BACS HW4 - 109006234

Credit: 109006278

March 12th 2023

Supporting Functions

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

plotFunct <- function(data, title){
    hist(data, prob=TRUE, main = title)
    lines(density(data), lwd = 2, col = "brown", main = title)
    abline(v=median(data), col="orange")
    abline(v=mean(data), col="green")
}
```

Problem 1.

(a) Create a normal distribution (mean=940, sd=190) and standardize it

```
dataset1 <- rnorm(1000, mean=940, sd=190)
rnorm_std <- standardize(dataset1)
```

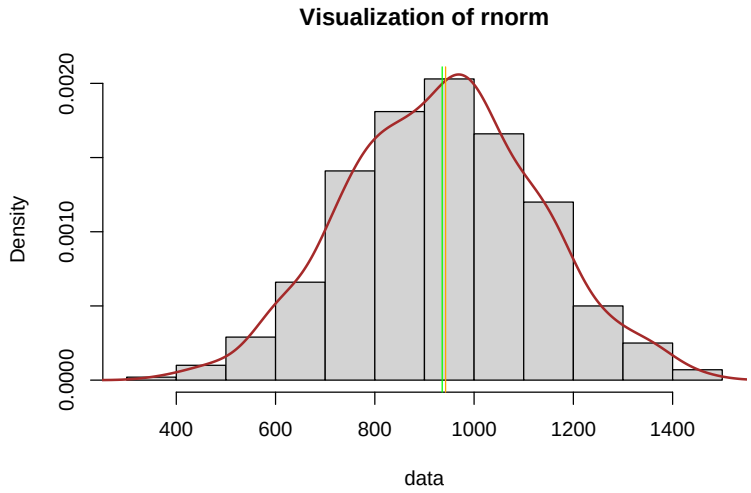
(i) What should we expect the mean and standard deviation of rnorm_std to be, and why?
As *rnorm_std* is a standardized normal distribution, the mean and standard deviation are expected to be 0 and 1 respectively. Based on the code below, we get what is expected, in this case, the computed mean is pretty much the same as 0. This can happen because the distribution is symmetric around 0.

```
## [1] "Mean of the rnorm_std:  2.2785408542586e-16"
```

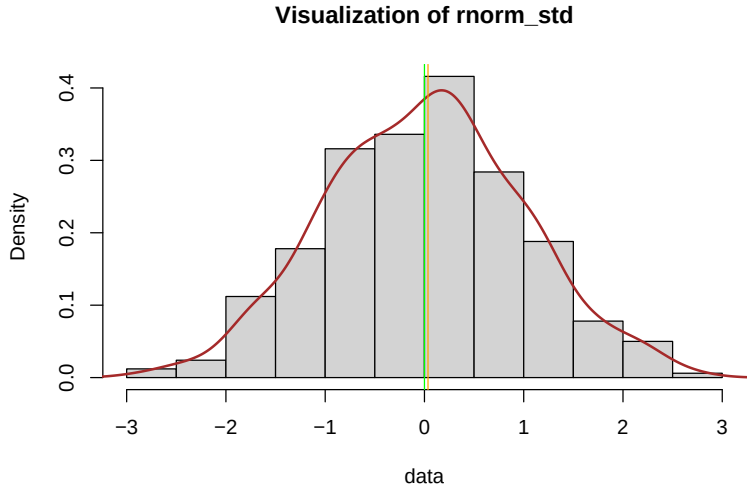
```
## [1] "Standard deviation of the rnorm_std:  1"
```

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

```
plotFunct(dataset1, "Visualization of rnorm")
```




```
plotFunct(rnorm_std, "Visualization of rnorm_std")
```



From both visualization above, it is evident that both visualization don't differ so much. We can observe that the shape of the distribution is similar to a bell-shape distribution.

(iii) What do we generally call distributions that are normal and standardized?
Based from both visualization of *dataset1* and *rnorm_std* above, we can conclude that it is a standard normal distribution or a z-distribution.

(b) Create a standardized version of minday discussed in question 3

