HW3

106022113

Question 1 : Standardize a Vector

(a) Normal Distribution

```
# Create Normal Distribution
norm <- rnorm(1000,mean = 940, sd = 190)
# Standardization
rnorm_std <- (norm-mean(norm))/sd(norm)
```

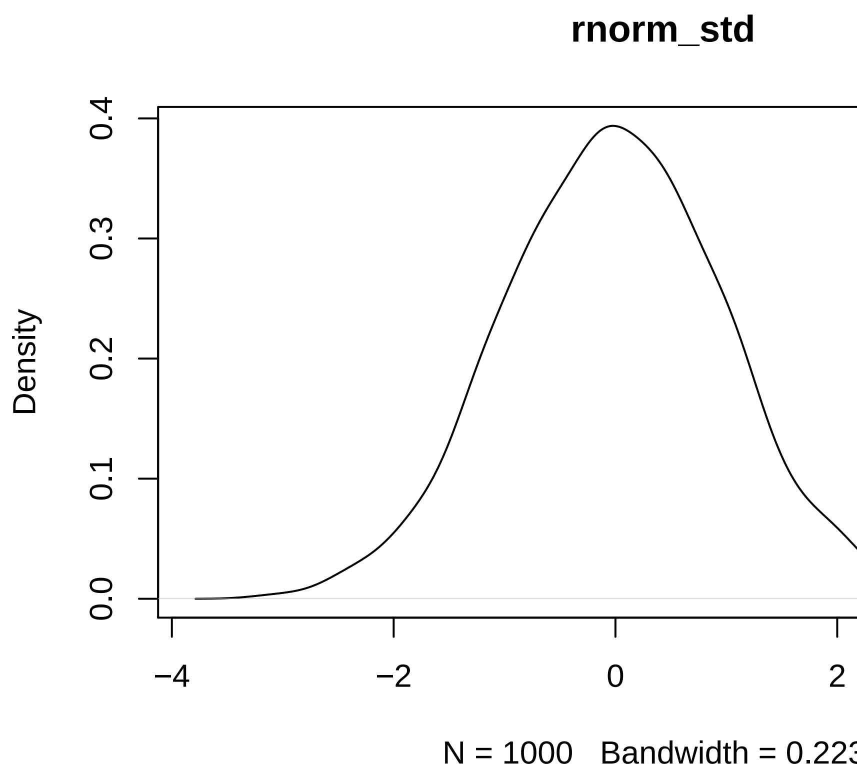
```
message("Mean: ",mean(rnorm_std), ", Standard Deviation: ", sd(rnorm_std))
```

(i) Expectation of the mean and standard deviaiton of rnorm_std

```
## Mean: -1.25801956742458e-16, Standard Deviation: 1
```

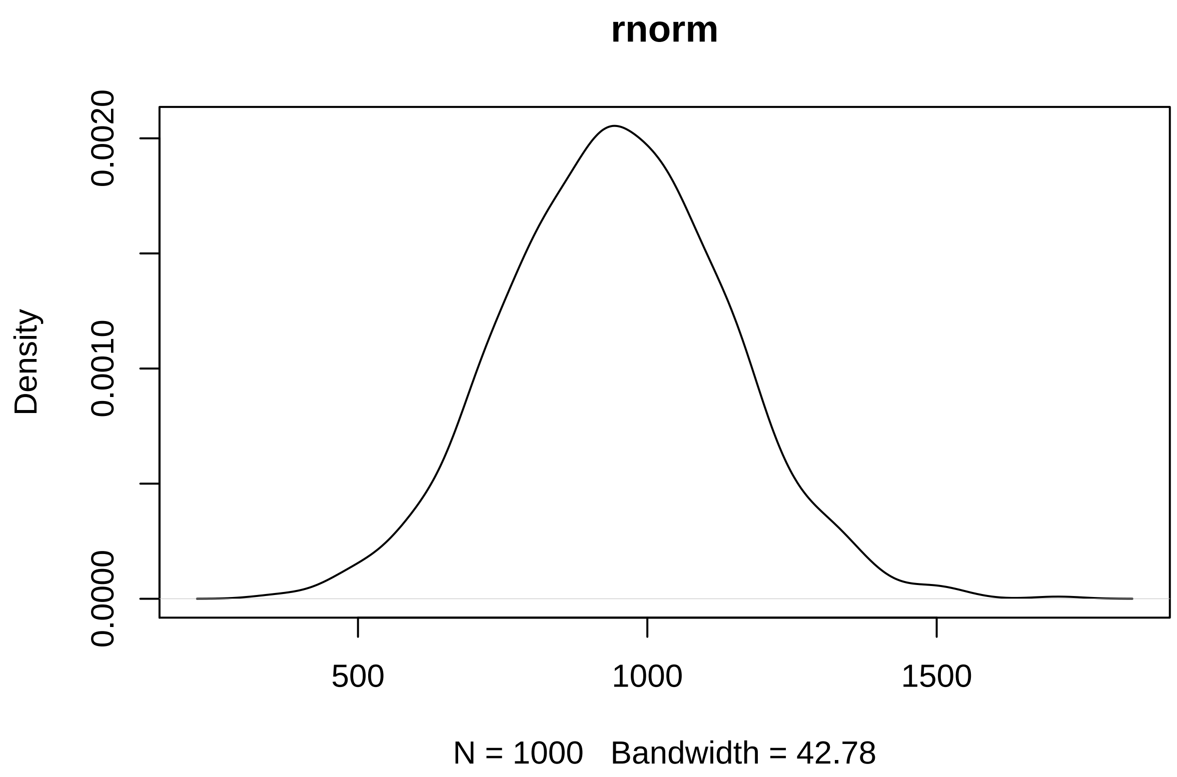
ANSWER:
Since standard normal distribution's mean is 0 and standard deviation is 1, performing standardization is to try erase the difference between the data and original mean by subtracting the mean and getting 0 as the mean of rnorm_std. Then, dividing the original data standard deviation is to transform the standardized data standard deviation to 1. Which is what the results are close to.

```
plot(density(rnorm_std),main="rnorm_std")
```



