

Pintando e bordando no R

ggplot2 e rmarkdown

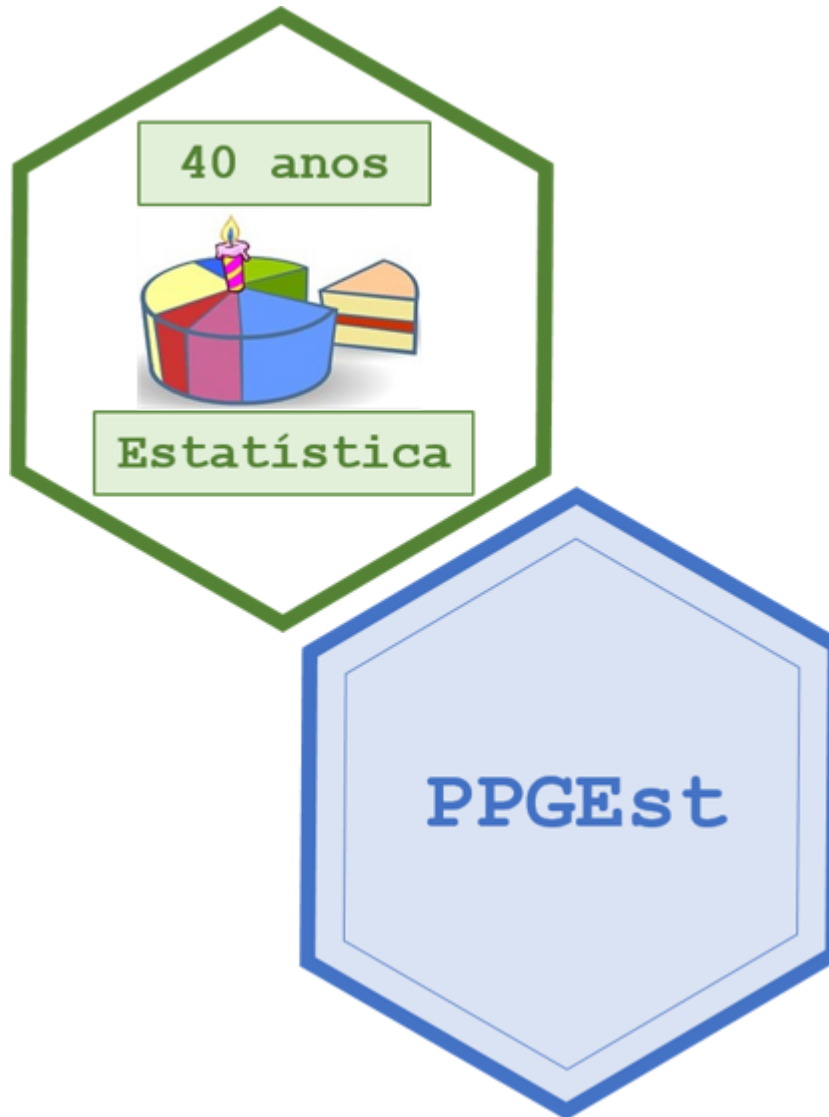
Rodrigo Citton P. dos Reis

em colaboração com Markus C. Stein, Márcia H. Barbian
e Silvana Schneider

Departamento de Estatística

IX SEMANÍSTICA e 1º Datathon da UFRGS
Porto Alegre, 15 de outubro de 2018

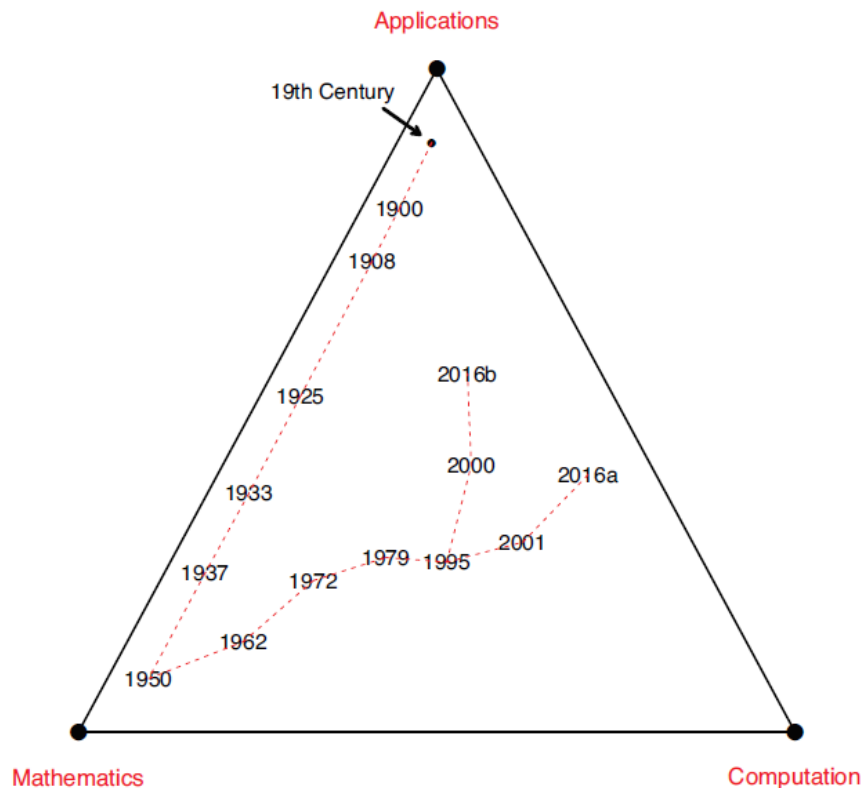




Estatística e Ciência dos dados



Estatística e Ciência dos dados



Fonte: Efron, B. e Hastie, T. (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge: Cambridge University Press.

MODERN DATA SCIENTIST

Data Scientist, the sexiest job of the 21st century, requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.

MATH & STATISTICS

- ☆ Machine learning
- ☆ Statistical modeling
- ☆ Experiment design
- ☆ Bayesian inference
- ☆ Supervised learning: decision trees, random forests, logistic regression
- ☆ Unsupervised learning: clustering, dimensionality reduction
- ☆ Optimization: gradient descent and variants

DOMAIN KNOWLEDGE & SOFT SKILLS

- ☆ Passionate about the business
- ☆ Curious about data
- ☆ Influence without authority
- ☆ Hacker mindset
- ☆ Problem solver
- ☆ Strategic, proactive, creative, innovative and collaborative

PROGRAMMING & DATABASE

- ☆ Computer science fundamentals
- ☆ Scripting language e.g. Python
- ☆ Statistical computing packages, e.g., R
- ☆ Databases: SQL and NoSQL
- ☆ Relational algebra
- ☆ Parallel databases and parallel query processing
- ☆ MapReduce concepts
- ☆ Hadoop and Hive/Pig
- ☆ Custom reducers
- ☆ Experience with xaaS like AWS