```
bookings <- read.table("G:/My Drive/111_2_BACS/HW4/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
```

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

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

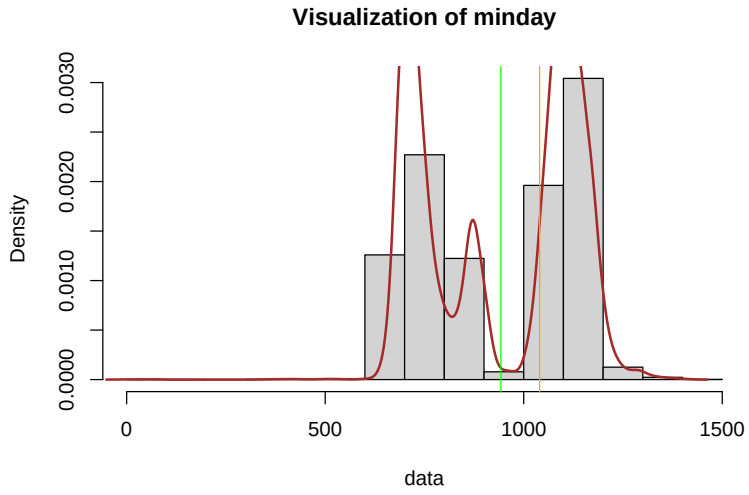
Similar to Q1a (i), the dataset *minday* has been standardized, meaning that both the mean and the standard deviation are going to be 0 and 1 respectively. From the code below, the result shows what is expected. In this case, the computed mean is pretty much the same as 0.

```
## [1] "Mean of the minday_std:  -4.25589034500073e-17"
```

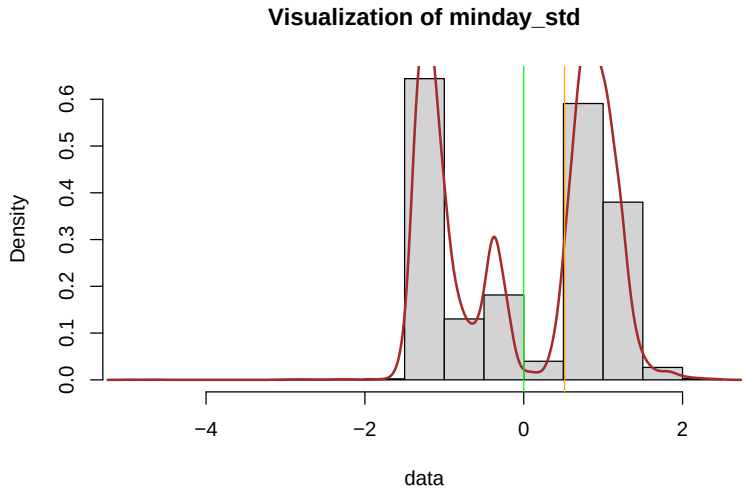
```
## [1] "Standard deviation of the minday_std:  1"
```

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

```
plotFunct(minday, "Visualization of minday")
```




```
plotFunct(minday_std, "Visualization of minday_std")
```



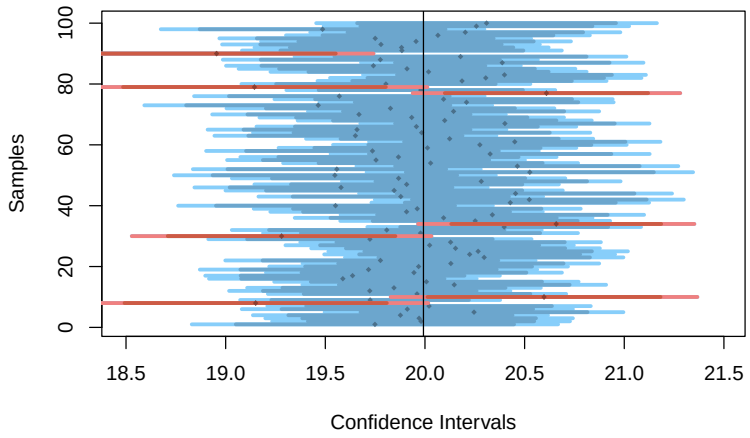
Similar to Q1a(ii), we can observe that the distribution is a non-normal distribution. Hence, standardization of a non-normal distribution results in a distribution with a mean of 0 and a standard deviation of 1. We can observe that the x-axis represents the possible values of the standardized variable, while the y-axis represents the probability density function of that variable.

Problem 2.

