kadaoui_alexandre_Shiny

Alexandre

23/12/2020

R Markdown

Dans le cadre de notre partiel, nous devons réaliser un total de 12 travaux retracant notre parcours et notre travail durant les 30 heures de cours.
Le travail à faire est le suivant :
- Une entête comportant un titre, un lien Github avec le ou les noms des auteurs.
- Une synthese de ce travail
- Un extrait commenté avec des parties de codes clé avec explication et commentaire.
- Une évalutation du travail avec nos 5 criteres.
- Une conclusion du travail

Definition des 5 critères de notations :

1) Effort de présentation :
2) Le knit est réalisable et bien présenté.
3) Explications simples et efficaces.
4) Le Code reproductible à d'autres DataFrame avec facilité.
5) Description des fonctions utilsés et du raisonnement.

Rpart

Travail réalisé par "Benjamin Guigon" le 12/11/2020.

GitHub: https://github.com/benjaminguigon/PSBX/blob/main/V2%20shiny.R (https://github.com/benjaminguigon/PSBX/blob/main/V2%20shiny.R)

Synthese :

Extrait commenté du code :

```
#install.packages('promises')
#install.packages('shiny')
#install.packages('plotly')
library(shiny)
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise
```

```
library(flexdashboard)
library(cowplot)
library(shinydashboard)
```

```
##
## Attaching package: 'shinydashboard'
```

```
## The following objects are masked from 'package:flexdashboard':
##
##     renderValueBox, valueBox, valueBoxOutput
```

```
## The following object is masked from 'package:graphics':
##
##     box
```

```
ui <- dashboardPage(skin = "red", # Choix de la couleur de l'en tête
    dashboardHeader(title = "Dashboard général"), # Création d'n dashboard général dans leque
l
    # l'on créera nos différentes partie de l'interface

    dashboardSidebar( "Choix des pages", # Création d'une sidebar pour pouvoir jongler entre
 plusieurs onglets dans le Dashboard

                      sidebarMenu( # Menu permettant de créer les différents onglets du dashb
oard
                                   # https://fontawesome.com/icons?from=io retrouver tous le
s icones disponibles
                                   menuItem("Iris", tabName = "iris", icon = icon("tree")), #
Onglet 1
                                   menuItem("Cars", tabName = "cars", icon = icon("car")), #
 Onglet 2
                                   menuItem("Machine Learning", tabName = "M_L", icon = icon(
"database")))), # Onglet 3
    #-------------------------------------------------------------------------
    dashboardBody( # Permet de créer une fenêtre principale
        tabItems( # Permet de créer plusieurs graph dans la même fenêtre mais pas dans les mê
mes onglets
            # Attention, il y a un "s" au tabItems pour spécifié qu'il y aura plusieurs Items
            #-----------------------------------------------------------------
            # Création de la page Iris
            tabItem("iris",
                    fluidPage(h1("Iris")), # Création d'un titre à la page avec les balises H
TML
                    box(plotOutput("correlation_plot"),width = 8), # Création d'une 1ère box
 pour afficher mon Graph de correlation
                    box(selectInput("features","Features:", c("Sepal.Width","Petal.Length","P
etal.Width")), width = 4), #
                    box(plotOutput("correlation_plot_2")) # Affichage d'un 2ème graph sur une
2ème box
                    ),
            #-----------------------------------------------------------------
            # Création de la page Cars
            tabItem("cars",
                    fluidPage(h1("Cars")),
                    dataTableOutput("carstable")
                    ),
            #-----------------------------------------------------------------
            # Création de la page Machine learning
            tabItem("M_L",
                    fluidPage(h1("Machine learning with Shiny"),
                              wellPanel(
                                  tabsetPanel(
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 1",
                                                 box(numericInput(
                                                     "SD",label = "Standard Deviation", value
= 3)),
                                                 box(numericInput(
                                                     "Sample",label = "Sample size", value =
50,step = 10)),
                                                 plotOutput("Tableau_1")
```

```
                                          ),
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 2",
                                                 plotOutput("Tableau_2")
                                                 ),
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 3",
                                                 plotOutput("Tableau_3")
                                                 ),
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 4",
                                                 plotOutput("Tableau_4")
                                                ),
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 5",
                                                 sliderInput(
                                                     "bins","Parameter",min = 1,max = 200,val
ue = 0, step = 10),
                                                 plotOutput("Tableau_5")
                                        ),
                                      #-------------------------------------------------------
--------------
                                        tabPanel("Exemple 6",
                                                 box(numericInput("Val","Valeur de k:", 20, m
in = 1, max = 20, step = 1),
                                                 sliderInput(
                                                     "bins_2","Parameter",min = 0,max = 50,va
lue = 0, step = 5)),
                                                 box(plotOutput("Tableau_6"))
                                        )
                                      #-------------------------------------------------------
--------------
                                    )),
                                ))
        )
    ),
)


server <- function(input, output) {

    set.seed(955)
    dat <- data.frame(cond = rep(c("A", "B"), each=10), xvar = 1:20 + rnorm(20,sd=3), yvar =
1:20 + rnorm(20,sd=3))
    n <- 20
    x1 <- rnorm(n); x2 <- rnorm(n)
    y1 <- 2 * x1 + rnorm(n)
    y2 <- 3 * x2 + (2 + rnorm(n))
    A <- as.factor(rep(c(1, 2), each = n))
    df <- data.frame(x = c(x1, x2), y = c(y1, y2), A = A)
    fm <- lm(y ~ x + A, data = df)

    output$correlation_plot = renderPlot({
        plot(iris$Sepal.Length, iris[[input$features]], xlab = "Sepal length",
```

```
             ylab = "Feature")

    })

    output$correlation_plot_2 = renderPlot({
        ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
            geom_point() + facet_grid(. ~ Species) +
            stat_smooth(method = "lm") +
            background_grid(major = 'y', minor = "none") +
            panel_border()

    })

    output$carstable = renderDataTable(mtcars)

    output$Tableau_1 = renderPlot({
         dat_1 <- data.frame(cond = rep(c("A", "B"), each=(input$Sample+input$Sample)), xvar
 = 1:input$Sample + rnorm(input$Sample,sd=input$SD), yvar = 1:input$Sample + rnorm(input$Samp
le,sd=input$SD))

         ggplot(dat_1, aes(x=xvar, y=yvar)) + geom_point(shape=2)


    })

    output$Tableau_2 = renderPlot({
        A = ggplot(dat, aes(x=xvar, y=yvar)) +geom_point(shape=2) + geom_smooth(method= lm)
        B = ggplot(dat, aes(x=xvar, y=yvar)) +geom_point(shape=1) +geom_smooth(method = loes
s)
        ggdraw()+ draw_plot(B, 0, 0, 1, .5) +draw_plot(A, 0, .5, 1, .5)
    })

    output$Tableau_3 = renderPlot({

        df <- data.frame(x <- rchisq(1000, 10, 10),
                         y <- rnorm(1000))

        ggplot(df, aes(x, y)) +
            geom_point(alpha = 0.5) +
            geom_density_2d() +
            theme(panel.background = element_rect(fill = '#ffffff'))

    })

    output$Tableau_4 = renderPlot({
        ggplot(data = cbind(df, pred = predict(fm)), aes(x = x, y = y, color = A)) + geom_poi
nt() + geom_line(aes(y = pred))
    })

    output$Tableau_5 = renderPlot({
        dfGamma = data.frame(nu75 = rgamma(100+input$bins, 0.75),nu1 = rgamma(100+input$bins,
1),nu2 = rgamma(100+input$bins, 2)) #Data frame de 3 colonnes chacune de 100 éléments
        dfGamma = stack(dfGamma) #Met bout à bout les 3 colonnes de longueur 100 pour former
 un 1 colonne de 300 éléments
        ggplot(dfGamma, aes(x = values)) + stat_density(aes(group = ind, color = ind),positio
n="identity",geom="line")
        #Affiche  les 3 colonne du data frame original pour afficher 3 courbes gamma associée
s aux 3 colonnes
```

```
    })

    output$Tableau_6 = renderPlot({

        #dchisq = Density
        x_dchisq <- seq(0, 5+input$bins_2, by = 0.1)
        y_dchisq <- dchisq(x_dchisq, df = input$Val)

        dftest = data.frame(y_dchisq1 <- dchisq(x_dchisq, df = 5), y_dchisq2 <- dchisq(x_dchi
sq, df = 6))
        dftest = stack(dftest)

        plot(y_dchisq)

    })

}
# lacement de l'application
shinyApp(ui = ui, server = server)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is instal
led, please make sure the phantomjs executable can be found via the PATH variable.
```

