BACS HW (Week 4)

108020024

due on 03/12 (Sun)

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.

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

```
# Create a normal distribution (mean=940, sd=190)
rnorm_std <- rnorm(n=1000, mean=940, sd=190)

standardize <- function(numbers) {
numbers <- (numbers - mean(numbers)) / sd(numbers)
return(numbers)
}

# Combining them into a composite dataset
rnorm_std <- standardize(rnorm_std)
```

i) What should we expect the mean and standard deviation of standardized to have a mean of 0 and a standard deviation of 1 to be, and why?

```
mean(rnorm_std)
```

```
## [1] 4.213958e-17
```

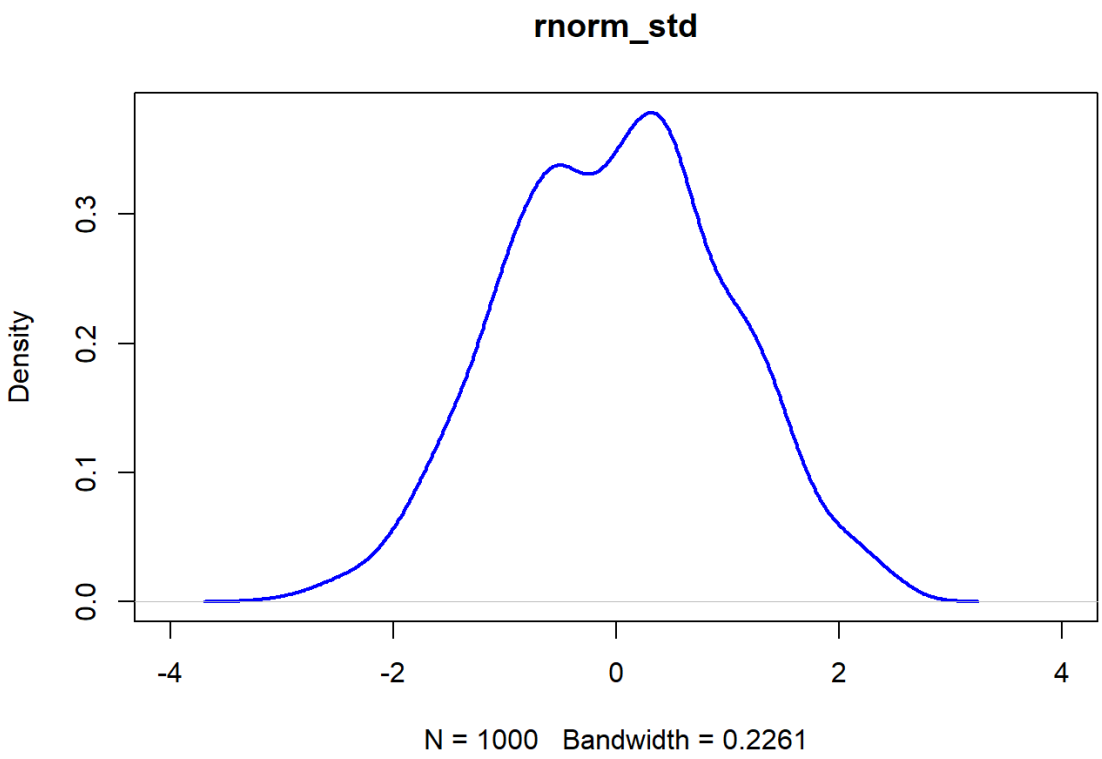
```
sd(rnorm_std)
```

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

The mean and standard deviation of rnorm_std should be 0 and 1. The reason is because rnorm_std had been standardized, and this means it will have a mean of 0 and a standard deviation of 1, this is what standardization do.

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

```
# Let’s plot the density function
plot(density(rnorm_std), col="blue", lwd=2,main = "rnorm_std", xlim = c(-4,4))
```



It should look like a standard normal distribution, bell shape.It's because we first create a normal distribution with mean=940 and sd=190, and standardized it so rnorm_std should look like a standard normal distribution.

