HW03

107070008

Question 1) (a)Create a normal distribution (mean=940, sd=190) and standardize it (let's call it rnorm_std)

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

```
standardized <-function(numbers) {
    std <-(numbers-mean(numbers)) / sd(numbers)
    return(std)
}
data <- rnorm(n = 10000, mean = 940, sd = 190)
rnorm_std <- standardized(data)
mean(rnorm_std)
```

```
## [1] -3.267236e-17
```

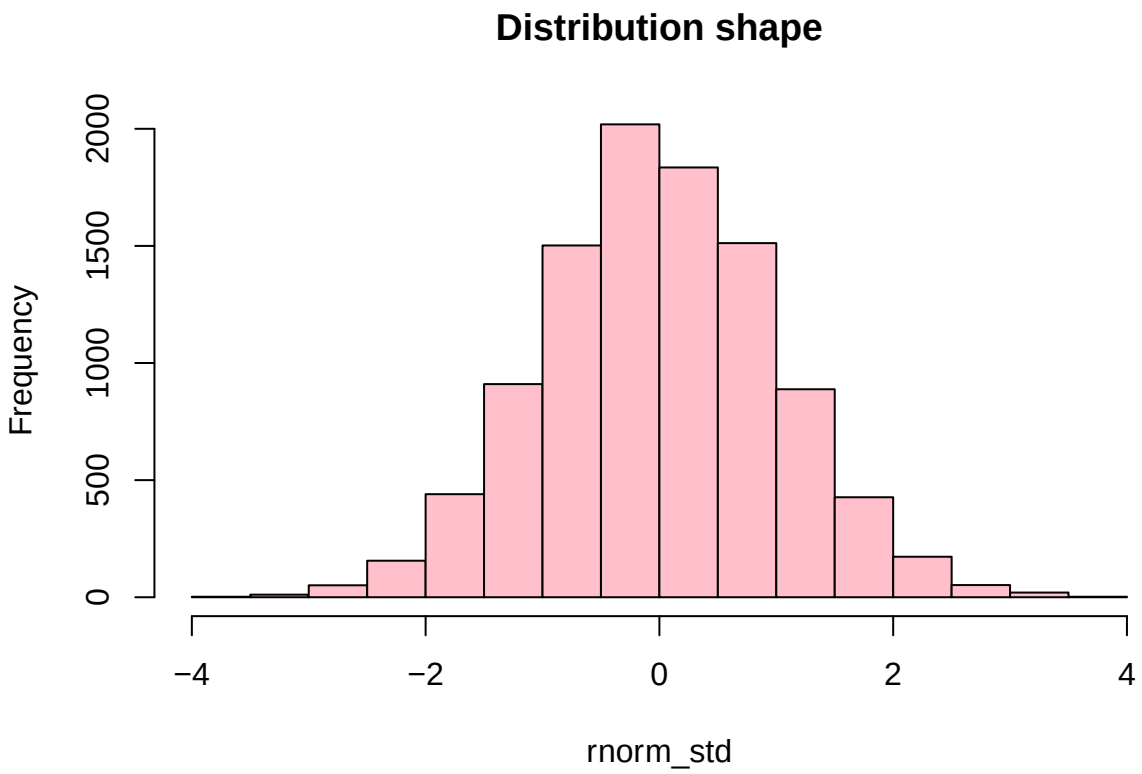
```
sd(rnorm_std)
```

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

ANS: `It's mean will be 0, and sd will be 1, because it is a normal distribution.`

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

```
hist(rnorm_std, col="pink", main = "Distribution shape")
```



ANS: `It looks like a bell, because it is noraml distrbution.`

iii)What do we generally call distributions that are normal and standardized? ANS: `Standard normal distibution, standard score, and z-score.`

(b)Create a standardized version of minday discussed in question 3 (let's call it minday_std)

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

```
bookings <- read.table("./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
```

```
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
minday_std <- standardized(minday)
mean(minday_std)
```

```
## [1] -4.25589e-17
```

