

# A GitBook template

Nikola Sekulovski

2021-09-27



# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Monte Carlo Simulations</b>	<b>7</b>
2.1	The Confidence Interval . . . . .	7
2.2	The Central Limit Theorem . . . . .	9
<b>3</b>	<b>Formulas</b>	<b>11</b>
3.1	Bayes theorem . . . . .	11
3.2	Normal PDF . . . . .	11
	<b>Literature</b>	<b>13</b>
	<b>Appendix A</b>	<b>15</b>



# Chapter 1

## Introduction

This is a template GitBook based on *A GitBook Example for Teaching and bookdown: Authoring Books and Technical Documents with R Markdown*.



## Chapter 2

# Monte Carlo Simulations

```
library(tidyverse)
```

### 2.1 The Confidence Interval

In this exercise I will try to repeat the example given by Gerko Vink

The main idea of this exercise is to illustrate the nature of the *Confidence Interval* as described by Neyman (1934)

We set a seed to make our results reproducible:

```
set.seed(6465)
```

- The first step is to take 100 samples (in this case of size 800) from a *normal distributuon* with  $\mu = 0$  and  $\sigma = 1$ :

```
samples <-plyr::rply(100, rnorm(800, 0, 1))
```

- Secondly, we need to calculate for the mean of each sample: the absolute bias; standard error lower bound of the 95% confidence interval and upper bound of the 95% confidence interval.

We can construct a function that does this:

Table 2.1: Here is a table of the samples

Mean	Bias	Std.Err	Lower	Upper	Covered
-0.0945589	0.0945589	0.0353553	-0.1639592	-0.0251585	0
0.0740058	0.0740058	0.0353553	0.0046055	0.1434062	0

```
samp_function <- function(x) {
  m <- mean(x)
  n <- length(x)
  se <- 1/sqrt(n)
  bias <- abs(-0 - m)
  df <- n - 1
  interval <- qt(.975, df) * se
  return(c(m, bias, se, m - interval, m + interval))
}

format <- c("Mean" = 0, "Bias" = 0, "Std.Err" = 0, "Lower" = 0, "Upper" = 0)
```

Now we use the constructed function `samp_function` on all 100 samples contained in the object `samples`. And we also add a new column to the results that indicates which CI of the respective samples does contain  $\mu$ .

```
results <- samples %>%
  vapply(., samp_function, format) %>%
  t %>%
  as_tibble %>%
  mutate(Covered = ifelse(Lower < 0 & Upper > 0, 1, 0))
```

We can also add a table with the sample statistics of the samples whose CI's do not contain  $\mu$ .

```
results %>%
  filter(Covered == 0) %>%
  kableExtra::kable(caption = "Here is a table of the samples" )
```

And finally we can also make a nice plot illustrating everything that we did so far.

