bacs_hw3

110071010

2024-03-09

110034002 have helped me notice the arguments of quantile(), so eventually I made it to know how to estimate the 95% CI using this function (I thought the function was only a data-description thing at first :(

Question 1

1a

Instruction

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

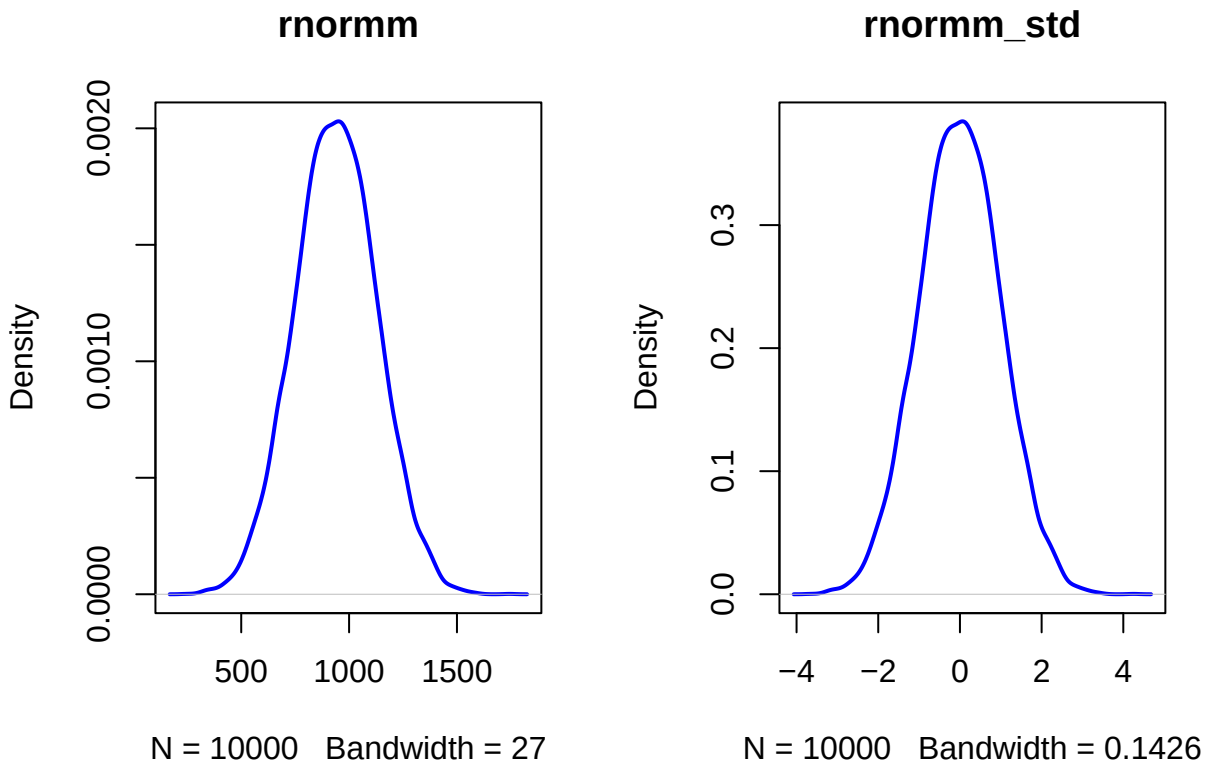
```
# Create a normal dist of size 10000 (called rnormm) using rnorm()
# I call it rnormm rather than rnorm to avoid confusion with later compstatslib function args
rnormm <- rnorm(10000, mean=940, sd=190)

# Standardize nd into rnormm_std
rnormm_std <- (rnormm-mean(rnormm))/sd(rnormm)

# Print mean and sd of rnormm_std
cat("Mean of rnormm_std: ", mean(rnormm_std), "\n")
cat("Sd of rnormm_std: ", sd(rnormm_std), "\n")
```

```
## Mean of rnormm_std:  2.08957e-16
## Sd of rnormm_std:  1
```

```
# Plot rnormm and the standardized rnormm_std
par(mfrow=c(1, 2))
plot(density(rnormm), main = "rnormm", lwd = 2 , col = "blue")
plot(density(rnormm_std), main = "rnormm_std", lwd = 2 , col = "blue")
```



(i) What should we expect the mean and standard deviation of rnormm_std to be, and why?
: The mean and standard deviation of normm_std are expected to be 0 and 1, respectively, as they represent the standardized data drawn from normal distribution.