(ii) Expectation of the shape of nrorm_std

```
plot(density(norm),main="rnorm")
```



ANSWER:
It should look like a normal distribution with mean 0 and standard deviation as 1. Because the standardization will remain the normal property of the data points. Hence, with the mean and standard deviation derived in (i), we can expect the plot to be shaped as described above.

(iii) Distributions that are normal and standardized *ANSWER*: Standard Normal Distribution is the name for distributions that are standardized and normal

(b) Minday

```
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"
```

```
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
```

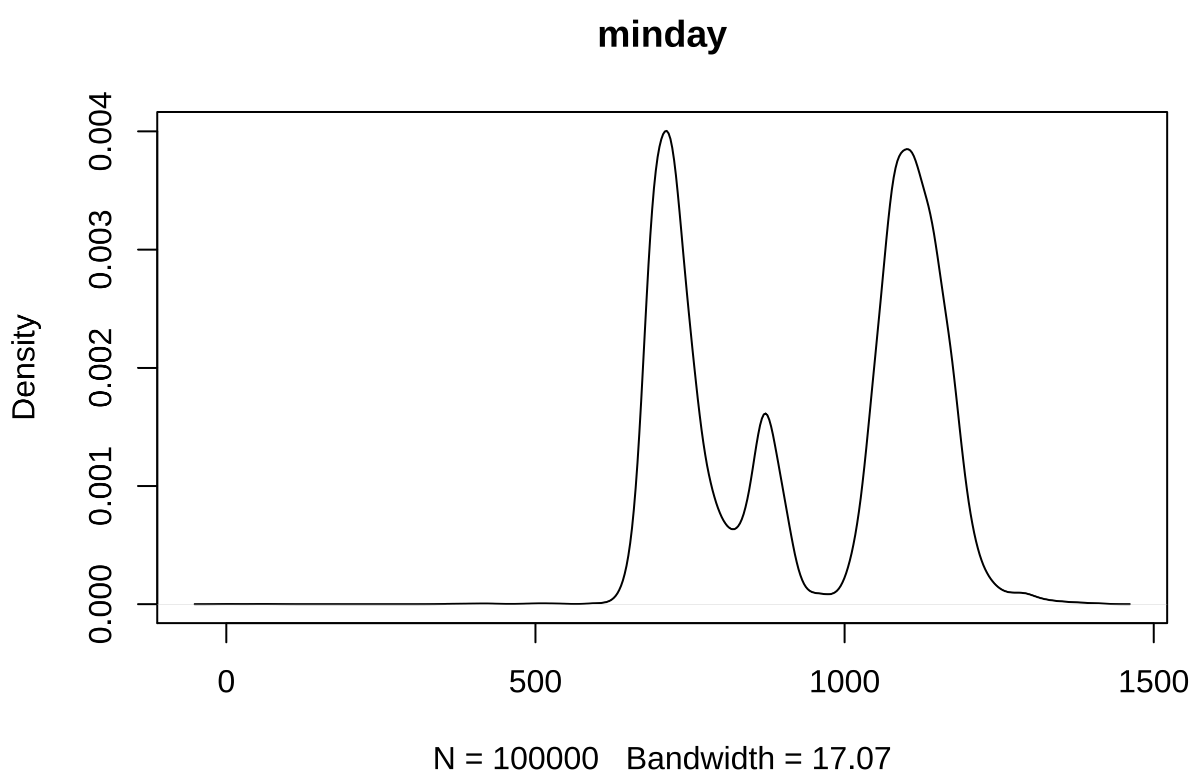
```r
minday_std <- (minday-mean(minday))/sd(minday)
paste("Mean of Minday_std: ",mean(minday_std), ", Standard Deviation: ",sd(minday_std))
```

(i)

```
## [1] "Mean of Minday_std:  -4.25589034500073e-17 , Standard Deviation:  1"
```