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

Standard normal distribution.

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

```
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
minday_std <- standardize(minday)
```

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

```
mean(minday_std)
```

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

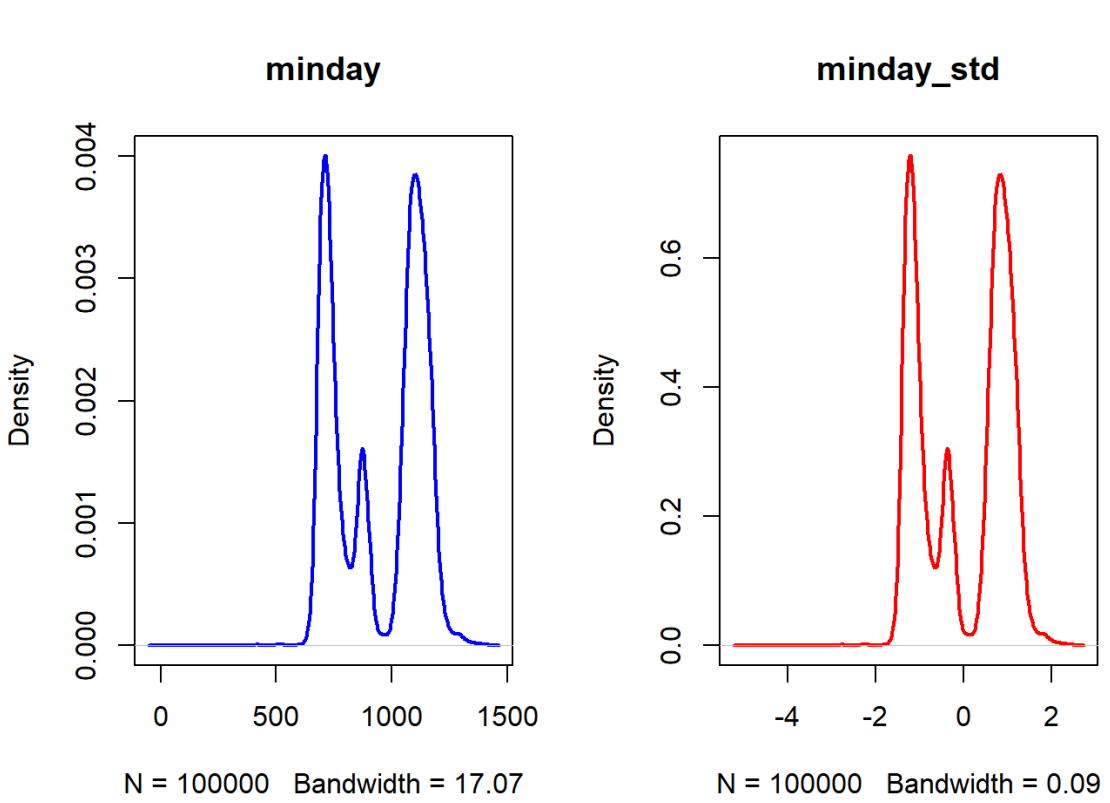
```
sd(minday_std)
```

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

The mean and standard deviation of minday_std should be 0 and 1. The reason is because minday_std had been standardized, and this means it will have a mean of 0 and a standard deviation of 1, this is what standardization do.

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

```
par(mfrow=c(1,2))
plot(density(minday), main="minday", col="blue", lwd=2)
plot(density(minday_std), main="minday_std", col="red", lwd=2)
```

