Statistical Inference Course Project

Balint SZEBENYI

2015-07-11

Overview

This document is my course project submission for the coursera course "Statistical Inference", which is part of the Data Science courses.

The project has got two objectives. One montre carlo simulation to estimate the basic characteristics of the exponential distribution. The second part analyzes the ToothGrowth data of R to illustrate an inference example.

As I am trying to learn dplyr and ggplot2 I have tried to answer the questions using these libraries.

The simulation

The simulation relies on the Central Limit Theorem and wants to show one can estimate distribution characteristics with a Monte Carlo simulation.

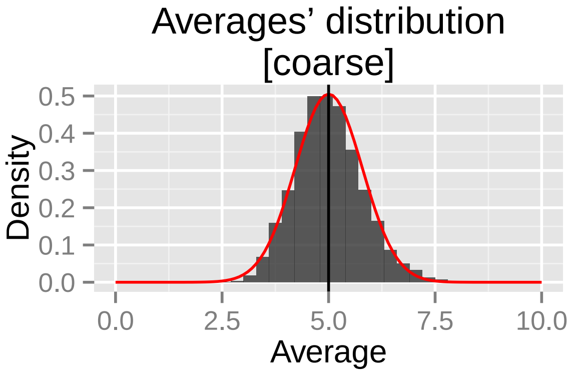
The simulation's parameters.

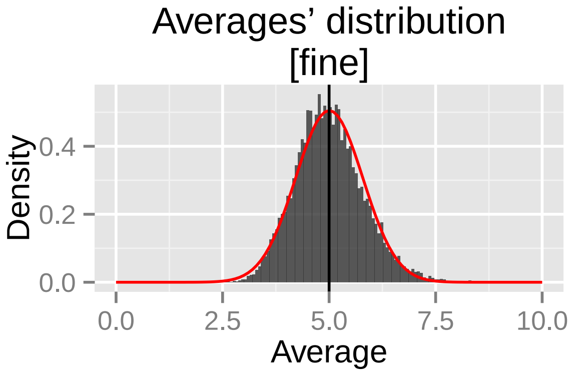
```r
lambda <- 0.2
theoretical_mean <- 1 / lambda
theoretical_sd <- 1 / lambda

number_of_simulated_items <- 40
number_of_simulations <- 10000
set.seed(seed = 128642)
```

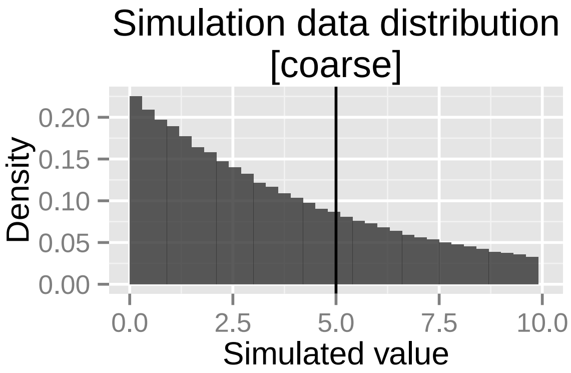
The simulation

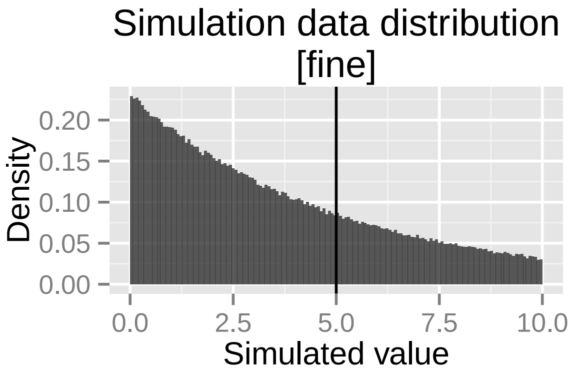
```r
simulated_x <- replicate(n = number_of_simulations,
                         expr = rexp(n = number_of_simulated_items,
                                     rate = lambda))
simulation_data <- melt(simulated_x)
simulation_data <- tbl_df(simulation_data)
colnames(simulation_data) <- c("item",
                               "run",
                               "value")
simulation_data <- simulation_data %>%
  mutate(item = rep(x = 1:number_of_simulated_items,
                    times = number_of_simulations))
simulation_data <- simulation_data %>%
  select(run, item, value)
```