(ii) What should the distribution (shape) of rnormm_std look like, and why?
: As the original rnormm, rnormm_std must come in a bell-shaped as well, since the standardization of a normally-distributed dataset does not change its shape.

(iii) What do we generally call distributions that are normal and standardized?
: Standard Normal Distribution

1b

Instruction

Create a standardized version of minday discussed in Question 3

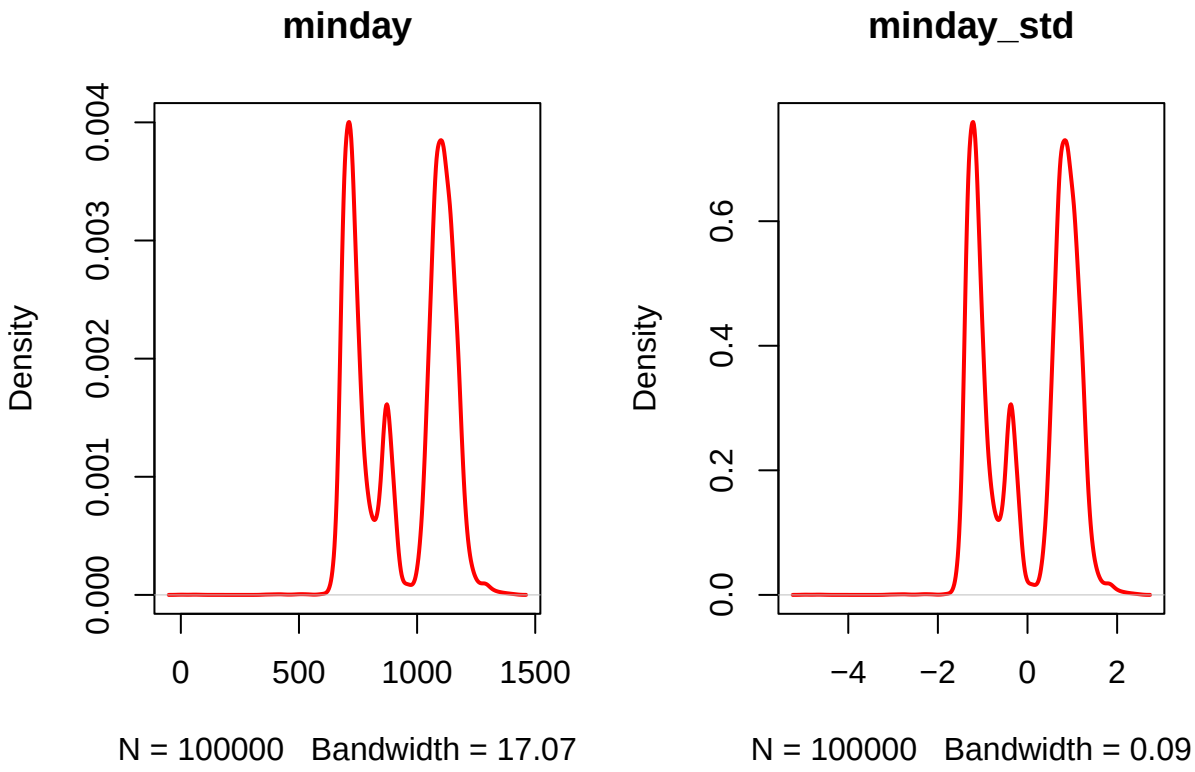
```
# Read Question 3's file and collect data for minday
bookings <- read.table("first_bookings_datetime_sample.txt", header=TRUE)
hours  <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$hour
mins   <- as.POSIXlt(bookings$datetime, format="%m/%d/%Y %H:%M")$min
minday <- hours*60 + mins
```

```
# Standardize minday into minday_std
minday_std <- (minday-mean(minday))/sd(minday)
```

```r
# Print mean and sd of rnorm_std
cat("mean of minday_std: ", mean(minday_std), "\n")
cat("sd of minday_std: ", sd(minday_std), "\n")
```

```
## mean of minday_std:  -4.25589e-17
## sd of minday_std:  1
```

```r
# Plot and compare minday and the standardized minday_std
par(mfrow=c(1, 2))
plot(density(minday), main = "minday", lwd = 2 , col = "red")
plot(density(minday_std), main = "minday_std", lwd = 2 , col = "red")
```



(i) What should we expect the mean and standard deviation of minday_std to be, and why?
: Standardization does not transform the underlying distribution structure of the data. So the mean and the standard deviation are still expected to be 0 and 1, respectively.

