Business Analytics using Statistical Modeling - Assignment 3

Question 1

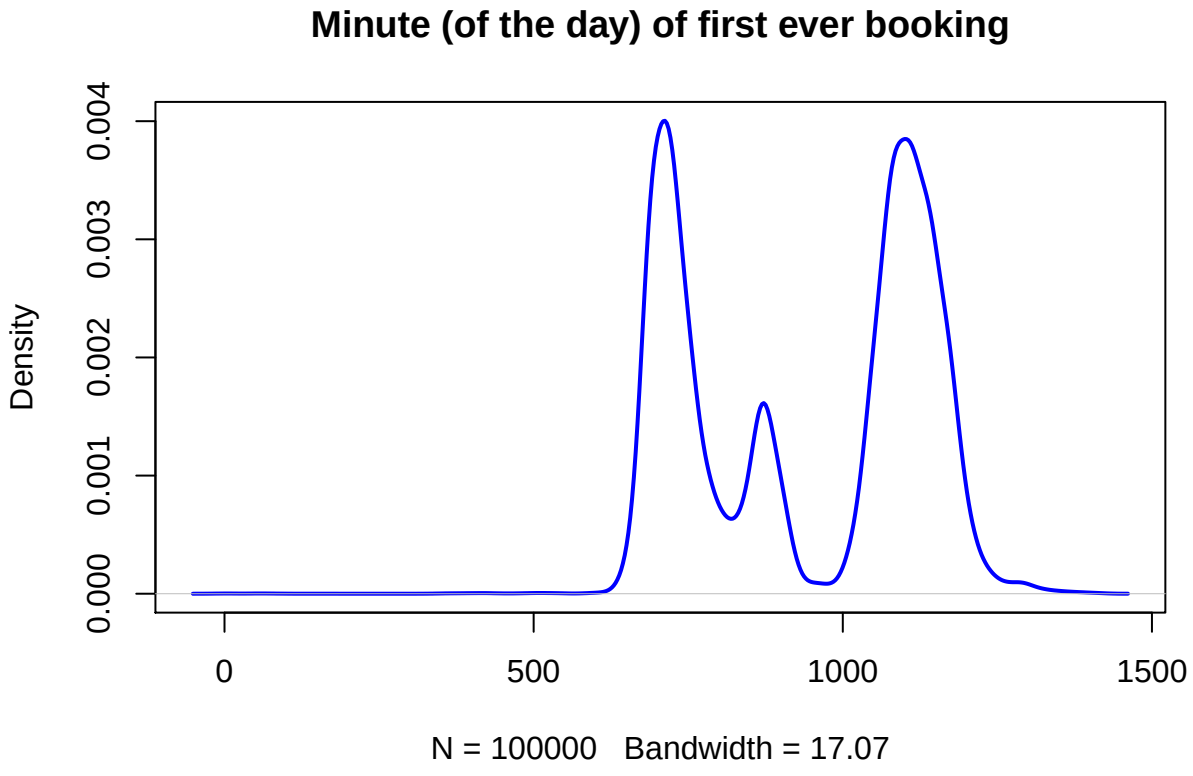
Explore the data

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.2.5
```

```
bookings <- fread('3-first_bookings_datetime_sample.txt')

datetime <- strptime(bookings$datetime, '%m/%d/%Y %H:%M')
hours <- datetime$hour
mins <- datetime$min
minday <- hours * 60 + mins
plot(density(minday), col = 'blue', lwd = 2,
     main = 'Minute (of the day) of first ever booking')
```



a) For which times of day do new members typically make their first restaurant booking?

i) Use traditional statistical methods to estimate the population mean of minday, its standard error, and its 95% confidence interval

```
sample_mean <- mean(minday)
sample_mean
```

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

```
standard_error <- sd(minday) / sqrt(length(minday))
standard_error
```

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

```
CI_95 <- c(sample_mean - 1.96 * standard_error, sample_mean + 1.96 * standard_error)
CI_95
```

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

ii) Use 2000 bootstrapped samples to estimate the 95% confidence interval of the mean

```
compute_sample_mean <- function(sample0){
  resample <- sample(sample0, length(sample0), replace = TRUE)
  return(mean(resample))
}
sample_means <- replicate(2000, compute_sample_mean(minday))
quantile(sample_means, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 941.2942 943.7311
```

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

i) Estimate the median of `minday`