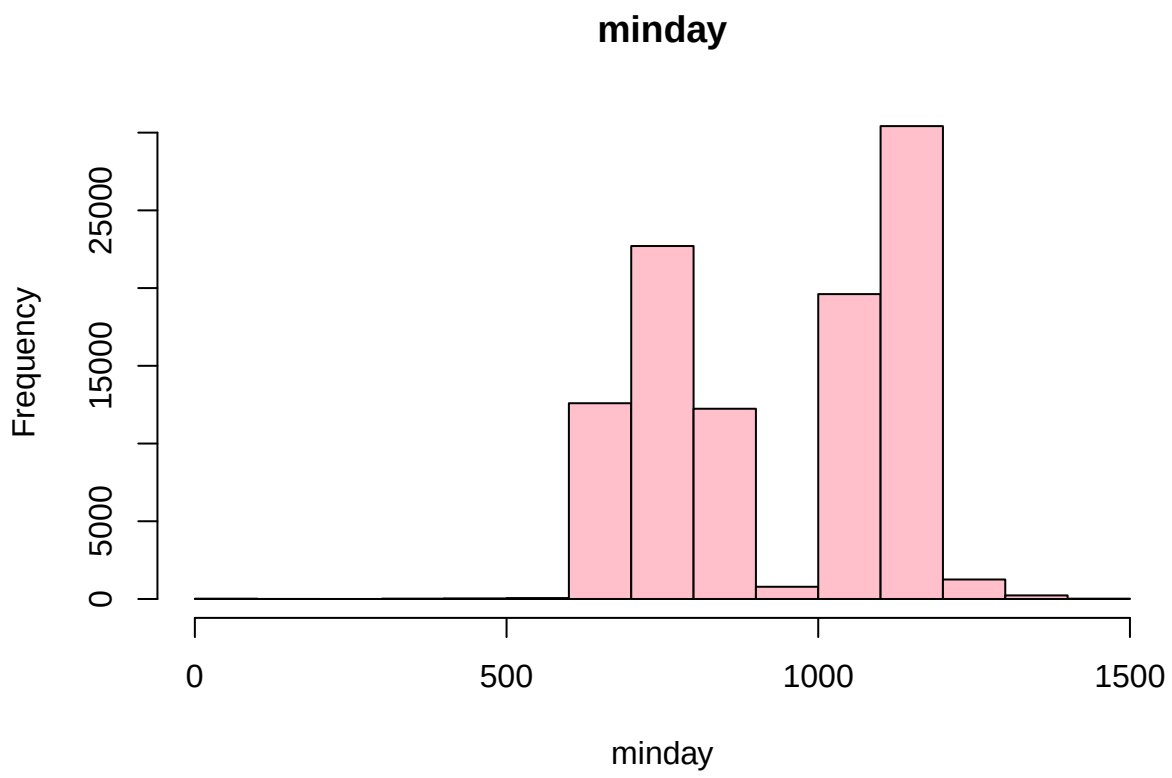
```
sd(minday_std)
```

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

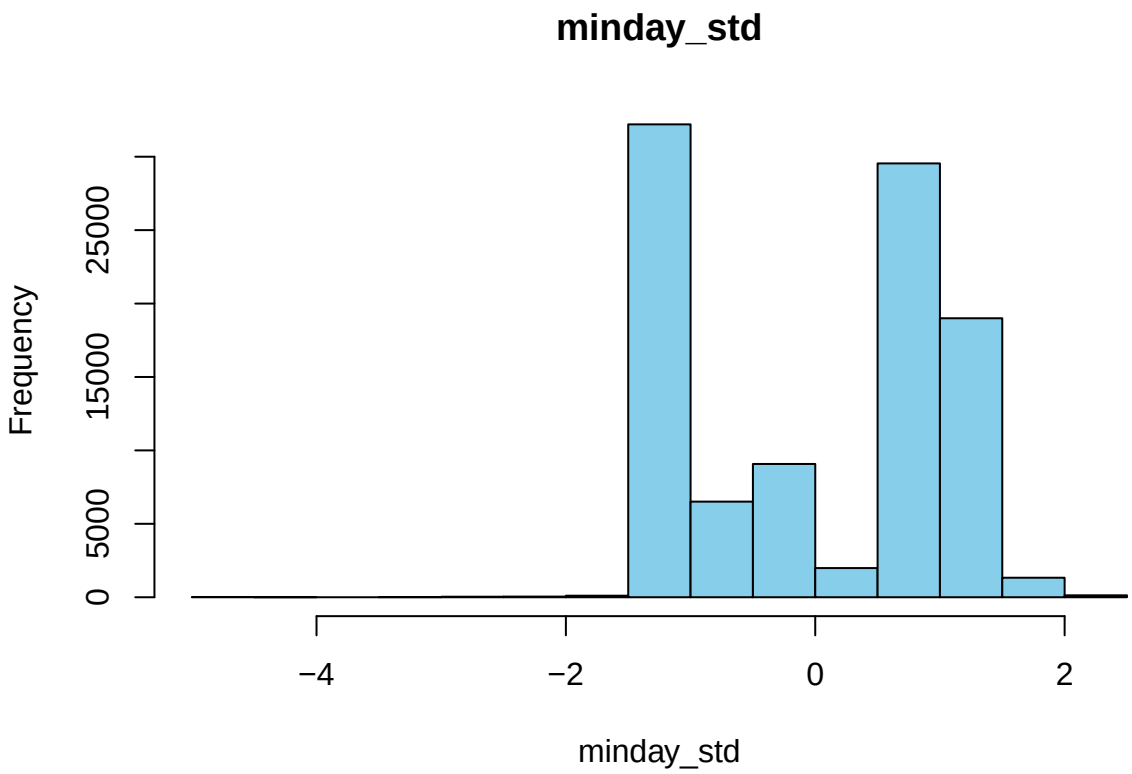
ANS: It's mean will not always be 0, because it's not a normal distribution.

