

# Comunicando seus resultados:

Criando apresentações com R

Beatriz Milz e Haydee Svab - R-Ladies São Paulo

20 de maio de 2019

(Atualizado em: 13/04/2019)

# Instrutoras

## Beatriz Milz



- Doutoranda em Ciência Ambiental no PROCAM/IEE/USP.
- Co-organizadora do R-Ladies São Paulo.
- Assessora Editorial na Revista Ambiente e Sociedade
- Bacharel em Gestão Ambiental (EACH/USP).
- Mestre em Ciências no PPG-Análise Ambiental Integrada (UNIFESP/Diadema).
- Email: beatriz.milz@hotmail.com
- Página: <https://beatrizmilz.github.io/>
- Github: @beatrizmilz
- Twitter: @BeaMilz

# Instrutoras

## Haydee Svab



- É pesquisadora em mobilidade urbana e ativista pela equidade de gênero e pelo direito à cidade.
- Atualmente é consultora do Banco Mundial, CEO da ASK-AR (consultoria em análise de dados), membro do Conselho Deliberativo da AEAMESP (Associação dos Engenheiros e Arquitetos de Metrô) e da comunidade Transparência Hacker.
- É co-fundadora dos grupos RLadies - São Paulo, do PoliGNU (Grupo de Estudos de Software Livre da Poli-USP) e da PoliGen (Grupo de Estudos de Gênero da Poli-USP).
- Cursa doutorado em Smart Cities pelo IME-USP, é mestra em Engenharia e Planejamento de Transportes (Poli-USP) e especialista em Democracia Participativa, Repúblicas e Movimentos Sociais (UFMG) além de ter formação em Engenharia Civil/Arquitetura pela USP (Programa Poli-FAU).

# Cronograma

- 1 - O que é o R-Ladies?
- 2 - Introdução sobre a linguagem de marcação markdown e sobre arquivos do tipo RMarkdown.
- 3 - Como criar uma apresentação utilizando o pacote Xaringan.
- 4 - Hospedagem da apresentação utilizando o Github Pages.

# O que é o R-Ladies?

R-Ladies é uma organização mundial que promove a diversidade de gênero na comunidade da linguagem R. R-Ladies São Paulo integra, orgulhosamente, a organização R-Ladies Global, em São Paulo.

## Como?

Através de meetups e mentorias em um ambiente seguro e amigável.

Nosso principal objetivo é promover a linguagem computacional estatística R compartilhando conhecimento, assim, quem tiver interesse na linguagem será bem-vinda, independente do nível de conhecimento.

Fonte: [About us - R-Ladies](#), [Meetup R-Ladies São Paulo](#)

# Para quem?

Nosso principal público-alvo são as pessoas que se identificam com o gênero feminino, portanto, mulheres cis, mulheres trans, bem como pessoas não-binárias e queer.

# Missão

Como uma iniciativa de diversidade, a missão das R-Ladies é alcançar uma representação proporcional de pessoas de gêneros atualmente sub-representados na comunidade R, incentivando, inspirando e capacitando-as.

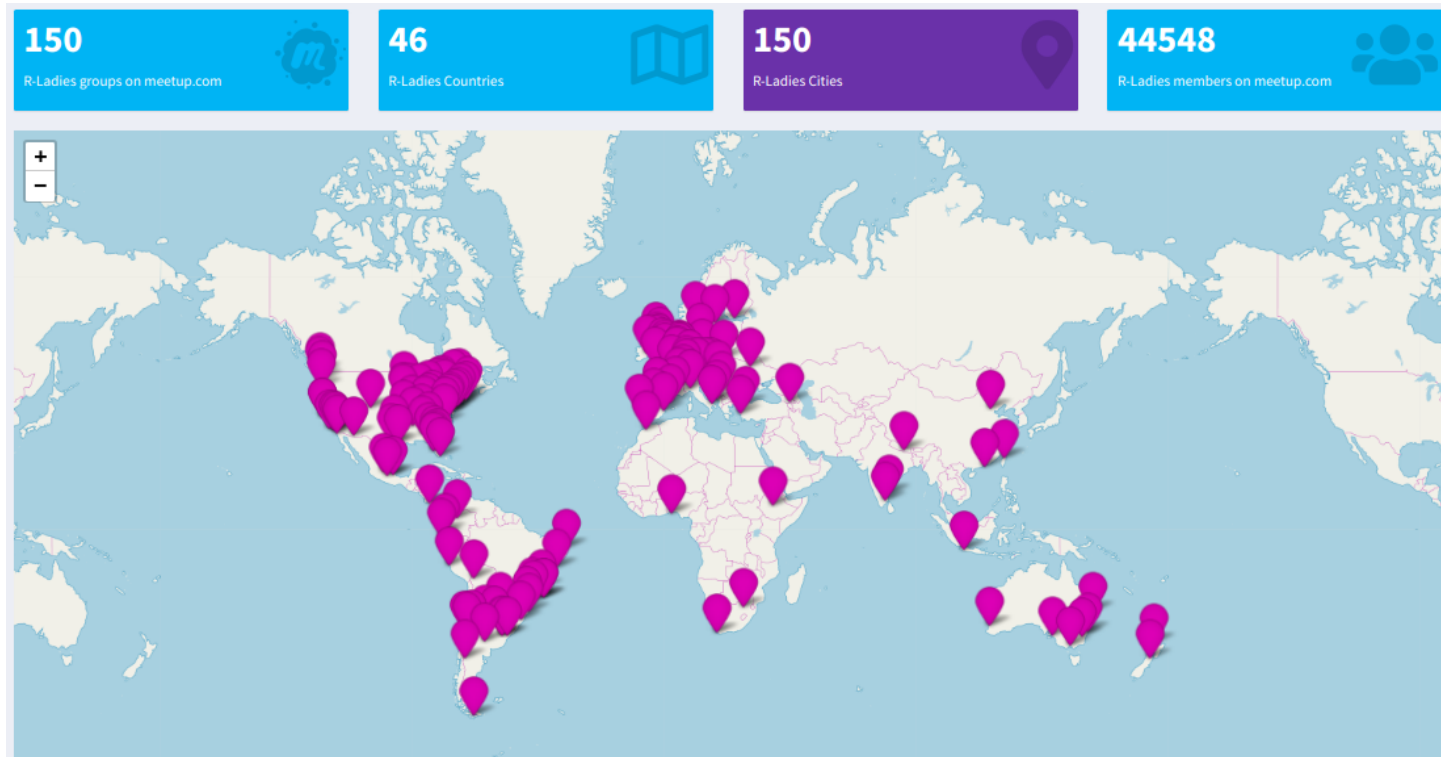
Fonte: [About us - R-Ladies](#), [Meetup R-Ladies São Paulo](#)

# Como o R-Ladies começou?



**Gabriela de Queiroz** fundou o R-Ladies no dia **1 de outubro de 2012**. Ela queria retribuir à comunidade depois de ir a vários encontros e aprender muito de graça. O primeiro encontro R-Ladies foi realizado em **San Francisco, Califórnia (Estados Unidos)**. Nos anos seguintes, mais capítulos do R-Ladies começaram em todo o mundo.