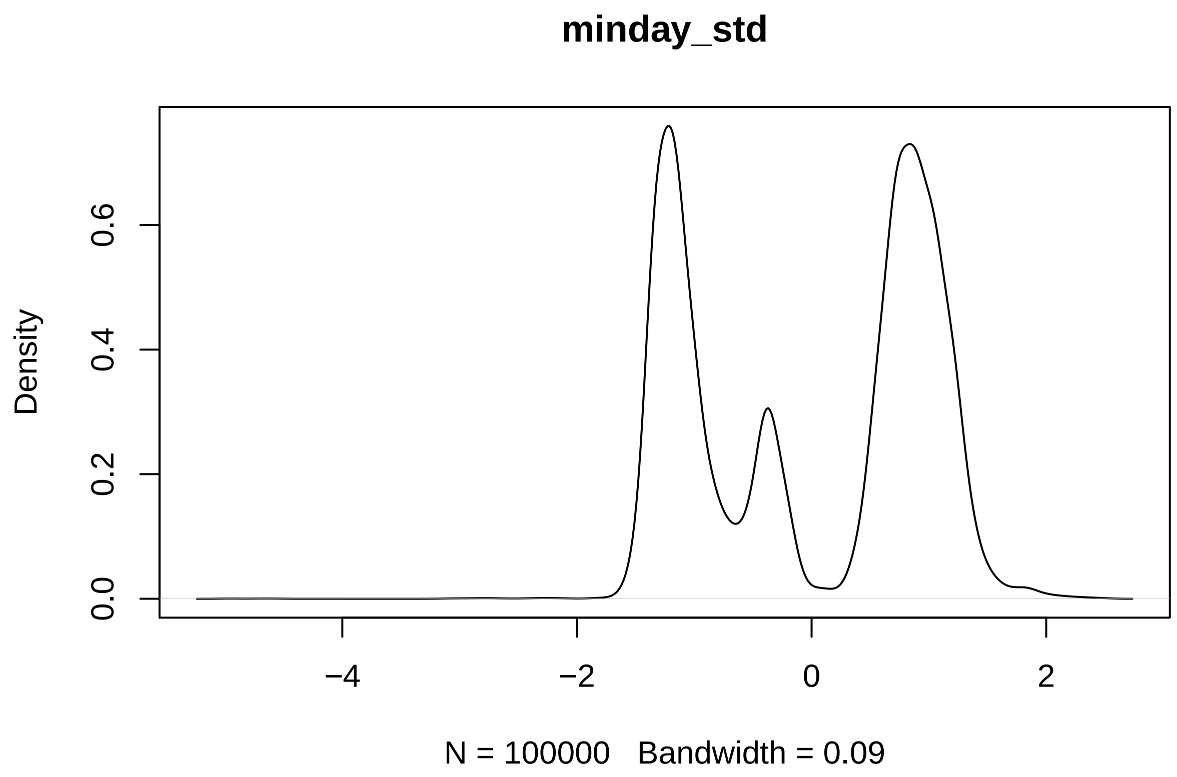
ANSWER: The mean of minday_std should be close to 0 since every element of minday is subtracted by the mean. The standard deviation should be close to 1 since every element is also divided by the original standard deviation.

```r
plot(density(minday),main = 'minday')
```



(ii)

```r
plot(density(minday_std),main = 'minday_std')
```



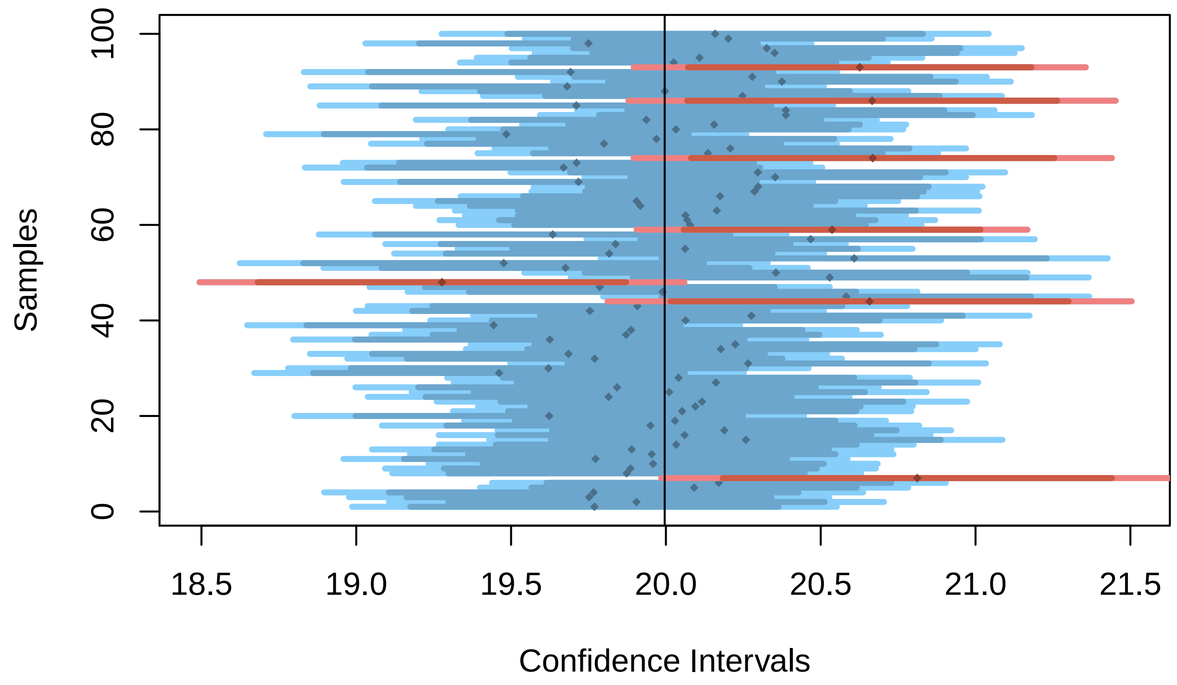
ANSWER: The shape should be the same as the original minday plot. Because the standardization process could be view as moving the plot horizontally and shrinking it's magnitude with the standard deviation. Hence, we can see that the two plots looks about the same with it's magnitude as the difference.

