

Descriptive Analytics-Part 6: Interactive dashboard (2/2)



Descriptive Analytics is the examination of data or content, usually manually performed, to answer the question “What happened?”. As this series of exercises comes to an end, the last part is going to be the development of a data product. Not

everybody is able to code in R, so it is useful to be able to make GUIs in order to share your work with non-technical people. This part may be a little challenging, since it requires some basic knowledge of the shiny package. The outcome of this set of exercises will be almost like [this](#) web app (some variables are missing because I had to reduce the size of the data set).

In order to be able to solve this set of exercises you should have solved the [part 0](#), [part 1](#), [part 2](#), [part 3](#), and [part 4](#) of this series but also you should run this [script](#) which contain some more data cleaning. In case you haven't, run this [script](#) in your machine which contains the lines of code we used to modify our data set. This is the tenth set of exercise of a series of exercises that aims to provide a descriptive analytics solution to the '2008' data set from [here](#). This data set which contains the arrival and departure information for all domestic flights in the US from 2008 has become the “iris” data set for Big Data. The goal of Descriptive analytics is to inform the user about what is going on at the dataset. Before proceeding, it might be helpful to look over the help pages for the `fluidPage`, `pageWithSidebar`, `headerPanel`, `sidebarPanel`, `selectInput`, `mainPanel`, `tabPanel`, `observe`, `verbatimTextOutput`, `renderPrint`, `shinyApp`.

For this set of exercises you will need to install and load the package shiny.

```
install.packages('shiny')
library(shiny)
```

I have also changed the values of the DaysOfWeek variable, if you wish to do that as well the code for that is :

```
install.packages('lubridate')
library(lubridate)
flights$DayOfWeek <-
wday(as.Date(flights$Full1_Date, '%m/%d/%Y'), label=TRUE)
```

Because the app requires some time to run, I have also removed the rows with missing values from the data set just to save some time.

```
flights <- flights[which(!is.na(flights['WeatherDelay'])),]
flights <- flights[which(!is.na(flights['ArrDelay'])),]
```

Answers to the exercises are available [here](#).

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page. Moreover it would be really nice of you to share the links of the apps you have developed. It would be a great contribution the community.



Learn more about Shiny in the online course [R Shiny Interactive Web Apps – Next Level Data Visualization](#). In this course you will learn how to create advanced Shiny web apps; embed video, pdfs and images; add focus and zooming tools; and many other functionalities (30 lectures, 3hrs.).

Exercise 1

Create the user interface and set as the header of the web app : “Descriptive Analysis”

Exercise 2

Create a side panel.

Exercise 3

Create two select list input control. The former will contain the variables: CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay. The latter will contain the variables :Dest, Origin, UniqueCarrier, TailNum, CancellationCode.

Exercise 4

Create a set of radio buttons used to select a plot from a list (Histogram, Scatter plot, Violin plot),and set as default plot the Histogram.

Exercise 5

Create a set of radio buttons used to select a plot from a list (bar plot, pie chart, rose wind),and set as default plot the bar plot.

Exercise 6

Create a main panel.

Exercise 7

Create in the main panel two tabs named “Delays” and “Categorical” that will contain the plots of the exercises 4 and 5 respectively.

Exercise 8

Now that we are done with the user interface, create the server side of the app. Create the output of the first tab, which will be the plots from exercise 4 in respect to the first set of variables from exercise 3 (notice that they are all continuous variables), bear in mind that at the scatter plot the x-axis should be the Full_Date and at the violin plot the x-axis should be the DayOfWeek as we did at the previous

set of exercises. (please check out the first tab of the app, to make things more clear).

Exercise 9

Create the output of the second tab, , which will be the plots from exercise 5 in respect to the second set of variables from exercise 3 from the exercise 5, use the knowledge you applied (or acquired at the previous exercises for the plots, make them as interesting as you can).(please check out the second tab of the app, to make things more clear).

Exercise 10

Launch the app.