Fonte: [About us - R-Ladies](#)



Fonte: [R-Ladies Shiny App](#) (13/04/2019)



# Código de conduta

O R-Ladies dedica-se a proporcionar uma experiência livre de assédio para todos.  
Não toleramos nenhuma forma de assédio aos participantes.

Código de conduta - R-Ladies

# R-Ladies no Brasil:

Atualizado em: 13/04/2019. Fonte: [R-Ladies Shiny App](#)

Show  entries

Search:

	city	dt_created	members
1	Florianópolis	2019-04-07	22
2	Salvador	2018-07-23	114
3	Niterói	2018-06-04	289
4	Curitiba	2018-05-29	151
5	Belo Horizonte	2018-04-20	509
6	São Paulo	2018-02-10	343
7	Porto Alegre	2017-10-30	218
8	Natal	2017-09-25	135
9	Americana	2017-06-24	28
10	Rio de Janeiro	2017-02-27	394

Showing 1 to 10 of 10 entries

Previous

1

Next 10 / 58

# Como saber mais?

Meetup

Twitter: [@RLadiesGlobal](#), [@RLadiesSaoPaulo](#)

Instagram: [@RLadiesSaoPaulo](#)

Facebook: [@RLadiesSaoPaulo](#)

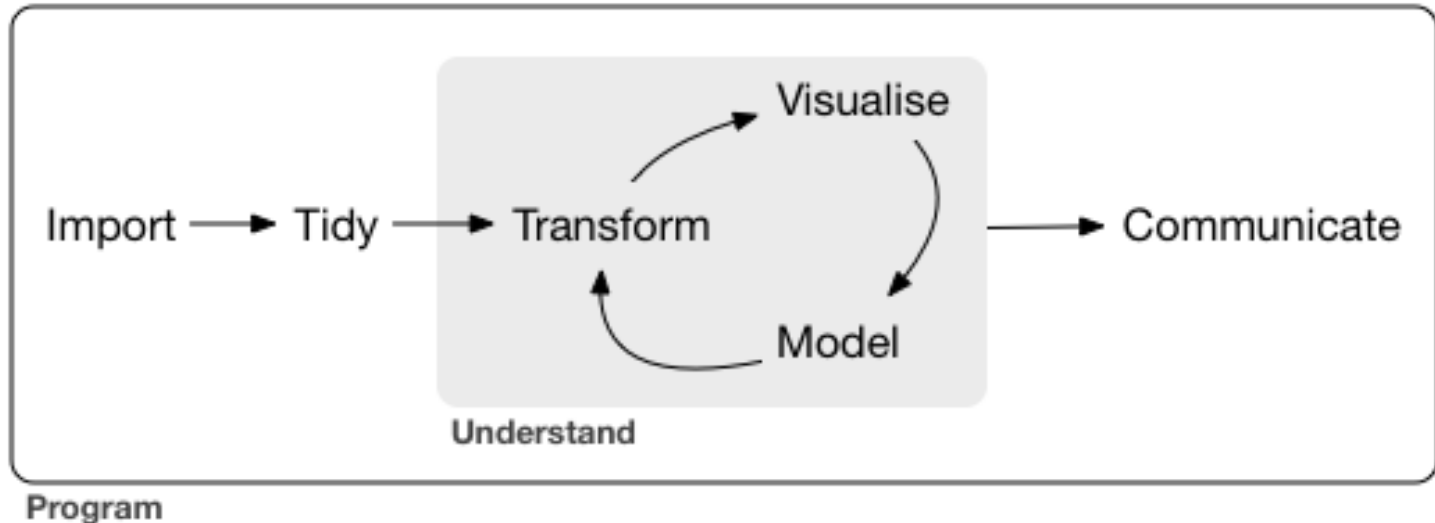
Website: <https://rladies.org/>

R-Ladies LATAM Blog (Latin America) - Em breve!

[R-Ladies - How do get involved](#)

**Introdução sobre a linguagem de marcação  
markdown e arquivos do tipo RMarkdown.**

# Por que apresentações são importantes?



Fonte: <http://r4ds.had.co.nz/>

- Uma das formas de comunicar os resultados!
- O que vocês utilizam em apresentações?

# Formas de criar apresentações no R:

- Pacote xaringan - Utiliza a biblioteca remark.js do JavaScript para criação de apresentações; Utiliza RMarkdown.
- Outros:
  - output: **beamer\_presentation** -> beamer (pdf)
  - output: **ioslides\_presentation** -> ioslides (HTML)
  - output: **slidy\_presentation** -> slidy (HTML)

Obs: para separar entre um slide e outro, usar três \*

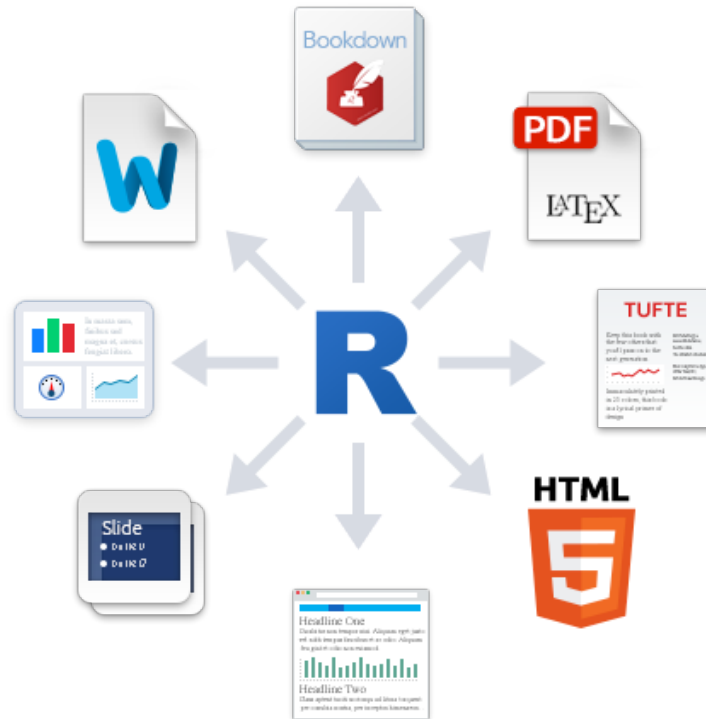
- Hoje iremos praticar como criar uma apresentação utilizando o R e o pacote Xaringan.

# Rmarkdown

- É um tipo de arquivo que suporta códigos em R, texto, markdown e outros formatos.
- Apresenta os seus códigos, o resultado, e também os seus comentários.
- Arquivos **.Rmd** são reprodutíveis.
- São interessantes para trabalhar em equipe, para comunicar seus resultados, e para desenvolver seus códigos em R.
- O markdown é uma linguagem de marcação simples.
- [Rmarkdown Cheatsheet](#)
- Apresentação Yihui: [Advanced R Markdown](#)
- Dica de livro: [R Markdown: The Definitive Guide](#)

# Rmarkdown

- Possibilita exportar diferentes tipos de arquivos.