Question 2:

```
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)

  # 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 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
```

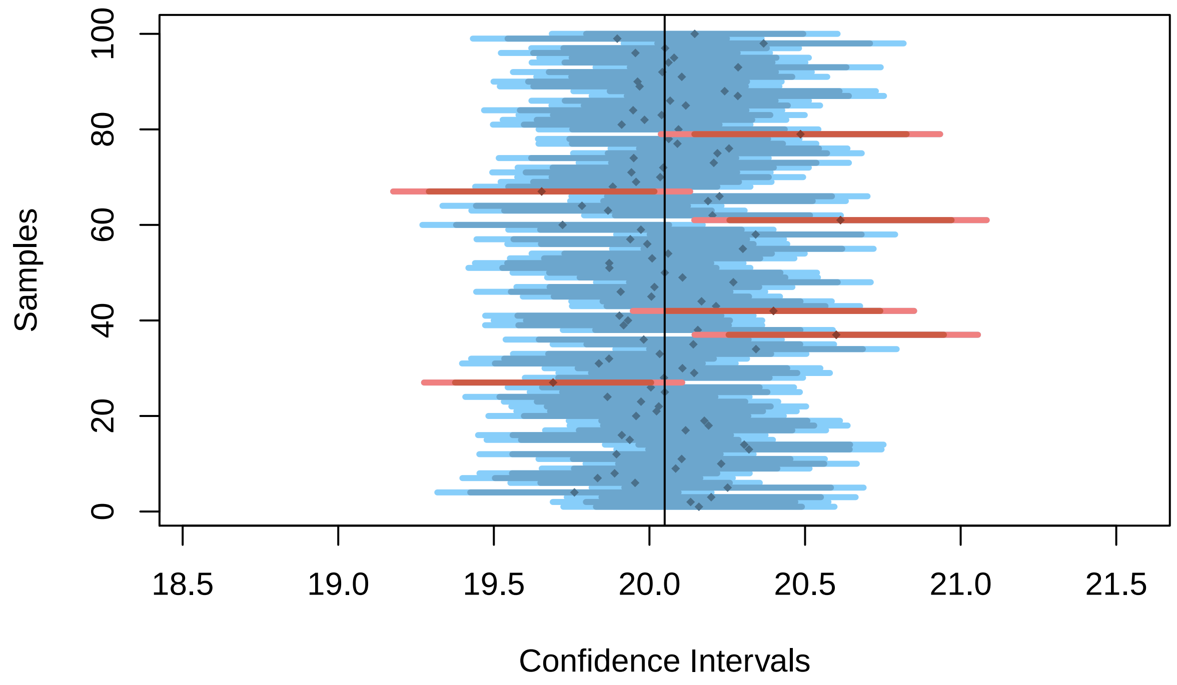


(i) Samples to NOT include the population mean in its 95% CI? $ANSWER: 100 - 100 * 0.95 = 5$
With 95% confidence we can assume that 95% of the samples will cover the population mean.

(ii) Samples to NOT include the population mean in its 99% CI? $ANSWER: 100 - 100 * 0.99 = 1$

(b) Rerun with 300 samples

```
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size=30000, distr_func=rnorm, mean=20, sd=
```



(i) Do we expect their 95% and 99% CI to become wider or narrower than before? *ANSWER*: NI expect the confidence interval to be narrower. It's because that with the increase of sample size, the error of the samples will be smaller due to the law of large numbers. Also, if we observe the formula to calculate the confidence interval