```
sample_median <- median(minday)
sample_median
```

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

ii) Use 2000 bootstrapped samples to estimate the 95% confidence interval of the median

```
compute_sample_median <- function(sample0){
  resample <- sample(sample0, length(sample0), replace = TRUE)
  return(median(resample))
}
sample_medians <- replicate(2000, compute_sample_median(minday))
quantile(sample_medians, probs = c(0.025, 0.975))
```

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

Question 2

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

```
rdata <- rnorm(n = 800, mean = 940, sd = 190)
rnorm_std <- (rdata - mean(rdata)) / sd(rdata)
```

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

We expect the new mean to be equal to 0.

Since the mean is the central tendency measurement of the data, when we subtract all the data points by its mean, the new data points will be centered at 0 (positive and negative numbers), and thus the new mean will be equal to 0.

We also expect the new standard deviation to be equal to 1.

Since after subtracting all data points by its mean, we divide it by the old standard deviation, the mean distance between the new data points and 0 (the new mean) becomes 1; so the new standard deviation is expected to be equal to 1.

```
mean(rnorm_std)
```

```
## [1] 9.369889e-17
```

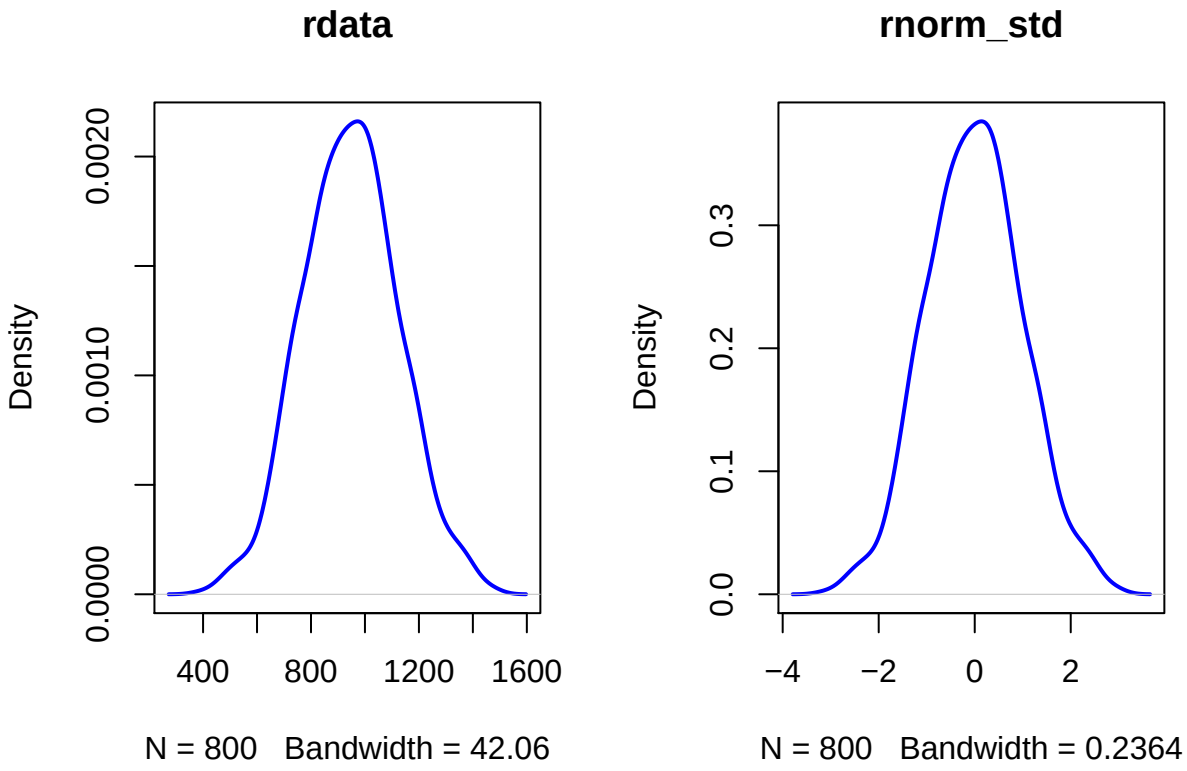
```
sd(rnorm_std)
```

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

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

The new distribution should look like the original distribution, only that it has different mean and standard deviation.