(ii)

```
hist(minday, col="pink", main = "minday")
```




```
hist(minday_std, col="skyblue", main = "minday_std")
```



ANS: The shape of minday_std is more similar to bell shape than original minday.

Question 2)

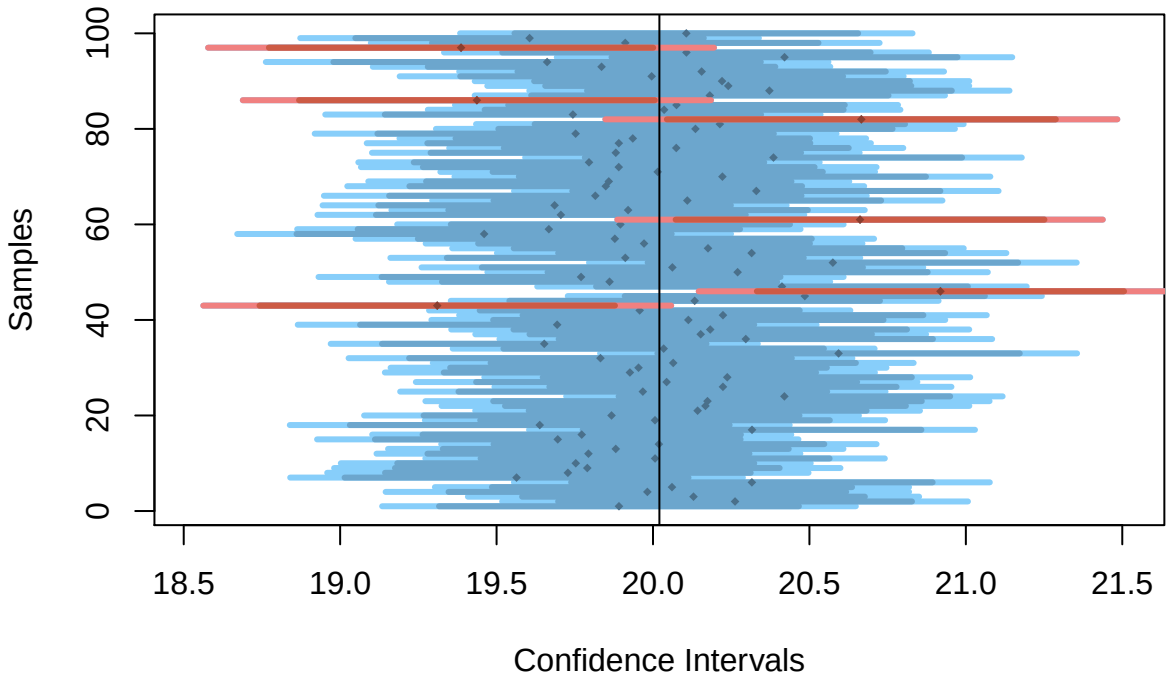
```r
# Visualize the confidence intervals of samples drawn from a population
#   e.g.,
#     visualize_sample_ci(sample_size=300, distr_func=rnorm, mean=50, sd=10)
#     visualize_sample_ci(sample_size=300, distr_func=runif, min=17, max=35)
visualize_sample_ci <- function(num_samples = 100, sample_size = 100,
                                pop_size=10000, distr_func=rnorm, ...) {
  # Simulate a large population
  population_data <- distr_func(pop_size, ...)
  pop_mean <- mean(population_data)
  pop_sd <- sd(population_data)

  # Simulate samples
  samples <- replicate(num_samples, sample(population_data, sample_size, replace=FALSE))

  # Calculate descriptives of samples
  sample_means = apply(samples, 2, FUN=mean)
  sample_stdevs = apply(samples, 2, FUN=sd)
  sample_stderrs <- sample_stdevs/sqrt(sample_size)
  ci95_low  <- sample_means - sample_stderrs*1.96
  ci95_high <- sample_means + sample_stderrs*1.96
  ci99_low  <- sample_means - sample_stderrs*2.58
  ci99_high <- sample_means + sample_stderrs*2.58
```

```
  # Visualize confidence intervals of all samples
  plot(NULL, xlim=c(pop_mean-(pop_sd/2), pop_mean+(pop_sd/2)),
       ylim=c(1,num_samples), ylab="Samples", xlab="Confidence Intervals")
  add_ci_segment(ci95_low, ci95_high, ci99_low, ci99_high,
                 sample_means, 1:num_samples, good=TRUE)

  # Visualize samples with CIs that don't include population mean
  bad = which(((ci95_low > pop_mean) | (ci95_high < pop_mean)) |
                ((ci99_low > pop_mean) | (ci99_high < pop_mean)))
  add_ci_segment(ci95_low[bad], ci95_high[bad], ci99_low[bad], ci99_high[bad],
                 sample_means[bad], bad, good=FALSE)

  Not95 <- which(((ci95_low > pop_mean) | (ci95_high < pop_mean)))
  Not99 <- which(((ci99_low > pop_mean) | (ci99_high < pop_mean)))
  cat("Not in 95% ci:", length(Not95))
  cat("\nNot in 99% ci:", length(Not99))

  # Draw true population mean
  abline(v=mean(population_data))
}

add_ci_segment <- function(ci95_low, ci95_high, ci99_low,
                           ci99_high, sample_means, indices, good=TRUE) {
  segment_colors <- list(c("lightcoral", "coral3", "coral4"),
                         c("lightskyblue", "skyblue3", "skyblue4"))
  color <- segment_colors[[as.integer(good)+1]]

  segments(ci99_low, indices, ci99_high, indices, lwd=3, col=color[1])
  segments(ci95_low, indices, ci95_high, indices, lwd=3, col=color[2])
  points(sample_means, indices, pch=18, cex=0.6, col=color[3])
}
```