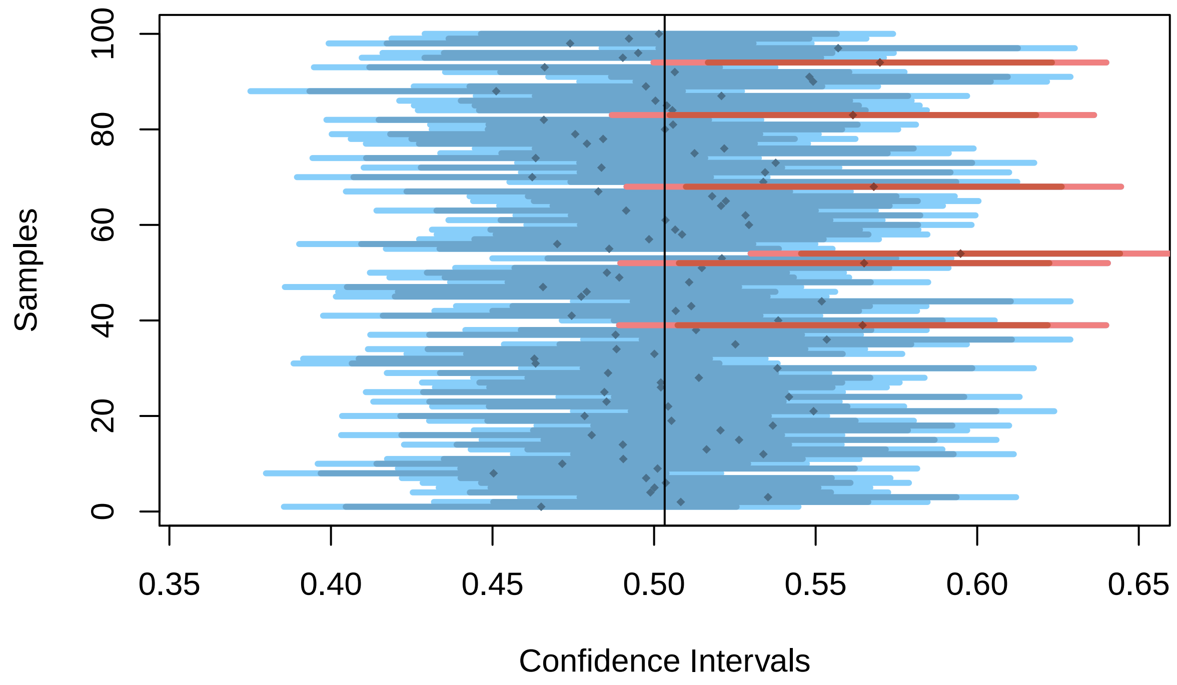
$$\mu + z * \frac{s}{\sqrt{n}}$$

We can still see that the confidence interval will be smaller with larger sample size.

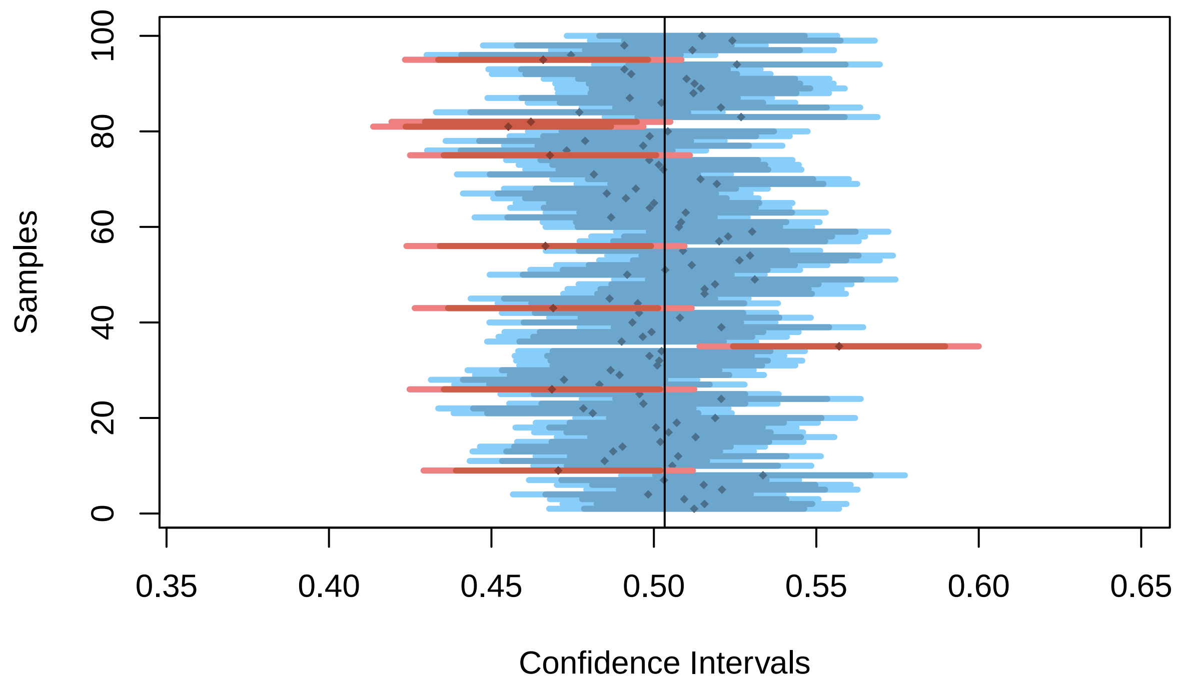
(ii) How many samples would we expect to NOT include the population mean in its 95% CI? *ANSWER*: $100 - 100 * 0.95 = 5$

(c) Run the sample using uniform distribution, do you expect (a),(b) answers to change?

```
#(a) Sample Size 100 with uniform
visualize_sample_ci(num_samples = 100, sample_size = 100, pop_size=10000, distr_func=runif)
```



