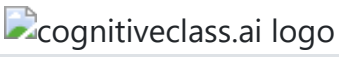
CheatSheet - DashBoard



Command	Syntax	Description	Example
Install shiny	`install.packages("name of the packae")`	Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R.	`install.packages("shiny")`
load shiny	`library("name of the packae")`	Run library(shiny) to load the shiny and make it available in your current R session.	`library(shiny)`
shinyUI	`shinyUI(ui)`	files to register a user interface with Shiny.	
fluidPage	`fluidPage(..., title = " " , ...,)`	To create rows within the grid you use the fluidRow() function.	`fluidPage(titlePanel("Motor Trend Car Road Tests Data"), verticalLayout())`
titlePanel	`titlePanel("title")`	An application title to display.	`titlePanel("Motor Trend Car Road Tests Data")`
vertical_layout	`vertical_layout(objects...)`	Lays out elements vertically, one by one below one another.	`verticalLayout(plotOutput("histPlot"),sliderInput( inputId = "bins", label = "Number of bins:", min = 1, max = 10, value = 5)`
sliderInput	`sliderInput(inputId, label, min, max, value)`	Constructs a slider widget to select a numeric value from a range.	`sliderInput("bins", "Number of bins:", 1, 10, 5)`
plotOutput	`plotOutput(outputId, width = "", height = "",...)`	Constructs a slider widget to select a numeric value from a range.	`sliderInput("bins", "Number of bins:", 1, 10, 5)`
Install ggplot2	`install.packages("name of the packae")`	ggplot2 is a plotting package that makes it simple to create complex plots from data in a data frame.	`install.packages("ggplot2")`
load ggplot	`library("name of the packae")`	Run library(ggplot) to load the ggplot and make it available in your current R session.	`library(ggplot)`
ggplot	`ggplot(object..)`	It can greatly improve the quality and aesthetics of your graphics, and will make you much more efficient in creating them.	`ggplot(mtcars, aes(x = mpg)) + geom_histogram(bins = input$bins)`
geom_histogram	`geom_histogram()`	Histograms (geom_histogram()) display the counts with bars.	`geom_histogram(bins = input$bins)`
theme	`theme(line,rect,text,...)`	Themes are a powerful way to customize the non-data components of your plots.	`theme(text = element_text(size = 20)`
shinyServer	`shinyServer(objects...)`	This generally involves creating functions that map user inputs to various kinds of output.	`shinyServer( function(input, output))`

renderPlot	`renderPlot(objects...)`	Renders a reactive plot that is suitable for assigning to an output slot.	`renderPlot({ggplot(mtcars, aes(x = mpg)) + geom_histogram(bins = input$bins)})`

Author(s)

D.M Naidu

Changelog

Date	Version	Changed by	Change Description
2020-08-11	1.0	D.M. Naidu	Initial Version