The distribution of the mean of 40 exponentials

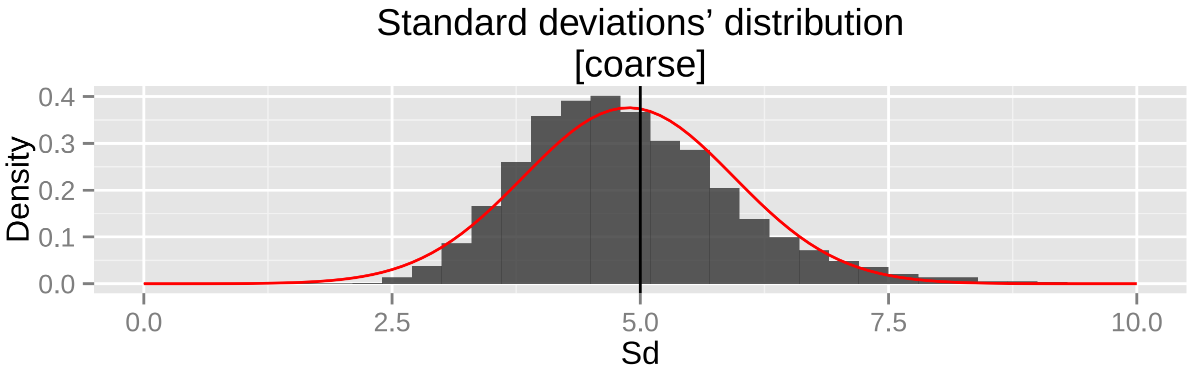


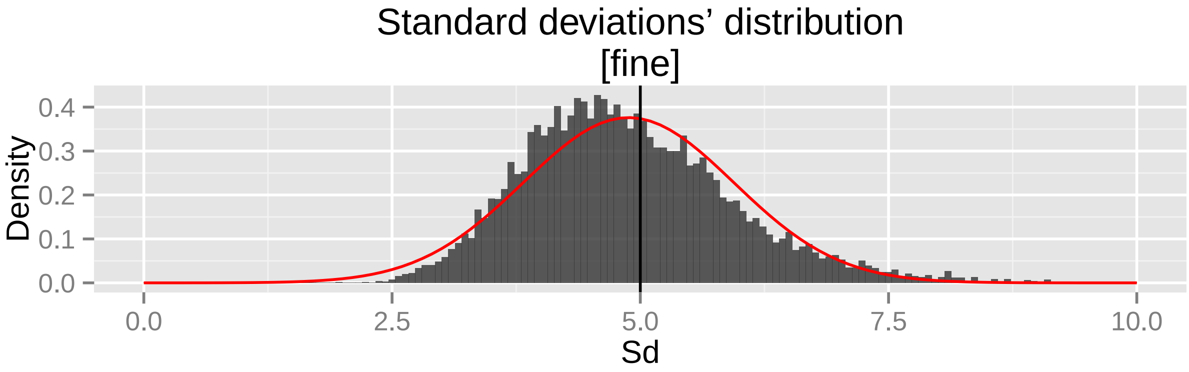






As it could be expected, the distribution is centered around 5.0029353, which lies really close to the theoretical mean of the simulated distribution 5. The red line shows the theoretical normal distribution, whole the bars represent the empirical values. The theoretical mean is highlighted by a black vertical line.





The standard deviation can be used to describe the variability of data. One could have chosen variance to fulfill this role, but I have chosen the square root of it - the result is the same. The original, simulated distribution's standard distribution parameter could be well estimated by using Monte Carlo simulation. The results are not as promising as in the case of the mean as the distribution is slightly skewed towards left compared to the normal (red) distribution. The estimated value is `4.8816731`, the theoretical standard deviation would be 5.

Appendix

First plot

```
averages <- simulation_data %>%
  group_by(run) %>%
  summarise(average = mean(value))

avg <- mean(averages$average)
sdd <- sd(averages$average)

coarse <- ggplot(data = averages,
                 aes(x = average)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 0.3,
                 alpha = 0.8) +
  stat_function(fun = dnorm,
                arg = list(mean = avg,
                           sd = sdd),
                color = "red") +
  geom_vline(xintercept = theoretical_mean) +
  xlim(0,10) +
  labs(title = "Averages' distribution\n[coarse]",
       x = "Average",
       y = "Density")
fine <- ggplot(data = averages,
                 aes(x = average)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 1/15,
                 alpha = 0.8) +
  stat_function(fun = dnorm,
                arg = list(mean = avg,
                           sd = sdd),
                color = "red") +
  geom_vline(xintercept = theoretical_mean) +
  xlim(0,10) +
  labs(title = "Averages' distribution\n[fine]",
       x = "Average",
       y = "Density")
coarse_all <- ggplot(data = simulation_data,
                 aes(x = value)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 0.3,
                 alpha = 0.8) +
  geom_vline(xintercept = theoretical_mean) +
  xlim(0,10) +
  labs(title = "Simulation data distribution\n[coarse]",
       x = "Simulated value",
       y = "Density")
fine_all <- ggplot(data = simulation_data,
                 aes(x = value)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 1/15,
                 alpha = 0.8) +
  geom_vline(xintercept = theoretical_mean) +
  xlim(0,10) +
  labs(title = "Simulation data distribution\n[fine]",
       x = "Simulated value",
       y = "Density")
grid.arrange(coarse, fine, coarse_all, fine_all)
```

Second plot

```
std_devs <- simulation_data %>%
  group_by(run) %>%
  summarise(std_dev = sd(value))

avg <- mean(std_devs$std_dev)
sdd <- sd(std_devs$std_dev)

coarse <- ggplot(data = std_devs,
                 aes(x = std_dev)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 0.3,
                 alpha = 0.8) +
  stat_function(fun = dnorm,
                arg = list(mean = avg,
                           sd = sdd),
                color = "red") +
  geom_vline(xintercept = theoretical_sd) +
  xlim(0,10) +
  labs(title = "Standard deviations' distribution\n[coarse]",
       x = "Sd",
       y = "Density")
fine <- ggplot(data = std_devs,
                 aes(x = std_dev)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 1/15,
                 alpha = 0.8) +
  stat_function(fun = dnorm,
                arg = list(mean = avg,
                           sd = sdd),
                color = "red") +
  geom_vline(xintercept = theoretical_sd) +
  xlim(0,10) +
  labs(title = "Standard deviations' distribution\n[fine]",
       x = "Sd",
       y = "Density")
grid.arrange(coarse, fine)
```