```
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)
```

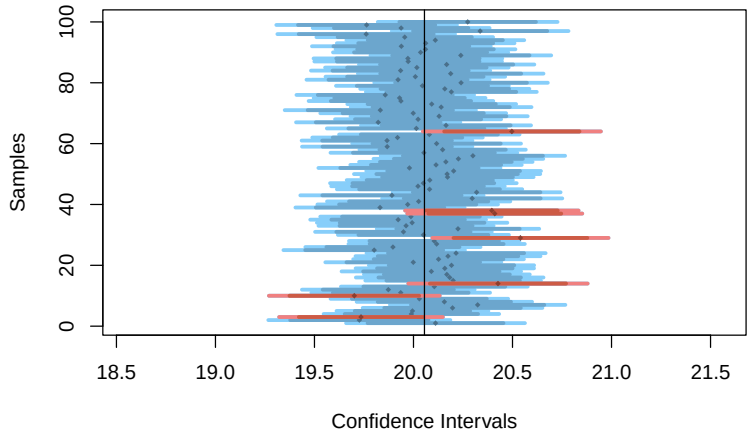


(i) How many samples do we expect to NOT include the population mean in its 95% CI
To determine the number of samples that we expect to not include the population mean in their 95% confidence intervals, we can use the binomial distribution with parameters n = 100 (the number of trials) and p = 0.05 (the probability of success in each trial). It can be calculated as: E(X) = n * p = 100 * 0.05 = 5. Therefore, we would expect approximately 5 out of the 100 samples to not include the population mean in their 95% confidence interval.

(ii) How many samples do we expect to NOT include the population mean in their 99% CI?
Using the formula above, E(X) = n * p = 100 * 0.01 = 1. Therefore, we would expect approximately 1 out of the 100 samples to not include the population mean in their 99% confidence interval.

(b) Rerun the previous simulation with the 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?

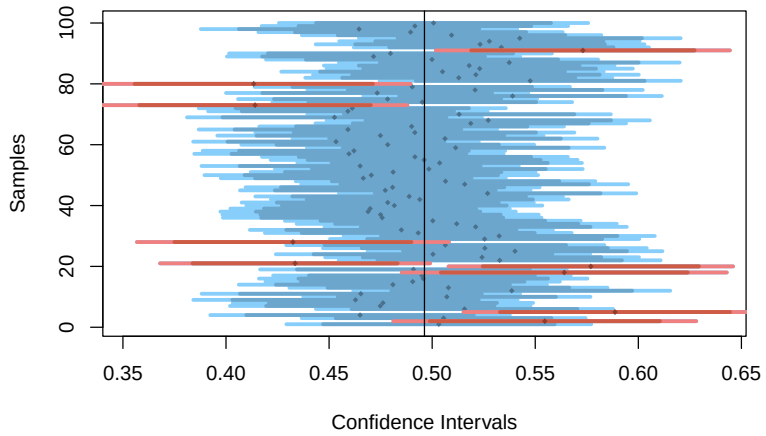
Based on the visualization in Q2(a) and Q2(b), we can observe that that this visualization with the bigger sample is much narrower that the one with a smaller sample. With the increase of sample size, the distribution becomes more centered in by the mean of the distribution.

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

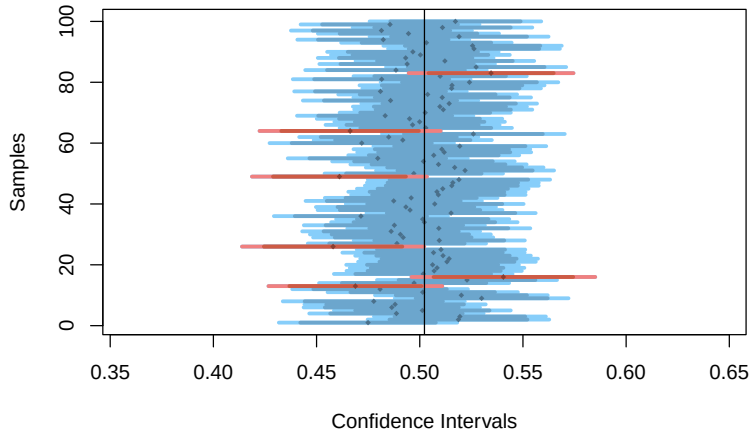
Same as the question above, it can be calculated as: E(X) = n * p = 100 * 0.05 = 5. Therefore, we would expect approximately 5 out of the 100 samples to not include the population mean in their 95% confidence interval.

(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)
```



In a uniform distribution population, we expect the width of the confidence interval to decrease as the sample size increases. This is because larger sample sizes provide more precise estimates of the population parameter, resulting in narrower confidence interval. In general, with a larger sample sizes and less variability, the number of sample size not included will be smaller. On the other hand, with smaller sample size and more variability, the number of sample size not included will be larger.