```
par(mfrow = c(1, 2))
plot(density(rdata), col = 'blue', lwd = 2, main = 'rdata')
plot(density(rnorm_std), col = 'blue', lwd = 2, main = 'rnorm_std')
```



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

Standard Normal Distributions

b) Create a standardized version of minday from above

```
minday_std <- (minday - sample_mean) / sd(minday)
```

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

We expect the mean to be equal to 0, and the standard deviation to be equal to 1. The explanation is the same as the answer of question 2.a.i. above.

```
mean(minday_std)
```

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

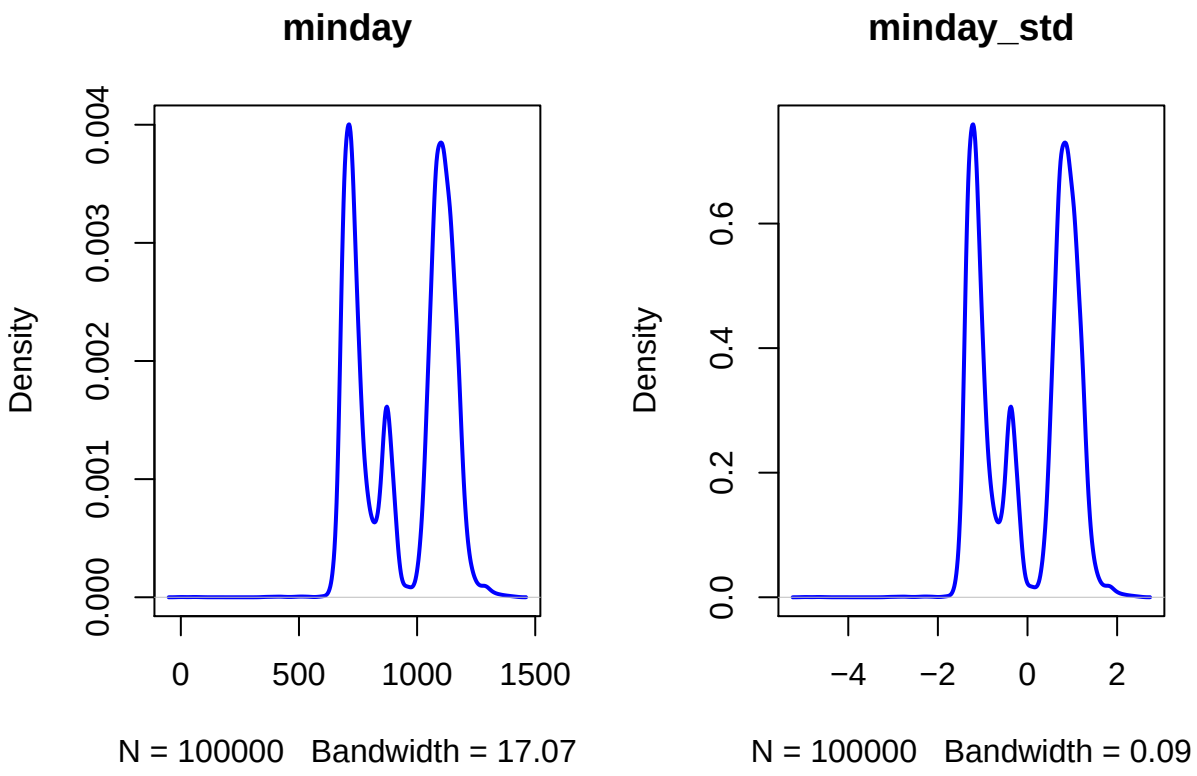
```
sd(minday_std)
```

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

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

It should also look like the original distribution, since we just standardize the values, without changing the shape.

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



Question 3