COMMUNICATION & VISUALIZATION

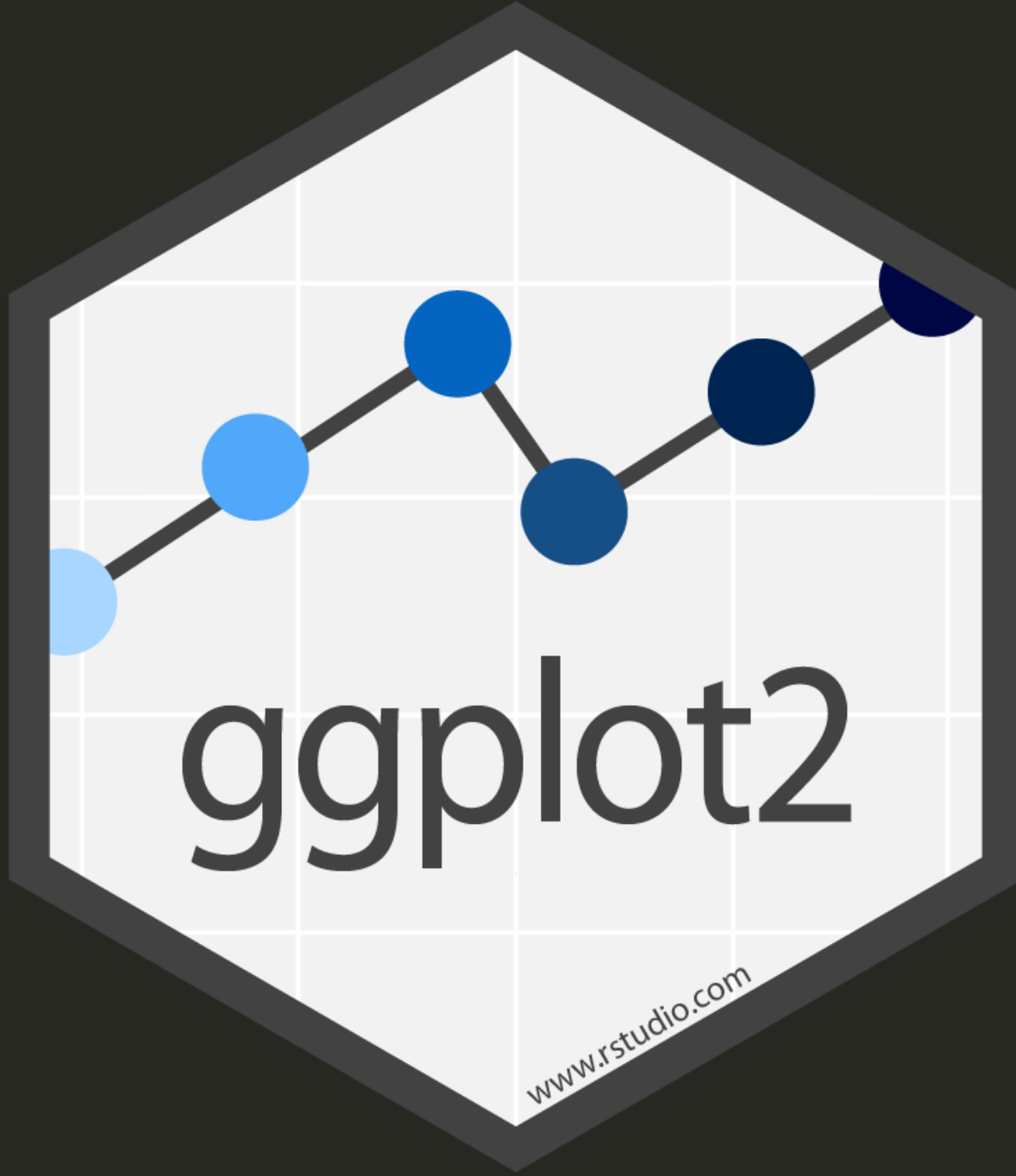
- ☆ Able to engage with senior management
- ☆ Story telling skills
- ☆ Translate data-driven insights into decisions and actions
- ☆ Visual art design
- ☆ R packages like ggplot or lattice
- ☆ Knowledge of any of visualization tools e.g. Flare, D3.js, Tableau





Você vai precisar de ...

- **R**
- **RStudio** é recomendável, mas não é necessário.
- Pacotes principais:
 - **ggplot2**
 - **rmarkdown**
- Pacotes auxiliares:
 - **knitr**
 - **plyr**
 - **gapminder**
- Banco de dados:
 - **gapminder** do pacote **gapminder**
- Arquivos de apoio:
 - <https://www.ufrgs.br/datathon>



O que é o ggplot2?



