App Documentation

Martin Richardson

October 28, 2017

The data

This application performs a cursory analysis of the 'mtcars' data set found in the R 'datasets' library. A preview of this data set is as follows:

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

So there are 32 observations, recording 11 quantities for 32 different models of car. The quantities measured are the miles-per-gallon rating ('mpg'), the number of engine cylinders ('cyl'), the engine displacement ('disp'), the engine horsepower ('hp'), the rear axle ratio ('drat'), the car weight ('wt'), the quarter-mile time ('qsec'), the engine type (V-shaped or straight) ('vs'), the transmission type ('am'), the number of forward gears ('gear'), and the number of carburetors ('carb').

The calculations

This app creates a 2-D scatter plot where the user can select any combination of three of the 11 variables from a drop-down list panel; two to be plotted on either axis, and a third to colour the points by. There is also a line drawn on this plot that comes from a linear regression built from user inputs.

This app calculates a linear regression model based on a user-input-defined formula. The criterion variable in this case is always mpg, and using a checkbox, the user can select any of the other variables to be predictors for mpg.

The R function used is lm(), which takes as its first argument a formula of the form "criterion ~ predictor_1 + predictor_2 +…" . These predictor variables are chosen by a list of checkboxes in the sidebar.

In addition to being used to draw a line on the plot, this linear model also has an 'R-squared' value calculated for it, which quantifies the amount of variance explained by the current regression model. This is displayed in the lower right corner of the app and also updates based on user choices.

Getting started

When the app first starts, the plot shows wt on the x-axis and mpg on the y-axis, and the regression line shown is based on the formula "mpg ~ wt". The default variable for the colour of the points is cyl. The user is free to make any adjustments to the drop-down lists and checkboxes that they wish, and any changes made by the user will be processed and displayed on the plot immediately. However, there must be at least one checkbox ticked or no plot will be displayed and instead an error message will appear.

The code

User Interface file

The UI file is structured as follows:

```r
shinyUI(fluidPage(

  tags$head(
             # Code to insert background image
  ),

  titlePanel(
              # The main title
  ),

  inputPanel(
              # Three user input drop-down lists, for selection of variables
              # x_var, y_var, and colour_var
  ),

  sidebarLayout(
    sidebarPanel(
                  # Checkboxes for choosing lm() formula predictors, where
                  # the variable names are the 11 variables of 'mtcars'
                  # aside from 'mpg'
    ),

    mainPanel(
               # plot displays alongside sidebar in the main panel
               # referred to as 'myplot'
    )
  ),

  mainPanel(
             # R-squared value displays after sidebar in the main panel
  )
))
```

Server file

The server file is structures as follows:

```
shinyServer(function(input, output) {

  output$myPlot <- renderPlot({
    # Create formula for linear model:
    # -> Take boolean vector based on checkbox sidebar i.e. c(input$cyl, input$disp, ...)
    # -> Create formula string (mpg ~ predictor_i + predictor_j + predictor_k + ...)

    # Create linear model via lm(formula, data = mtcars)

    # Make plot via plot(x = mtcars[[input$x_var]], ...)

    # Add regression line to plot via abline()

    # Retrieve and display R-squared value from linear model
    })
  })
```