

Using Shiny Apps in Teaching Statistics

Diana Šimić

September 8, 2021

Where do we start

<https://github.com/disimic/biostat2021> (<https://github.com/disimic/biostat2021>)

What do we need for the workshop?

- Install R from <https://www.r-project.org/> (<https://www.r-project.org/>)
- Install RStudio Desktop from <https://www.rstudio.com/products/rstudio/> (<https://www.rstudio.com/products/rstudio/>)
- Open the script `install_packages.R` in RStudio and run



How did you manage? [menti.com 8363 7556](https://menti.com/83637556)

What do you expect from the workshop?



menti.com 8363 7556

Let's see some examples

Critical value for the confidence interval

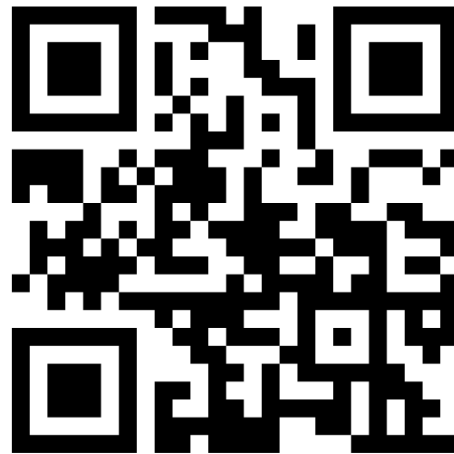
- Open the github.com/disimic/biostat2021
- Click on Links.md
- Click on Critical value for a confidence interval
- Open menti.com and answer the question



[menti.com 8363 7556](https://menti.com/83637556)

Correlation and the linear regression

- Open the github.com/disimic/biostat2021
- Click on Links-md
- Click on Correlation and linear regression
- Try out tabs “Correlation Coefficient” and “The Least Squares”
- Open menti.com and answer the question



[menti.com 8363 7556](https://menti.com/83637556)

How close did you get?

- Open the github.com/disimic/biostat2021
- Click on Links.md
- Click on Correlation and linear regression
- On tab One more example find the “real” values of regression coefficients.

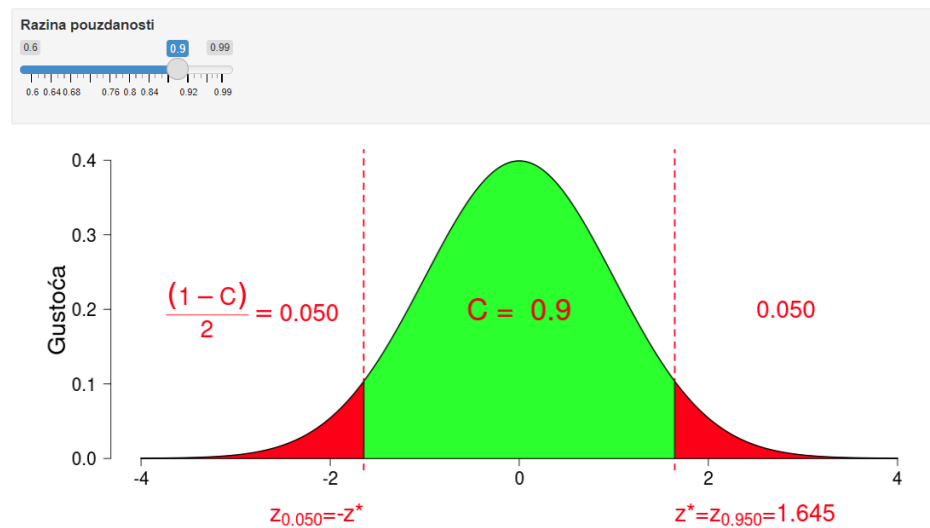
Parts of a Shiny App

Critical Value for a Confidence Interval

Diana Šimić

September 2021

How do we find the critical value for computing a margin of error for a confidence interval based on normal distribution? This animation displays the relationship between the confidence level and the critical value. For a confidence level C , the critical value is the $C + \frac{1-C}{2}$ -th quantile of the standard normal distribution. The green area is C , and the two red areas together are $1 - C$.