(a)Simulate 100 samples (each of size 100), from a normally distributed population of 10,000:

(i)How many samples do we expect to NOT include the population mean in its 95% CI? (ii)How many samples do we expect to NOT include the population mean in their 99% CI?

```
visualize_sample_ci(num_samples = 100, sample_size = 100,
                    pop_size=10000, distr_func=rnorm, mean=20, sd=3)
```

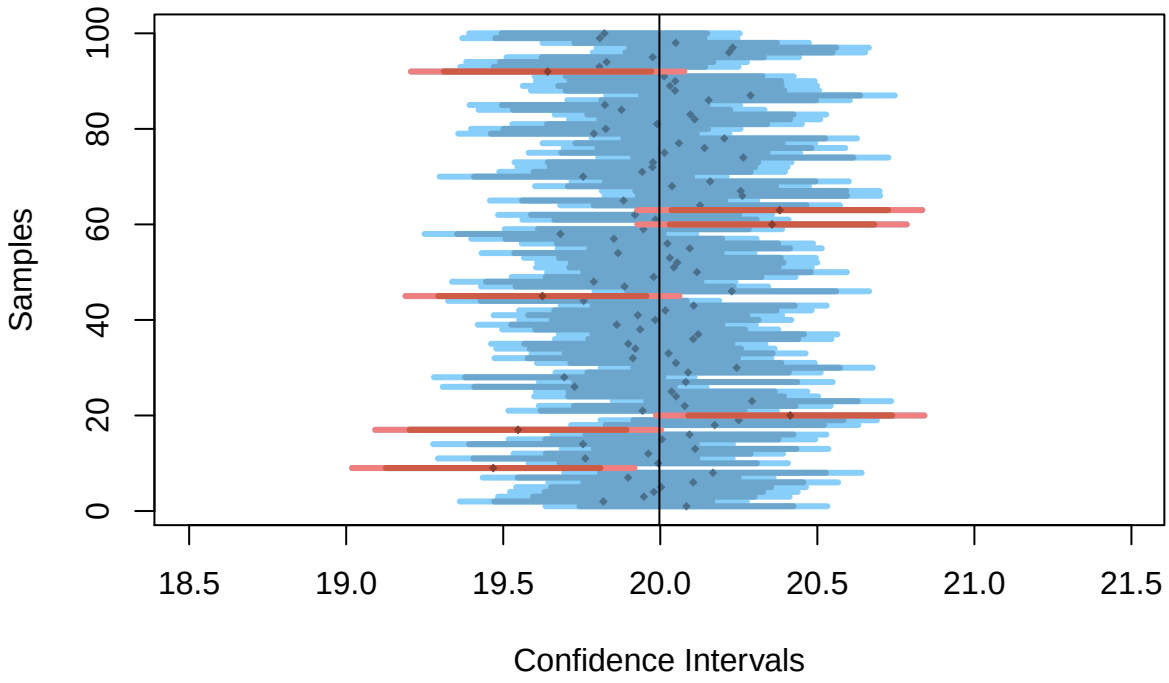



```
## Not in 95% ci: 6
## Not in 99% ci: 1
```

