

# Criando uma aplicação para análise de dados em 10 minutos

Wesley Moura

Veja aqui como será nosso produto final:

<https://wesleymoura.shinyapps.io/ClusteringWithShiny>

Para testar, crie um arquivo .CSV com cabeçalho e insira os dados que gostaria de usar como teste, exemplo:

```
idade,salario
20,2500
30,7000
22,2548
25,5247
40,8965
60,5475
80,5424
52,9854
45,4215
49,4217
57,8652
24,3254
20,6524
45,4521
47,5657
65,4587
54,9652
55,5555
21,4255
27,1200
```

Usaremos o Shiny para esta tarefa. Shiny é um framework web para criação de aplicações em R. Ou seja, criaremos nosso software em R e o publicaremos em nossa conta do Shiny.

Eu particularmente estou usando esta ferramenta para criar protótipos interativos de algumas análises, mas você pode usar o Shiny até mesmo para criar seu próprio software de análise de dados e comercializá-lo!

A estrutura de um projeto feito no Shiny é bem simples. Basicamente existem dois arquivos principais no projeto: **ui.R** e **server.R**. Como o próprio nome já diz, usamos o arquivo ui.R para criar os controles de interação com o usuário, como botões e filtros. Já o arquivo server.R é usado para as tarefas de backend, como tratamento de dados e criação de modelos.

Podemos trocar informações entre os arquivos ui.R e server.R por meio de parâmetros. Você também pode criar seus próprios códigos em javascript para customizar algum controle em específico e até integrar com outros frameworks para visualização de dados, como o d3.js.

Consulte mais informações sobre a arquitetura do Shiny, exemplos e tutoriais em <http://shiny.rstudio.com>.

Bom vamos lá, antes de começarmos certifique-se de que já instalou o R e o R Studio em seu computador. Os respectivos links para download são: <https://www.r-project.org> e

<https://www.rstudio.com>. Não custa nada lembrar, R é a linguagem de programação que vamos usar neste tutorial e R Studio é o ambiente integrado de desenvolvimento (em inglês, IDE).

### Do zero ao seu primeiro produto de análise de dados em 10 minutos

Primeiro abra o R Studio e crie um novo projeto Shiny. Tão simples quanto **file -> new Project -> new directory -> Shiny web application** (ou tudo isso em português, se você escolheu idioma português durante a instalação do R Studio).

Note que seu projeto será inicializado com os dois arquivos citados anteriormente.

Ok, no segundo passo vamos criar nossa conta no Shiny. Para isso acesse <http://www.shinyapps.io/>, escolha seu plano gratuito para criar a conta e em seguida faça seu primeiro login. Ao fazer seu primeiro login defina também a URL de sua conta.

Eis o que temos até aqui:

- Projeto Shiny criado no R Studio
- Conta criada no Shiny Apps

Ok. Antes de irmos para o próximo passo, deixe-me contar o que vamos fazer. Basicamente criaremos uma aplicação de *clustering*. O objetivo da nossa aplicação será identificar grupos (ou padrões) no conjunto de dados informado pelo usuário. Nossa aplicação pode ser útil, por exemplo, para identificação de outliers (padrões incomuns) em conjunto de dados com várias dimensões (variáveis).

Se você quer saber mais sobre *clustering* acesse o post em que ensino a implementar algoritmos de cluster do zero em <https://hackinganalytics.com/2016/08/20/implementing-clustering-from-scratch>

Na prática, o usuário da nossa aplicação fará upload de um arquivo que contém as informações para análise e informará a quantidade de clusters que deseja criar. Nossa aplicação processará os dados com o algoritmo K-Means e devolverá a análise de cluster, contendo os dados brutos, um gráfico que mostra os clusters e uma última dela contendo métricas de qualidade dos nossos clusters (na verdade mostraremos apenas uma métrica conhecida como silhueta).

Continuando, vamos instalar o pacote Shiny em nossa máquina e conectá-lo à conta que criamos. Isso nos permitirá fazer deploy da nossa aplicação diretamente pelo R Studio de uma forma bem simples.

Ao fazer login pela primeira vez na sua conta do Shiny você deve ter sido direcionado para uma tela de início contendo algumas instruções iniciais. Pois bem, seguiremos o passo a passo desta tela. Abra o R Studio e siga as etapas abaixo.

