

UNIVERSIDADE FEDERAL DE PELOTAS  
DEPARTAMENTO DE ECONOMIA  
CURSO DE CIÊNCIAS ECONÔMICAS

## **SÉRIE CADERNOS ECONÔMICOS**

### **Gráficos dinâmicos no R: Um exemplo com *shiny* e *dygraphs***

*Texto didático n. 10*

Autor: Angelo Salton

PELOTAS

Junho 2016

# Gráficos dinâmicos no R: Um exemplo com *shiny* e *dygraphs*

Angelo Salton<sup>1</sup>

5 de junho de 2016

## Introdução

Este é um exercício de construção de gráficos dinâmicos no software estatístico R. Este tipo de visualização pode ser útil especialmente em apresentações com múltiplas séries de tempo distribuídas sobre os mesmos eixos. *Shiny* é um pacote voltado para a construção de aplicações para web com o R. *Dygraphs* é um pacote que permite a construção de gráficos interativos. Trabalhar com o *RStudio* tornará o exercício mais fácil, através da sua interface.

## Método

Inicialmente, vamos carregar os pacotes necessários. Pode ser necessário uma instalação prévia, através do comando:

```
install.packages(c("dygraphs", "shiny", "xts")):
```

```
library(dygraphs)
library(shiny)
library(xts)
```

Para este exemplo, vamos carregar uma base de dados (presente na instalação padrão do R) que relaciona a obrigatoriedade do uso do cinto de segurança com acidentes de trânsito fatais na Grã-Bretanha. Em seguida, farei algumas transformações para aproximar as grandezas entre as séries. Por fim, vamos visualizar as primeiras observações com a função `head()`:

```
data("Seatbelts")
Seatbelts <- as.xts(Seatbelts)
Seatbelts <- Seatbelts[,-c(2,7)]
Seatbelts$kms <- Seatbelts$kms/10
Seatbelts$PetrolPrice <- Seatbelts$PetrolPrice*1000
head(Seatbelts)
```

##		DriversKilled	front	rear	kms	PetrolPrice	law
##	jan 1969	107	867	269	905.9	102.9718	0
##	fev 1969	97	825	265	768.5	102.3630	0
##	mar 1969	102	806	319	996.3	102.0625	0
##	abr 1969	87	814	407	1095.5	100.8733	0
##	mai 1969	119	991	454	1182.3	101.0197	0
##	jun 1969	106	945	427	1239.1	100.5812	0

---

<sup>1</sup> Mestre em Economia Aplicada pelo Programa de Pós-Graduação em Organizações e Mercados (PPGOM) da Universidade Federal de Pelotas. E-mail: [gsalton4@hotmail.com](mailto:gsalton4@hotmail.com).

As aplicações do shiny contém dois objetos fundamentais: **ui** e **server**, criadas através das funções `shinyUI()` e `shinyServer()`. A primeira parte do código nos permite determinar como será o layout da interface do usuário. Podemos adicionar elementos como caixas de seleção, barras deslizantes e até mesmo espaços para entrada de dados. Mais adiante veremos que é possível usar esses elementos para manipular nossa visualização:

```
ui <- shinyUI(fluidPage(  
  
  # O título  
  titlePanel("Acidentes de Trânsito: Grã-Bretanha, 1969-84"),  
  
  # Define uma barra lateral com opções  
  sidebarLayout(  
    sidebarPanel(  
      checkboxInput("opt1", label = "Mortes no trânsito", value = TRUE),  
      checkboxInput("opt2", label = "Colisões frontais e traseiras", value = TRUE)  
    ),  
    checkboxInput("opt3", label = "Preço dos combustíveis e km rodados", value =  
TRUE),  
    textOutput("legends")  
  ),  
  
  # Reserva espaço para o gráfico  
  mainPanel(  
    dygraphOutput("dygraph", width = "auto")  
  )  
))
```

A próxima parte do código diz respeito à parte do servidor, ou seja, a parte responsável por gerar o gráfico a partir dos dados, além de determinar como os elementos da interface se relacionam com a visualização. A função `dyVisibility()` determina quais séries serão exibidas no gráfico, de acordo com a ordem das variáveis na base. Temos que implementá-la aqui pois ainda não está disponível na versão atual do pacote `dygraphs`. Ela espera um vetor lógico como argumento, assim vamos definir uma função auxiliar `visible()` para exibir (ou ocultar) grupos de séries. `visible()`, por sua vez, recebe como argumentos os estados das três caixas de seleção definidas anteriormente: *verdadeiro* quando marcadas e *falso* caso contrário.

