HW4

111078513

2023-03-11

Question 1()

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

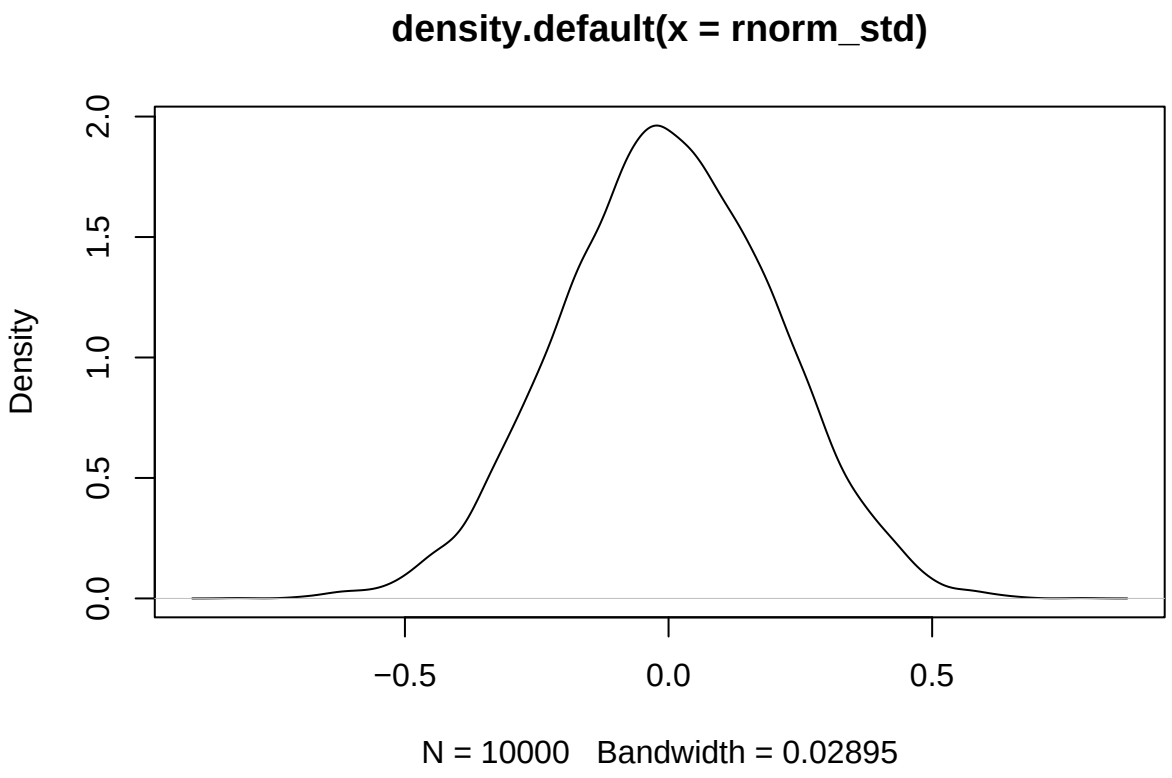
(i) What should we expect the mean and standard deviation of rnorm_std to be, and why?

```
data <- rnorm(10000, 940, 190)
data_mean <- mean(data)
data_std <- mean(data)
rnorm_std <- (data-data_mean)/data_std
```

We expected that the mean and std from rnorm_std would be 940 and 190 seperately because they were samples from a normal distribution with mean 940 and std 190.

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

```
plot(density(rnorm_std))
```



It just looks like a bell shape and I think that's because it came from another bell-shape distribution.

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

Z-distributon(or Standard Normal Distribution).