#### Passo 1: Instalar Shiny

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

#### Passo 2: Carregar o pacote Shiny

```
library('shiny')
```

### Passo 3: Instalar RS Connect

```
install.packages('rsconnect')
```

### Passo 2: Autorizar conta

```
rsconnect::setAccountInfo(name='sua_conta',  
                           token='seu_token',  
                           secret='<SECRET>')
```

Tudo configurado! Agora vamos criar nossa aplicação.

Para facilitar, vou disponibilizar os fontes dos arquivos ui.R e server.R. O código já está comentado.

#### Arquivo ui.R

```
library(shiny)  
  
# Aqui entra toda a camada de apresentação da aplicação (frontend)  
shinyUI(fluidPage(  
  # Título da aplicação  
  headerPanel("Aplicação de clustering"),  
  
  # Botão para escolher arquivo para análise  
  fileInput('myFile', 'Choose CSV File',  
            accept=c('text/csv',  
                     'text/comma-separated-values,text/plain',  
                     '.csv'))  
),  
  
  # Sidebar com um slider input para escolher o número de clusters  
  sidebarPanel(  
    sliderInput("k",  
               "Number of clusters:",  
               min = 1,  
               max = 10,  
               value = 3)  
),  
  
  # Monta as tabs e insere os objetos dentro de cada uma  
  mainPanel(  
    tabsetPanel(  
      tabPanel("Dataset", tableOutput("contents")),  
      tabPanel("Cluster Plot", plotOutput("myPlot")),  
      tabPanel("Silhueta", tableOutput("mySilhouette"))  
    )  
  )  
)  
)
```

#### Arquivo server.R

```

library(shiny)
library(cluster)

# Aqui entra toda a lógica da aplicação (backend)
shinyServer(function(input, output) {

# Monta um dataset com os dados do arquivo escolhido pelo usuário
myData <- reactive({
  inFile <- input$myFile
  if (is.null(inFile)) return(NULL)
  data <- read.csv(inFile$datapath, header = TRUE)
  data
})

# Processa o algoritmo K-Means para encontrar clusters nos dados
# Note que estamos usando o parâmetro de qtde de clusters definido pelo user
myCluster <- reactive({
  if (is.null(myData())) return(NULL)
  # generate clusters based on input$k from ui.R
  kmeans(myData(), input$k, iter.max = 10, nstart = 1)
})

# Calcula a silhueta de cada elemento
mySilhouette <- reactive({
  if (is.null(myData())) return(NULL)

  dissE <- daisy(myData())
  sk <- silhouette(myCluster()$cluster, dissE)

  df <- data.frame(sk[,c(1,2,3)])
  df
})

# Criar um elemento chamado output$contents com os dados do arquivo
# Note que este objeto (assim como os objetos criados a seguir) é usado no
# arquivo ui.R
output$contents <- renderTable({
  myData()
})

# Criar um elemento chamado output$myPlot com o gráfico dos clusters
output$myPlot <- renderPlot({
  if (is.null(myData())) return(NULL)
  clusplot(myData(), myCluster()$cluster, color=TRUE, shade=TRUE, labels=2,
lines=0, main="Identificação dos grupos")
})

# Criar um elemento chamado output$mySilhouette com os dados da silhueta
output$mySilhouette <- renderTable({
  mySilhouette()
})

})

```

Ok pessoal. Agora no último passo vamos rodar nossa aplicação local para ver se está funcionando e em seguida publicá-la em nossa conta do Shiny.

Para rodar a aplicação na sua máquina local digite:

```
runApp()
```

Se tudo der certo sua aplicação deve ser carregada no seu navegador padrão.

Agora vamos fazer o deploy da aplicação com o código abaixo:

```
library(rsconnect)
rsconnect::deployApp()
```

Se tudo der certo acesse sua conta do Shiny e clique em applications -> all -> clique no nome da sua aplicação -> URL

Sucesso! Sua análise está na web! Não deixe de explorar todos os controles de aplicação existentes na sua conta. Você definitivamente deveria considerar o Shiny como uma ferramenta para tirar do papel aquele seu projeto para criar um produto de análise de dados.