```
# 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])
}
```

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

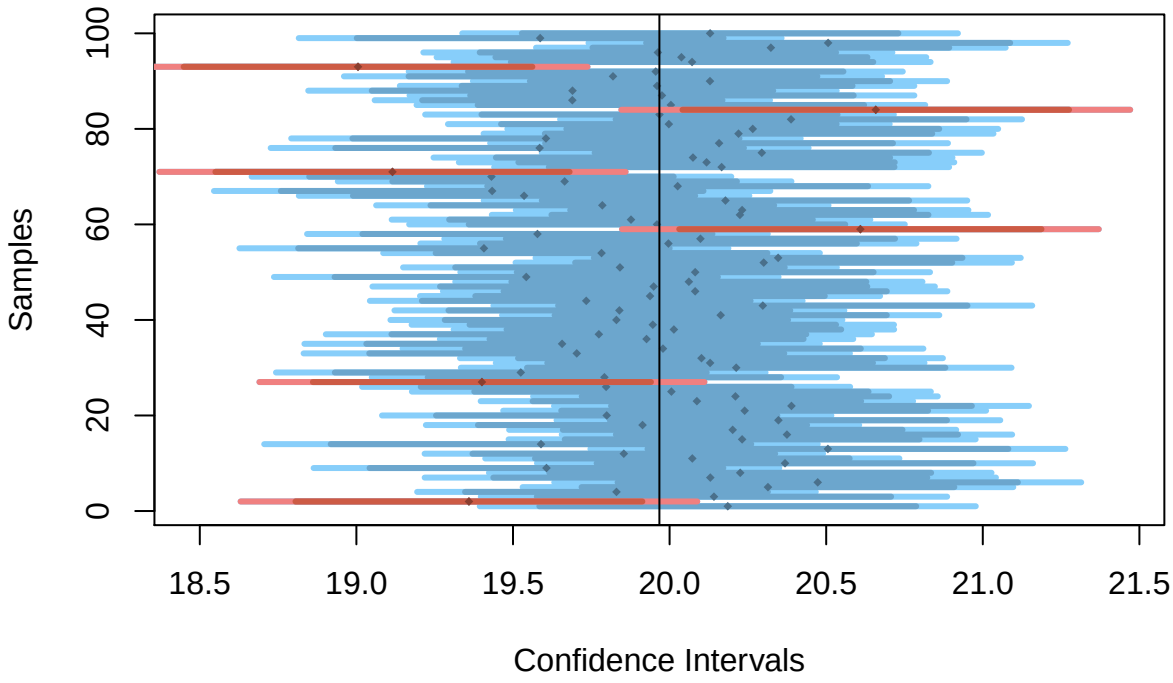
i) How many samples do we expect to NOT include the population mean in its 95% CI?

We expect 5% of the samples (5% * 100 = 5 samples) to not 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 1% of the samples (1% * 100 = 1 sample) to not include the population mean in its 99% CI.

