

# Visualizando Insights: Construção de um dashboard Interativo para Análise de Dados

Breno Fernando Guerra Marrão A97768

Tales André Rovaris Machado A96314

Tiago Passos Rodrigues A96414

Orientação: Cecília Castro, Departamento de Matemática

As técnicas de visualização de dados são parte integrante das áreas de Business Intelligence. A apresentação dos resultados de análises de dados através de gráficos é uma maneira bastante persuasiva e impactante na comunicação do conteúdo dos dados, dando suporte à decisão, de forma rápida e esclarecedora.

O programa R dispõe de **packages** que facilitam a construção de gráficos de grande qualidade, dinâmicos ou estáticos e dispõe, ainda, de software que permite a construção de **dashboards** interativos.

## 1

Passos para a construção de um **dashboard**:

1. aquisição dos dados
2. análises estatísticas e escolha de gráficos e tabelas a incorporar no **dashboard** opções

### 1.1

O conjunto de dados “lung” do **package** R **survival** contém dados de sobrevivência para pacientes com cancro no pulmão em estado avançado que participaram num ensaio clínico de um novo tratamento.

inst: código da instituição

time: Tempo de sobrevivência em dias

status: status de censura 1 = censurado, 2 = morto (usualmente a codificação é 0 para censurado e 1 para morte) :: Modificar

age: idade em anos

sex: Male=1 Female=2

ph.ecog: Pontuação de desempenho ECOG conforme avaliado pelo médico. 0 = assintomático, 1 = sintomático, mas completamente deambulatório, 2 = acamado <50% do dia, 3 = acamado > 50% do dia, mas não acamado, 4 = acamado

ph.karno: pontuação de desempenho de Karnofsky (mau = 0-bom = 100) avaliado pelo médico

pat.karno: Pontuação de desempenho de Karnofsky conforme avaliado pelo paciente

meal.cal: Calorias consumidas nas refeições

wt.loss: Perda de peso nos últimos seis meses (libras) :: Modificar para kg

### 1.1.1

A pontuação Karnofsky Performance Score (KPS) fornece uma medida do estado funcional de um paciente ou capacidade de realizar atividades da vida diária (autocuidado, mobilidade e capacidade de trabalho) sendo uma forma de avaliar o status de desempenho de pacientes com cancro.

A classificação do KPS é de 0-100. Uma pontuação 100 indica funcionamento normal e uma pontuação 0 indica morte.

Esta classificação pode ser usada para ajudar a orientar as decisões de tratamento, pois os pacientes com pontuações KPS mais baixas podem ser menos capazes de tolerar certos tratamentos.

## 2

Perguntas de investigação relacionadas com este conjunto de dados (modelos de sobrevivência e gráficos comparativos das funções de sobrevivência para grupos diferentes de indivíduos):

I Qual é a distribuição do tempo de sobrevivência dos pacientes com cancro do pulmão e que fatores influenciam essa distribuição?

1. A idade afeta significativamente o tempo de sobrevivência em pacientes com cancro de pulmão? E após o controle de outros fatores relevantes, como sexo, pontuação de desempenho ECOG e pontuação de desempenho de Karnofsky?
2. Qual é o efeito da perda de peso nos últimos seis meses no tempo de sobrevivência em pacientes com cancro de pulmão?
3. A pontuação de desempenho de Karnofsky, avaliada pelo médico, prediz o tempo de sobrevivência em pacientes com cancro do pulmão?
4. Existe uma diferença significativa no tempo de sobrevivência entre homens e mulheres com cancro do pulmão? E após o controle de outras covariáveis, como idade, classificação ECOG ou pontuação de Karnofsky?
5. O consumo de calorias nas refeições afeta o tempo de sobrevivência em pacientes com cancro de pulmão?
6. Estimar o tempo de sobrevivência em pacientes com cancro do pulmão com base na sua idade, sexo, classificação ECOG e pontuação de Karnofsky.

Exemplo de análise para questão 6.

```
# carregar os packages survival e survminer
```

```

library(survival)

library(survminer)

# Ler os dados

lung <- lung

# Ajustar um modelo de Cox proportional hazards com age, sex,
#classificação ECOG , e pontuação Karnofsky como preditores

fit <- coxph(Surv(time, status) ~ age + sex + ph.ecog + ph.karno, data = lung)

#####

#A função coxph ajusta um modelo de riscos proporcionais de Cox aos dados,
#que estima a taxa de
# risco para cada variável preditora.
#A taxa de risco representa o risco relativo de um evento (por exemplo, morte)
# para pacientes # com um aumento de uma unidade
#na variável preditora, controlando as outras variáveis no modelo.
#A saída dessa função é um objeto da classe "coxph",
# que pode ser usada para fazer previsões,
#calcular curvas de sobrevivência e realizar outras análises.

#####

# survfit function para definir um objeto da classe "survfitcox" "survfit"

surv_object <- survfit(fit)