(b)Rerun the previous simulation with the same number of samples, but larger sample size

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

```
visualize_sample_ci(num_samples = 100, sample_size = 300,
                    pop_size=10000, distr_func=rnorm, mean=20, sd=3)
```

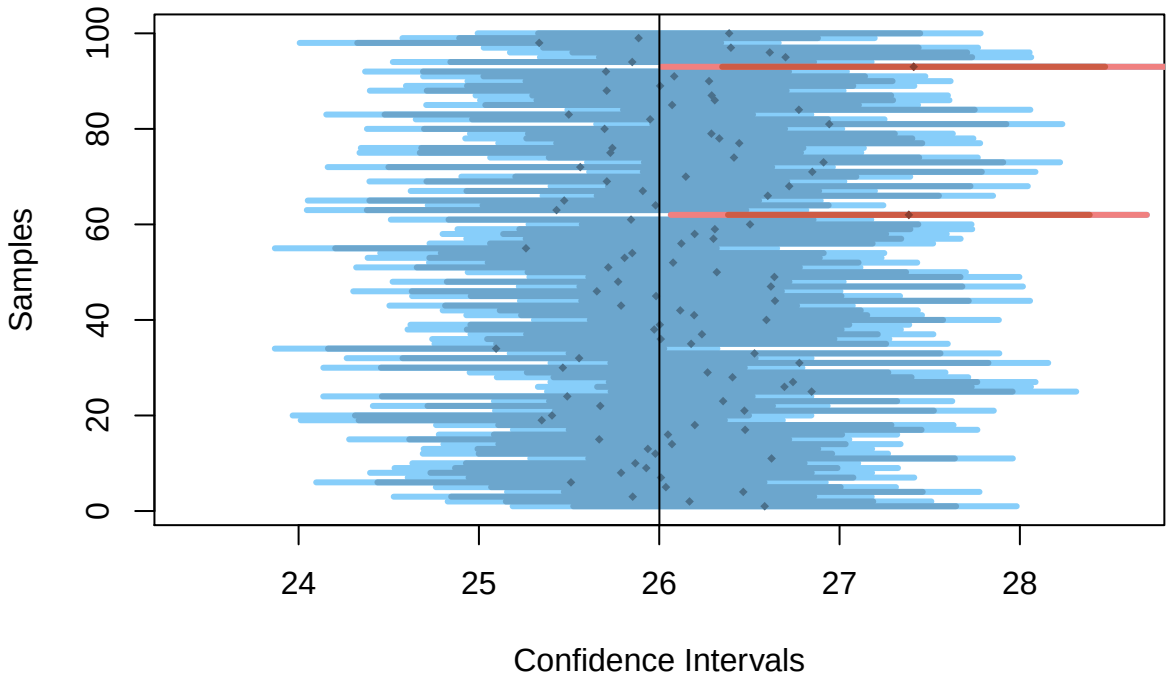



```
## Not in 95% ci: 7
## Not in 99% ci: 1
```

ANS: `Yes, it's 99% ci and 95% ci are narrower than before.When the number of sample increase our datas will become more reliable, and the width of confidence interval will decrease.`