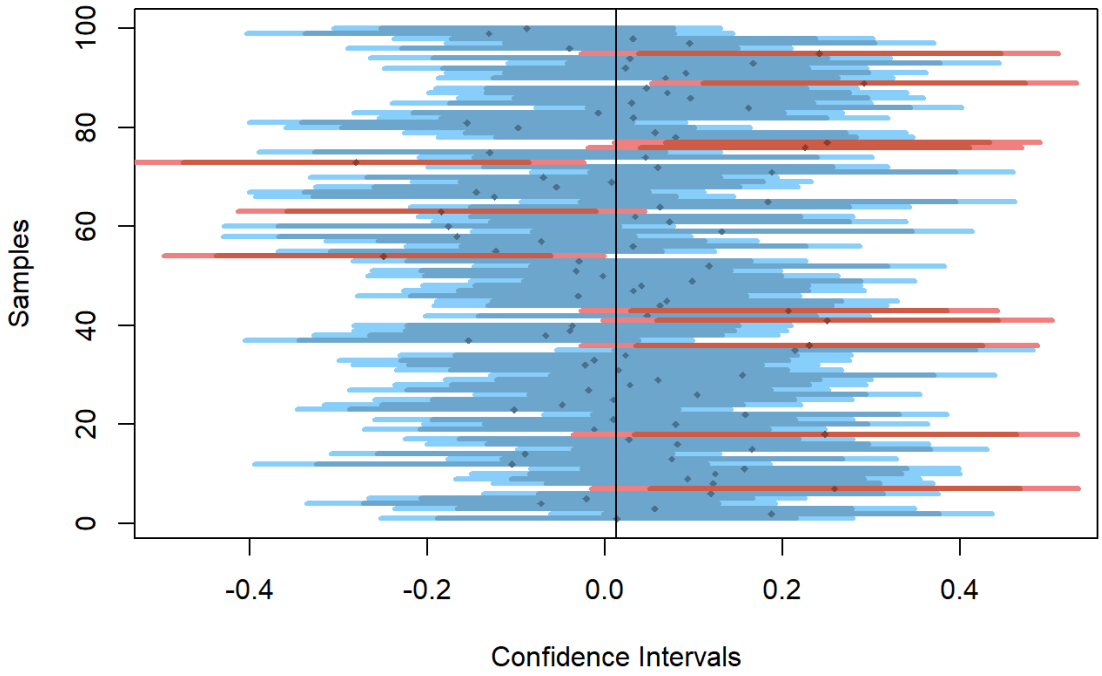



```
par(mfrow=c(1,1))
```

The two distribution looks the same, with different scale. This is because doing standardization will not change the shape of distribution, but it will rescale the distribution to mean = 0, sd = 1.

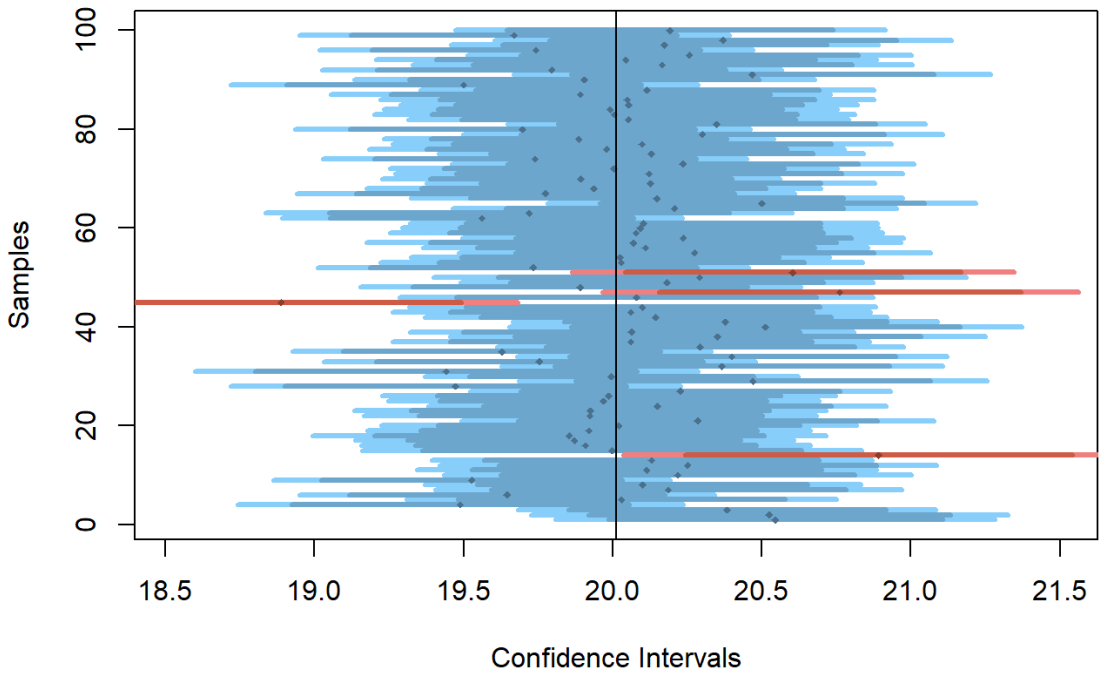
Question 2) Install the compstatslib package from Github (see class notes) and run the plot_sample_ci() function that simulates samples drawn randomly from a population. Each sample is a horizontal line with a dark band for its 95% CI, and a lighter band for its 99% CI, and a dot for its mean. The population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

```
library(compstatslib)
plot_sample_ci()
```



