HW4_108020033

Che-Wei, Chang

2023-03-09 helped by 108020031

Question 1) Let's reexamine how to standardize data: subtract the mean of a vector from all its values, and divide this difference by the standard deviation to get a vector of standardized values.

```r
# Create a standard function to standardize the data
standardize <- function(numbers) {
  numbers <- (numbers - mean(numbers)) / sd(numbers)
  return(numbers)
}
```

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?

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

```r
# Create the data set rnorm_std
rnorm_part_a <- rnorm(n = 300, mean = 940, sd = 190)
rnorm_std <- standardize(rnorm_part_a)
mean(rnorm_std)
```

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

```
## [1] -1.034517e-16
```

```r
sd(rnorm_std)
```

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

Ans:

1. When we let the data minus mean and divide the standard deviation, we do the standardization. Therefore, we will expect the mean is 0 standard deviation is 1

2. Since we do standardization, rnorm_std will be bell - shaped curve and its mean is 0 and standard deviation is 1

3. We generally call distributions that are normal and standardized "standard normal distributions".

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?

```
# read the table
bookings <- read.table("first_bookings_datetime_sample.txt", header = TRUE)
bookings$datetime[1 : 9]
```

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

```
## [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"
```

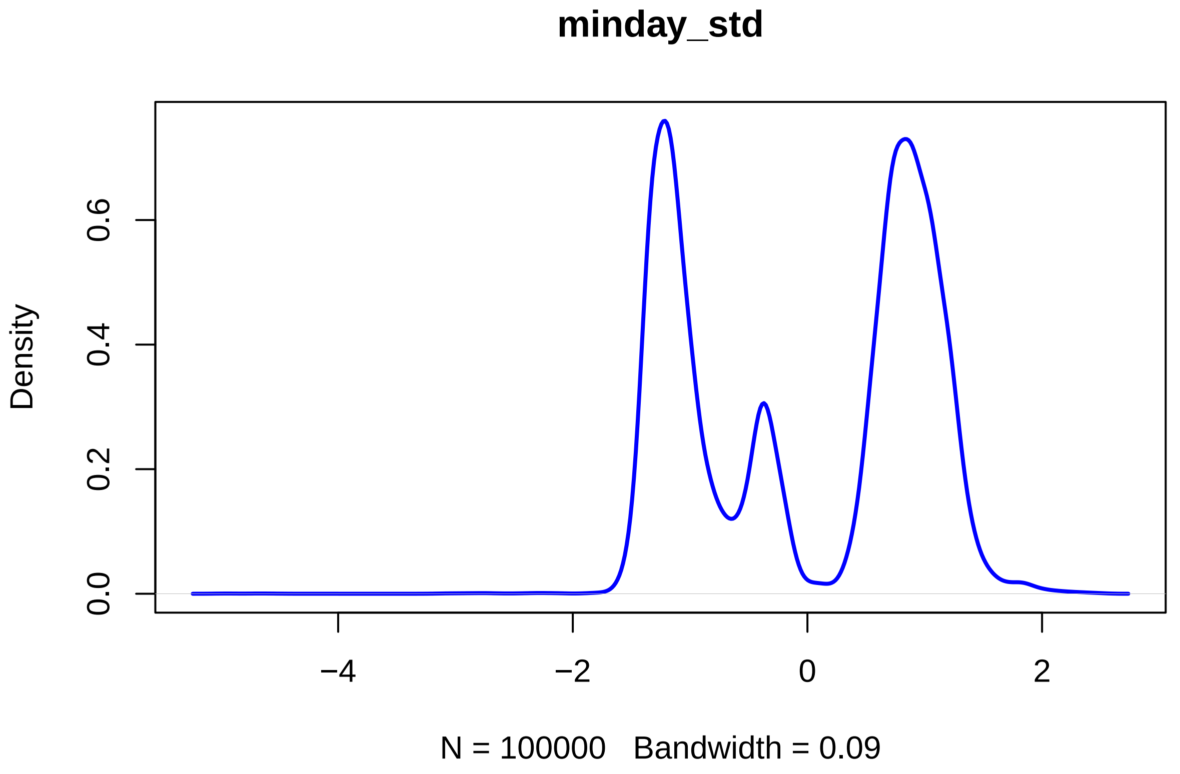
```
hours <- as.POSIXlt(bookings$datetime, format = "%m/%d/%Y %H:%M")$hour
mins <- as.POSIXlt(bookings$datetime, format = "%m/%d/%Y %H:%M")$min
minday <- hours * 60 + mins

# do standardization
minday_std <- standardize(minday)
mean(minday_std)
```