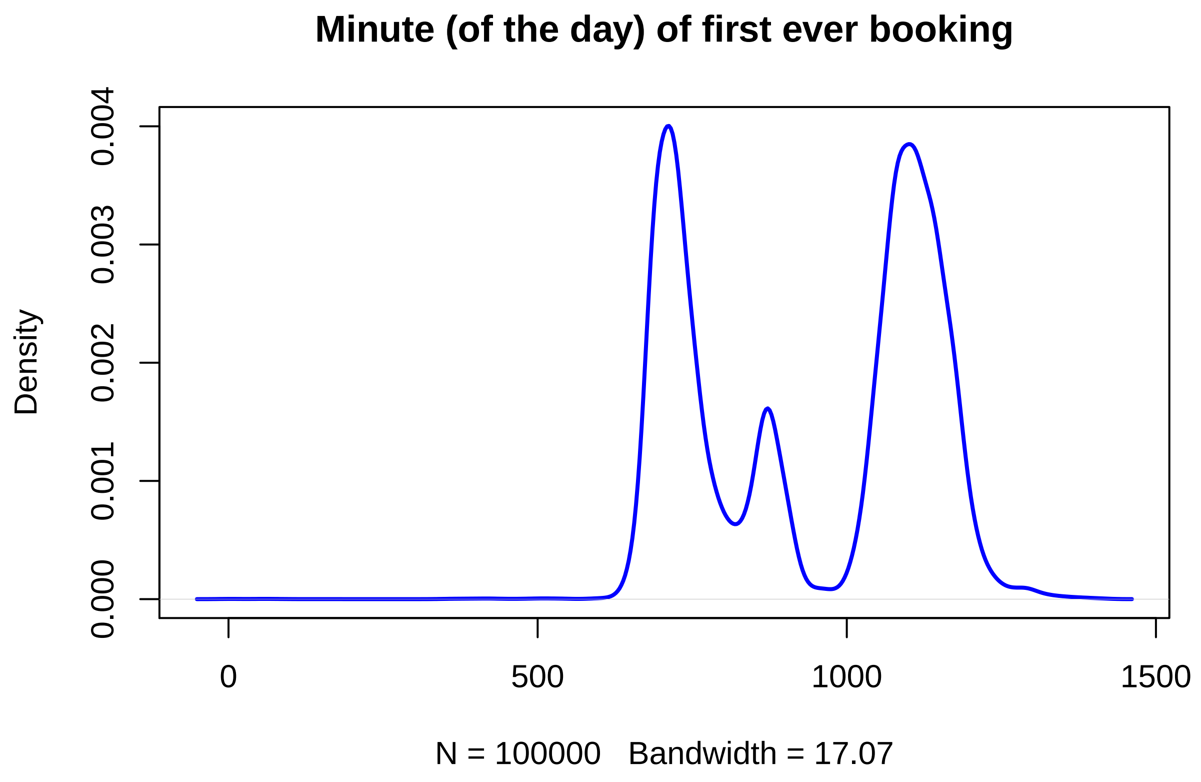
```
#(b) Sample Size 300 with uniform
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size=30000, distr_func=runif)
```



ANSWER: Since the distribution isn't normal, but follows a distribution with same probability, the confidence interval for larger sample size will not alter. Hence, we can see that the red lines for the two plots are approximately the same. Moreover, it will not follow the rules of being normal, so it will definitely be different from (a), (b).

Question 3 EZTABLE

```
bookings <- read.table("first_bookings_datetime_sample.txt", header=TRUE)
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)
```

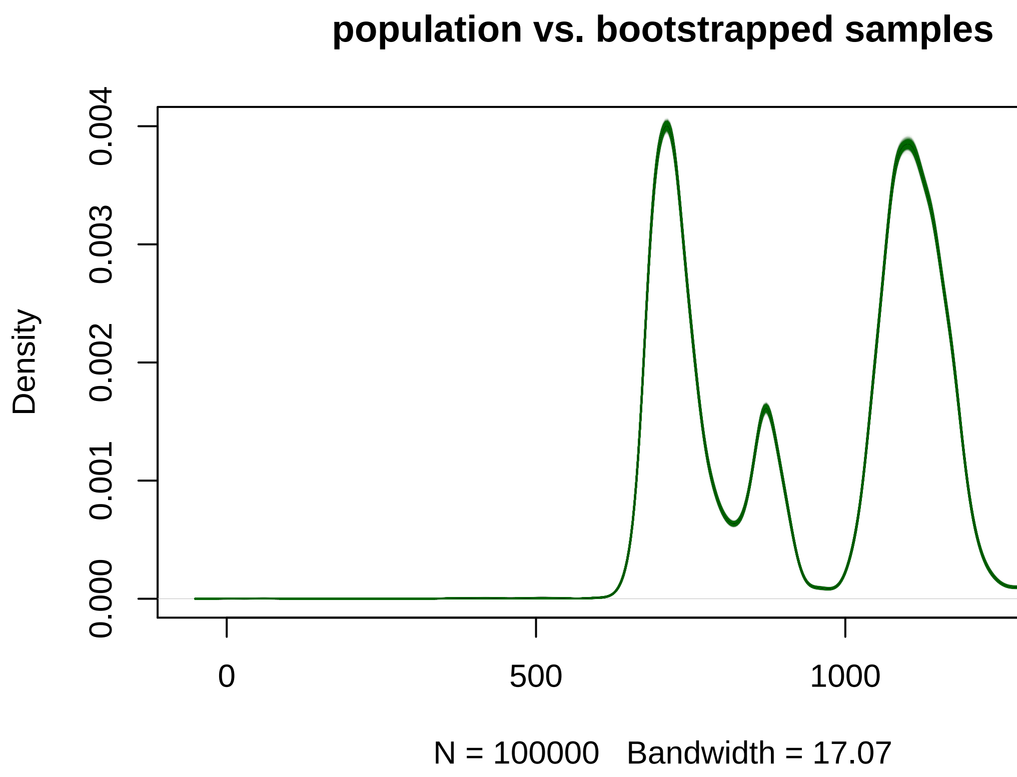


(a) Average booking time for new members

```
#paste("Mean: ", mean(minday),", Standard Error ",sd(minday)/(sqrt(length(minday)), "95% Confidence Int
```

