

Visualizar e armazenar os dados

Daniel C. Mota e Kleber G. Abitante

Salvar uma cópia da área de trabalho

- Para salvar todos os objetos do ambiente do R (imagem do ambiente), pode ser utilizada a função `save.image()`:
`save.image("area_trabalho.RData")`

Carregar uma cópia da área de trabalho

- Para carregar um arquivo RData (com todos os objetos do ambiente do R), pode ser utilizada a função `load()`:
`load("area_trabalho.RData")`
- **Atenção:** ao carregar um arquivo RData, se já houver variáveis no ambiente com o mesmo nome das variáveis do arquivo RData carregado, as variáveis do arquivo irão sobrescrever as variáveis que estão no ambiente.

Exercício 1 (2 min.)

- a. Carregue na memória do R o arquivo “pib_ipca.csv” usando a função `read.csv2`.
- b. Carregue na memória do R o arquivo “res_prim.xlsx” usando a função `read_excel` do pacote `readxl`.
- c. Salve uma imagem da área de trabalho em um arquivo RData usando a função `save.image()`.
- d. Limpe a área de trabalho usando o comando `rm(list=ls())`.
- e. Carregue a imagem da área de trabalho usando a função `load()`.

Salvar apenas uma variável da área de trabalho

- Use a função `saveRDS()` para salvar apenas uma variável do ambiente:
`saveRDS(x, "arquivo.RDS")`

Carregar um arquivo RDS

- Use a função `readRDS()` para ler um arquivo RDS e atribuir o resultado a uma variável:

```
w <- readRDS("arquivo.RDS")
```

Exercício 2 (2 min.)

- a. Salve um arquivo RDS contendo apenas a tabela de PIB e IPCA do exercício anterior, usando a função `saveRDS()`.
- b. Apague o conteúdo do ambiente do R usando as funções `rm(list=ls())`.
- c. Carregue o arquivo RDS salvo anteriormente em uma variável com o nome diferente da variável que foi salva.

- Existem alguns *packages* para exportar dados para o Excel.
- O *package* `openxlsx` é um dos mais versáteis.
- Ele permite que sejam exportadas várias variáveis para o mesmo arquivo, podendo ser uma em cada aba do arquivo ou na mesma aba.
- Também é possível adicionar conteúdo em um arquivo Excel já existente.

Exportar para Excel - exportar uma única variável para um arquivo novo

```
library(jsonlite)
library(openxlsx)
url1 <-
"https://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados/ultimos/10?formato=json"
serie_433 <- fromJSON(url1)
serie_433$valor <- as.numeric(serie_433$valor)
# exportar um unico objeto para um arquivo Excel novo
write.xlsx(serie_433, "serie_433.xlsx")
```

Exportar para Excel - exportar múltiplas variáveis para um arquivo novo

```
# url do INPC
url2 <-
"https://api.bcb.gov.br/dados/serie/bcdata.sgs.188/dados/ultimos/10?formato=json"
serie_188 <- fromJSON(url2)
serie_188$valor <- as.numeric(serie_188$valor)
# criar um workbook (equivalente a um arquivo Excel)
wb <- createWorkbook()
# adicionar uma aba para o IPCA
addWorksheet(wb, "IPCA")
# adicionar conteudo na aba do IPCA
writeData(wb, sheet = "IPCA", x = serie_433)
# adicionar uma aba para o INPC
addWorksheet(wb, "INPC")
# adicionar conteudo na aba do INPC
writeData(wb, sheet = "INPC", x = serie_188)
# salvar o arquivo Excel
saveWorkbook(wb, "indice_precos.xlsx")
```

Exportar para Excel - exportar uma variável para arquivo existente

```
# url do INCC
url3 <-
"https://api.bcb.gov.br/dados/serie/bcdata.sgs.192/dados/ultimos/10?formato=json"
serie_192 <- fromJSON(url3)
serie_192$valor <- as.numeric(serie_192$valor)
# carregar um arquivo ja existente
wb <- loadWorkbook("indice_precos.xlsx")
# adicionar uma aba para o IPCA
addWorksheet(wb, "INCC")
# adicionar conteudo na aba do INCC
writeData(wb, sheet = "INCC", x = serie_192)
# criar o arquivo Excel
saveWorkbook(wb, "indice_precos.xlsx", overwrite = TRUE)
```

Exercício 3 (3 min.)