(b)

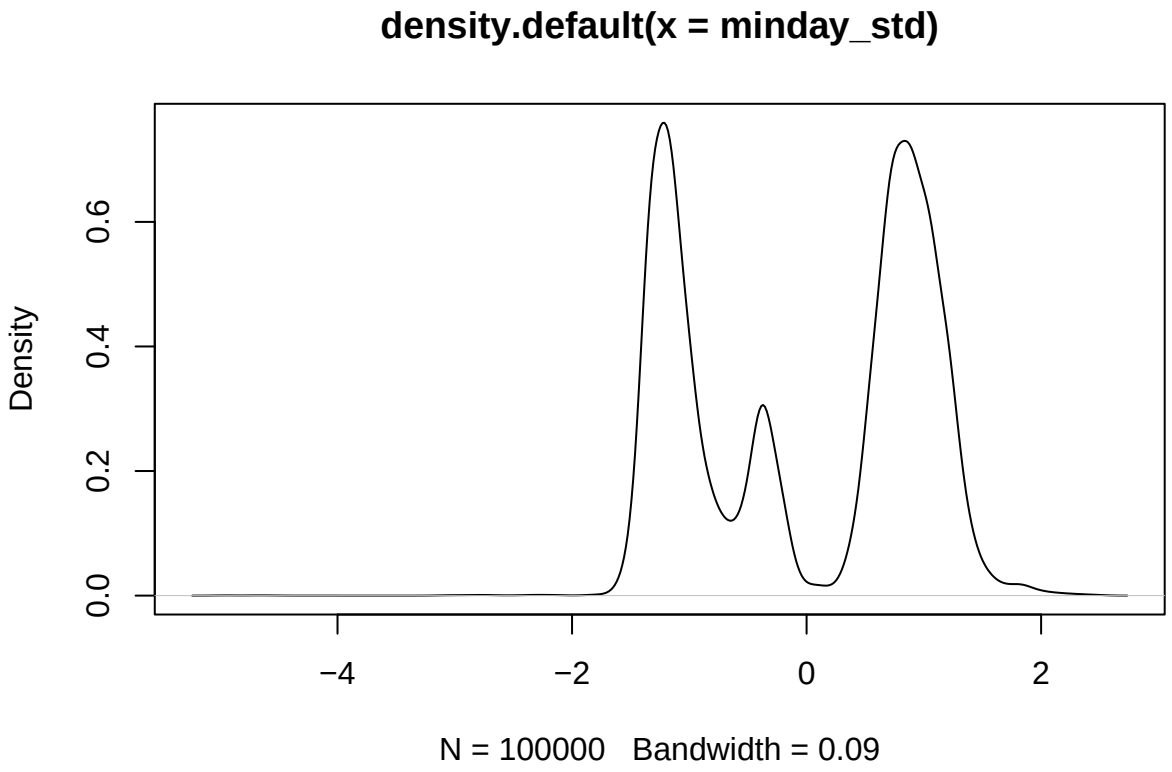
```
bookings <- read.table("first_bookings_datetime_sample.txt", header=TRUE)
hours  <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$hour
mins   <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$min
minday <- hours*60 + mins
minday_mean <- mean(minday)
minday_sd <- sd(minday)
minday_std <- (minday - minday_mean)/minday_sd
```

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

Based on the plot we drew, we can expect the mean of minday in May to fall between 700 and 1200. Furthermore, we anticipate that the standard deviation of midday in May will be less than 200, given that the majority of the data falls between 700 and 1200.
By setting an interval of three times the standard deviation above and below the mean, we can capture nearly the entire data set.

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

```
plot(density(minday_std))
```