(ii) What should the distribution of minday_std look like compared to minday, and why?
: As explained in (i) as well as shown in graph above, the two distributions look identical.

Question 2

Instruction

Install the "compstatslib" package from CRAN 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 compstatslib package from Github

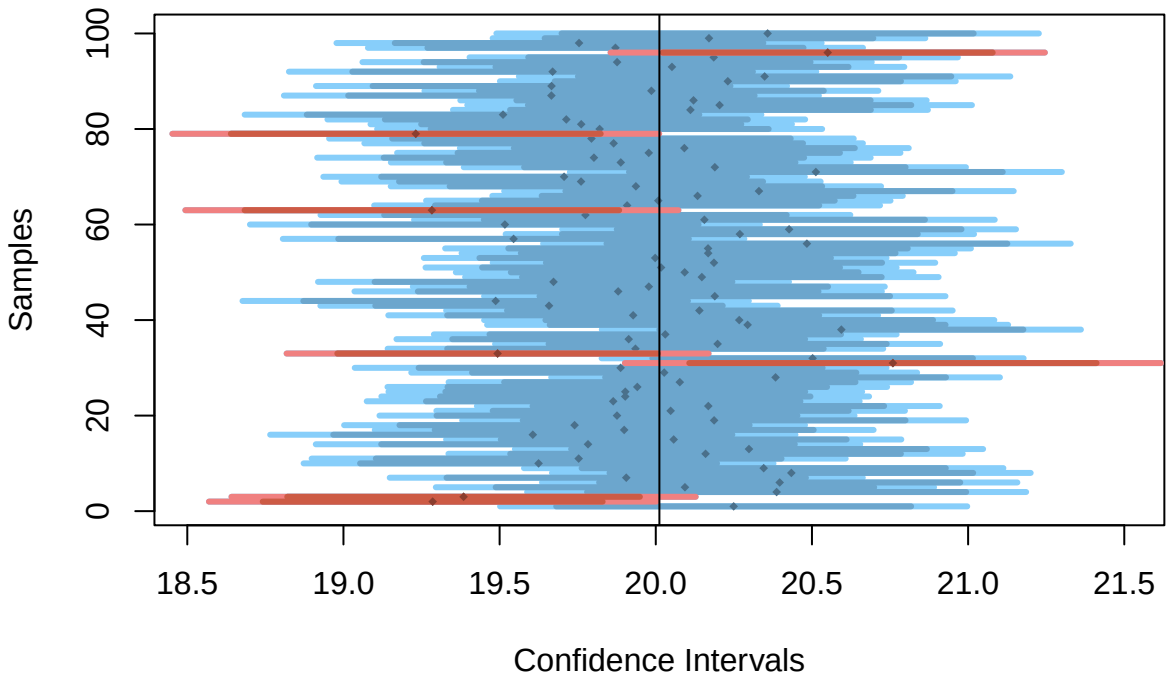
```
# install.packages("remotes")
# remotes::install_github("soumyaray/compstatslib")
library("compstatslib")
```

2a

Instruction

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?
: number of samples 100 * 5 % = 5