# Markdown

- O Rmarkdown utiliza a linguagem de marcação Markdown.
- O markdown é uma linguagem de marcação simples.
- [Rmarkdown Cheatsheet 2.0](#) e [Rmarkdown Cheatsheet](#).
- Elaborada para ser fácil de ler, fácil de escrever.

# Markdown

Negrito: **\*\*bold\*\***

Itálico: *\_italics\_*

Adicionar títulos

# Título 1

## Título 2

### Título 3

## Título 1

### Título 2

#### Título 3

# Fazendo uma lista

Você pode fazer uma lista escrevendo com hífen ou asteriscos, como a seguir:

- \* **negrito com asteriscos duplos**
- \* *itálico com underline*
- \* `fonte do tipo código com crase`

ou dessa forma:

- **negrito com asteriscos duplos**
- *itálico com underline*
- `fonte do tipo código com crase`

Cada um vai aparecer como:

- **negrito com asteriscos duplos**
- *itálico com underline*
- `fonte do tipo código com crase`

Referência: [SW Carpentry](#)

# Fazendo uma lista enumerada

Você pode fazer uma lista numerada usando somente números. Você pode usar o mesmo número quantas vezes quiser:

1. **negrito com asteriscos duplos**

1. itálico com underline

1. `fonte do tipo código com crase`

Isso irá aparecer como:

1. **negrito com asteriscos duplos**

2. itálico com underline

3. `fonte do tipo código com crase`

Referência: [SW Carpentry](#)

# Mais de Markdown

Você pode fazer um hiperlink dessa forma: `[**texto para mostrar**]`  
(<http://the-web-page.com>).

Você pode incluir arquivo de imagem dessa forma: `![**título**]`  
(<http://url/for/file>)

Se você sabe como escrever equações em LaTeX, ficará feliz em saber que pode usar `$$` para inserir equações matemáticas, como:

```
# $$y = \mu + \sum_{i=1}^p \beta_i x_i + \epsilon$$
```

$$y = \mu + \sum_{i=1}^p \beta_i x_i + \epsilon$$

Referência: [SW Carpentry](#)

# Mais sobre Markdown:

Consulte a [Cheatsheet!](#)

# Como criar uma apresentação utilizando o pacote Xaringan

# Introdução sobre o Xaringan

- O xaringan é um pacote do R para criar apresentações usando o [remark.js](#) através do Rmarkdown.



- Como pronuncia Xaringan? Veja a [resposta do autor do pacote](#) e [Video](#) (1 minuto e 04 segundos)
- Mais sobre o xaringan:
  - [Apresentação da documentação](#)
  - [Livro R Markdown \(Capítulo 7\)](#)
  - [Remark.js Wiki](#)
  - [Xaringan Wiki](#)



# Criador do xaringan

- O xaringan foi criado pelo [Yihui Xie](#). Ele é desenvolvedor de software na RStudio.
- Desenvolveu os seguintes pacotes: [knitr](#), [animation](#), [bookdown](#), [blogdown](#), [pagedown](#), [xaringan](#), e [tinytex](#).



Thanks Yihui for this amazing package!

# Pré-requisitos:

- R
- RStudio
- Pacote **Rmarkdown** \* O R instala esse pacote automaticamente quando necessário.
- Pacote **Xaringan**
- Pacote **knitr**
- Google Chrome ou Chromium.

# Instalar o Xaringan

- Existem duas formas de instalar:

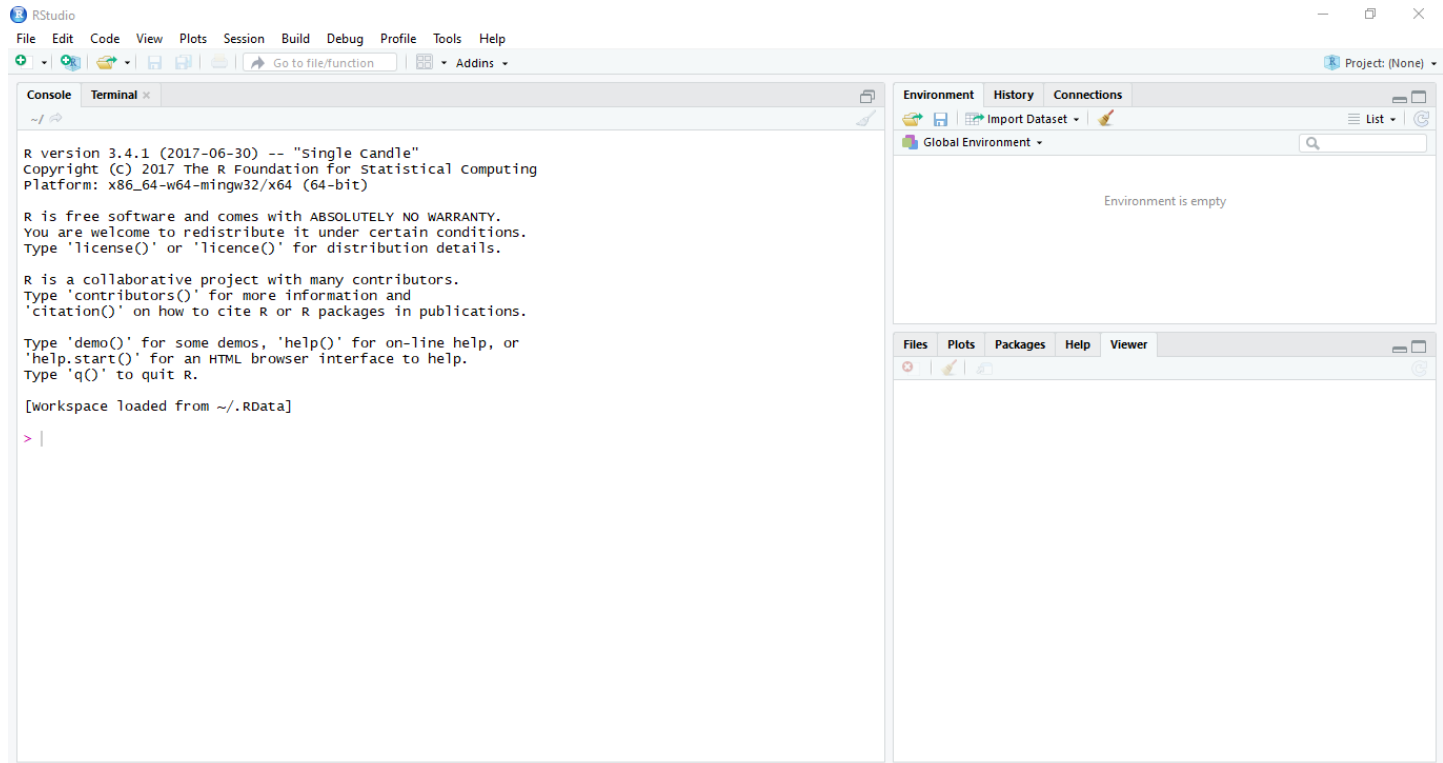
CRAN:

```
install.packages("xaringan")
```