A função `renderDygraph()` recebe os dados e opções de visualização e gera o gráfico dinâmico para a aplicação Shiny. O pacote `dygraphs` possui funções implementadas que permitem adicionar opções de interatividade para os gráficos, como por exemplo a função `dyShading()`, que permite sombrear uma área do gráfico relativa à determinado período. Aqui, uso ela para demarcar o período em que foi determinada a obrigatoriedade do uso do cinto de segurança em vias britânicas.

```
server <- shinyServer(function(input, output){  
  
  dyVisibility <- function(dygraph, visibility = TRUE){  
    dygraph$x$attrs$visibility <- visibility  
  }
```

```

    dygraph
  }

  visible <- function(x,y,z){
    vis <- logical(ncol(Seatbelts))
    ifelse(x==TRUE, vis[1] <- TRUE, vis[1] <- FALSE) # opt1
    ifelse(y==TRUE, vis[c(2,3)] <- TRUE, vis[c(2,3)] <- FALSE) # opt2
    ifelse(z==TRUE, vis[c(4,5)] <- TRUE, vis[c(4,5)] <- FALSE) # opt3
    return(vis)
  }

  output$dygraph <- renderDygraph({
    dygraph(Seatbelts[,-6], xlab = "Tempo", ylab = "", main = "") %>%
    dyRangeSelector(height = 20) %>%
    dyHighlight(highlightCircleSize = 0, hideOnMouseOut = T,
               highlightSeriesOpts = list(strokeWidth=2),
               highlightSeriesBackgroundAlpha = 0.4) %>%
    dyVisibility(visible(input$opt1, input$opt2, input$opt3)) %>%
    dyShading(from = "1983-2-1", to = "1984-12-1") %>%
    dyLegend(show = "always", labelsDiv = "legends")
  })
})

```

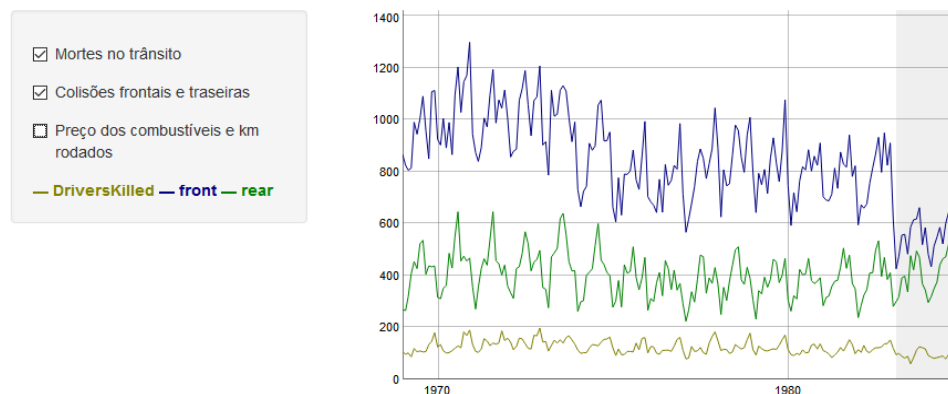
Com tudo definido, vamos executar a aplicação.

```
shinyApp(ui = ui, server = server)
```

## Exemplo

Se tudo der certo, uma página da web irá se abrir com o gráfico e os controles interativos.

### Acidentes de Trânsito: Grã-Bretanha, 1969-84



Neste exemplo, é possível verificar, de maneira preliminar, uma queda nas estatísticas de colisões frontais após a promulgação da lei que obriga o uso do cinto de segurança.