- Title and static text and images (does not change)
- Input (can be changed dinamically)
- Output (depends on/changes with the Input)

A simple Shiny app using Rmarkdown

- From the File menu select New File => R Markdown
- Type in a title and your name
- On the left choose Shiny
- For Default Output Format choose Shiny Document
- Click OK
- In the window of the new script click “Run Document”
 - when asked provide a file name Example-1.Rmd

Parts of an R Markdown Shiny document

- YAML header
- text (formatted using markdown)
- R code chunks (this is where the magic happens)

YAML header

```
---  
title: ""  
author: ""  
date: ''  
output: html_document  
runtime: shiny  
---
```

Just fill in title, author, and date, and don't touch anything else :-)

R Markdown text

Function

Format

Section headers:

#, ##, ### at the beginning of a line

Paragraph:

two spaces at the end of a line

Bulleted list:

start a new line with a - or * or +

Nested lists:

indent the list

Numbered list:

start a new line with number 1. etc.

Italics:

put text between characters *

Bold:

put text between characters **

R Code chunks

- Code chunks start with `` `` {r}`, and end with `` ```
- Options can be added within the braces
- Code in a chunk is run, and the output is included in the document

For example chunk:

```
` `` {r echo=TRUE}  
2+2  
` ``
```

will result in:

```
2+2
```

```
## [1] 4
```

Shiny means reactive

- Shiny package makes an R Markdown document reactive
- `inputPanel()` is used to get input from a reader. Arguments are comma separated input controls
- `renderXXXX()` is used to create output that is automatically changed when input changes. XXXX can be Text, Plot, Image, Table etc. for different types of output

Shiny input functions 1/3

```
checkboxInput("Flag", "Is it blue?", value=FALSE) renderPrint(paste("And the result is:", input$Flag))
```

Is it blue? ☐

```
[1] "And the result is: FALSE"
```

```
selectInput("Select", "Select:", choices=c("A", "B", "C"), selected = "A", multiple = FALSE)
```

Select:

```
[1] "And the result is: A"
```


Shiny input functions 2/3

```
radioButtons("rad.id", "Select:", choices=c("A", "B", "C"), selected = "A",  
inline=FALSE)
```

Select:

- A ☒
- B ☐
- C ☐

```
[1] "And the result is: A"
```

```
numericInput("Number", "How long is it?", min = 1, max = 100, step = 1, value = 10 )
```

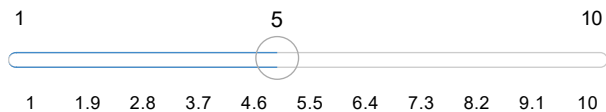
How long is it?

```
[1] "And the result is: 10"
```

Shiny input functions 3/3

```
sliderInput("Slide1", "How much?", min=1, max=10, step=0.1, value=5)
```

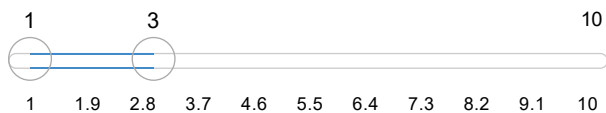
How much?



```
[1] "And the result is: 5"
```

```
sliderInput("Slide2", "What are the limits?", min=1, max=10, step=0.1, value=c(1,3))
```

How much?



```
[1] "And the result is: 1 to 3"
```

Back to playing ...



[menti.com 8363 7556](https://menti.com/83637556)

Let's get working

- Open the github.com/disimic/biostat2021
- Click on Links.md
- Click on Activity - Create your first Shiny document
- Try to create a Shiny document that looks and reacts exactly like Example 2

Adding more structure

- This is just a one page document.
- More structure can be added using the package flexdashboard
- Open the document “LinearRegression_en.Rmd”

The header

```
---  
title: "Correlation and linear regression"  
output:  
  flexdashboard::flex_dashboard:  
    orientation: columns  
    vertical_layout: fill  
runtime: shiny  
---
```

Before, we had a simple line

```
output: html_document
```

Dashboard structure

- With flexdashboard we can add structure
- the first level headers introduce tabs (with one # or double underscores)
- the second level headers introduce columns (with two ## or single underscore)
- the third level headers introduce titles within columns, i.e. rows

Special second order header:

```
Input {.sidebar}
```

creates the left column with slightly different format.

Flexdashboard options

With options in the YAML header we can change the layout of a dashboard

- orientation can be columns or rows
- vertical_layout can be fill or scroll

We can add options after column (row) headings

- {data-orientation=columns | rows}
- {data-width=}
- {data-height=}

And now to your ideas

Assignment:

- develop an idea for a Shiny app that you would like to use in teaching your classes.
- think about what are inputs, and what are outputs of that Shiny app
- Open the github.com/disimic/biostat2021
- Click on Links.md
- Click on Activity - Add your ideas to a Google doc
- In the Google doc write your
 - name or nick
 - short description of your Shiny idea
 - identify inputs and outputs