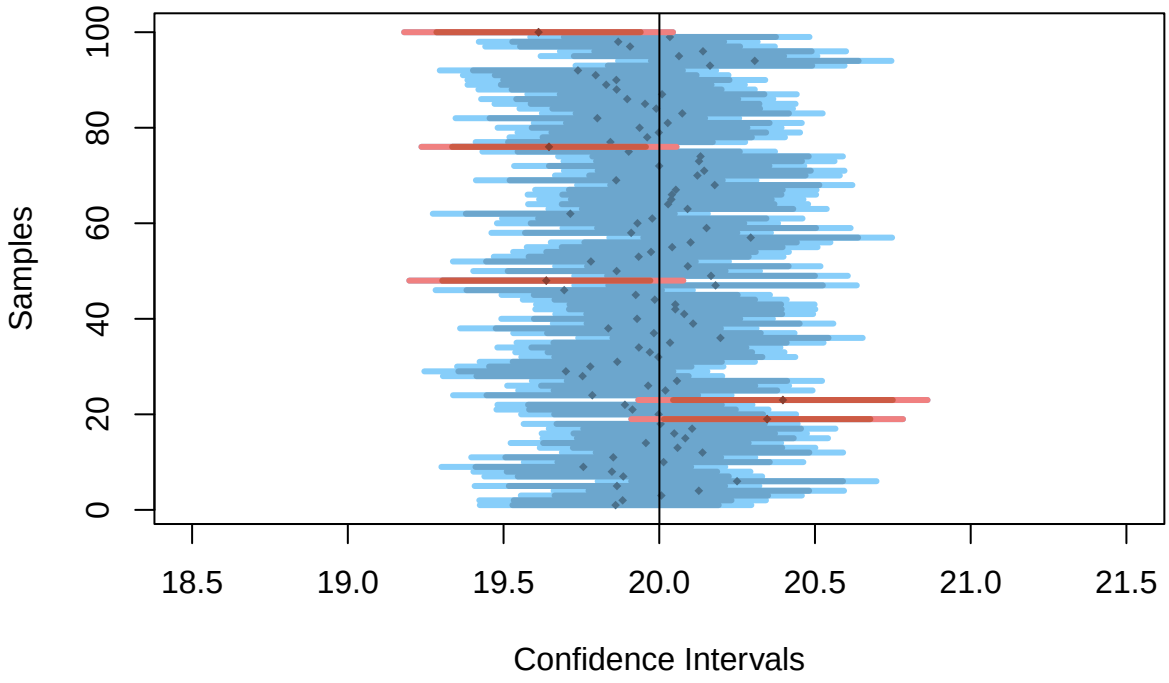
(ii.) How many samples do we expect to NOT include the population mean in their 99% CI?
: number of samples 100 * 1 % = 1

2b

Instruction

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?
: Narrower. Larger sample size lowers the standard error, and lower standard error further narrows the CIs. If comparing two graphs above, one can easily tell the one with larger sample size has visibly narrower colored intervals.

(ii) This time, how many samples (out of the 100) would we expect to NOT include the population mean in its 95% CI?
: number of samples 100 * 5 % = 5

2c

If we ran the above two examples (a and b) using a uniformly distributed population, how do you expect your answers to (a) and (b) to change, and why?
: The answers remain basically the same, as the Central Limit Theorem states that, given a sufficiently large sample size, the distribution of the sample means will approximate a normal distribution, regardless of the shape of the population distribution.

Question 3

Getting prepared

set up functions

```
compute_sample_mean <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  mean(resample)
}
compute_sample_median <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  median(resample)
}
```

read first_bookings_datetime_sample.txt

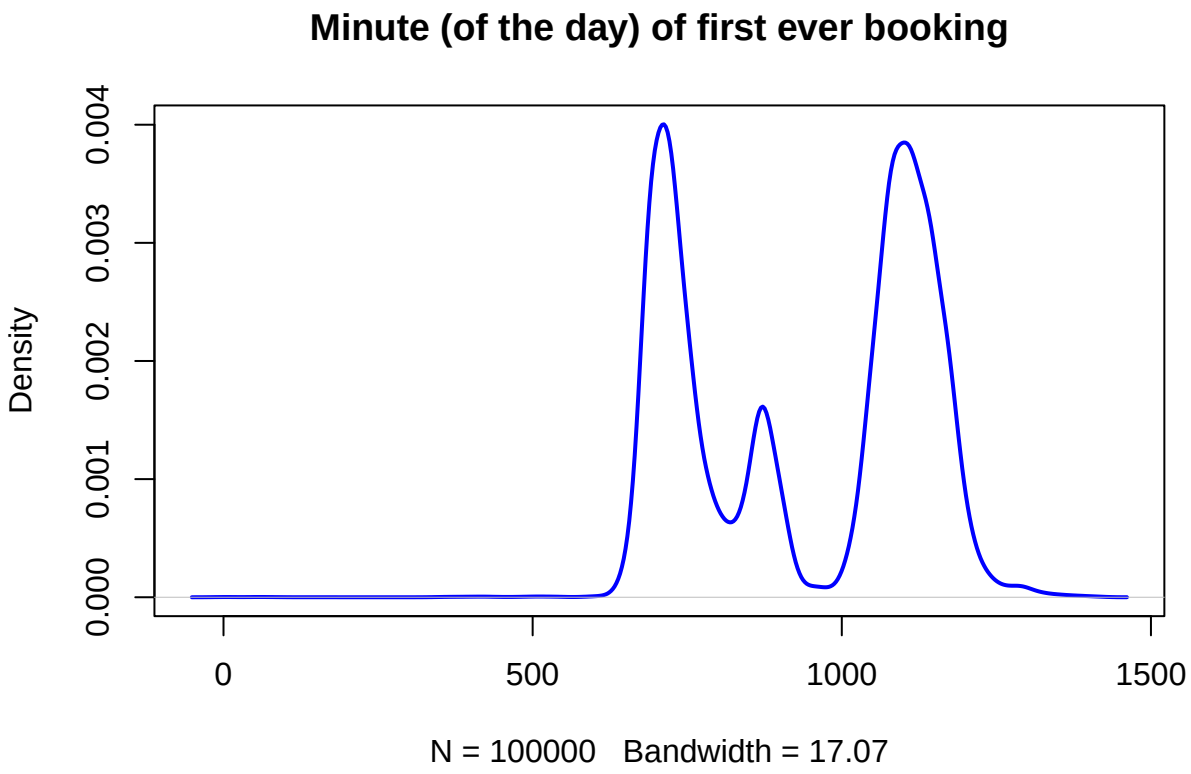
```
bookings <- read.table("first_bookings_datetime_sample.txt", header=TRUE)
bookings$datetime[1:9]
```

```
## [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"
```