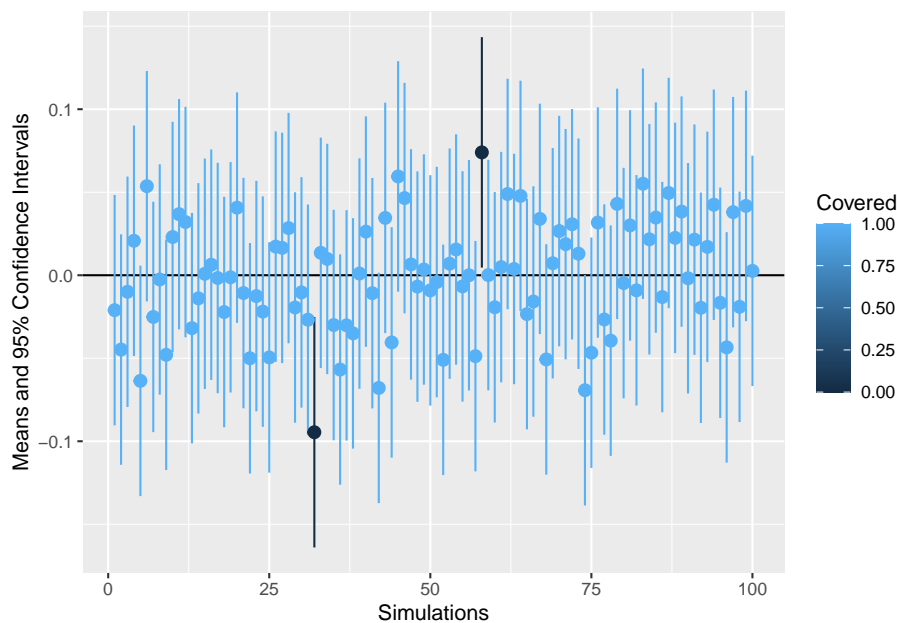
```
lims <- aes(ymax = results$Upper, ymin = results$Lower)
ggplot(results, aes(y=Mean, x=1:100, colour = Covered)) +
  geom_hline(aes(yintercept = 0)) +
```



```
geom_pointrange(lims) +
xlab("Simulations") +
ylab("Means and 95% Confidence Intervals")
```

```
## Warning: Use of 'results$Upper' is discouraged. Use 'Upper' instead.
```

```
## Warning: Use of 'results$Lower' is discouraged. Use 'Lower' instead.
```



In this case only two out of 100 CI's do not include the true population mean.

## 2.2 The Central Limit Theorem

Here we will also try to illustrate the Central Limit Theorem, in it's most basic form, with a very simple example.

First we draw 1000 samples (again of size 800), from , say, a *Poisson* distribution, of course we could've drawn them from a uniform or an exponential as well.

```
samples_2 <- samples <-plyr::rply(1000, rpois(800, 2))
```

Now we calculate the mean for each sample:

```
means <- samples_2 %>%  
  lapply(., mean) %>%  
  as.data.frame() %>%  
  t()
```

And now we plot a histogram of the resulting means:

```
hist(t(means))
```

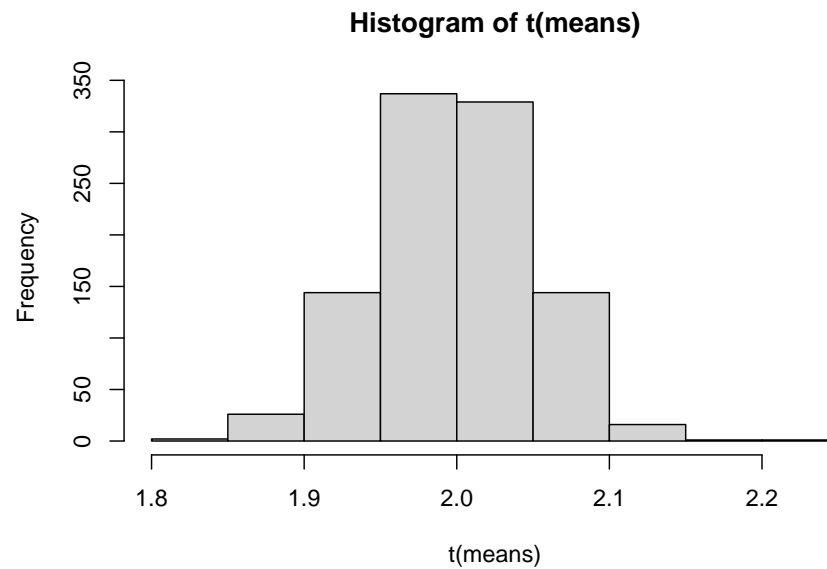


Figure 2.1: Histogram of the sampling distribution of the mean

## Chapter 3

# Formulas

### 3.1 Bayes theorem

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \quad (3.1)$$

### 3.2 Normal PDF

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (3.2)$$

This is how we refer to equations: -see equation (3.2)



# Literature



# Appendix A