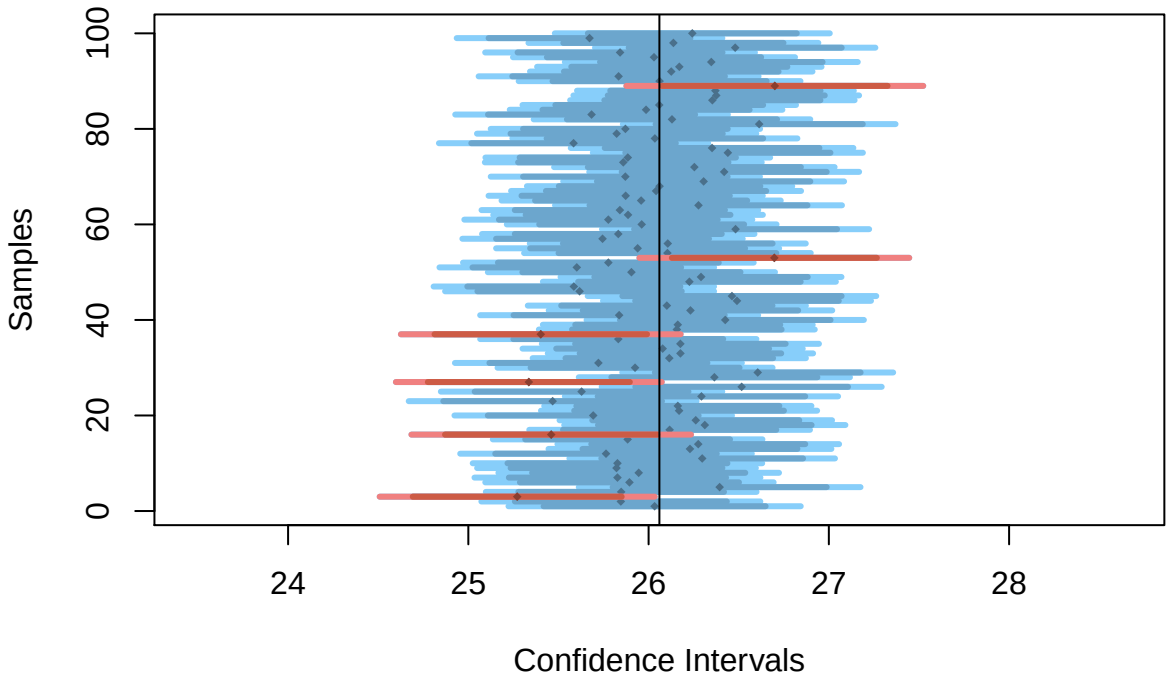
(c)If we ran the above two examples (a and b) using a uniformly distributed population (specify distr_func=runif for visualize_sample_ci), how do you expect your answers to (a) and (b) to change, and why?

```
visualize_sample_ci(num_samples = 100, sample_size = 100,
                    pop_size=10000, distr_func=runif, min=17, max=35)
```




```
## Not in 95% ci: 2
## Not in 99% ci: 2
```

```
visualize_sample_ci(num_samples = 100, sample_size = 300,
                    pop_size=10000, distr_func=runif, min=17, max=35)
```




```
## Not in 95% ci: 6
## Not in 99% ci: 1
```

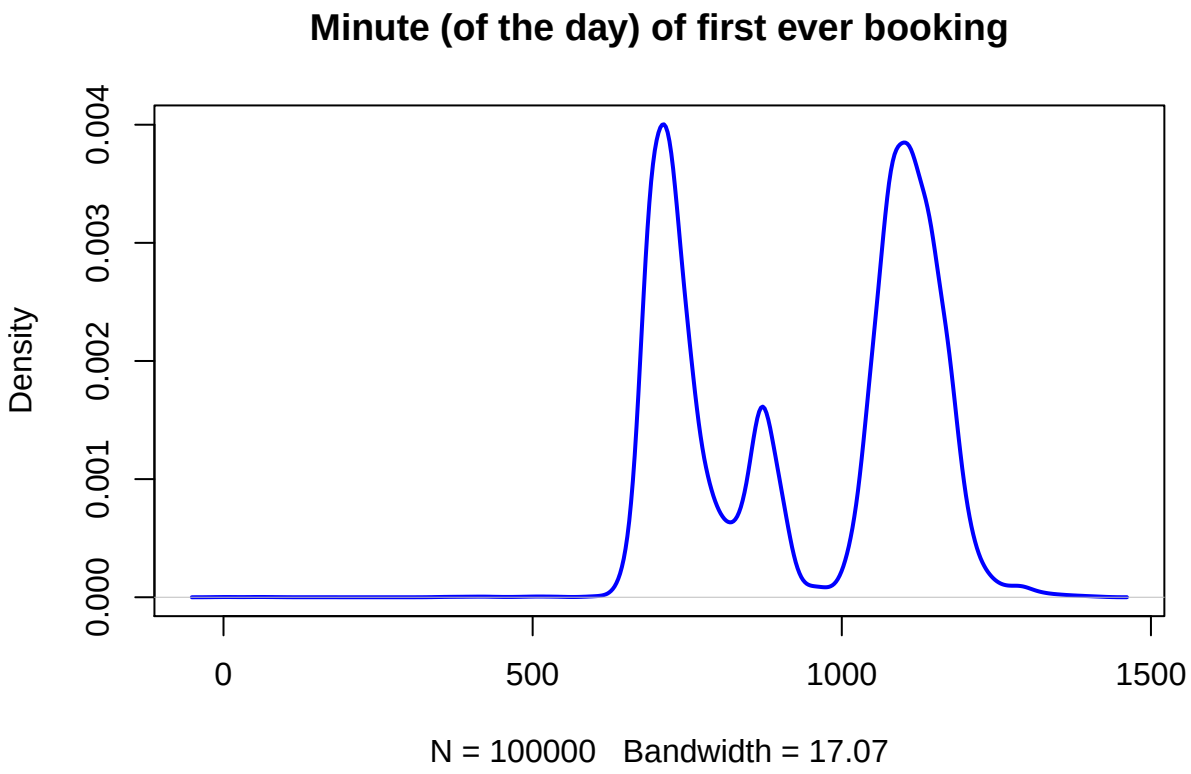
ANS: Compared to (a) and (b), they both hava wider width when sample_size is 100, and have narrower width when sample_size is 300.