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

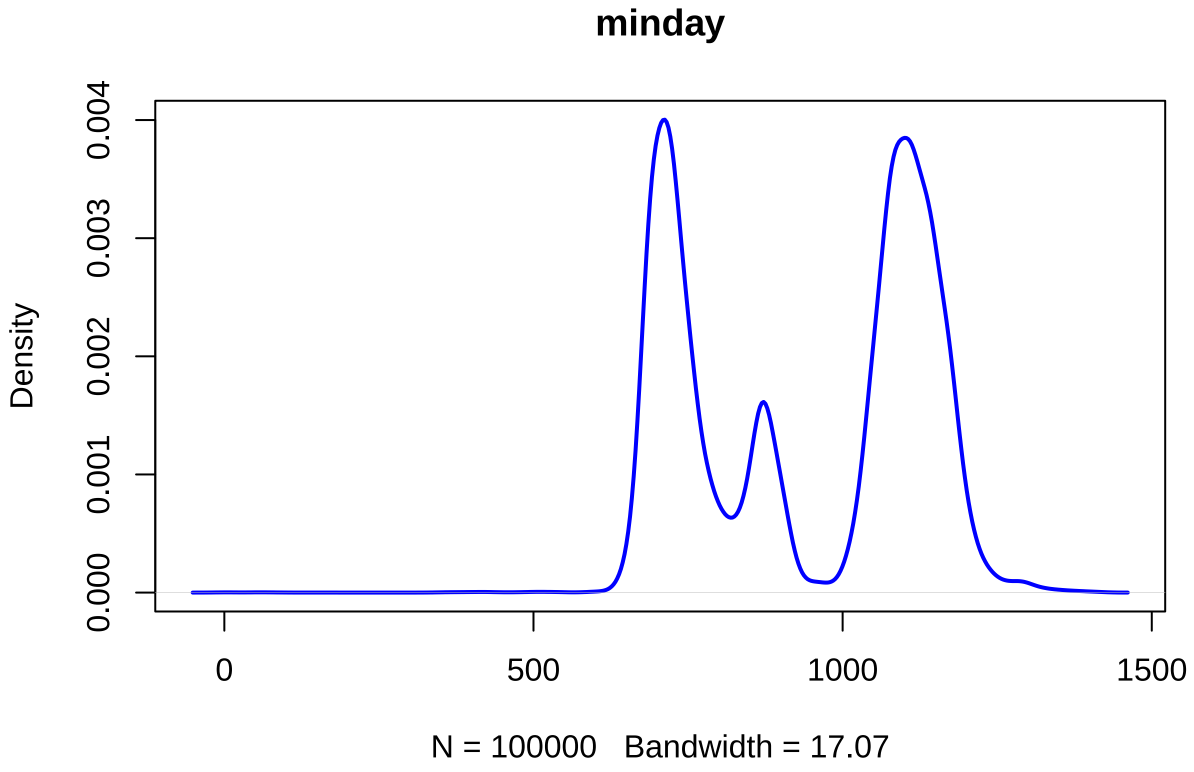
```
sd(minday_std)
```

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

```
# show the plot
plot(density(minday_std), main = "minday_std", col = "blue", lwd = 2)
```



```r
plot(density(minday), main = "minday", col = "blue", lwd = 2)
```



Ans:

1. When we let the data minus mean and divide the standard deviation, we do the standardization. Therefore, we will expect the mean is 0 standard deviation is 1.

