

By definition, the answer should still be 5. We can also prove it by simulation!

```
#simulation uniform distribution
bad_vec <- c()
for (i in 1:10000){
  bad = visualize_sample_ci(num_samples = 100, sample_size = 300, pop_s
ize=10000, distr_func=runif,min=10, max=30,draw = F,ci=95)
  bad_vec[i] <- length(bad)
}
#Use mean as estimator
cat (mean(bad_vec),"\n")

## 4.7168
```

(c) If we ran the above two examples (a and b) using a uniformly distributed population, how do would you expect your answers to (a) and (b) to change?

```
visualize_sample_ci(num_samples = 100, sample_size = 100, pop_size=1000
0, distr_func=runif, min=10, max=30)
```



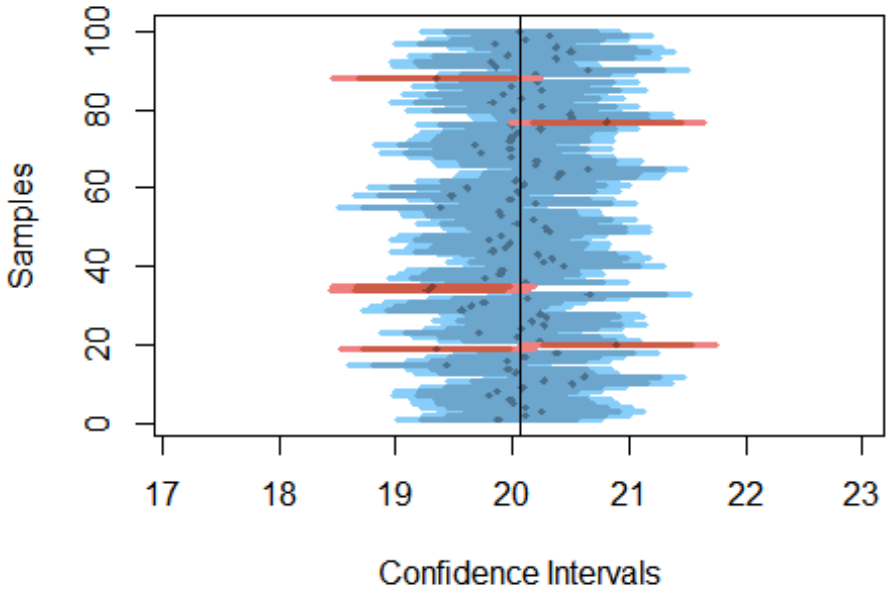
```
#simulation uniform distribution
bad_vec <- c()
for (i in 1:10000){
  bad = visualize_sample_ci(num_samples = 100, sample_size = 100, pop_s
ize=10000, distr_func=runif,min=10, max=30,draw = F,ci=95)
  bad_vec[i] <- length(bad)
}
#Use mean as estimator
cat (mean(bad_vec),"\n")

## 5.2081
```

I think the result may not change. From the above experiment, the number of samples (out of the 100) NOT include the population mean in its 95% CI is still close to 5.

```
#simulation uniform distribution. 99% CI
bad_vec <- c()
for (i in 1:10000){
  bad = visualize_sample_ci(num_samples = 100, sample_size = 100, pop_s
ize=10000, distr_func=runif,min=10, max=30,draw = F,ci=99)
  bad_vec[i] <- length(bad)
}
#Use mean as estimator
cat (mean(bad_vec),"\n")

## 1.1267
```

The number of samples (out of the 100) NOT include the population mean in its 99% CI is still close to 1.

```
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size=1000
0, distr_func=runif, min=10, max=30)
```



By comparing the above two plots, the latter one is narrower, which is the same answer as (b).