Github:

```
devtools::install_github("yihui/xaringan")
```

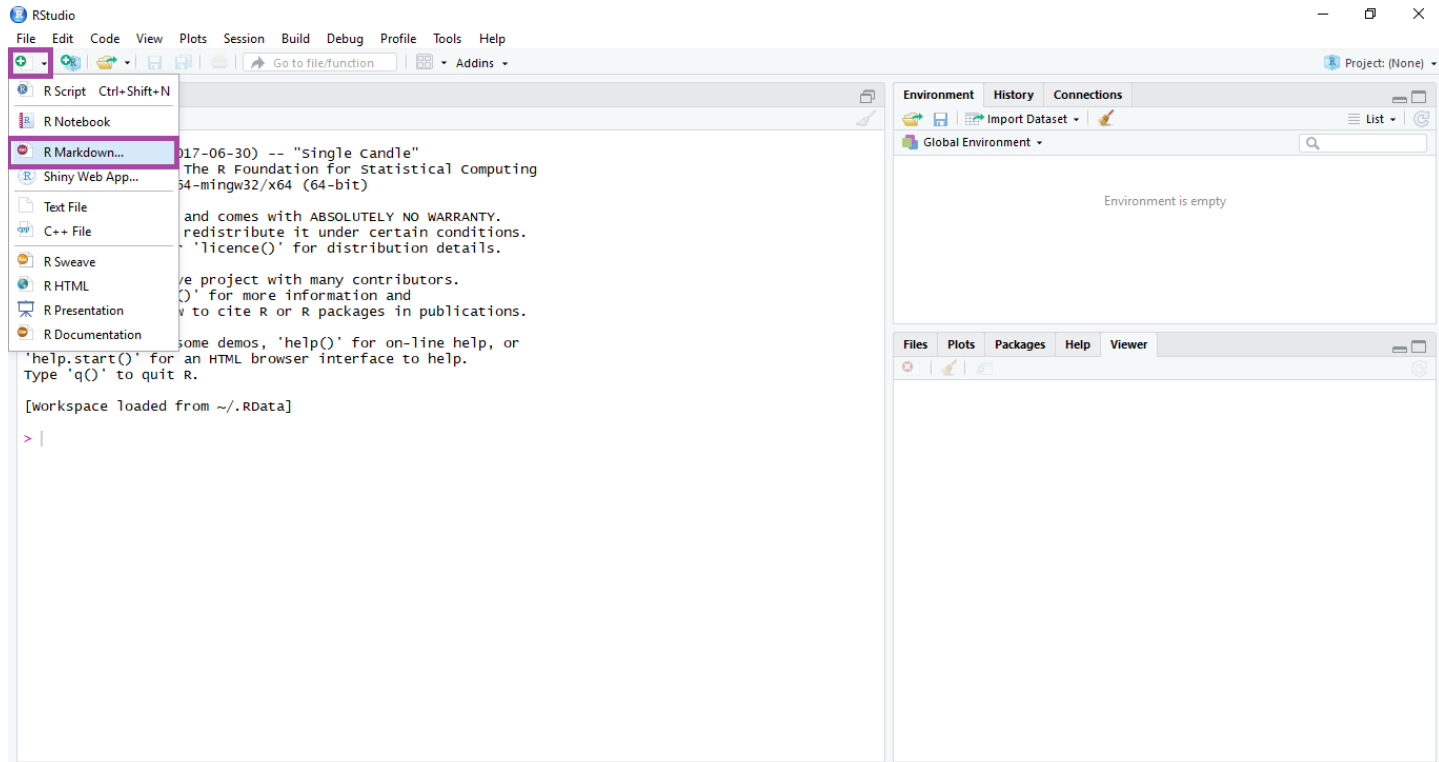
# Inicie o RSTUDIO:



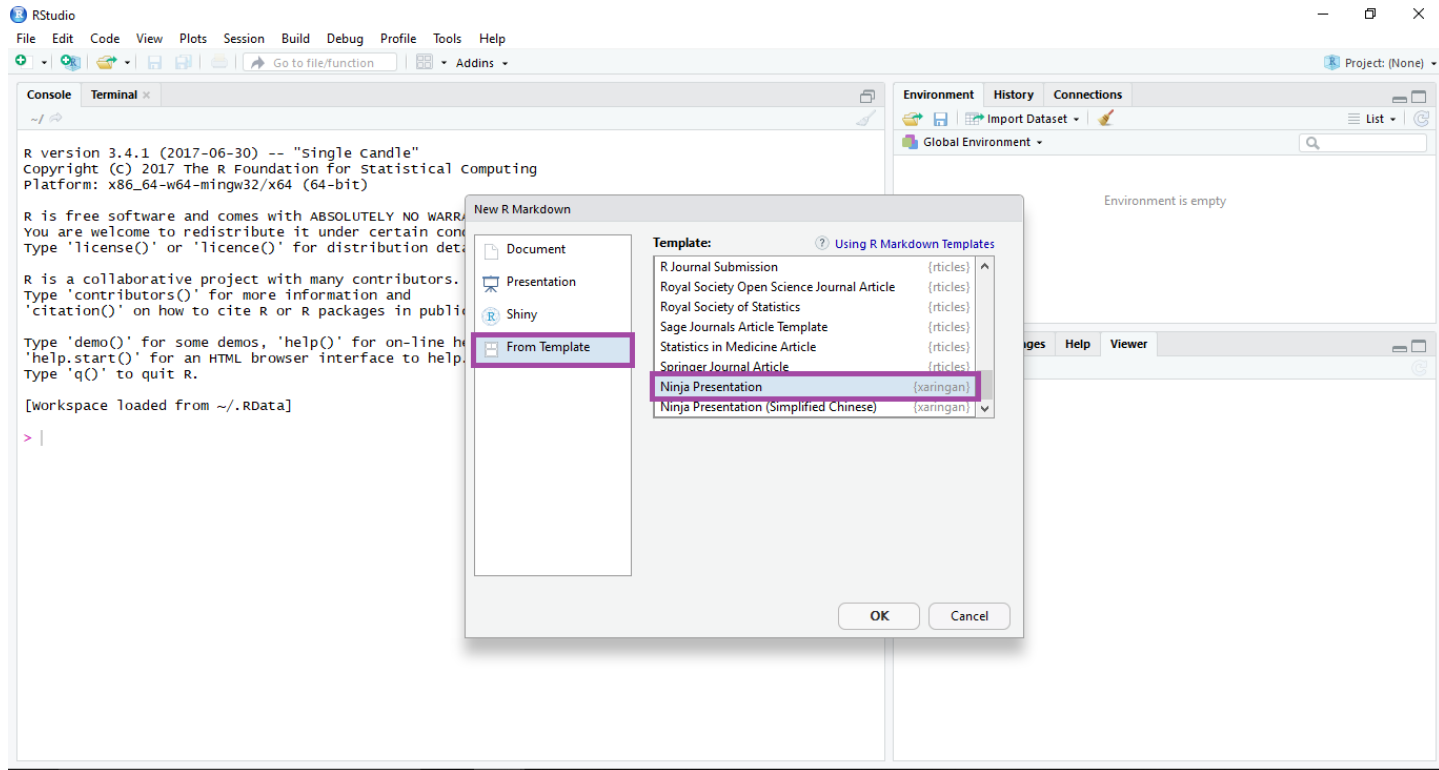
- Instale os pacotes:

```
install.packages("rmarkdown")
install.packages("knitr")
devtools::install_github("yihui/xaringan")
```