2. Because the standardization won't change the shape, it looks like same as minday. Some differences are that the mean changes to 0 and standard deviation changes to 1.

Question 2) Install the compstatslib package from Github (see class notes) and run the plot_sample_ci() function that simulates samples drawn randomly from a population. Each sample is a horizontal line with a dark band for its 95% CI, and a lighter band for its 99% CI, and a dot for its mean. The population mean is a vertical black line. Samples whose 95% CI includes the population mean are blue, and others are red.

```
# import the compstatslib
library(compstatslib)
```

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

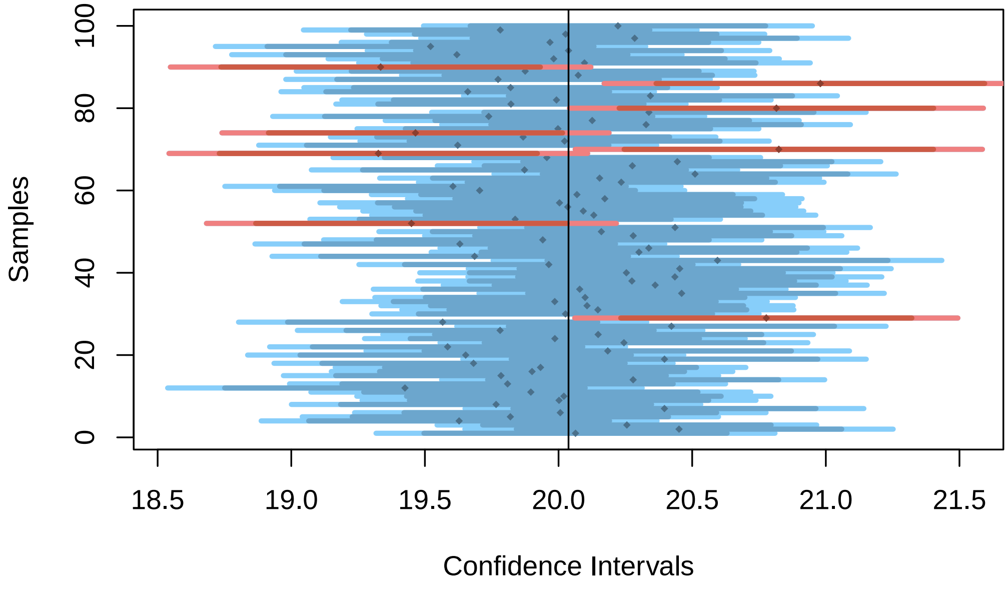
plot__sample__ci(num__samples = 100, sample__size = 100, pop__size=10000,

distr__func=rnorm, mean=20, sd=3)

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

```
plot_sample_ci(num_samples = 100, sample_size = 100, pop_size = 10000, distr_func = rnorm, mean = 20, s
```

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



Ans:

