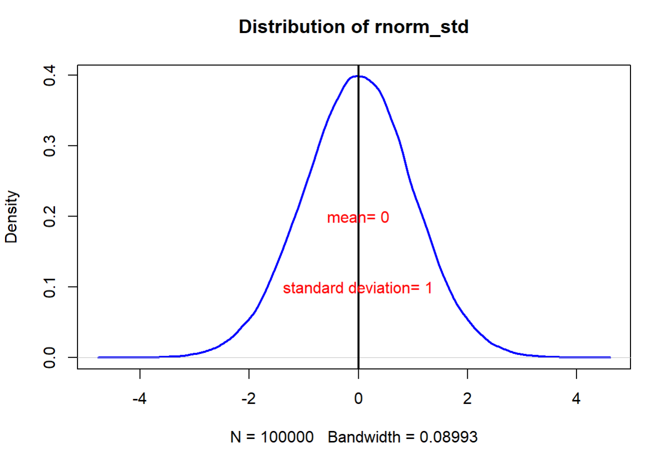
BACS HW4 106070020

The students who help me: 106023021, 106000199

Question 1

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

```r
#(a)
standardize<-function(num){
  num<-(num-mean(num))/sd(num)
  return(num)
}
rnorm_new<-rnorm(n=100000, mean=940, sd=190)
rnorm_std<-standardize(rnorm_new)
a<-paste("mean=",ceiling(mean(rnorm_std)))
b<-paste("standard deviation=",ceiling(sd(rnorm_std)))
{plot(density(rnorm_std), col="blue", lwd=2,
      main = "Distribution of rnorm_std")

  # Add vertical lines showing mean and median
  abline(v=mean(rnorm_std),lwd = 2)
  text(x=0, y=0.2,a, col="red")
  text(x=0, y=0.1,b, col="red")}
```



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

Answer: We should expect the mean be 0 and the standard deviation be 1, since the expected value of the mean and the standard deviation of the rnorm_std is 0 and 1.

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

Answer: rnorm_std should look like a bell shape, which is similar with the normal distribution, and it is because each sample is from normal distibution with the mean=940 and sd=190.

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

Answer: We usually call distributions that are normal and standardized after they are standardized, and it is also known as the Z-distribution. (Each observation in the distribution is minus the mean and then divided by the standard deviation)

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

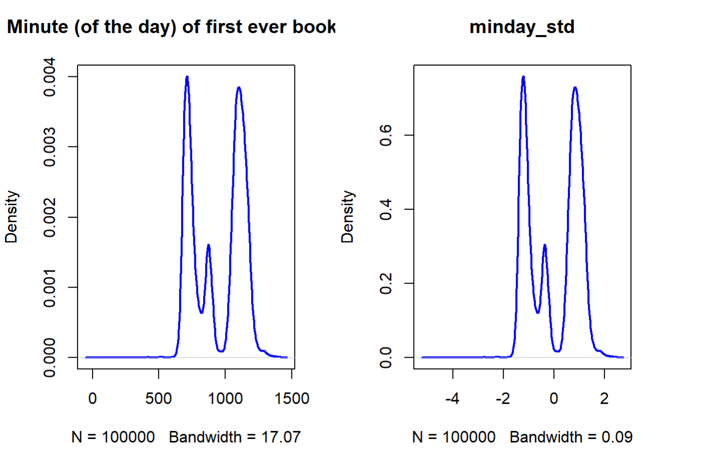
```r
#(b)
op=par(mfrow=c(1,2))
bookings <- read.table("C:/Users/eva/Downloads/first_bookings_datetime_sample.txt", header=TRUE)
#bookings$datetime[1:9]
hours  <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$hour
mins   <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$min
minday <- hours*60 + mins
plot(density(minday), main="Minute (of the day) of first ever booking", col="blue", lwd=2)
mean(minday)
```

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

```r
sd(minday)
```

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

```r
standardize<-function(num){
  num<-(num-mean(num))/sd(num)
  return(num)
}
minday_std<-standardize(minday)
plot(density(minday_std), main="minday_std", col="blue", lwd=2)
```




```r
par(op)
```

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

Answer: we should expect the mean is around 942 and the standard deviation is about 190, as the sample mean and sample standard deviation are 942.4964 and 189.6631. According to the Law of Large Numbers, the larger the sample size, the higher the chance that the arithmetic mean will be close to the expected value. Also, In most cases, the parent standard deviation is estimated by randomly sampling a certain amount of samples and calculating the sample standard deviation. The sample variance is an unbiased estimator of the parent variance.