# Crie um novo arquivo Rmarkdown:

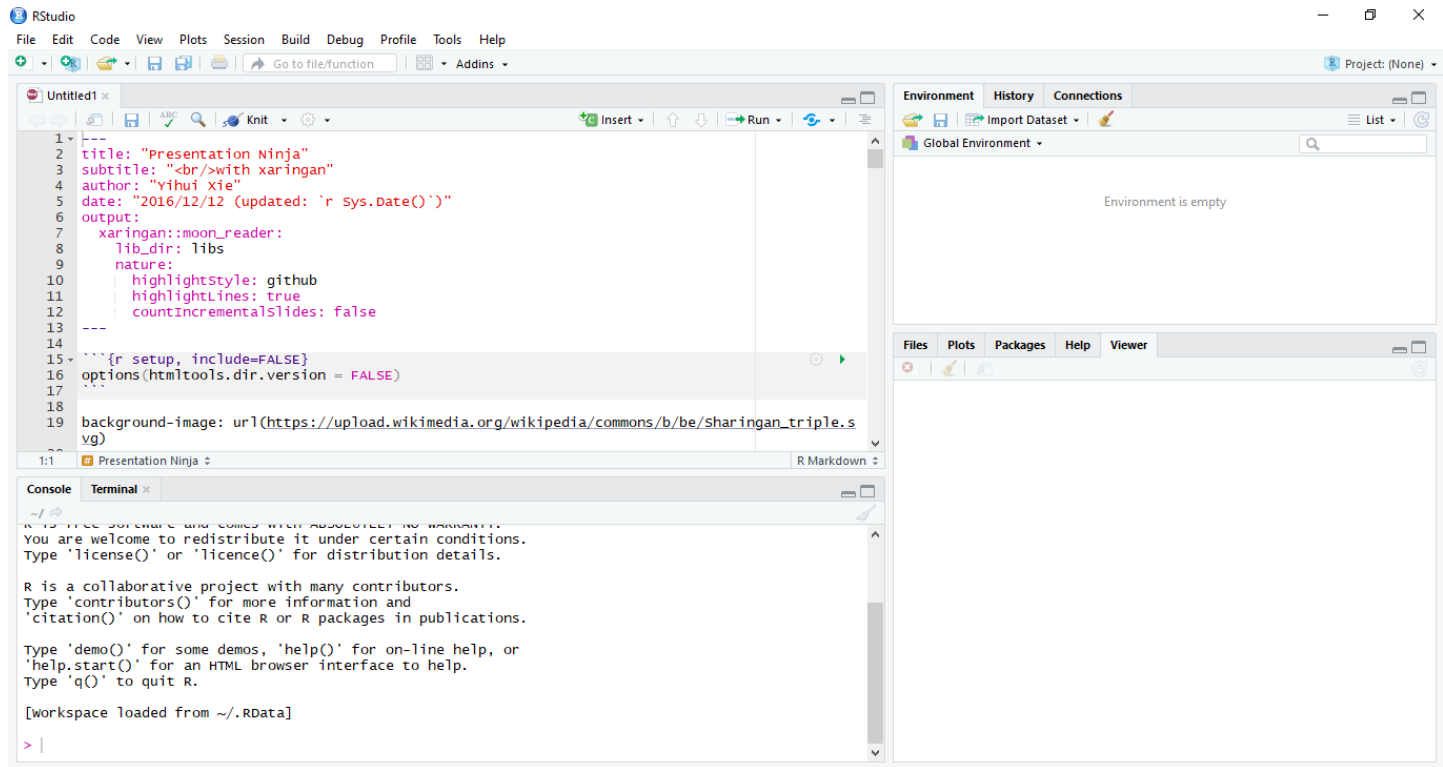


# Selecione o template Ninja Presentation:



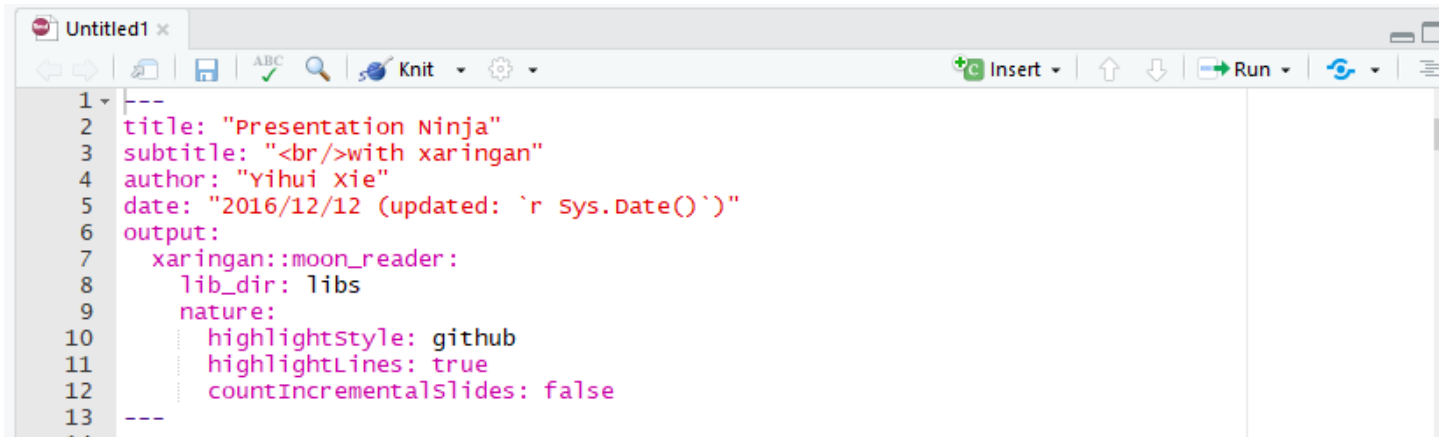
- Dica: para facilitar posteriormente, crie a sua apresentação com o nome de `index.Rmd`.

# Será criado um RMarkdown com o Ninja Presentation:



# yaml - metadados do documento:

- yet another markup language
- Cuidado com a **identação**
- Controla muitas das configurações do documento.