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?

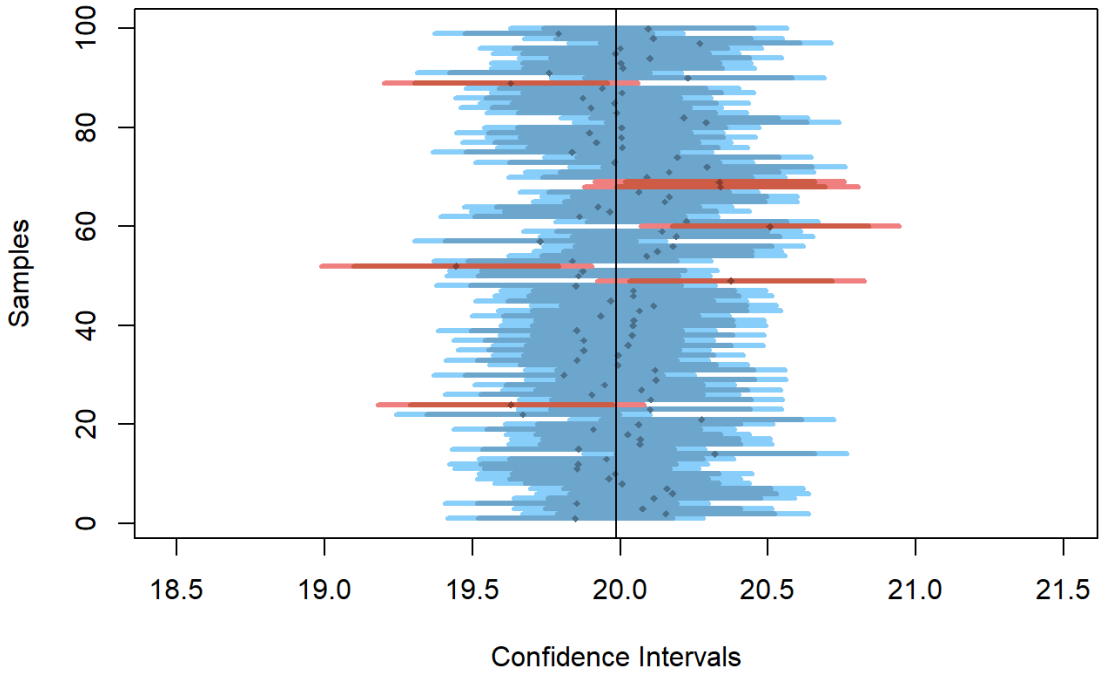
A: 5 samples, the meaning of 95% CI is, for example take 100 different samples and compute a 95% confidence interval for each sample, then approximately 95 of the 100 confidence intervals will contain the true mean.

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

A: 1 sample, same reason with (a).

