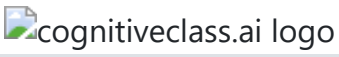
CheatSheet - Dashboard Application



Command	Syntax	Description	Example
Install shiny	`install.packages("name of the package")`	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 package")`	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")`
sidebarLayout	`sidebarLayout(sidebarPanel, mainPanel, position = c("left", "right"))`	Create a layout (sidebarLayout()) with a sidebar (sidebarPanel()) and main area (mainPanel()).	`sidebarLayout( sidebarPanel( sliderInput("obs", "Number of observations:",0,1000,500 ), mainPanel(plotOutput("distPlot") )`
sidebarPanel	`sidebarPanel(..., width = 4)`	The sidebarPanel() containing input controls.	`sidebarPanel( sliderInput("obs", "Number of observations:",0,1000,500 )`
varSelectInput	`varSelectInput( inputId, label, data, selected = " ")`	Create a select list that can be used to choose a single or multiple items from the column names of a data frame.	`varSelectInput("continuous_variable", "Select Continuous Variable", data = select(mtcars, -categorical_varibles), selected = "mpg")`
numericInput	`numericInput( inputId, label, value)`	Create an input control for entry of numeric values.	`numericInput("bins","Number of bins",min = 2, max = 20,value = 10)`
radioButtons	`radioButtons(inputId, label, choices, selected = NULL)`	Create a set of radio buttons used to select an item from a list.	`radioButtons("hist_fill", "Histogram fill:", choices = c("default", "blue"))`
mainPanel	`mainPanel(objects...)`	Create a main panel containing output elements that can in turn be passed to sidebarLayout.	`mainPanel(tabsetPanel( tabPanel("Distribution of Numerical Variables",plotOutput("p1"), plotOutput("p2") )`
tabsetPanel	`tabsetPanel(objects...)`	Tabsets are useful for dividing output into multiple independently viewable sections.	`tabsetPanel( tabPanel("Distribution of Numerical Variables",plotOutput("p1"),plotOutput("p2"))`

plotOutput	`plotOutput(outputId, width = "", height = "",...)`	Constructs a slider widget to select a numeric value from a range.	`plotOutput("p1")`
Install ggplot2	`install.packages("name of the package")`	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 package")`	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 = !!input$continuous_variable))`
geom_histogram	`geom_histogram()`	Histograms (geom_histogram()) display the counts with bars.	`geom_histogram(bins = input$bins)`
geom_boxplot	`geom_boxplot()`	The boxplot compactly displays the distribution of a continuous variable.	`ggplot(mtcars, aes(y = !!input$continuous_variable)) +geom_boxplot()`
coord_flip	`coord_flip(xlim = NULL, ylim = NULL, expand = TRUE)`	Flipped cartesian coordinates so that horizontal becomes vertical, and vertical, horizontal. This is primarily useful for converting geoms and statistics which display y conditional on x, to x conditional on y.	`ggplot(mtcars, aes(y = !!input$continuous_variable)) + geom_boxplot() + coord_flip()`
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({p <- ggplot(mtcars, aes(x = !!input$continuous_variable))})`

Author(s)

D.M Naidu

Changelog

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