(i) Population Mean, Standard Error, 95% CI

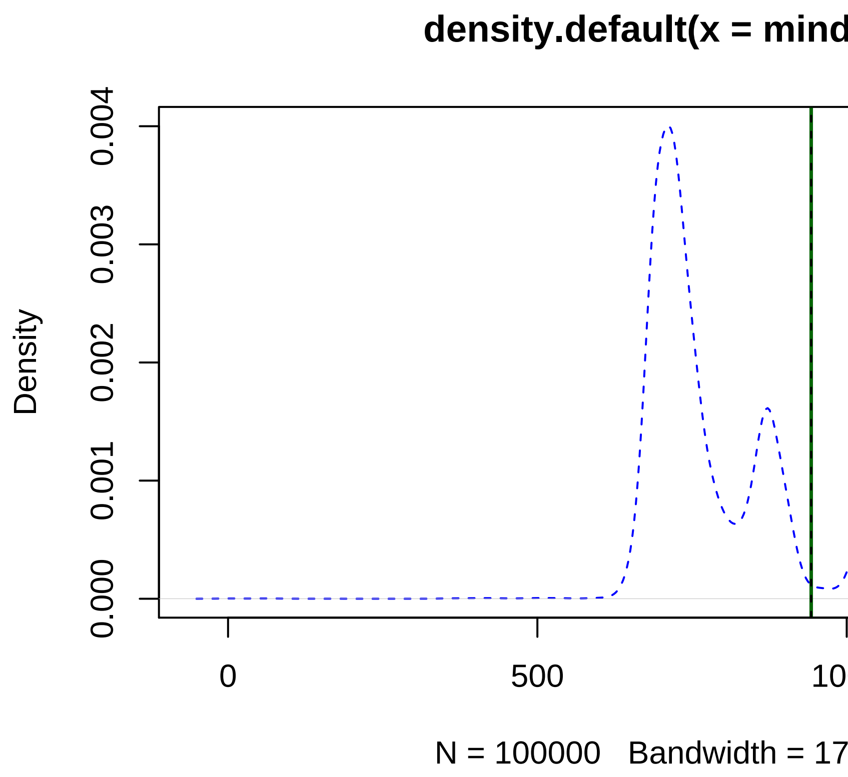
```
resamples <- replicate(2000,sample(minday,length(minday),replace = TRUE))
plot(density(minday),lwd=0,lim=c(0, 0.009), main="population vs. bootstrapped samples")
plot_resample_density <- function(sample_i){
  lines(density(sample_i),col=rgb(0.0, 0.4, 0.0, 0.01))
}
#Draw the plot!
apply(resamples,2, FUN = plot_resample_density)
```



(ii) Produce 2000 new samples

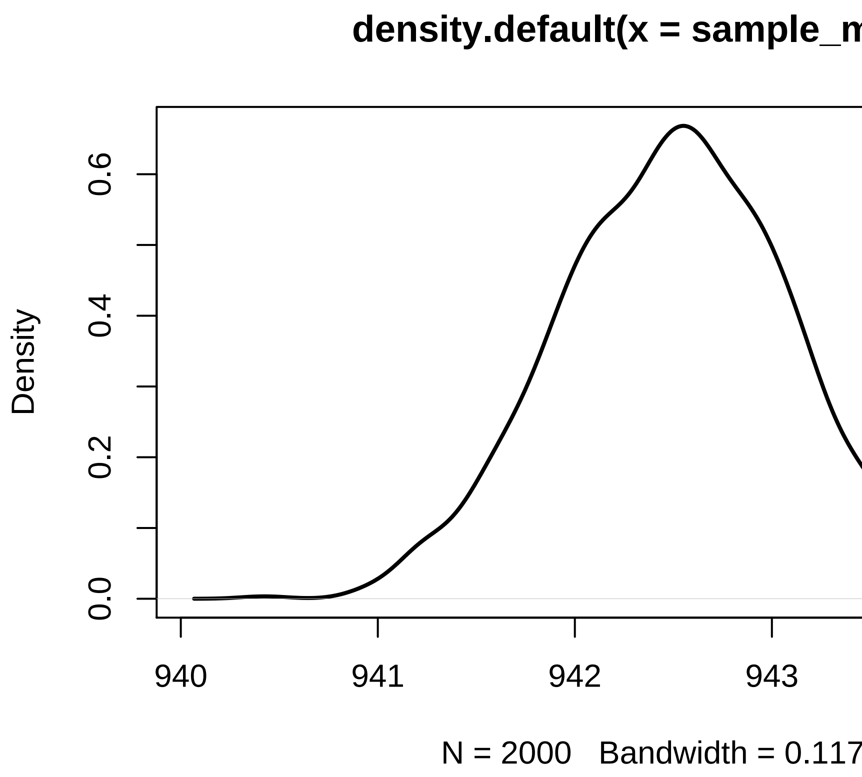
```
## NULL
```

```
plot(density(minday),col = 'blue',lty = "dashed")
plot_resample_mean <- function(sample_i){
  abline(v=mean(sample_i),col = rgb(0.0,0.4,0.0,0.01))
  return (mean(sample_i))
}
#draw all bootstrap sample means
sample_means <- apply(resamples,2,FUN = plot_resample_mean)
#draw minday mean
abline(v = mean(minday),lty="dashed")
```



(iii) Visualize the means of the new samples

```
# The range of sample Mean
plot(density(sample_means),lwd=2)
```



(iv) Estimate the 95% Confidence Interval

```
# 95% Confidence Interval:
quantile(sample_means,probs = c(0.05,0.95))
```

```
##       5%      95%
## 941.5341 943.5053
```

(b) Half of the members arrive at restaurant?