- a. Salve as duas tabelas carregadas no exercício 1 (“pib_ipca.csv” e “res_prim.xlsx”) em um único arquivo Excel (em abas diferentes) usando o *package* `openxlsx`.

- Para exportar para CSV, sugiro utilizar a função `write_csv2()` do *package* `readr`.

```
write_csv2(serie_192, "serie_192.csv")
```

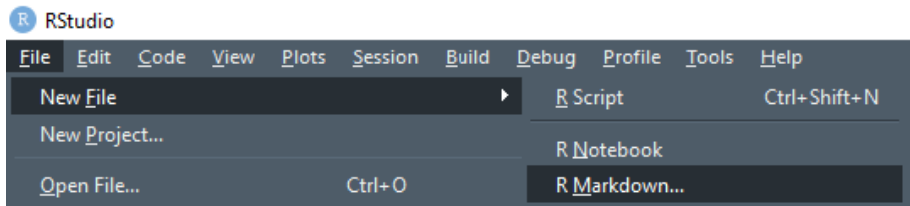
Exercício 4 (1 min.)

- a. Exporte a tabela gerada pela carga do arquivo “pib_ipca.csv” para um arquivo CSV usando a função `write_csv2()` do *package* `readr`.

- O R Markdown é um formato de arquivo para construir documentos dinâmicos com o R¹

¹GROLEMUND, G. Introduction to R Markdown. 2014. Disponível em: https://rmarkdown.rstudio.com/articles_intro.html. Acesso em: 13 mar. 2023

Relatório em R Markdown



Relatório em R Markdown

```
---
title: "Rotinas de R para acompanhamento da conjuntura econômica"
author: "Kleber"
date: "2024-07-09"
output: word_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

← Tipo de documento

← Chunk

← Título 2

Relatório em R Markdown - *chunk*

- a. O *chunk* é delimitado por três crases no começo e no início dele. Entre chaves, deve ser colocada a letra `r` seguido do nome do *chunk* (que não pode se repetir ao longo do documento) e a customização do *chunk* por meio de *options*:

```
```{r nomeDoChunk, options(facultativo)}  
```
```

- b. As principais *options* disponíveis são:
- `include = FALSE`: evita que o código e os resultados apareçam no arquivo final;
 - `echo = FALSE`: evita apenas que o código apareça no arquivo final. Os resultados irão aparecer;
 - `message = FALSE`: evita que mensagens que são geradas pelo código apareçam no arquivo final;
 - `warning = FALSE`: evita que warnings que são gerados pelo código apareçam no arquivo final; e
 - `fig.cap = "..."`: adiciona um título para um gráfico.

- 8. Você tem duas alternativas para executar e carregar as suas variáveis no R Markdown:
 - Criar as variáveis dentro de um ou mais *chunks* (dentro do R Markdown);
 - Criar as suas variáveis fora do R Markdown, salvá-las em um arquivo .RData e carregar esse arquivo no R Markdown.

Relatório em R Markdown - variáveis criadas no *chunk*

- No primeiro *chunk*:

```
{r tabela, echo=F, message = FALSE}
# definir que o separador de decimal é a vírgula
options(OutDec= ",")
library(jsonlite)
library(flextable)
library(ggplot2)
# url da serie 433
url433 <- "https://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados/ultimos/12?formato=json"
# baixar os dados
ipca <- fromJSON(url433)
# transformar a coluna do saldo para numero
ipca$valor <- as.numeric(ipca$valor)
# criar coluna de data
ipca$data <- as.Date(ipca$data, "%d/%m/%Y")
# ordenar por data
ipca <- ipca[order(ipca$data),]
# formatar a data
ipca$data <- format(ipca$data, "%m/%Y")
# transformar em fator
ipca$data <- factor(ipca$data, levels=ipca$data)
# gerar o grafico da serie
g <- ggplot(ipca, aes(data, valor)) +
  geom_line(color="blue", group=1) +
  labs(x = NULL, y = "%", title="IPCA (%)") +
  scale_x_discrete(breaks = ipca$data[c(1,5,9,12)])
# renomear as colunas
colnames(ipca) <- c("Mês","IPCA (%)")
# gerar a tabela do IPCA
df <- regulartable(ipca, cwidth=1)
# alterar o separador de decimal da tabela para vírgula e o separador de milhar para ponto
colformat_double(df, decimal.mark = ",", big.mark = ".")
```

Relatório em R Markdown - variáveis criadas no *chunk*

- No segundo *chunk*: `{r tabela, echo=F, message = FALSE}`
`print(g)`

Relatório em R Markdown - variáveis criadas em outro *script*

- Em um *script* fora do RMarkdown:

```
library(jsonlite)
library(ggplot2)
# url da serie 433
url433 <- "https://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados/ultimos/12?formato=json"
# baixar os dados
ipca <- fromJSON(url433)
# transformar a coluna do saldo para numero
ipca$valor <- as.numeric(ipca$valor)
# criar coluna de data
ipca$data <- as.Date(ipca$data, "%d/%m/%Y")
# ordenar por data
ipca <- ipca[order(ipca$data),]
# formatar a data
ipca$data <- format(ipca$data, "%m/%Y")
# transformar em fator
ipca$data <- factor(ipca$data, levels=ipca$data)
# gerar um grafico
g <- ggplot(ipca, aes(data, valor)) +
  geom_line(color="blue", group=1) +
  labs(x = NULL, y = "%", title="IPCA (%)") +
  scale_x_discrete(breaks = ipca$data[c(1,5,9,12)])
```

Relatório em R Markdown - variáveis criadas em outro *script*

- No primeiro *chunk*:

```
{r tabela, echo=F, message = FALSE}
setwd("C:/Users/Kleber/Documents/Curso_R/base_dados")
options(OutDec= ",")
library(flextable)
load("area_trabalho_ipca.RData")
colnames(ipca) <- c("Mês","IPCA (%)")
df <- regulartable(ipca, cwidth=1)
# alterar o separador de decimal da tabela para vírgula e o separador de
milhar para ponto
colformat_double(df, decimal.mark = ",", big.mark = ".")
```

- No segundo *chunk*:

```
{r grafico, echo=F, fig.width = 7}
print(g)
```

Relatório em R Markdown - algumas convenções

- a. #: Título 1
- b. ##: Título 2
- c. ###: Título 3
- d. **italico**: itálico
- e. ****negrito****: negrito
- f. quebra de linha: 2 espaços e ENTER.
- g. 1. item
- h. + subitem (2 tabs de recuo)
- i. inserir figuras: ``
`{width="1000"}`

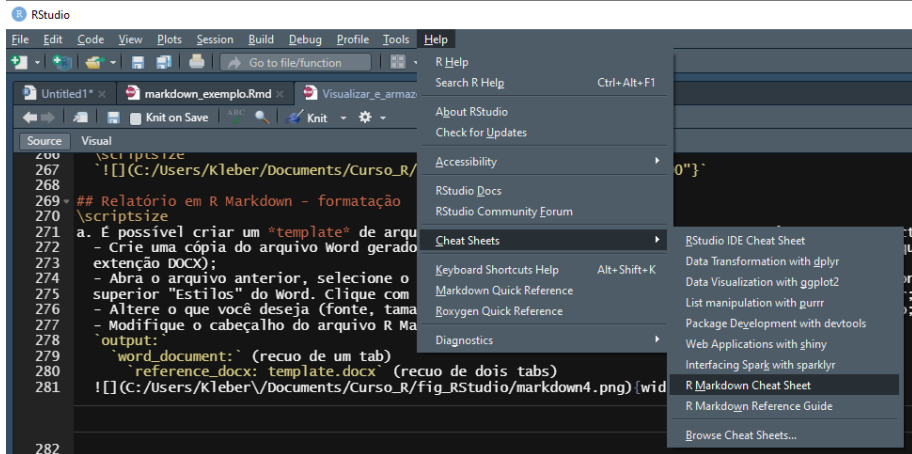
É possível incluir variáveis geradas no *chunk* no meio do texto usando o código “r variável” envolto em crases.

Relatório em R Markdown - formatação

- É possível criar um *template* de arquivo Word para que o R Markdown gere um arquivo com as características do *template*:
- Crie uma cópia do arquivo Word gerado pelo R e renomeie ele para “template.docx” (pode ser qualquer nome com extensão DOCX);
- Abra o arquivo anterior, selecione o que você deseja alterar e procure o estilo que está selecionado na barra superior “Estilos” do Word. Clique com o botão direito no estilo selecionado e clique em Modificar;
- Altere o que você deseja (fonte, tamanho da fonte, cor da fonte, ...), salve e feche o documento; e
- Modifique o cabeçalho do arquivo R Markdown:
output:
word_document: (reco de um tab)
reference_docx: template.docx (reco de dois tabs)