1. Since 100 x 100 x (1 - 0.95) = 500, I think that there are 500 samples NOT include the population mean in its 95% CI

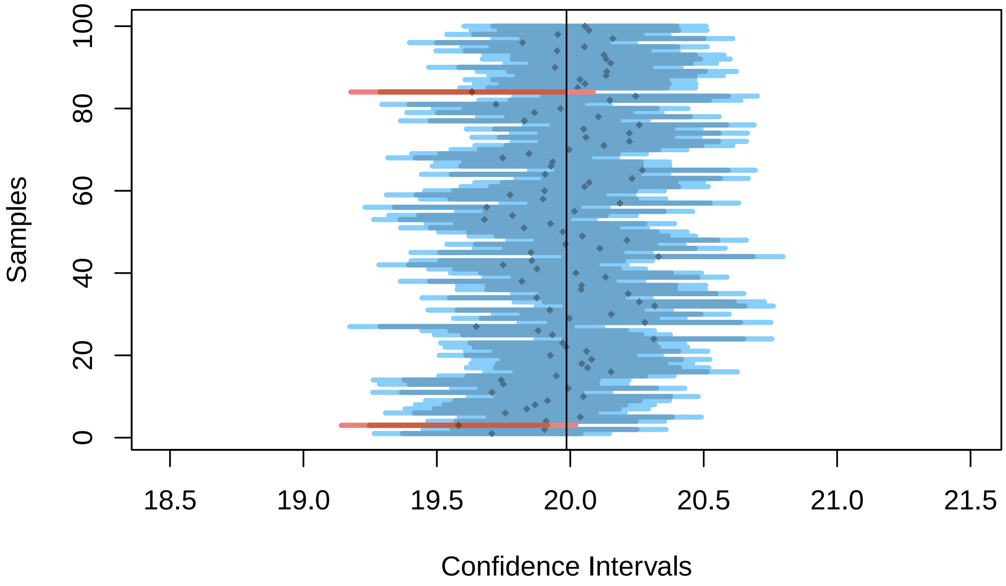
2. Since 100 x 100 x (1 - 0.99) = 100, I think that there are 100 samples NOT include the population mean in its 99% CI

b) Rerun the previous simulation with the same number of samples, but larger sample size (sample_size=300):

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?

```
plot_sample_ci(num_samples = 100, sample_size = 300, pop_size = 10000, distr_func = rnorm, mean = 20, s
```

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



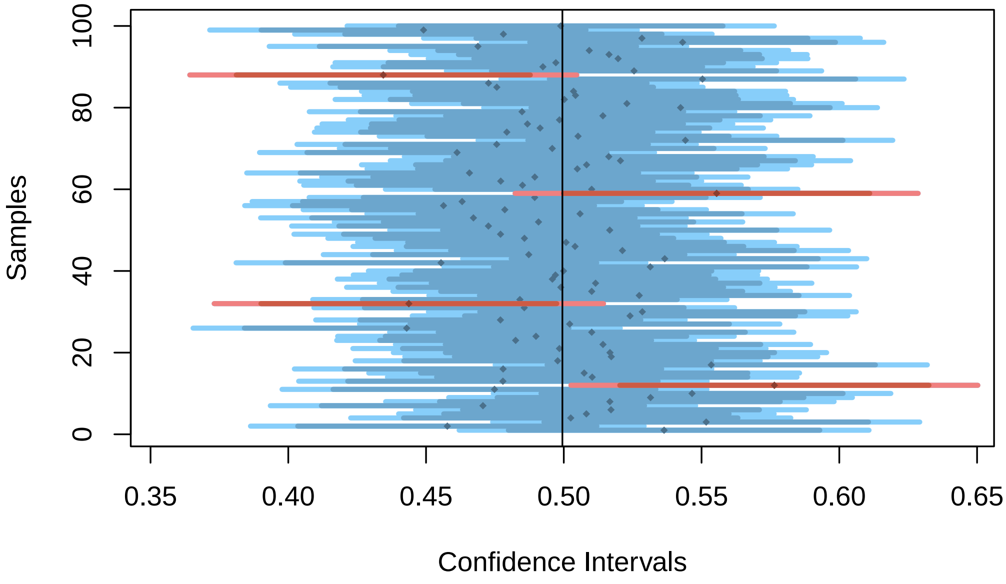
Ans:

1. By the plot, we can see that when we increase the size of each sample, 95% and 99% CI will become narrower than before.