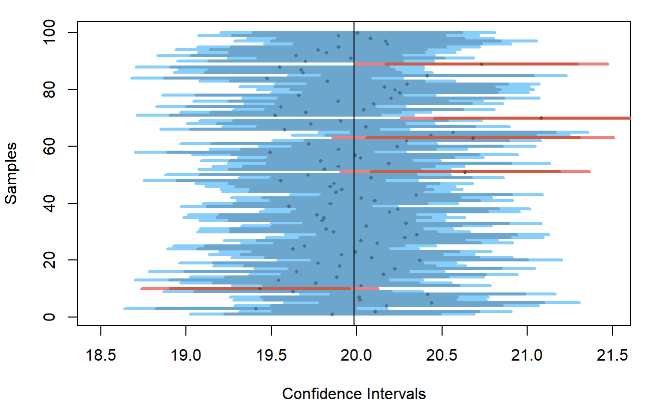
(ii) What should the distribution of minday_std look like compared to minday, and why?

Answer: The plot minday_std should look alike with the distribution minday, because at the beginning, the minday minus its mean, which will not change the shape.Then, the minday is divided by the standard deviation, which just change its scale. Thus, the standardize overall will not change the shape of the minday, so the shape of minday_std is similar with that of minday.

Question 2:

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

```r
# Visualize the confidence intervals of samples drawn from a population
#   e.g.,
#     visualize_sample_ci(sample_size=300, distr_func=rnorm, mean=50, sd=10)
#     visualize_sample_ci(sample_size=300, distr_func=runif, min=17, max=35)
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])
}
visualize_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?

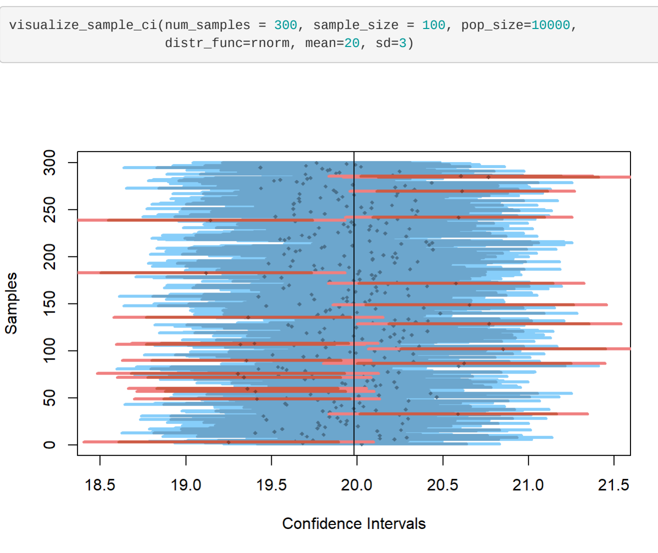
Answer: We expect 100*0.05=5 smaples not included in the population mean in its 95% CI.

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

Answer: We expect 100*0.01=1 smaples not included in the population mean in its 99% CI.

(b) Rerun the previous simulation with larger samples (sample_size=300):

```r
visualize_sample_ci(num_samples = 300, sample_size = 100, 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?

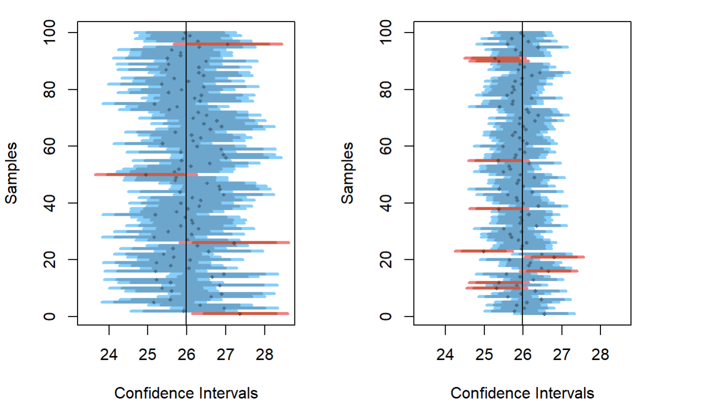
Answer: The 95% C.I = mean ± 1.96 * (σ / √n), and the 99% C.I= mean ± 2.58 * (σ / √n), as the n become greater, the 95% and 99% CI will become narrorer than before.

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

Answer: 5 samples out of 100 (5%) would we expect to NOT 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 distr_func=runif for visualize_sample_ci), how do you expect your answers to (a) and (b) to change, and why?

```r
#(c)
op2=par(mfrow=c(1,2))
visualize_sample_ci(sample_size=100, distr_func=runif, min=17, max=35)
visualize_sample_ci(sample_size=300, distr_func=runif, min=17, max=35)
```




```r
par(op2)
```

Answer: As the size of the sample increase, the 95% and 99% CI will still become narrorer than before in this situation. The mean of the distribution is (17+35)/2= 26, and the standard deviation is (35-12)^2/12=44.08. The calculation of 95% and 99% CI still mean ± 1.96 * (σ / √n) and mean ± 2.58 * (σ / √n). Thus, the answer to (a) and (b) will not change. (eg. 5 samples out of 100 (5%) would we expect to NOT include the population mean in its 95% CI.)