```
output:  
word_document:  
reference_docx: template.docx
```

Relatório em R Markdown - mais informações



Exercício 5 (3 min.)

- a. Crie um arquivo R Markdown no formato Word (DOCX).
- b. Carregue para memória do R, dentro de um *chunk*, o arquivo “pib_ipca.csv” que está na pasta Material.
- c. Crie uma tabela mostrando todos os valores do arquivo.
- d. Gere o relatório.

Dica: use o argumento `j` da função `colformat_double` para informar os números das colunas que você deseja que tenham vírgula como separador de decimal e atribua o resultado para um novo objeto. Em seguida, use o argumento `j` da função `colformat_num` para informar que você não deseja separador de decimal (`big.mark = ""`) na coluna de ano.

- O relatório feito em PDF no R Markdown permite que sejam utilizados comandos em LaTeX na elaboração do documento.
- O LaTeX é um sistema de criação de fontes (tipografia) de alta qualidade. Inclui funcionalidades destinadas à produção de documentação técnica e científica.²
- O LaTeX é um software livre e muitos outros software utilizam ele como uma funcionalidade de seus softwares.

²THE LATEX PROJECT. LaTeX – A document preparation system. 2023. Disponível em: <https://www.latex-project.org/>. Acesso em: 15 mar. 2023

- a. Com LaTeX, é possível realizar formatação diretamente no arquivo:

- Alterar a fonte:

```
output:  
  pdf_document:  
    latex_engine: xelatex  
  mainfont: Arial
```

- Alterar o tamanho da fonte:

(<https://latex-tutorial.com/changing-font-size/>) ou

```
\fontsize{12}{14.4}
```

```
\selectfont
```

onde `\fontsize{tamanho da fonte}{espaçamento entre linhas}`

Regra de bolso para o espaçamento entre linhas: $1,2 \times$ o tamanho da fonte.

- Cor da fonte:

```
\textcolor{blue}{Isto é um texto em azul}
```

- É possível escrever equações em LaTeX:

```
\begin{equation}
```

```
x_{t} = \beta_{0} + \beta_{1}y_{t} + \epsilon_{t}
```

```
\end{equation}
```

$$x_t = \beta_0 + \beta_1 y_t + \epsilon_t \quad (1)$$

```
\begin{equation}
```

```
var\_pib\_real_{t} = \frac{pib\_nominal_{t}}{pib\_nominal_{t-1}} \times
```

```
\frac{1}{1+deflator_{t}}
```

```
\end{equation}
```

$$var_pib_real_t = \frac{pib_nominal_t}{pib_nominal_{t-1}} \times \frac{1}{1 + deflator_t} \quad (2)$$

(https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols)

- Para gerar tabelas em PDF, pode ser utilizada função `kable()` do *package* `knitr`:

```
{r cars, echo=F}
library(jsonlite)
# url da serie do IPCA
urlIPCA <-
"https://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados/ultimos/10?formato=js
# baixar a serie
ipca <- fromJSON(urlIPCA)
# transformar a coluna de valor para numerico
ipca$valor <- as.numeric(ipca$valor)
# gerar tabela
knitr::kable(ipca, col.names = c("Data", "IPCA (%))")
```


Exercício 6 (3 min.)

- a. Crie um arquivo R Markdown no formato PDF.
- b. Carregue para a memória do R, dentro de um *chunk*, o arquivo “pib_ipca.csv”.
- c. Crie uma tabelas das séries, usando a função `kable` do package `knitr`.

- Este método permite gerar uma apresentação em Powerpoint.

Apresentação em R Markdown - Powerpoint

- a. Para utilizar um modelo próprio:
 - 1 Abrir o modelo pretendido, vá no menu Página Inicial e clique em Layout;
 - 2 Verifique se o modelo possui pelo menos os seguintes *layouts*:
 - Slide de Título ("Title Slide");
 - Cabeçalho de Seção ("Section Header");
 - Título e Conteúdo ("Title and Content");
 - Duas Partes de Conteúdo ("Two Content");
 - Comparação ("Comparison");
 - Conteúdo com Legenda ("Content with Caption"); e
 - Branco ("Blank").
 - 3 Se não possuir, clique em Exibir (menu superior) e depois em Slide Mestre. Insira os *layouts* que faltam. Isso pode ser feito da seguinte forma:
 - crie uma Apresentação em Branco e vá em Exibir, Slide Mestre;
 - copie os *layouts* que estão faltando para o slide mestre do seu template.
 - 4 Renomeie os *layouts* para os nomes em inglês que constam no tópico 2 (vá em Exibir Slide Mestre e clique com o botão direito em cada *layout* e escolha renomear);
 - 5 Salve o modelo como Modelo do Powerpoint (.POTX) (template_pwrt.potx); e
 - 6 Altere o cabeçalho do arquivo R Markdown:

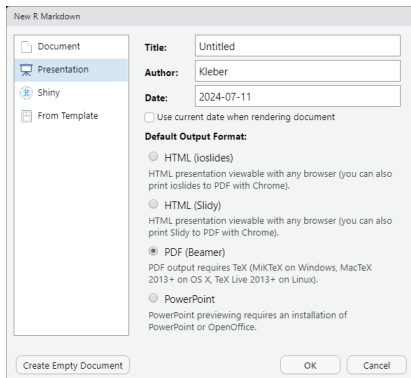
```
output:
  powerpoint_presentation:
    reference_doc: template_pwrt.potx
---
```

Exercício 7 (3 min.)

- a. Gere um arquivo RMarkdown no formato de apresentação do Powerpoint, usando o template padrão dele.
- b. Carregue para a memória do R, dentro de um *chunk*, o arquivo “pib_ipca.csv”, crie um gráfico de linha do PIB e gere o arquivo da apresentação.

Apresentação em R Markdown - Beamer

- Para gerar a apresentação no R Markdown com formato de saída “Beamer”, é necessário instalar o aplicativo “MikTeX”, mas este aplicativo está com problema de funcionamento no banco e não tem perspectiva de conserto.



- O modelo padrão gera um PDF na forma de slides, com fundo em branco.
- Mas é possível utilizar os seguintes modelos de *template*:
(https://deic.uab.cat/~iblanes/beamer_gallery/index_by_theme.html)
- Esta apresentação foi feita com o *template* “Madrid”, com customização do rodapé:

```
output:  
  beamer_presentation:  
    theme: "Madrid"
```

- Uma forma interessante de visualizar dados de forma interativa é por meio do pacote `shiny` (<https://shiny.posit.co/>), o qual cria um painel interativo;
- Aqui no banco, é possível criar um painel usando o `shiny` e colocá-lo para produção na nuvem do banco, por intermédio do Deinf.

Obrigado!