A screenshot of a code editor window titled 'Untitled1'. The editor displays a YAML document with the following content:

```
1 ---
2 title: "Presentation Ninja"
3 subtitle: "<br/>with xaringan"
4 author: "Yihui Xie"
5 date: "2016/12/12 (updated: `r Sys.Date()`)"
6 output:
7   xaringan::moon_reader:
8     lib_dir: libs
9     nature:
10       highlightStyle: github
11       highlightLines: true
12       countIncrementalSlides: false
13 ---
```

The code is color-coded: dashes are grey, strings are red, and other text is black. The editor has a toolbar at the top with icons for undo, redo, save, search, and Knit, along with a 'Run' button.

- title: título
- subtitle: subtítulo
- author: autor
- date: data
- output: formato (Ex: html\_document, pdf\_document, etc). No caso do Xaringan, o output é **xaringan::moon\_reader**
  - lib\_dir: pasta onde colocará bibliotecas necessárias (ex. remark-latest.min.js)

Leia também: [Output Formats - R Markdown: The Definitive Guide](#)



# Quais são os temas disponíveis?

- Depende da versão do xaringan que está usando:

```
names(xaringan::list_css())
```

```
## [1] "chocolate-fonts" "chocolate"      "default-fonts"
## [4] "default"          "duke-blue"       "fc-fonts"
## [7] "fc"               "hygge-duke"      "hygge"
## [10] "kunoichi"         "lucy-fonts"      "lucy"
## [13] "metropolis-fonts" "metropolis"      "middlebury-fonts"
## [16] "middlebury"       "ninjutsu"        "rladies-fonts"
## [19] "rladies"          "robot-fonts"     "robot"
## [22] "rutgers-fonts"    "rutgers"         "shinobi"
## [25] "tamu-fonts"       "tamu"            "uo-fonts"
## [28] "uo"              "uol-fonts"       "uol"
```

Alguns exemplos de disponíveis: [xaringan themes](#)

# Colocar o tema RLadies

```
output:  
  xaringan::moon_reader:  
    css: ["rladies", "rladies-fonts"]
```

- Criado por/Made by: [Alison Presmanes Hill](#)
- Se você entende de CSS, é possível adaptar os temas e também criá-los :)

# Outro tema RLadies: Kunoichi

- É esse tema que estou usando :)

```
output:  
  xaringan::moon_reader:  
    css: [kunoichi, ninjutsu, rladies-fonts]
```

- Criado por/Made by: [Emi Tanaka](#)

# RemarkJS

- O xaringan utiliza o seguinte arquivo javascript:  
<https://remarkjs.com/downloads/remark-latest.min.js>
- Por padrão, a apresentação só funcionará online!
- Como deixar o RemarkJS offline?

```
xaringan::summon_remark() # o xaringan irá fazer o download  
# da versão mais recente do remark e salvar na pasta /libs
```

O yaml deve conter as seguintes linhas:

```
output:  
  xaringan::moon_reader:  
    chakra: libs/remark-latest.min.js  
    lib_dir: libs
```

## Outra dica: Encoding

Para evitar problemas de codificação, adicione o encoding no yaml:

```
encoding: "UTF-8"
```

Leia o texto do blog do Yihui: <https://yihui.name/en/2018/11/biggest-regret-knitr/>

# Exemplo de yaml desta apresentação

```
---
title: "Comunicando seus resultados: "
subtitle: "Criando apresentações com R"
author: "Beatriz Milz e Haydee Svab - R-Ladies São Paulo"
date: "20 de maio de 2019 <br> (Atualizado em: `r format(Sys.Date(), for
encoding: "UTF-8"
output:
  xaringan::moon_reader:
    chakra: libs/remark-latest.min.js
    css: [kunoichi, ninjutsu, rladies-fonts]
    lib_dir: libs
    nature:
      highlightStyle: github
      highlightLines: true
      countIncrementalSlides: false
---
```

# Chunks de Código R

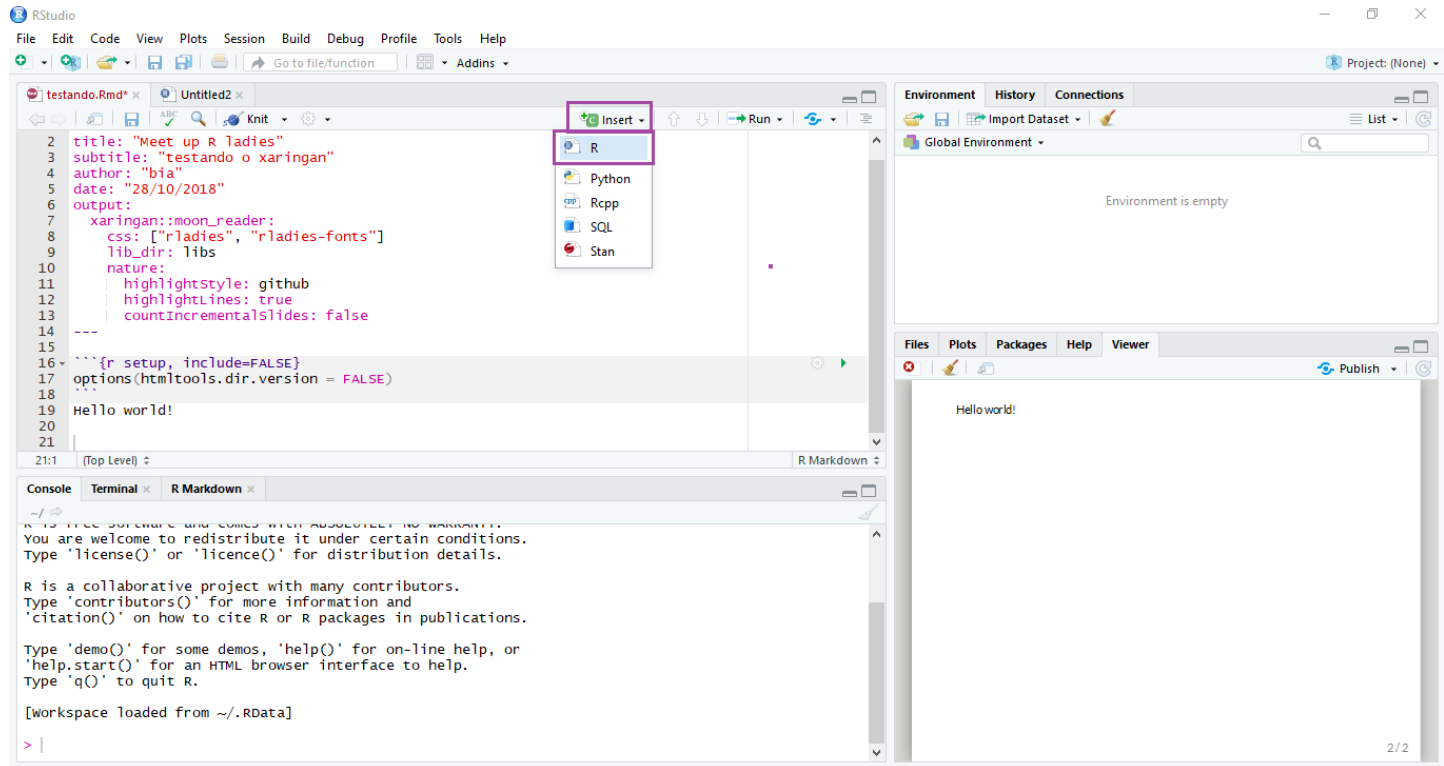
- É um campo onde podemos inserir código de R em um arquivo Rmarkdown:

```
```${r chunk_name}  
...  
```
```



- É uma boa ideia dar a cada chunk um nome, pois ele podem ajudar a corrigir erros, e, se quaisquer gráficos são produzidos, os nomes dos arquivos serão baseados no nome do chunk de código que os produzirem.