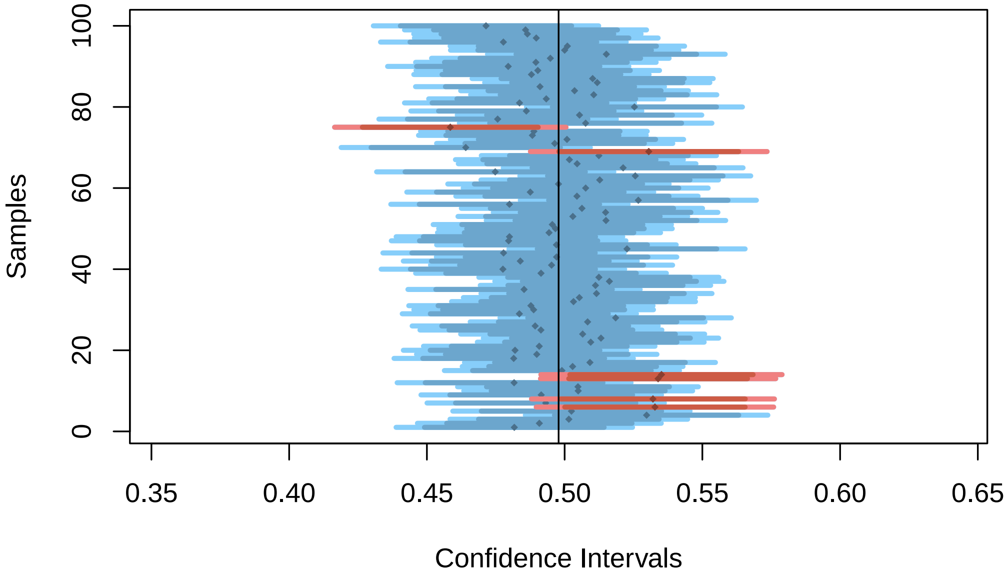
2. 300 x 100 x (1 - 0.95) = 1500

```
plot_sample_ci(num_samples = 100, sample_size = 100, pop_size = 10000, distr_func = runif)
```

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



```
plot_sample_ci(num_samples = 100, sample_size = 300, pop_size = 10000, distr_func = runif)
```



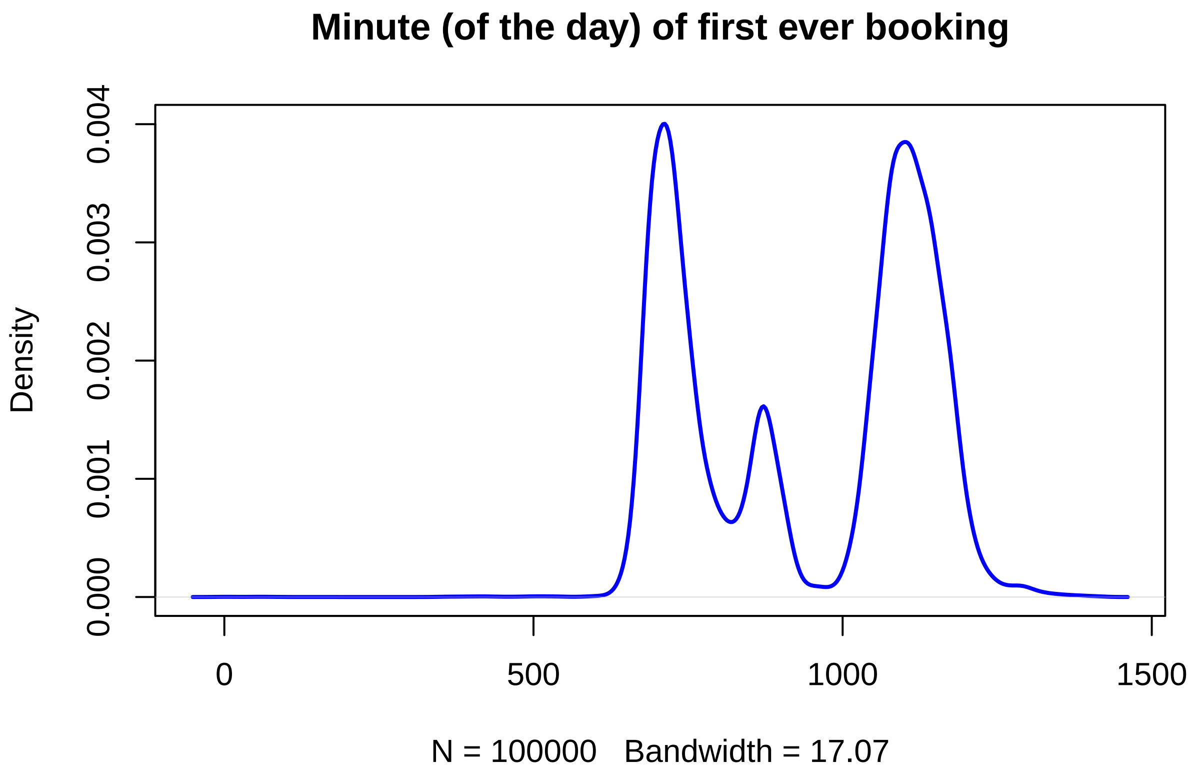
Ans: When the number of the samples are same, the standard deviation of uniform distribution is smaller than the standard deviation of normal distribution. Therefore, the confidence interval of uniform distribution is smaller than confidence interval of normal distribution.