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

```
plot_sample_ci(num_samples = 100, sample_size = 300, pop_size=10000, 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?

The 95% and 99% CI to become narrower than before, this is because Confidence Interval of the mean:

$$\bar{x} \pm Z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$$

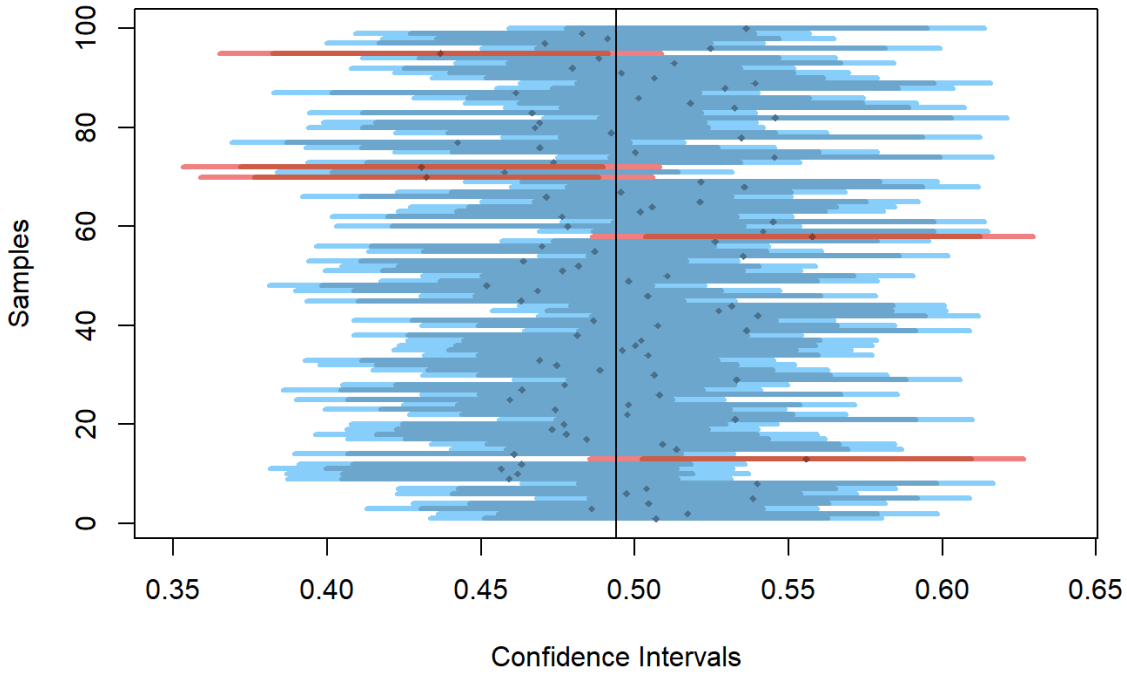
Where $\bar{x}$ is sample mean, s is sample standard deviation α is the confidence level, and n is the sample size. Since size of each sample has increased, the $\sqrt{n}$ increased, and because it's on denominator the value of $Z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$ decrease, the result of confidence interval become 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?

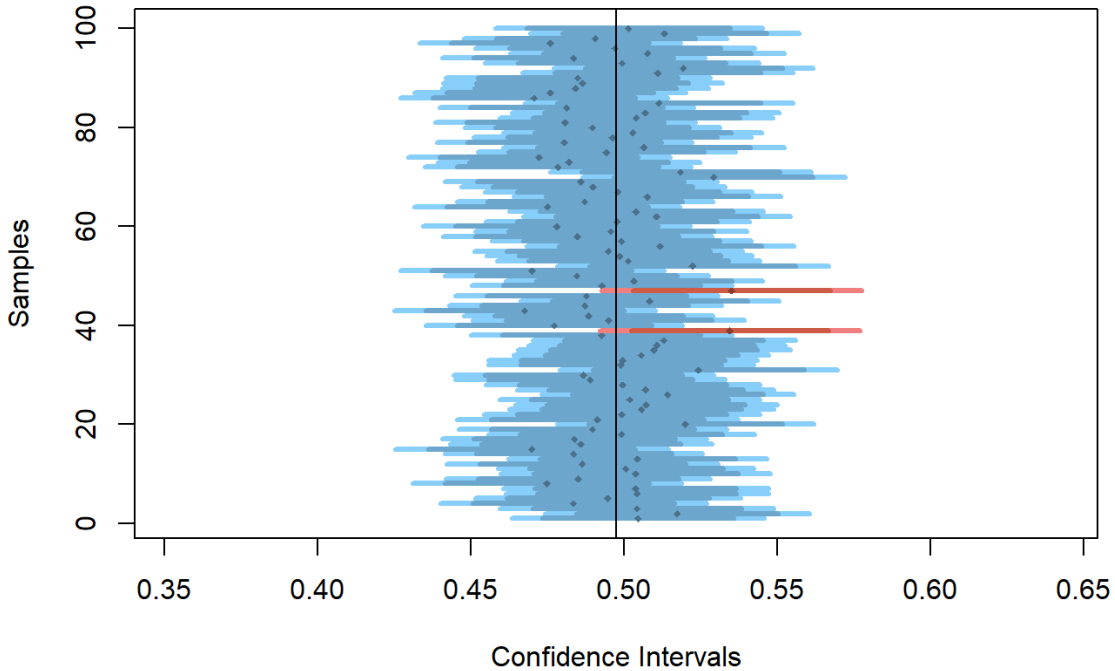
A: 5 samples

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 = 100, pop_size=10000, distr_func=runif)
```