Problem 3.

(a) What is the "average" booking time for new members making their first restaurant booking?

```
#Supporting Functions
plot_sample <- function(sample) {
  lines(density(sample), col="blue")
  return(mean(sample))
}

plot_sample_median <- function(sample) {
  abline(v=median(sample), col="#6cec5e")
  return(median(sample))
}
```

(i) Estimate the population mean of minday, its standard error, and the 95% confidence interval (CI) of the sampling means

```
## [1] "Mean of the minday:  942.49635"
```

```
## [1] "Standard deviation error of the minday:  0.599767314943967"
```

```
## [1] "The 95% Confindence Interval is between 941.32080606271 and 943.67189393729"
```

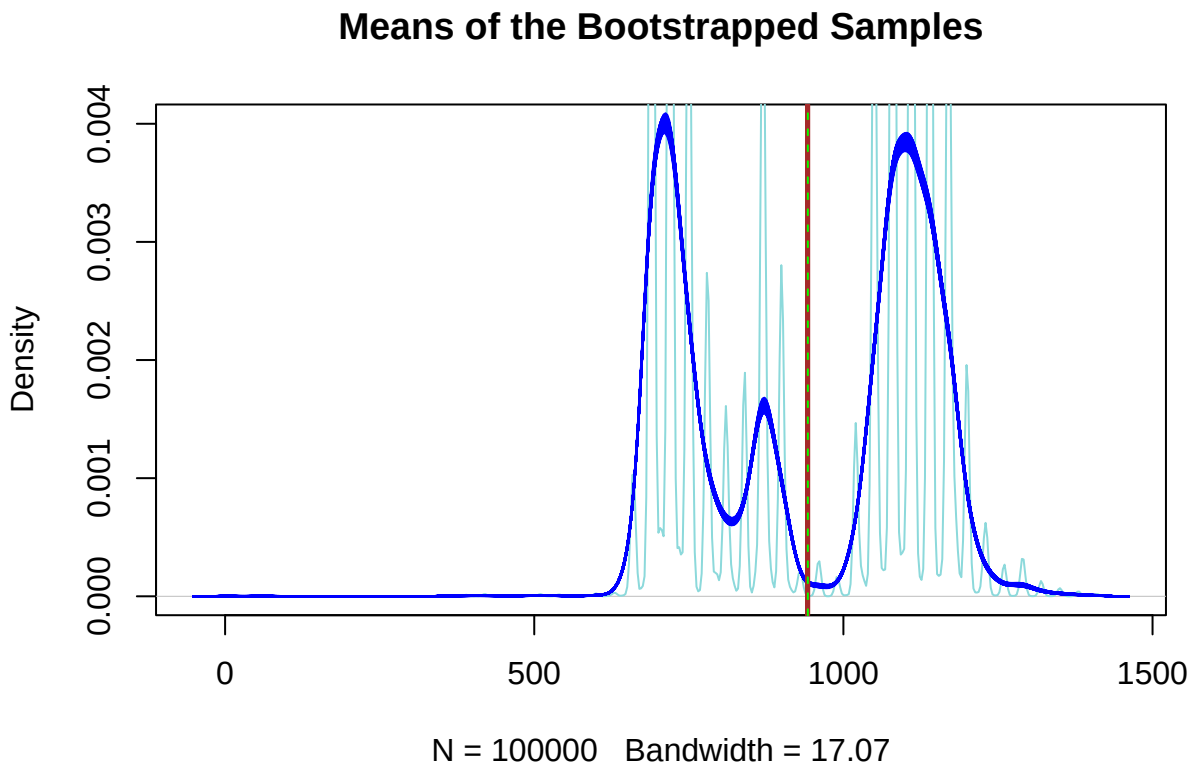
(ii) Bootstrap to produce 2000 new samples from the original sample

```
resamples <- replicate(3000, sample(minday, length(minday), replace=TRUE))
```