Question 3) The company EZTABLE has an online restaurant reservation platform that is accessible by mobile and web. Imagine that EZTABLE would like to start a promotion for new members to make their bookings earlier in the day.We have a sample of data about their new members, in particular the date and time for which they make their first ever booking (i.e., the booked time for the restaurant) using the EZTABLE platform. Here is some sample code to explore the data:

```r
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"
```

```r
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?

(use minday, which is the absolute minute of the day from 0-1440)

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

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

iii) Visualize the means of the 2000 bootstrapped samples

```
# i)
mean_val <- mean(minday)
std <- sd(minday)
n <- length(minday)

# set the significance level be 0.05
alpha <- 0.05
t_cri <- qt(1 - alpha/2, df = n - 1)

# count the CI
lower_bound <- mean_val - t_cri * std / sqrt(n)
upper_bound <- mean_val + t_cri * std / sqrt(n)

# print the mean and standard error
cat("mean: ", mean_val, "\n")
```

iv) Estimate the 95% CI of the bootstrapped means using the quantile function

```
## mean:  942.4964
```

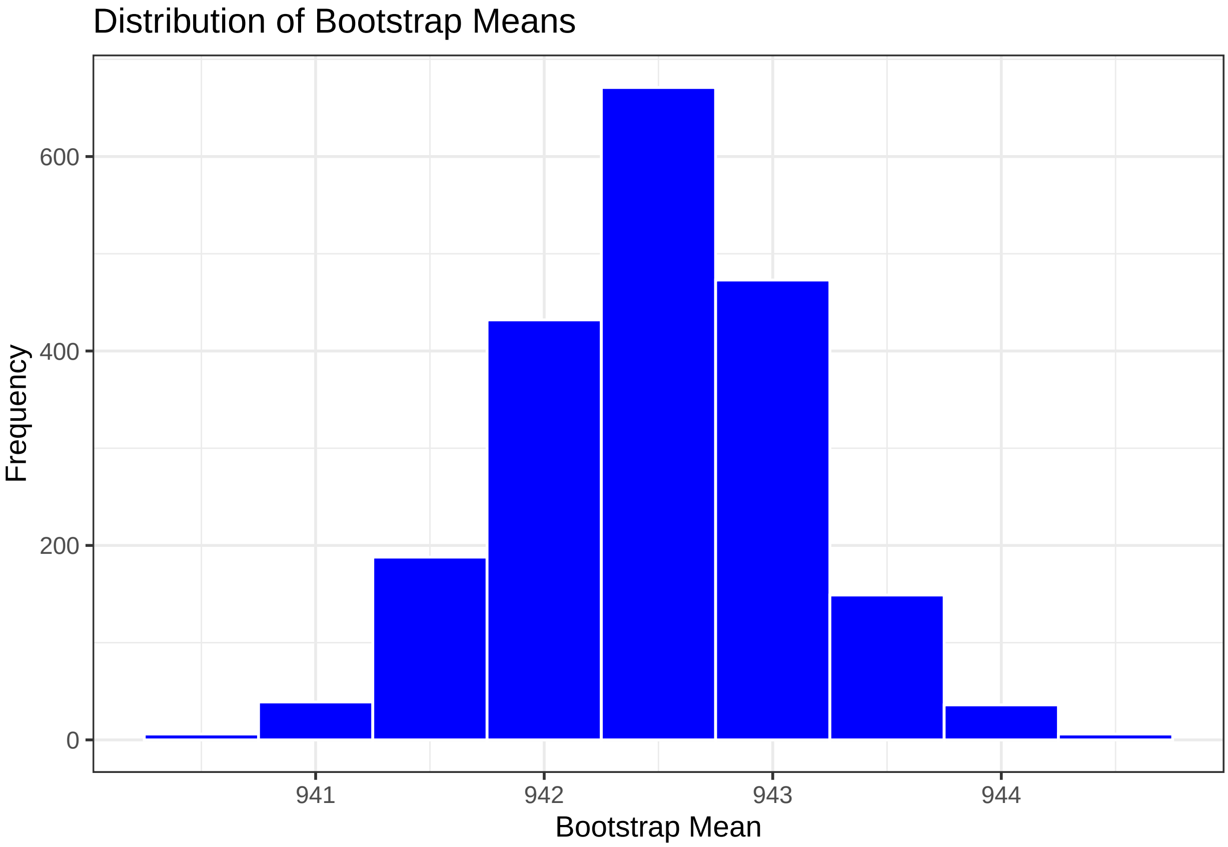
```
cat("standard error: ", std,"\n")
```

```
## standard error:  189.6631
```

```
# print the 95% CI
cat("The 95% CI of minday: [", lower_bound, ",", upper_bound, "].\n")
```

```
## The 95% CI of minday: [ 941.3208 , 943.6719 ].
```

```
# ii)
bootstrap_samples <- replicate(2000, sample(minday, replace = TRUE))

# iii)
# import the liberary ggplot2
library(ggplot2)

# count the mean of the bootstrap samples
bootstrap_means <- apply(bootstrap_samples, 2, mean)

# create a data frame
df <- data.frame(mean = bootstrap_means)

# show the result
ggplot(df, aes(x = mean)) +
  geom_histogram(binwidth = 0.5, color = "white", fill = "blue") +
  labs(x = "Bootstrap Mean", y = "Frequency",
       title = "Distribution of Bootstrap Means") +
  theme_bw()
```



```
# iv)
# count the 95% confidence intervals of bootstrap_means
CI <- quantile(bootstrap_means, c(0.025, 0.975))