Question 3 :

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.

```r
x<-mean(minday) #sample mean
s<-sd(minday)/sqrt(length(minday)) #standard_error
CI_95<-c(x-1.96*s,x+1.96*s)
x #sample mean
```

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

```r
s #standard_error
```

```
## [1] 0.5997673
```

```r
CI_95 #95% C.I
```

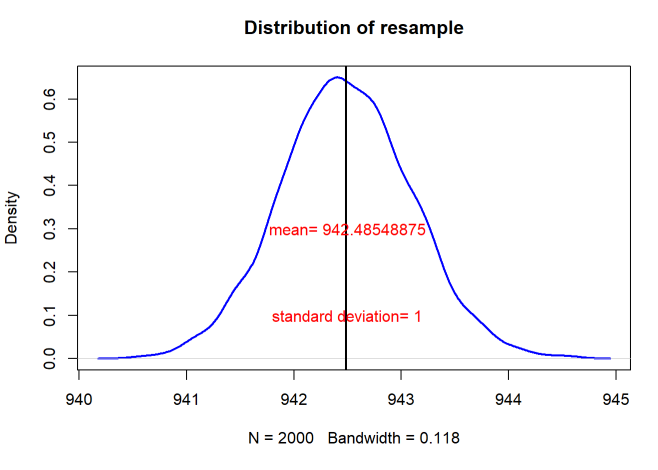
```
## [1] 941.3208 943.6719
```

(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)
}

res<-replicate(2000, compute_sample_mean(minday))
```

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

```r
a<-paste("mean=",mean(res))
b<-paste("standard deviation=",ceiling(sd(res)))
{plot(density(res), col="blue", lwd=2,
      main = "Distribution of resample")

  # Add vertical lines showing mean and median
  abline(v=mean(res),lwd = 2)
  text(x=942.5, y=0.3,a, col="red")
  text(x=942.5, y=0.1,b, col="red")}
```



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

```r
x1<-mean(res) #sample mean
s1<-sd(res)/sqrt(length(res)) #standard_error
CI_951<-c(x-1.96*s,x+1.96*s)
x1 #sample mean
```

```
## [1] 942.4855
```

```r
s1 #standard_error
```

```
## [1] 0.01357811
```

```r
CI_951 #95% C.I
```

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

(b)By what time of day, have half the new members of the day already arrived at their restaurant?

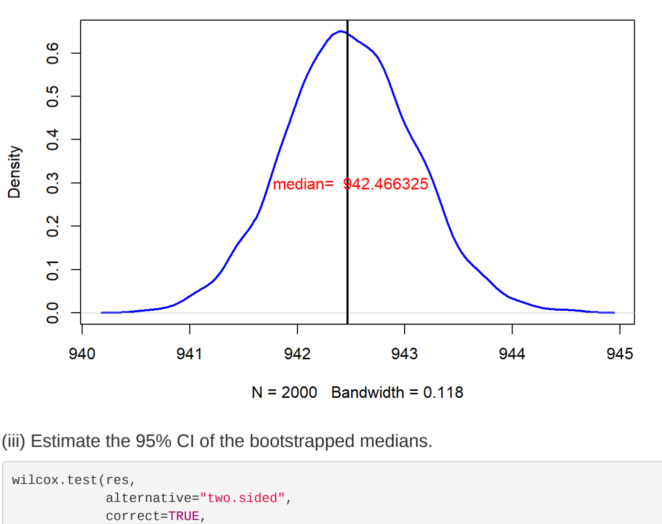
(i) Estimate the median of minday

```r
wilcox.test(minday,
            alternative="two.sided",
            correct=TRUE,
            conf.int=TRUE,
            conf.level=0.95)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  minday
## V = 4999450015, p-value < 2.2e-16
## alternative hypothesis: true location is not equal to 0
## 95 percent confidence interval:
##  930 930
## sample estimates:
## (pseudo)median
##            930
```

(ii) Visualize the medians of the 2000 bootstrapped samples

```r
a<-paste("median= ",median(res))
{plot(density(res), col="blue", lwd=2,
      main = "Distribution of resample")

  # Add vertical lines showing median
  abline(v=median(res),lwd = 2)
  text(x=942.5, y=0.3,a, col="red")
  }
```



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

```r
wilcox.test(res,
            alternative="two.sided",
            correct=TRUE,
            conf.int=TRUE,
            conf.level=0.95)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  res
## V = 2001000, p-value < 2.2e-16
## alternative hypothesis: true location is not equal to 0
## 95 percent confidence interval:
##  942.4548 942.5091
## sample estimates:
## (pseudo)median
##        942.482
```