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



The answer for (a) and (b) will not change. Still expect 5 samples not in 95% CI and 1 sample not in 99% CI, with the same explanation above. Also by looking the formula of Confidence Interval of mean $\bar{x} \pm Z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$, the result of confidence interval become narrower than before with the same reason in (b).

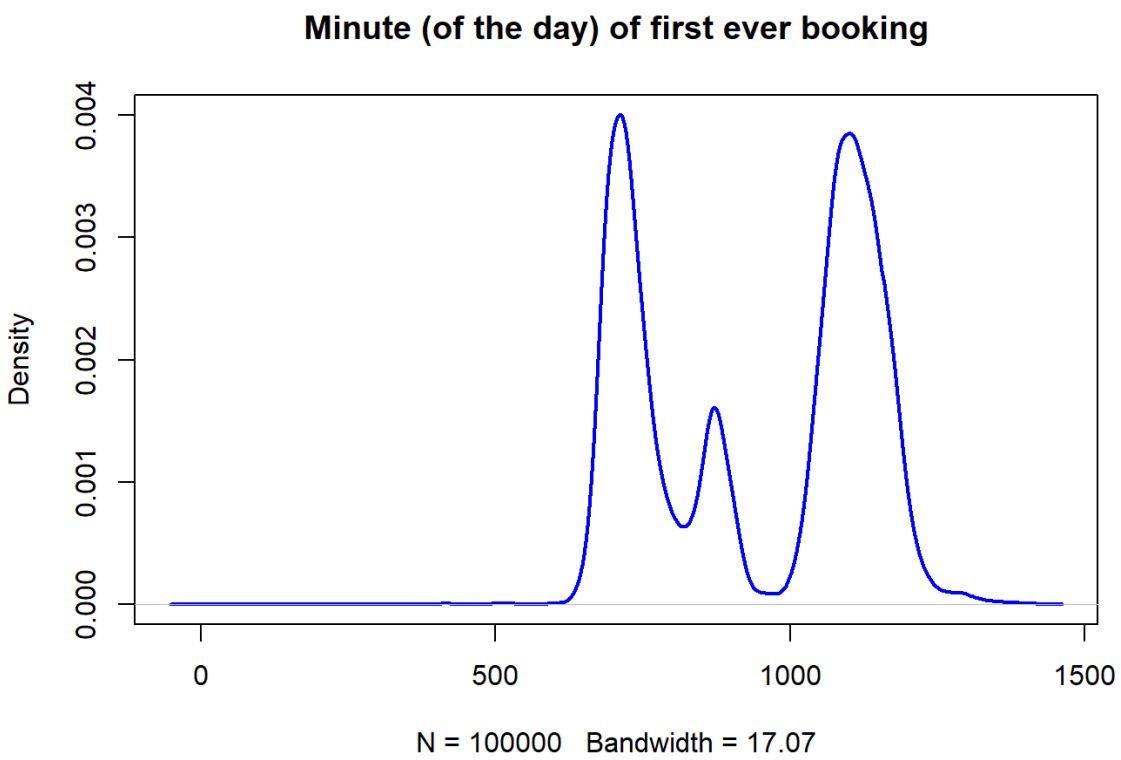
Question 3) The company EZTABLE has an online restaurant reservation platform that is accessible by mobile and web. Imagine that EZTABLE would like to start a promotion for new members to make their bookings earlier in the day.