```

```
# Plot the survival curve using the ggsurvplot function
```

```
ggsurvplot(surv_object, data = lung, risk.table = TRUE, xlab = "Time (days)",  
ylab = "Survival Probability",  
ggtheme = theme_bw())
```

7. Existe diferença significativa no tempo de sobrevivência entre pacientes com diferentes classificações de desempenho do ECOG?
8. Comparar a pontuação de desempenho de Karnofsky, avaliada pelo paciente, com a classificação do médico.
9. Avaliar o impacto dos fatores institucionais, conforme indicado pelo código da instituição, no tempo de sobrevivência em pacientes com cancro do pulmão.
10. Existe relação entre a pontuação de desempenho de Karnofsky e o número de calorias consumidas nas refeições?

Exemplo de gráfico e indicador para 10.

```
# carregar survival package e ler os dados  
library(survival)  
  
lung <- lung  
  
# fazer um scatter plot de Karnofsky vs. calorias da refeição  
#####  
  
library(highcharter)  
  
highchart()%>%  
  hc_add_series(lung, "scatter", hcaes(x = meal.cal, y = ph.karno))
```

```

l <- na.omit(lung)

# calcular o coeficiente de correlação entre as duas
#variáveis depois de omitir os casos com observações NA (omissas)

correlation <- cor(l$meal.cal, l$ph.karno)
cat("Correlation coefficient:", round(correlation, 2))

#####

```

### 3 Desenho do dashboard

Escolher a melhor forma de visualização dos dados. Inclui decidir que gráficos e modelos usar e como organizá-los para obter o máximo impacto.

### 4 Desenvolvimento

Grande variedade de opções.

Exemplo Shiny App

```

library(shiny)
library(ggplot2)
library(dplyr)
library(survival)

# Import lung data
lung_data <- lung

```

```

# Create UI

ui <- fluidPage(

  titlePanel("Lung Data Dashboard"),

  sidebarLayout(

    sidebarPanel(

      selectInput("age", "Select Age Group:",
        choices = c("All", unique(lung_data$age)))

    ),

    mainPanel(

      plotOutput("lung_plot")

    )

  )

)

# Create Server

server <- function(input, output) {

  # Filter data based on age group selection

  lung_data_filtered <- reactive({

    if(input$age == "All"){

      return(lung_data)

    } else {

      return(filter(lung_data, age == input$age))

    }

  })

```

```
# Create lung data plot

output$lung_plot <- renderPlot({

  ggplot(lung_data_filtered(), aes(x = meal.cal, y = wt.loss)) +

    geom_point() +

    labs(title = "Lung Data Plot")

})

}

# Run Shiny App

shinyApp(ui, server)
```

## 5 Teste

Após o desenvolvimento do **dashboard**, este deve ser testado para garantir que está a funcionar corretamente. Tal inclui verificar os elementos visuais e garantir que os dados estão a ser representados com precisão.

Manter a integridade gráfica: A integridade gráfica refere-se basicamente a manter a verdade sobre os dados. Significa ser objetivo sobre os valores e não fazê-los parecer de uma certa maneira que beneficie a análise.

## 6 Implementação

Depois do **dashboard** ter sido testado e quando estiver pronto para ser utilizado, poderá ser divulgado. Isso pode envolver hospedar o painel num site (ver RStudio).



# Aspetos a considerar no desenvolvimento da aplicação

1. Conte uma história com os dados
2. Forneça contexto
3. Tente fornecer o máximo de informação, mesmo que esta pareça óbvia
4. Nos gráficos, identifique conveniente os eixos e unidades de medida. Forneça valores de comparação. A regra geral aqui é usar as comparações mais comuns, por exemplo, comparação com uma meta definida, com um período anterior ou com um valor projetado.
5. Não tente colocar todas as informações na mesma página
6. Selecione o tipo certo de **dashboard**
7. Use o tipo certo de gráfico
8. Dê prioridade à simplicidade
9. Tenha cuidado com as cores - escolha apenas algumas cores
10. Seja consistente com os rótulos e a formatação dos dados
11. Use elementos interativos

## 7 Documentação

Understanding ShinyApps:

<https://www.r-bloggers.com/2022/10/understanding-shinyapps/>

<https://dplyr.tidyverse.org/>

<https://ggplot2.tidyverse.org/>

<https://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html>

<https://plotly.com/r/>

<https://rmarkdown.rstudio.com/>

<https://pkgs.rstudio.com/flexdashboard>

BOM TRABALHO!