# Como criar um chunk de código?

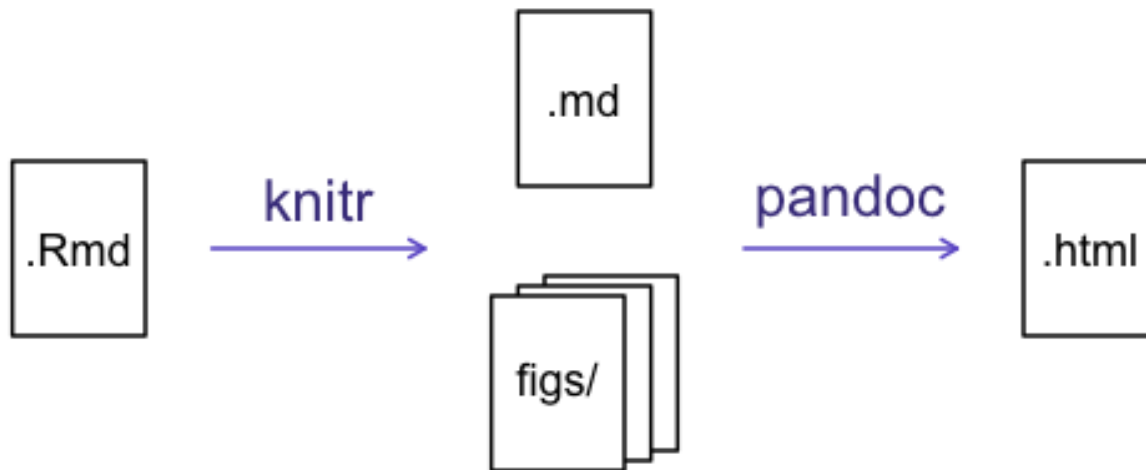


Dica: Atalho do teclado para criar chunks: **Ctrl + ALT + I**.

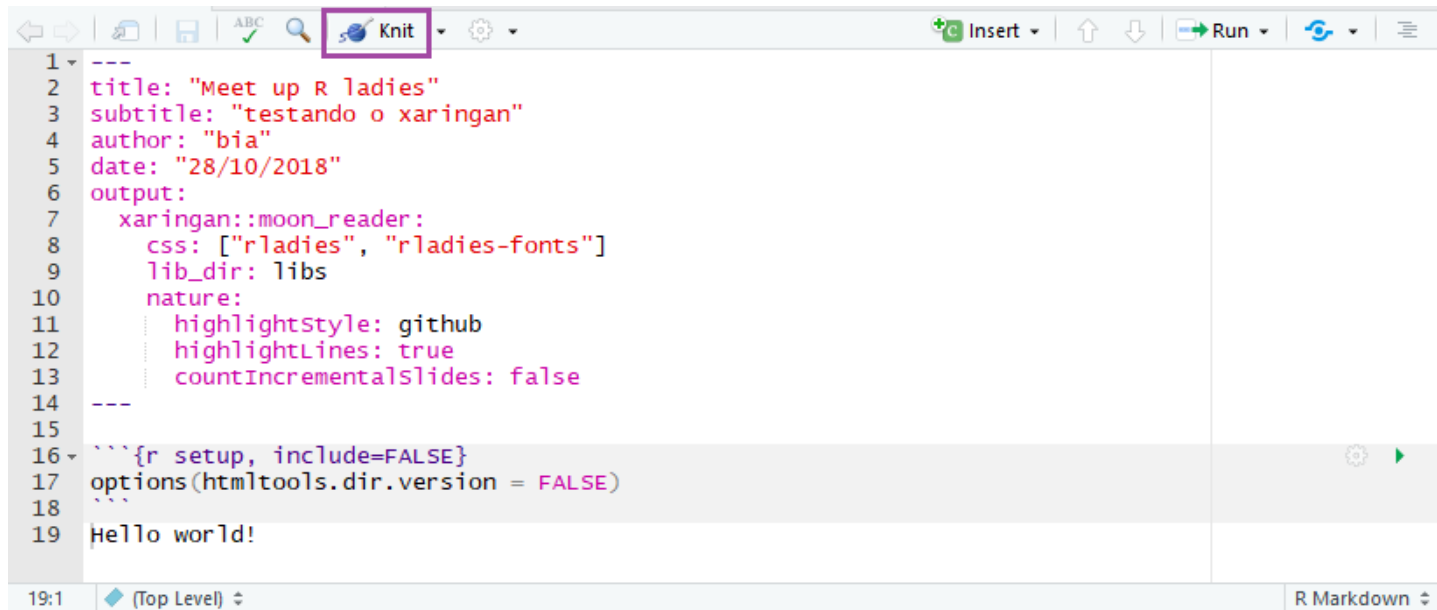


# Como as coisas são compiladas?

- Quando você pressiona o botão “Knit HTML”, o documento R Markdown é processado pelo knitr e um documento Markdown é produzido (como, também, um potencial conjunto de arquivos de figuras): o código em R é executado e substituído pelo input e output; se figuras são produzidas, links para essas figuras serão incluídos.
- A seguir, os documentos Markdown e de figuras são processados pela ferramenta pandoc, que converte o arquivo Markdown em um arquivo html, com as figuras anexadas.