data exploration

```
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

# Get a glimpse into how minday is distributed
plot(density(minday), main="Minute (of the day) of first ever booking", col="blue", lwd=2)
```



3a

Instruction

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

```
#mean <- compute_sample_mean(sample0)
mean <- mean(minday)
se <- sd(minday)/sqrt(length(minday))
ci95 <- mean + c(-1.96*se, 1.96*se) # 95% CI

cat("Sample Mean:", mean, "\n")
cat("Standard Error:", se, "\n")
cat("95% Confidence Interval of the Sampling Means:",
    "\n","Lower Bound =", ci95[1], "\n",
    "Upper Bound =", ci95[2], "\n")
```

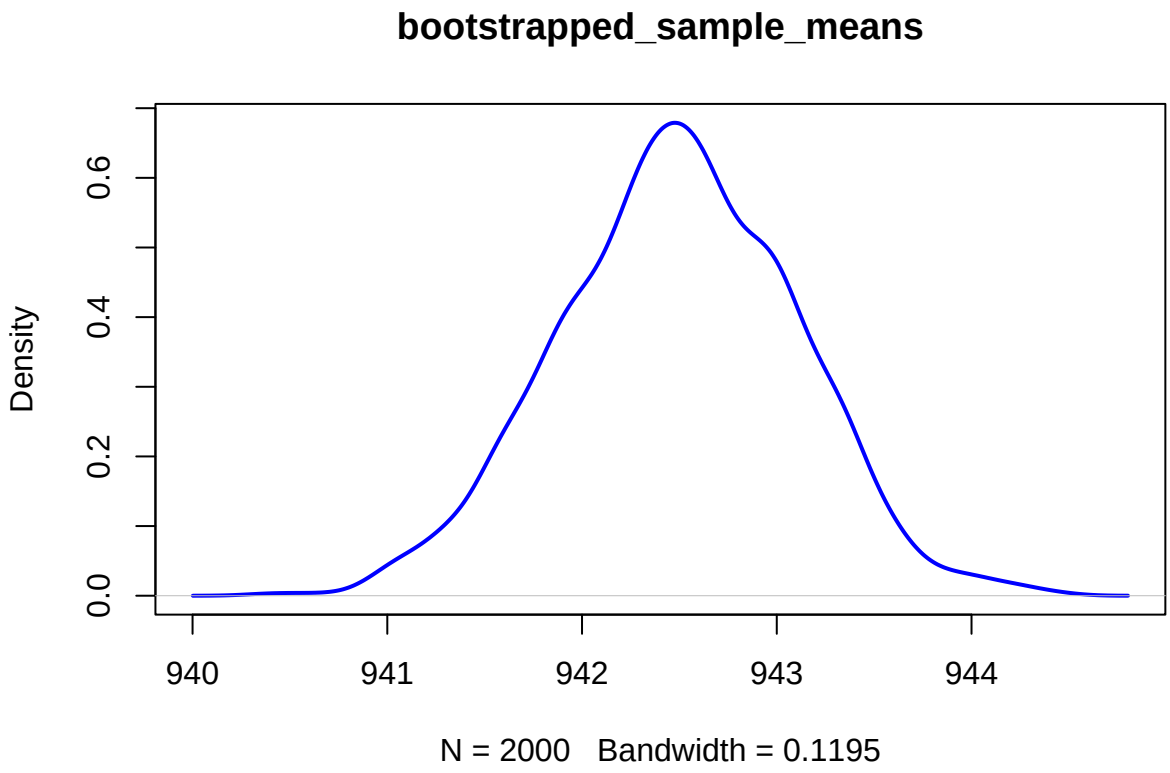
```
## Sample Mean: 942.4964
## Standard Error: 0.5997673
## 95% Confidence Interval of the Sampling Means:
##  Lower Bound = 941.3208
##  Upper Bound = 943.6719
```