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:

```
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
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

```
mean(minday)
```

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

```
sd(minday)
```

```
## [1] 189.6631
```

```
c(mean(minday)-1.96*sd(minday)/sqrt(length(minday)),mean(minday)+1.96*sd(minday)/sqrt(length(minday)))
```

```
## [1] 941.3208 943.6719
```

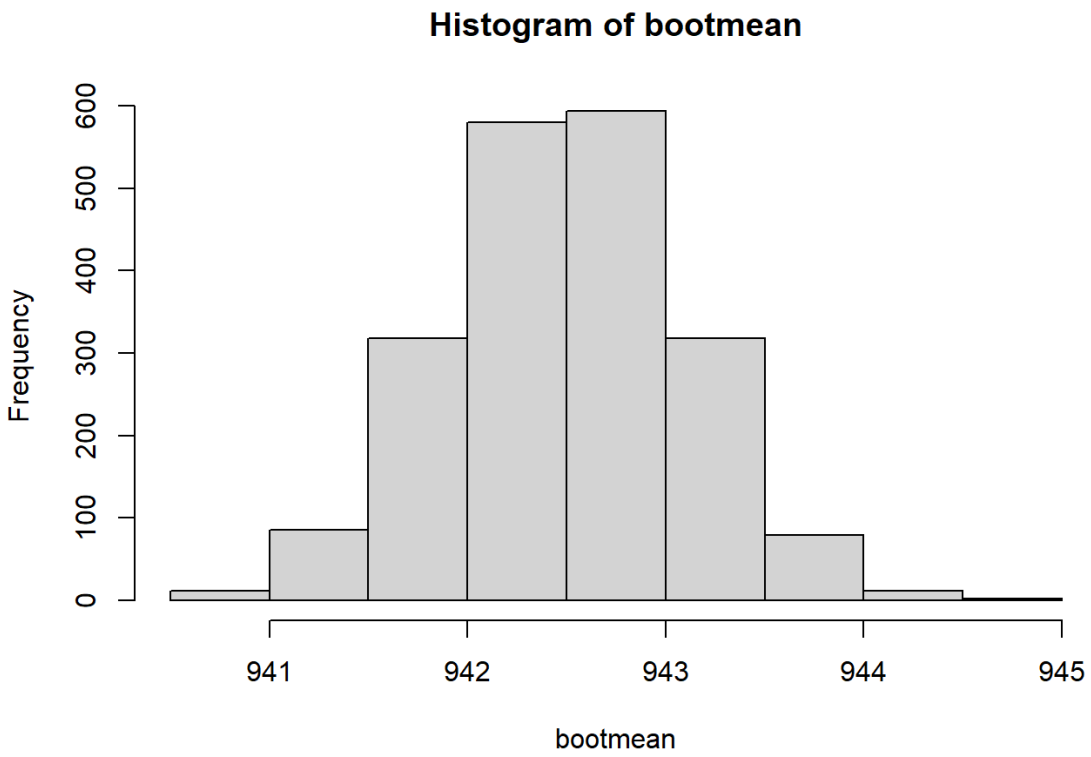
The sampling means is 942.4964, sample standard error is 189.6631, and the 95% confidence interval of the sampling means is (941.3208, 943.6719)

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)
}
bootmean <- replicate(2000,compute_sample_mean(minday))
```

iii) Visualize the means of the 2000 bootstrapped samples

```
hist(bootmean)
```