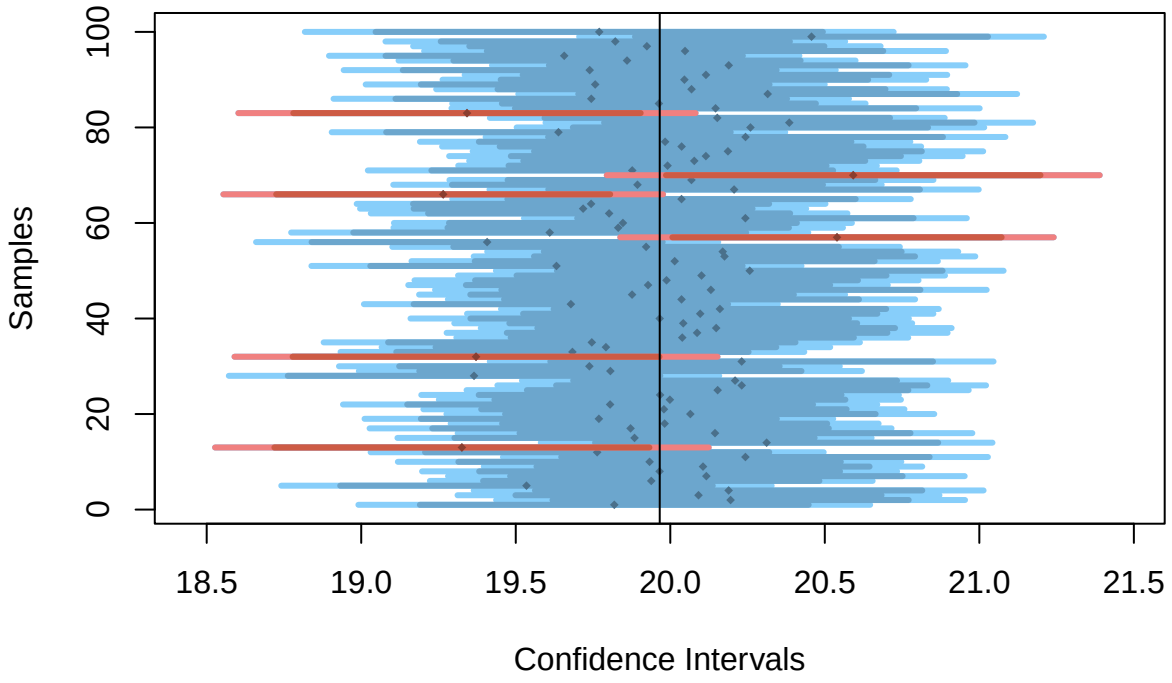


The two plots looked identical because each data point underwent the same translation and scaling.

Question 2)

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

```
library(compstatslib)
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?

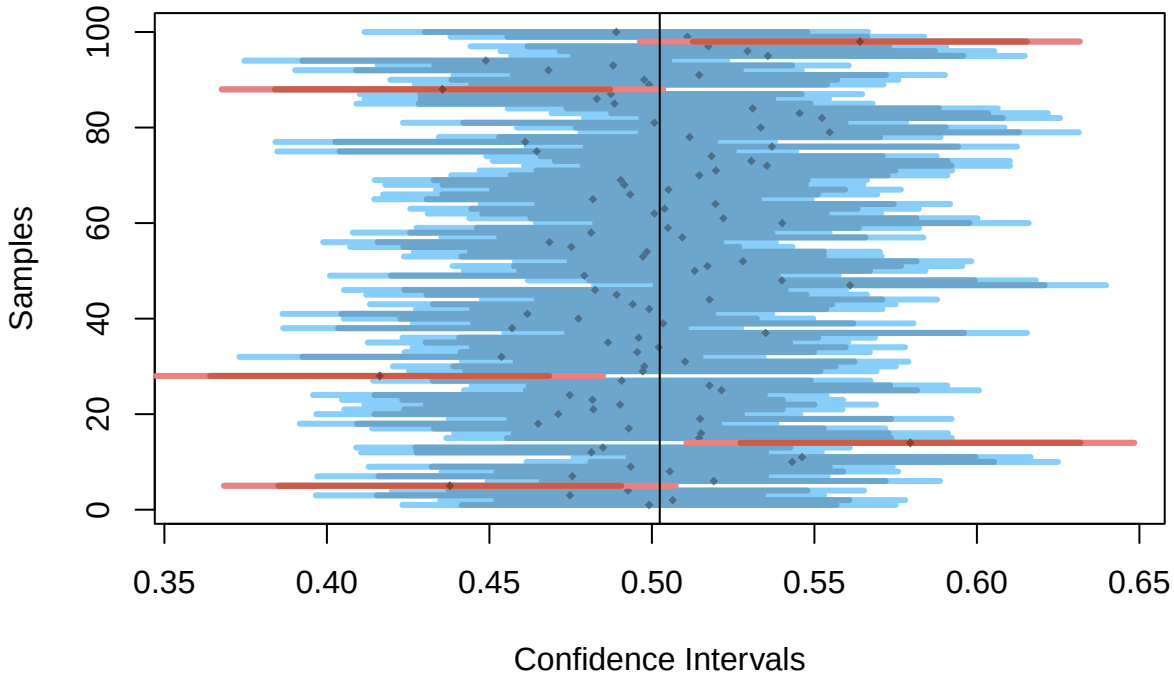
We expect that there are nearly 5 samples not to include the population mean in its 95% CI.

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

We expect that there are nearly 1 samples not to include the population mean in its 95% CI.

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



(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?

We expect their 95% and 99% CI to 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?

We still expect there are nearly 5 samples not to include the population mean in its 95% CI.

(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?

The answer we get would not have any change.
Although population follows uniform distribution, according to the central limit theorem, when the sample size is sufficiently large, the distribution of the sample mean approaches the normal distribution.
As mention above, we expect the result would not have any change.

Question 3)

(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.