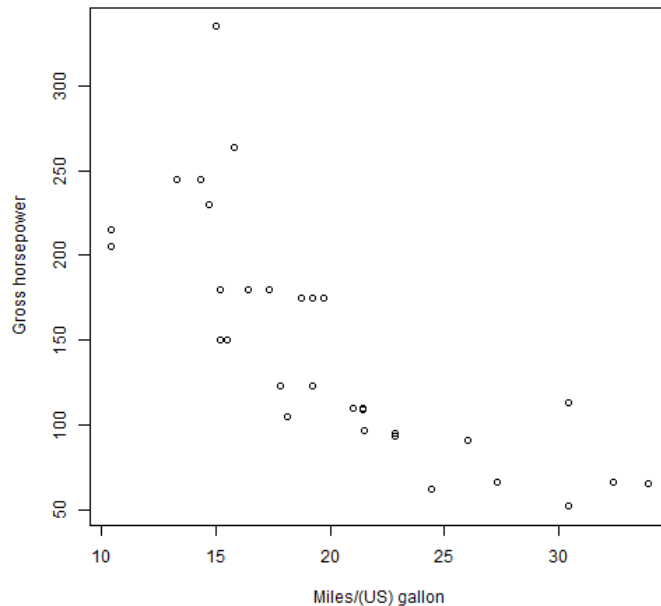
- Implementação em **R** da **Gramática dos Gráficos** (*Grammar of the Graphics*; **gg**) de **Leland Wilkinson**¹ por **Hadley Wickham**².
- **Pergunta:** que é um gráfico estatístico?
 - **gg:** é um mapeamento dos dados a partir de **atributos estéticos** (cores, formas, tamanho) de **formas geométricas** (pontos, linhas, barras)

[1] Wilkinson, L. (2005). *The Grammar of Graphics*. Springer; 2nd edition.

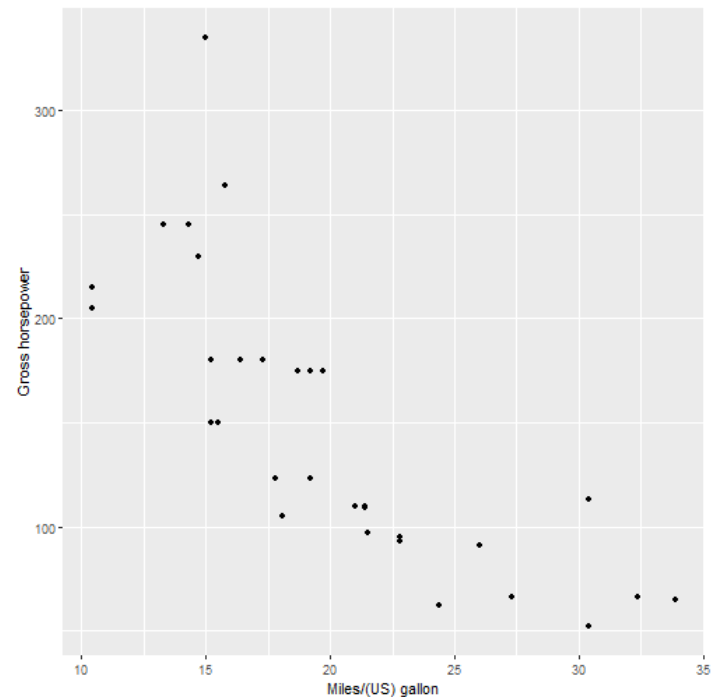
[2] Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer; 2nd edition.

Por que ggplot2?