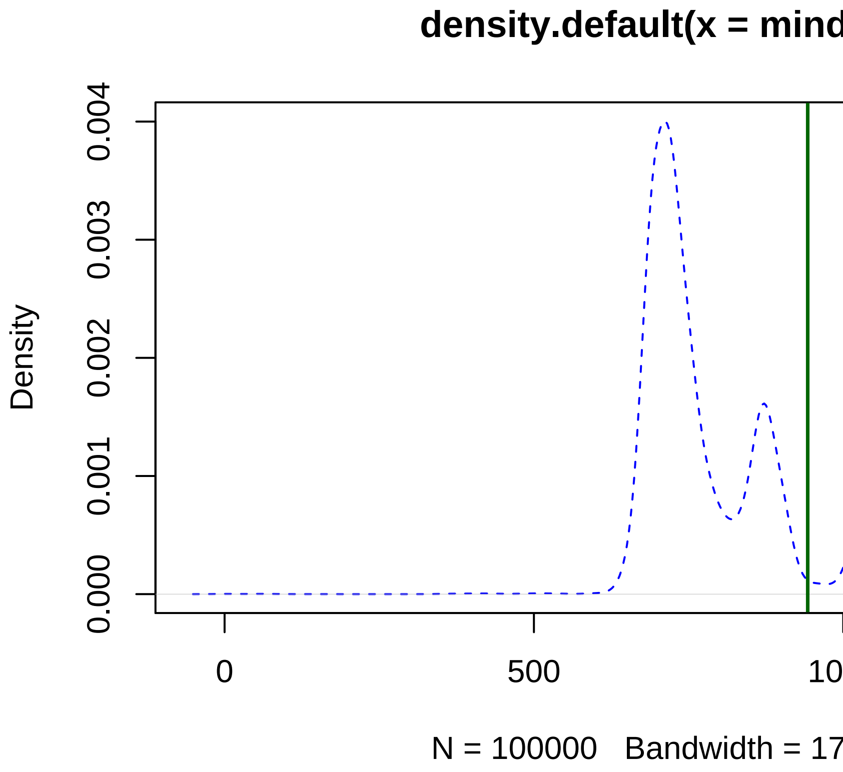
```
paste("Median of Minday: ",median(minday))
```

(i) Estimate Median

```
## [1] "Median of Minday:  1040"
```

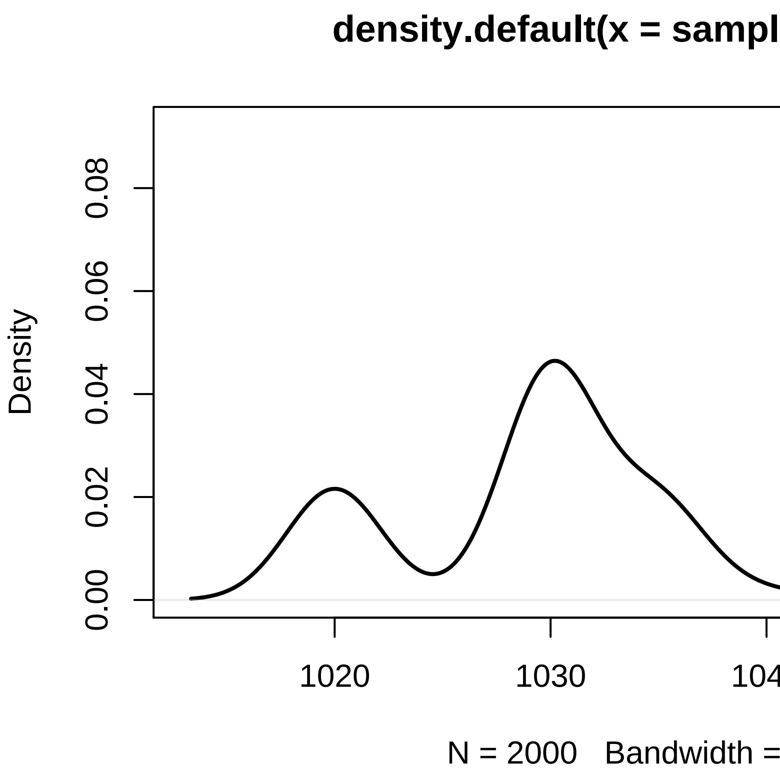
```
plot(density(minday),col = 'blue',lty = "dashed")
plot_resample_median <- function(sample_i){
  abline(v=mean(sample_i),col = rgb(0.0,0.4,0.0,0.01))
  return (median(sample_i))
}
```

```
#draw all bootstrap sample medians
sample_medians <- apply(resamples,2,FUN = plot_resample_median)
#draw minday median
abline(v = median(minday),lty="dashed")
```



(ii) Visualize Medians of Bootstrap samples

```
# The range of sample median
plot(density(sample_medians),lwd=2)
```



(iv) 95% Confidence Interval of Sample Medians

```
# 95% Confidence Interval:
quantile(sample_medians,probs = c(0.05,0.95))
```

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