# print the result
cat("95% CI: [", round(CI[1], 2), ", ", round(CI[2], 2), "]")
```

```
## 95% CI: [ 941.27 ,  943.68 ]
```

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

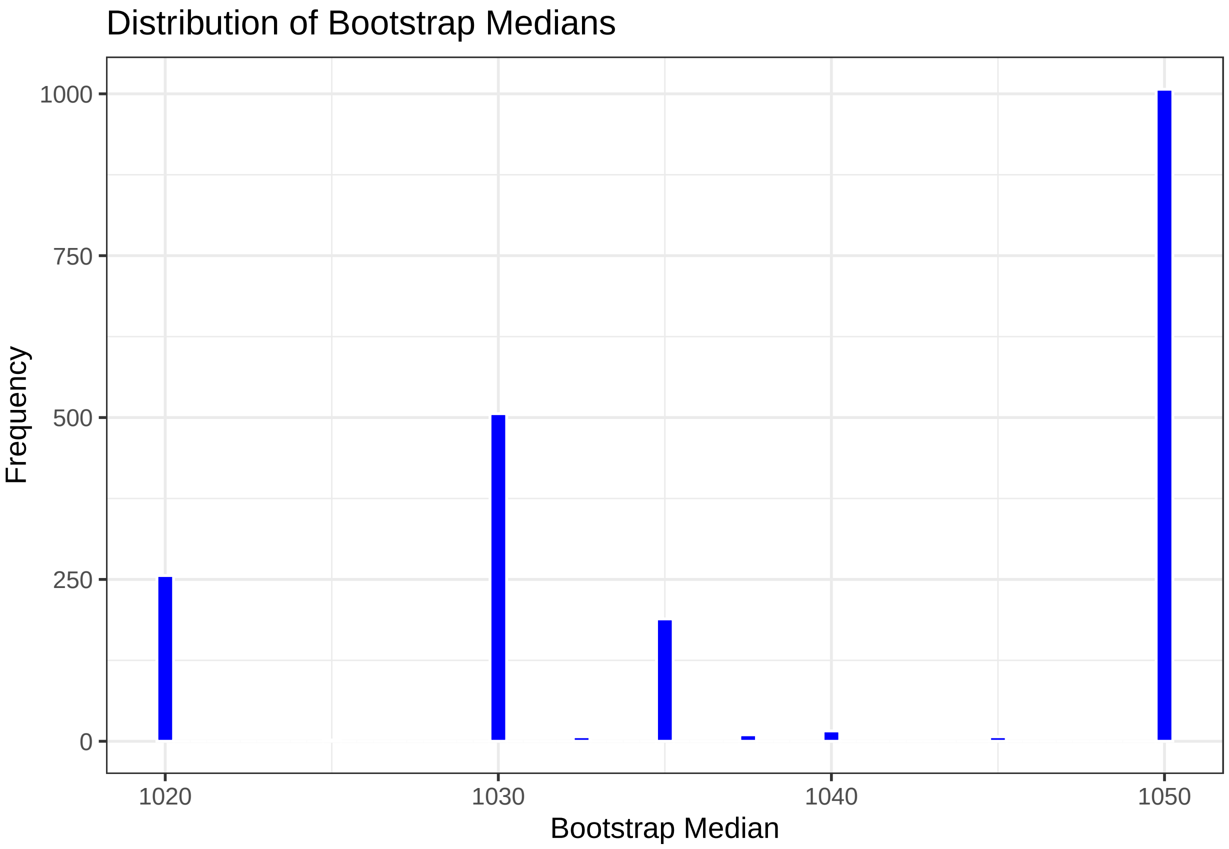
ii) Visualize the medians of the 2000 bootstrapped samples

```
# i)
# count the median of minday
median_of_minday <- median(minday)
cat("median of minday: ", median_of_minday,"\n")
```

iii) Estimate the 95% CI of the bootstrapped medians using the quantile function

```
## median of minday:  1040
```

```
# ii)
# count medians of every bootstrap samples
bootstrap_medians <- apply(bootstrap_samples, 2, median)

# create the data frame
df <- data.frame(median = bootstrap_medians)

# show the result
ggplot(df, aes(x = median)) +
  geom_histogram(binwidth = 0.5, color = "white", fill = "blue") +
  labs(x = "Bootstrap Median", y = "Frequency",
       title = "Distribution of Bootstrap Medians") +
  theme_bw()
```



```r
# iii)
# count the 95% CI of bootstrap_medians
ci <- quantile(bootstrap_medians, c(0.025, 0.975))

# print the result
cat("95% CI: [", round(ci[1], 2), ", ", round(ci[2], 2), "]", "\n")
```

```
## 95% CI: [ 1020 ,  1050 ]
```