```
visualize_sample_ci(num_samples = 100, sample_size = 100, pop_size = 10000,
                    distr_func = rnorm, mean = 20, sd = 3)
```



b) Rerun the previous simulation with larger samples (300 each)

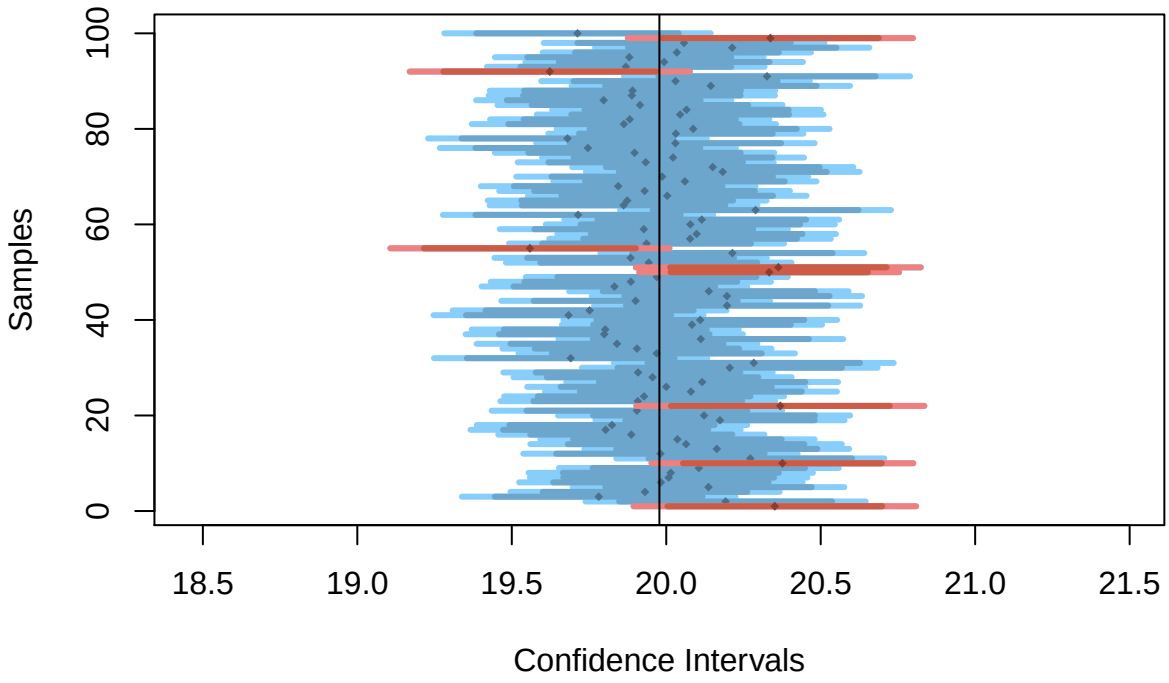
i) Now that the size of each sample has increased, do we expect their 95% and 99% CI to become wider or narrower than before?

We expect their 95% and 99% CI to become narrower, because the larger the sample size the smaller the standard error.

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 before, we still expect 5% of the samples (5% * 100 = 5 samples) to not include the population mean in its 95% CI.

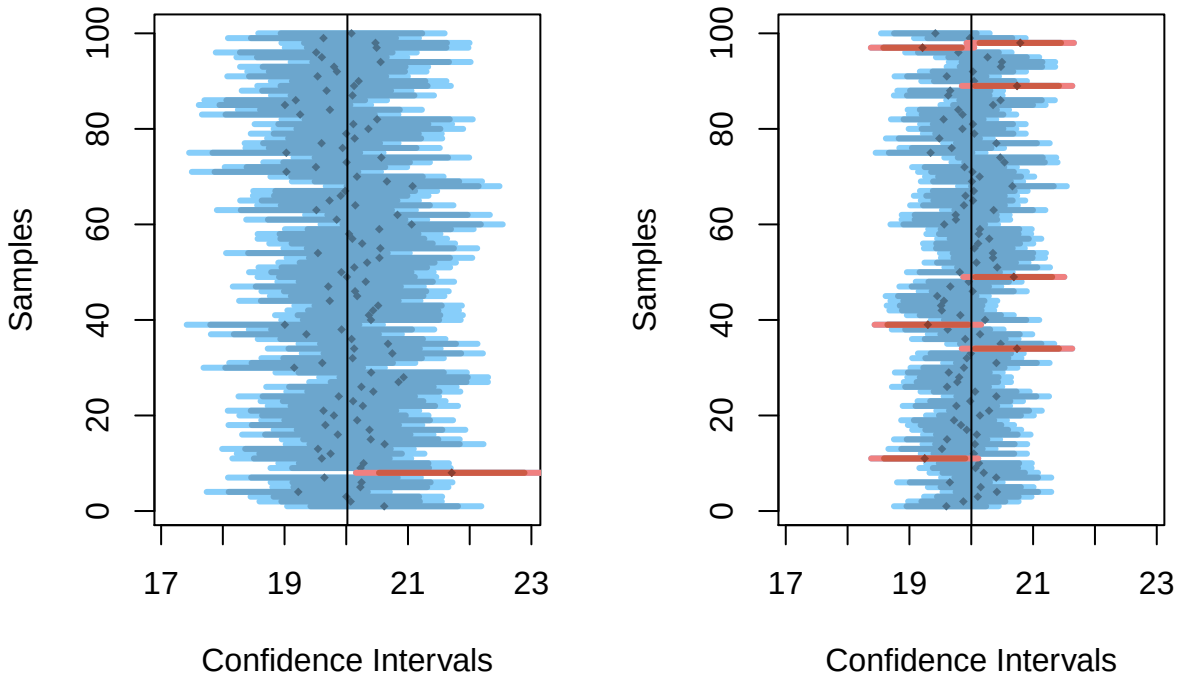
```
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size = 10000,
                    distr_func = rnorm, mean = 20, sd = 3)
```



c) If we ran the above two examples (a and b) using a uniformly distributed population, how do would you expect your answers to (a) and (b) to change?

The answers are expected to be the same: We expect 5% of the samples to not include the population mean in its 95% CI, and 1% of the samples to not include the population mean in its 99% CI. Also, we expect the larger sample size will result in smaller standard errors, thus a narrower CI.

```
par(mfrow = c(1, 2))
visualize_sample_ci(num_samples = 100, sample_size = 100, pop_size = 10000,
                    distr_func = runif, min = 10, max = 30)
visualize_sample_ci(num_samples = 100, sample_size = 300, pop_size = 10000,
                    distr_func = runif, min = 10, max = 30)
```


Samples
0
20
40
60
80
100
17
19
21
23
Confidence Intervals
Samples
0
20
40
60
80
100
17
19
21
23
Confidence Intervals