Question 3)

```
bookings <- read.table("./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
```

```
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)What is the "average" booking time for new members making their first restaurant booking? (i)Use traditional statistical methods to estimate the population mean of minday, its standard error, and the 95% confidence interval (CI) of the sampling means

```
mean(minday)
```

```
## [1] 942.4964
```

```
std_error <- sd(minday)/sqrt(length(minday))
std_error
```

```
## [1] 0.5997673
```

```
cat("95% confidence level is from",
    mean(minday)-1.96*std_error, "to", mean(minday)+1.96*std_error)
```

```
## 95% confidence level is from 941.3208 to 943.6719
```

(ii)Bootstrap to produce 2000 new samples from the original sample

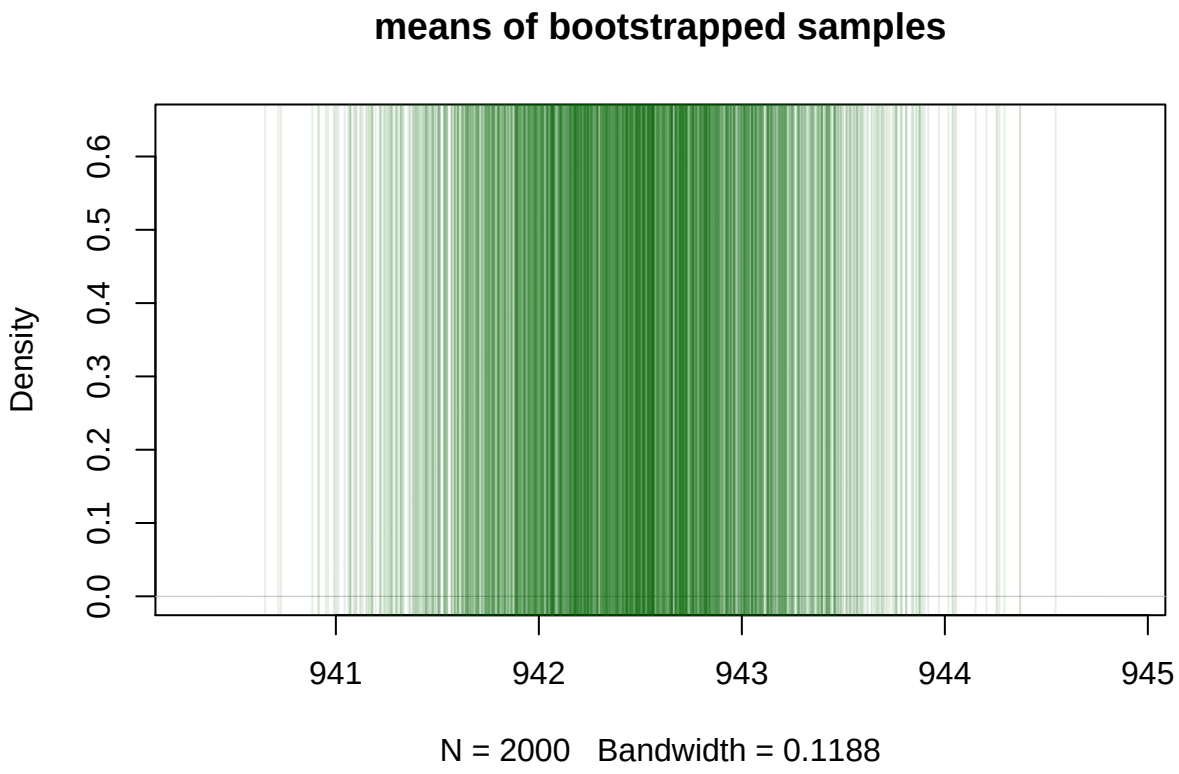
```
compute_sample_mean <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  mean(resample)
}
compute_sample_mean(minday)
```

```
## [1] 942.2733
```

```
sample_means <- replicate(2000, compute_sample_mean(minday))
```

(iii)Visualize the means of the 2000 bootstrapped samples

```
plot(density(sample_means), lwd=0, col=rgb(0.0, 0.4, 0.0, 0.01),
     main="means of bootstrapped samples")
plot_sample_mean<-function(sample_i) {
  abline(v=sample_i, col=rgb(0.0, 0.4, 0.0, 0.1))
}
resamples_ <- data.matrix(sample_means)
apply(resamples_, 2, FUN = plot_sample_mean)
```




```
## NULL
```

(iv) Estimate the 95% CI of the bootstrapped means.