(iii) Visualize the means of the 2000 bootstrapped samples

```
plot(density(minday), lty="dashed", main="Means of the Bootstrapped Samples")
lines(density(resamples), col="#8cd9db", lwd=1)
sample_means <- apply(resamples, 2, FUN=plot_sample)

abline(v=sample_means, col="brown", lwd=1)
abline(v=mean(sample_means), col="red", lwd=1, lty="dashed")
abline(v=mean(minday), col="green", lwd=1, lty="dashed")
```



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

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

```
##     2.5%    97.5%
## 941.3406 943.7140
```

(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

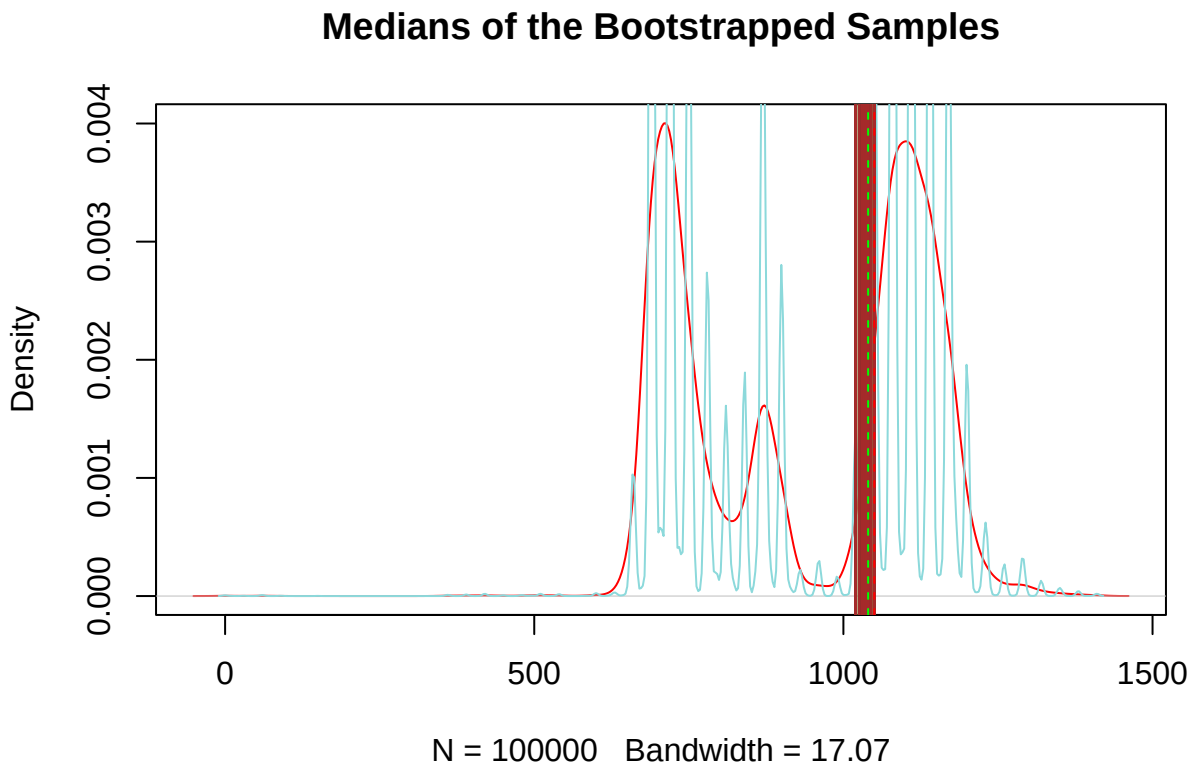
```r
median(minday)
```

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

(ii) Visualize the medians of the 2000 bootstrapped samples

```r
plot(density(minday), col="red", main="Medians of the Bootstrapped Samples")
lines(density(resamples), col="#8cd9db", lwd=1)
sample_medians <- apply(resamples, 2, FUN=plot_sample_median)

abline(v=sample_medians, col="brown", lwd=1)
abline(v=median(sample_medians), col="red", lwd=1, lty="dashed")
abline(v=median(minday), col="green", lwd=1, lty="dashed")
```



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

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

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