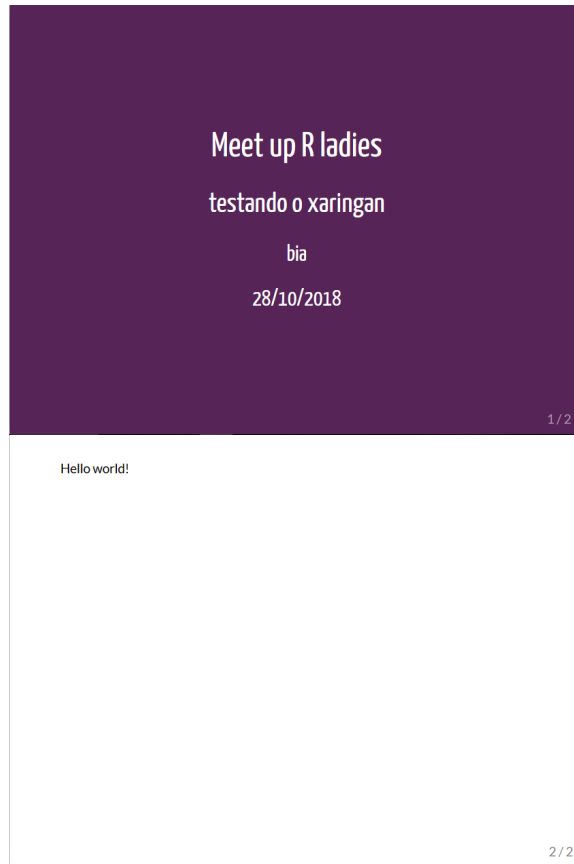
# Exemplo do yaml:



```
1 ---
2 title: "Meet up R ladies"
3 subtitle: "testando o xaringan"
4 author: "bia"
5 date: "28/10/2018"
6 output:
7   xaringan::moon_reader:
8     css: ["rladies", "rladies-fonts"]
9     lib_dir: libs
10    nature:
11      highlightstyle: github
12      highlightLines: true
13      countIncrementalSlides: false
14 ---
15
16 {r setup, include=FALSE}
17 options(htmltools.dir.version = FALSE)
18
19 Hello world!
```

- Aperte o Knit e...

# Resultado:



- O knit exportará um arquivo do tipo html.
- Agora o tema da apresentação está configurado!
- Aperte H ou ? para conferir a lista de atalhos do xaringan.

# DEU ERRO NO KNITR!

Calma! É bem comum dar erro ao compilar o arquivo pois **qualquer** código errado causa isso. Algumas coisas pra se observar:

- Yaml header - se algo estiver errado, vai dar erro ao compilar. Muito cuidado com a indentação!
- Código nos chunks - teste o Run All Chunks e ver se dá erro em algum específico. Se descobrir, tente verificar o que há de errado no código (pode ser a falta de um parênteses, vírgula, entre outros).



# Opções de chunk

Há uma variedade de opções que podem afetar como os chunks de código são tratados.

- Use `echo=FALSE` para evitar que o próprio código apareça.
- Use `results="hide"` para evitar que qualquer resultado seja impresso.
- Use `eval=FALSE` para mostrar o código, mas ele não é avaliado.
- Use `warning=FALSE` e `message=FALSE` para ocultar qualquer mensagens de avisos produzidas.
- Use `out.width` para controlar o tamanho de figuras produzidas (Ex: `out.width = "100%"`).

Assim, você pode escrever:

```
```{r load_libraries, echo=FALSE, message=FALSE}  
library("dplyr")  
library("ggplot2")  
```
```

Referência: [SW Carpentry](#)

# Opções globais de chunk

Muitas vezes, haverá algumas opções que você irá querer repetir seu uso; para isso, você pode configurar as opções globais de chunk, como:

```
```{r global_options, echo=FALSE}  
knitr::opts_chunk$set(fig.path="Figs/", message=FALSE, warning=FALSE,  
                        echo=FALSE, results="hide", fig.width=11)  
```
```

A opção `fig.path` define onde as figuras serão salvas. A `/` é importante, pois sem ela, as figuras seriam salvas no local padrão, mas só com nomes que começam com `Figs`.

Referência: [SW Carpentry](#)

# Código R no texto

Você pode fazer com que todo número em seu relatório seja reproduzível.

```
#Use `r` e ` ` para um in-line code chunk .  
#O código será executado e  
#substituído com o valor do resultado.
```

```
`r code`
```

Ex:

```
`r round(3.14159265359, 2)`
```

3.14

Não deixe que esse chunk in-line seja separado através das linhas.

Referência: [SW Carpentry](#)

# Código em R no texto

Outro exemplo:

```
# O dataframe mtcars possui `r nrow(mtcars)` carros.
```

Exemplo: O dataframe mtcars possui 32 carros.



# Exemplo: adicionando um gráfico

- Crie um chunk, e escreva o código que você geralmente utiliza para criar um gráfico :)

```
library(tidyverse)
ggplot(data = mtcars, mapping = aes(x = wt, y = mpg)) + geom_point() +
  stat_smooth(method = 'lm') + facet_grid(am ~ cyl)
```

## Exemplo: adicionando uma tabela - Dica: Usar a função DT::datatable()

```
DT::datatable(rladies_br)
```

Show  entries

Search:

|    | city           | dt_created | members |
|----|----------------|------------|---------|
| 1  | Florianópolis  | 2019-04-07 | 22      |
| 2  | Salvador       | 2018-07-23 | 114     |
| 3  | Niterói        | 2018-06-04 | 289     |
| 4  | Curitiba       | 2018-05-29 | 151     |
| 5  | Belo Horizonte | 2018-04-20 | 509     |
| 6  | São Paulo      | 2018-02-10 | 343     |
| 7  | Porto Alegre   | 2017-10-30 | 218     |
| 8  | Natal          | 2017-09-25 | 135     |
| 9  | Americana      | 2017-06-24 | 28      |
| 10 | Rio de Janeiro | 2017-02-27 | 394     |

Showing 1 to 10 of 10 entries

50 / 58

# Adicionando imagens usando o Knitr

Função: `knitr::include_graphics()`

Exemplo:

```
knitr::include_graphics("https://www.r-project.org/Rlogo.png")
```



# Adicionando imagens usando Knitr: opções de Chunk

- `out.width = "50%"` - tamanho da imagem



Opções do KnitR

# Adicionando imagens usando Knitr: opções de Chunk

- Alinhamento da figura

- `fig.align = 'center', 'default', **'left**, 'right'`



- `fig.align = 'center', 'default', 'left', **'right**`



# Adicionando imagens usando Knitr: opções de Chunk

- Legenda de figura
- `fig.cap="Logo R"`



R Logo

# Preview da apresentação

```
xaringan::inf_mr("index.Rmd")
```

- Post do Yihui sobre isso: <https://yihui.name/en/2019/02/ultimate-inf-mr/>
- Comigo não funciona com o tema Kunoichi :(

# Salvar a versão em PDF

- O autor do pacote disse que a forma mais confiável é abrir no Google Chrome ou Chromium, e salvar como PDF.
- Porém, existe essa função do pacote `pagedown`, que é recente e ainda será melhorada (pode não funcionar muito bem).
- Precisa ter o Chrome instalado.
- Comigo funciona :D

```
remotes::install_github('rstudio/pagedown')
```

```
pagedown::chrome_print("index.html")
```

Referência: [Apresentação do Yihui na RStudio Conf 2019: pagedown - Creating beautiful PDFs with R Markdown and CSS](#)



## Referências

- [SW Carpentry](#)
- [Apresentação do Yihui na RStudio Conf 2019: pagedown - Creating beautiful PDFs with R Markdown and CSS](#)

# Obrigada!

Apresentação elaborada com [Xaringan](#) e [Kunoichi](#).

Agradecimentos especiais:

- [Bruna Wundervald](#)
- [Julio Trecenti](#), [Curso-R](#) e [13 a Mostra de Estatística - IME USP](#) pelo mini-curso de Rmarkdown.

... ?