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

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

(iii) Visualize the means of the 2000 bootstrapped samples

```
plot(density(resamples_mean), main = "bootstrapped_sample_means",col="blue", lwd=2)
```



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

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

```
##     2.5%    97.5%
## 941.2864 943.6229
```

3b

Instruction

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

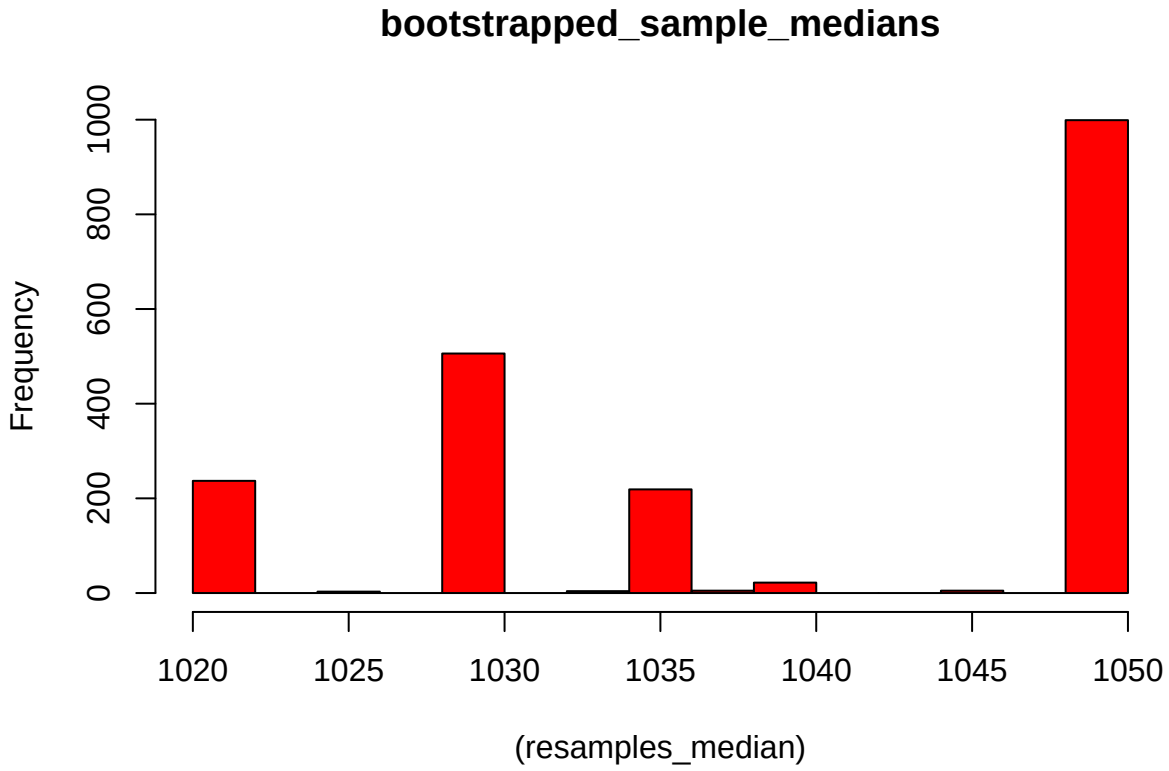
(i) Estimate the median of minday

```r
median <- median(minday)
cat("Sample Median:", median, "\n")
```

```
## Sample Median: 1040
```

(ii) Visualize the medians of the 2000 bootstrapped samples

```r
resamples_median <- replicate(2000, compute_sample_median(minday))
hist((resamples_median),main = "bootstrapped_sample_medians",col="red")
```



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

```r
quantile(resamples_median,c(0.025,0.975))
```

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