```
minday_mean <- mean(minday)
minday_mean
```

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

```r
minday_std <- sd(minday)
minday_std
```

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

```r
minday_95CI <- c(minday_mean - 2*minday_std, minday_mean + 2*minday_std)
minday_95CI
```

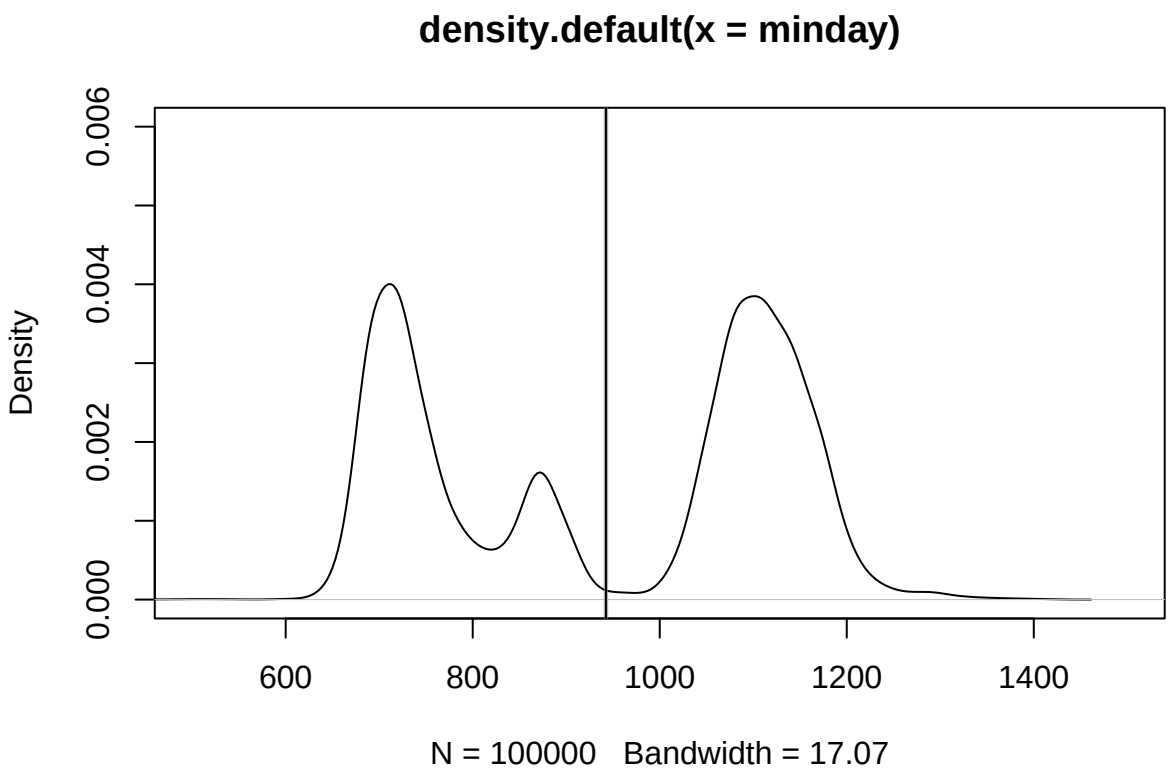
```
## [1]  563.1702 1321.8225
```

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

```r
compute_sample_mean <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  mean(resample)
}
resamples_mean <- replicate(2000,compute_sample_mean(minday))
```

(iii) Visualize the means of the 2000 bootstrapped samples.