```
quantile(sample_means, probs = c(0.05, 0.95))
```

```
##       5%      95%
## 941.4782 943.4597
```

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

```
hour <- round(median(minday)/60, 0)
minute <- median(minday)%%60
cat(hour,":",minute)
```

```
## 17 : 20
```

(i)Estimate the median of minday

```
median(minday)
```

```
## [1] 1040
```

(ii)Visualize the medians of the 2000 bootstrapped samples

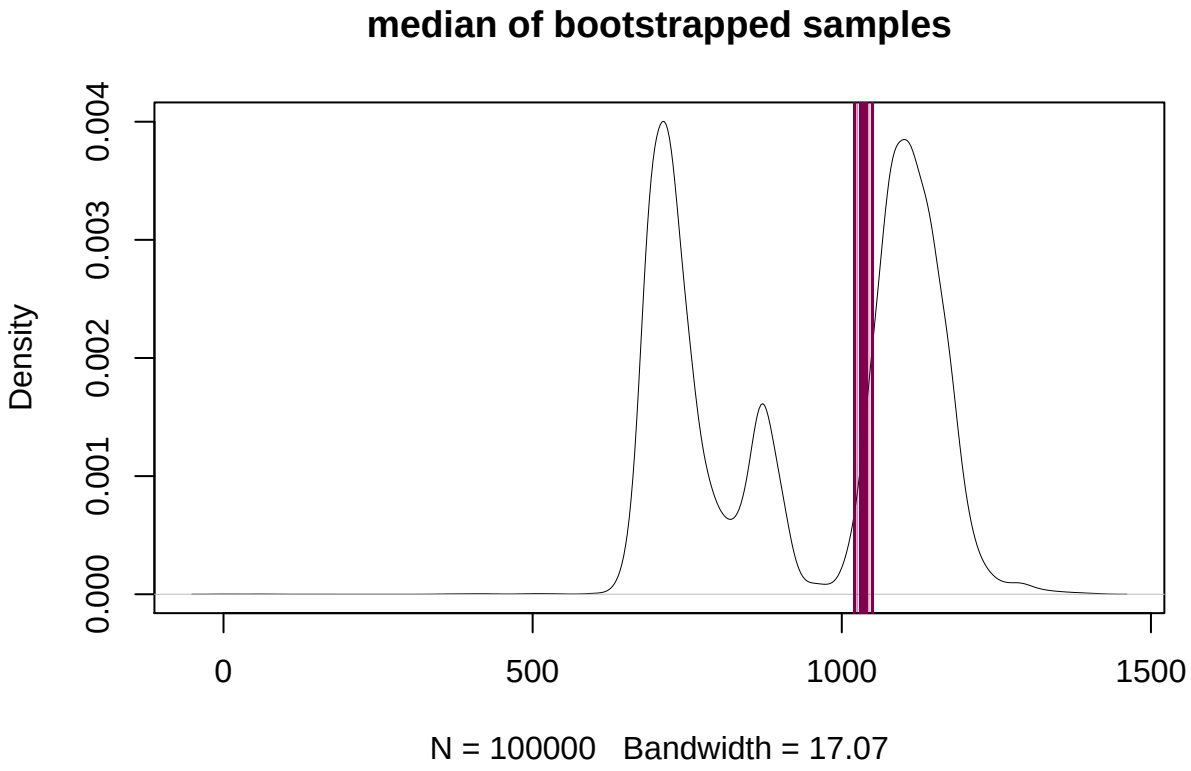
```
compute_sample_median <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  median(resample)
}
compute_sample_median(minday)
```

```
## [1] 1030
```

```
samples_medians <- replicate(2000, compute_sample_median(minday))

compute_sample_median(minday)
```

```
## [1] 1030
```

```
samples_medians <- replicate(2000, compute_sample_median(minday))

plot(density(minday), lwd=0, main="median of bootstrapped samples")
plot_resample_median<-function(sample_i) {
  abline(v=sample_i, col=rgb(0.5, 0.0, 0.3, 0.25))
}
resamples_m <- data.matrix(samples_medians)
apply(resamples_m , 2, FUN = plot_resample_median)
```




```
## NULL
```

(iii)Estimate the 95% CI of the bootstrapped medians

```
quantile(samples_medians, probs = c(0.05, 0.95))
```

```
##   5%  95%
## 1020 1050
```