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

```
mean(bootmean)
```

```
## [1] 942.4986
```

```
quantile(bootmean, probs=c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 941.3309 943.7182
```

The bootstrapped means and the 95% confidence interval of the bootstrapped means is the above values.

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

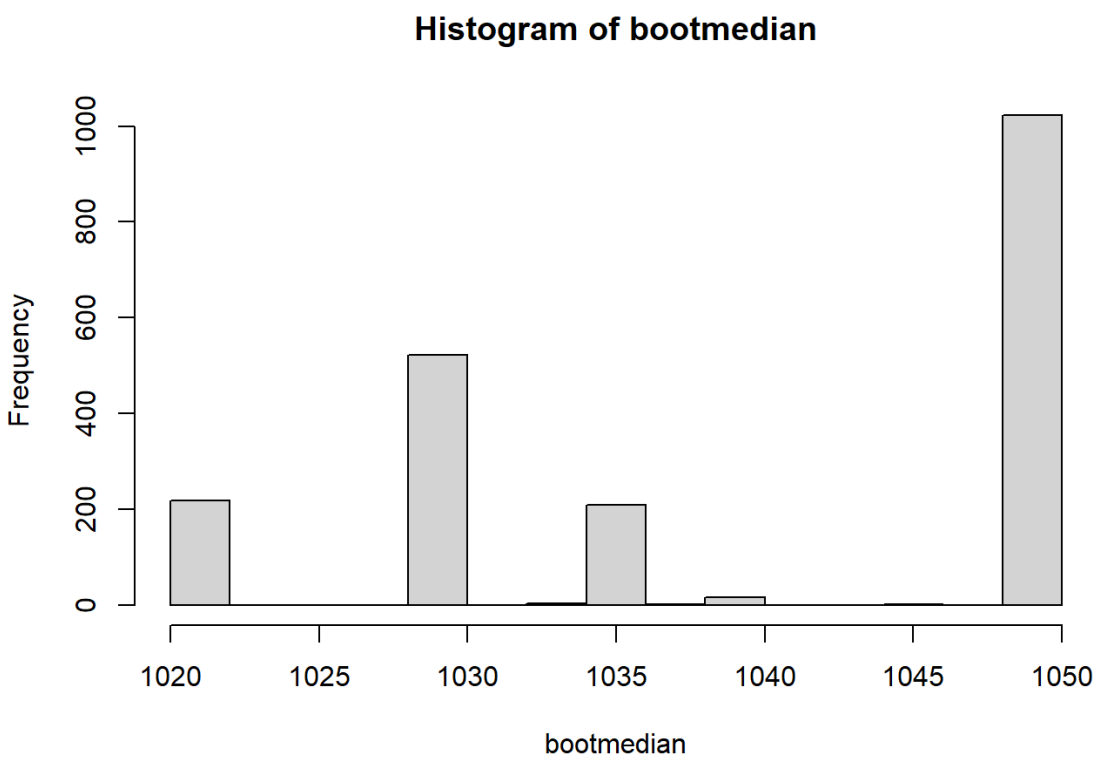
```
median(minday)
```

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

ii) Visualize the medians of the 2000 bootstrapped samples

```
compute_sample_median <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  median(resample)
}
bootmedian <- replicate(2000,compute_sample_median(minday))

hist(bootmedian)
```



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

```
quantile(bootmedian, probs=c(0.025, 0.975))
```

```
##  2.5% 97.5%
##  1020  1050
```

95% CI of the bootstrapped medians is (1020, 1050).

The answer is that when 1050, the store have half the new members of the day already arrived at their restaurant.