Shiny applications not supported in static R Markdown documents

Evalutation du travail :

1) Effort de présentation : La présentation du dash oard shiny en lui-même est très soignée et présente parfaitement ses différents onglets et diverses graphiques intéractifs.
2) Le knit est réalisable et bien présenté : Une fois toutes les librairies téléchargées et installées, le knit est très facile à réaliser et les interactions du dashboard shiny fonctionnent parfaitement.
3) Explications simples et efficaces : Les explications, bien que très claires pour la création de l’apparence et la structure du dashboard, disparaissent par la suite dès l’utilisation de la fonction **function(input, output)**. De

fait, la mise en place des modèles destinés à être affichés dans le dashboard shiny, bien que très complets et efficaces à l'utilisation, manquent grandement d'explication et peuvent sembler très flou.
4) Le Code reproductible à d'autres DataFrame avec facilité : De par la densité et l'absence d'explication du code se rapportant à l'implémentation des modèles dans le dashboard, il serait complexe de reproduire ce code pour l'appliquer à des cas différents. La première partie concernant la mise en place de la structure du dashboard est quand à elle par contre très compréhensible et facilement repodctible afin de générer diverses structures de dahsboards.
5) Description des fonctions utilsés et du raisonnement : Les fonctions clefs tulisées dans la formation du dashboard ont ici été les fonctions **dashboardPage**, **dashboardSidebar**, **menuItem** etc… Dans le cas du contenu des dashboards en eux mêmes, les fonctions utilisées afin de représenter les données sont par exemple **ggplot** dans la réalisation des graphiques.

Conclusion :

En conclusion, bien que le dashboard soit d'une très grande qualité et très bien réalisé, aussi bien dans son apparence et sa structure que dans osn interactivité, le manque d'explication concernant l'implémentation des modèles en tant que matière dans les dashboard présente un certain obstacle à la compréhension globale du fonctionnement de ce code.