```r
plot(density(minday), lwd=1, xlim = c(500, 1500), ylim=c(0, 0.006))
abline(v = resamples_mean, col=rgb(0.7, 0.7, 0.7, 0.01))
abline(v = mean(resamples_mean))
```



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

```r
quantile(resamples_mean, probs=c(0.025, 0.975))
```

```
##     2.5%    97.5%
```

```
## 941.2651 943.6773
```

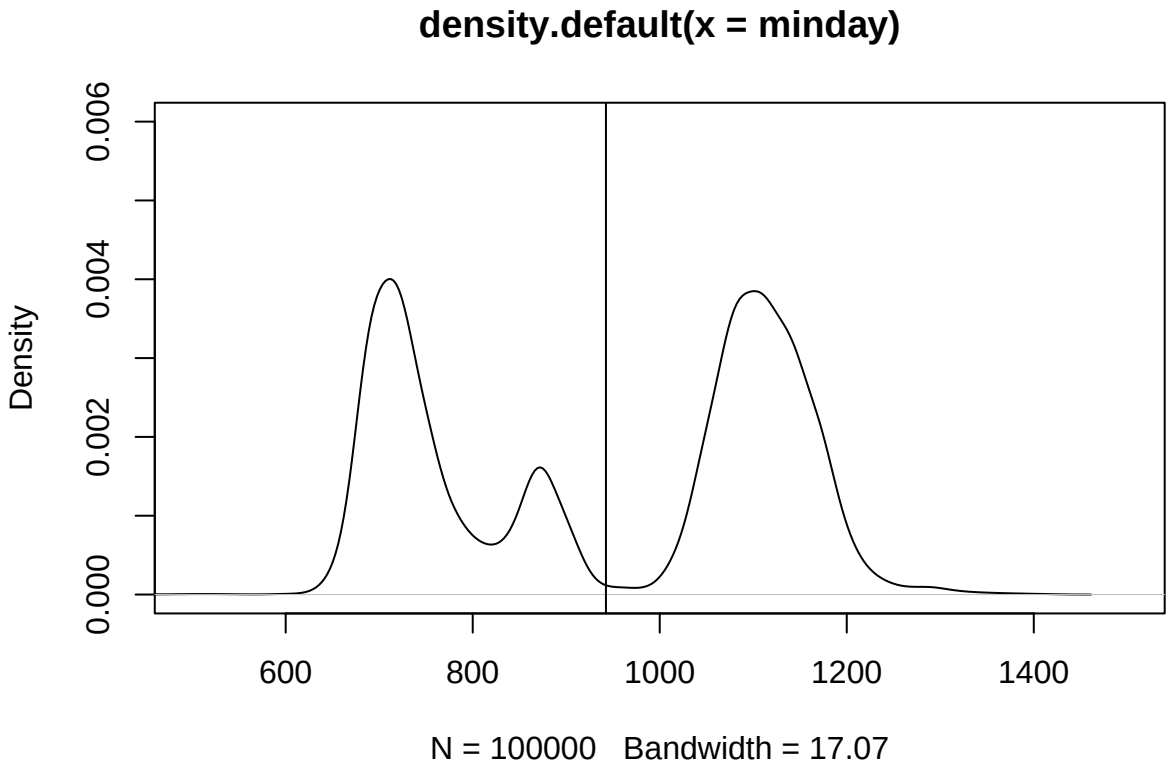
(b)

(i) Estimate the median of minday

```
resamples_mean_median <- quantile(resamples_mean, probs=c(0.5))
```

(ii) Visualize the medians of the 2000 bootstrapped samples.

```
plot(density(minday), lwd=1, xlim = c(500, 1500), ylim=c(0, 0.006))
abline(v = resamples_mean_median)
```



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

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

```
##     2.5%    97.5%
## 941.2651 943.6773
```