Um gráfico do pacote **graphics**

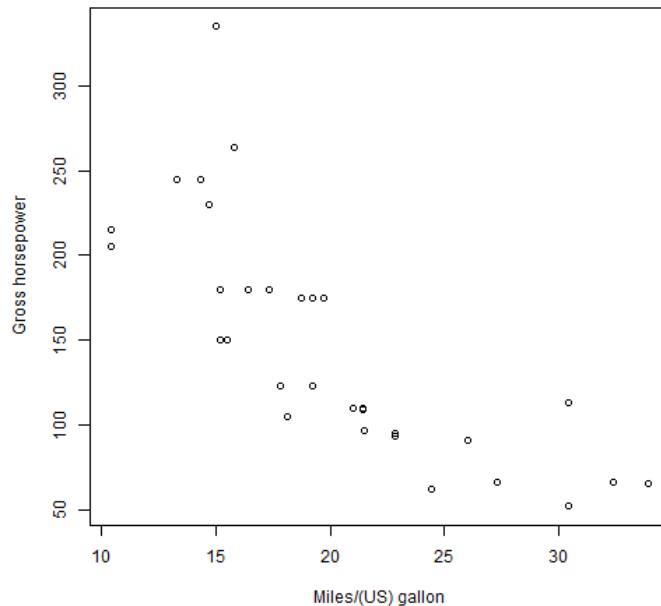


Um gráfico do pacote **ggplot2**

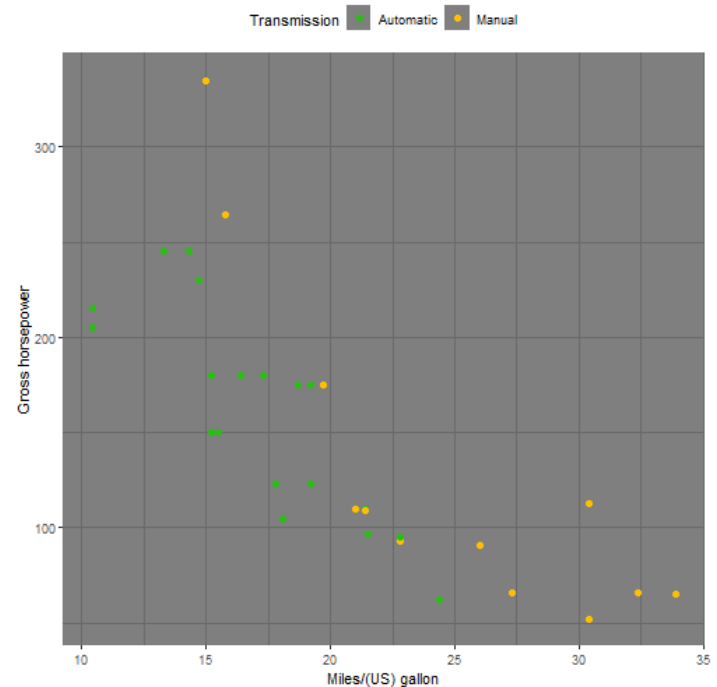


Por que ggplot2?

Um gráfico do pacote **graphics**



Um gráfico do pacote **ggplot2**



Camadas!

- No **ggplot2**, um gráfico é feito por camadas.





Instalando e carregando o ggplot2

- Instalando o pacote **ggplot2**

```
install.packages("ggplot2")
```

- Carregando o pacote **ggplot2**

```
library(ggplot2)
```

O banco de dados

- Carregando o banco de dados **gapminder**

```
# install.packages("gapminder")  
library(gapminder)  
gapminder
```

```
## # A tibble: 1,704 x 6  
##   country      continent  year lifeExp      pop gdpPercap  
##   <fct>        <fct>    <int>  <dbl>    <int>    <dbl>  
## 1 Afghanistan Asia      1952   28.8   8425333    779.  
## 2 Afghanistan Asia      1957   30.3   9240934    821.  
## 3 Afghanistan Asia      1962   32.0  10267083    853.  
## 4 Afghanistan Asia      1967   34.0  11537966    836.  
## 5 Afghanistan Asia      1972   36.1  13079460    740.  
## 6 Afghanistan Asia      1977   38.4  14880372    786.  
## 7 Afghanistan Asia      1982   39.9  12881816    978.  
## 8 Afghanistan Asia      1987   40.8  13867957    852.  
## 9 Afghanistan Asia      1992   41.7  16317921    649.  
## 10 Afghanistan Asia      1997   41.8  22227415    635.  
## # ... with 1,694 more rows
```

O banco de dados

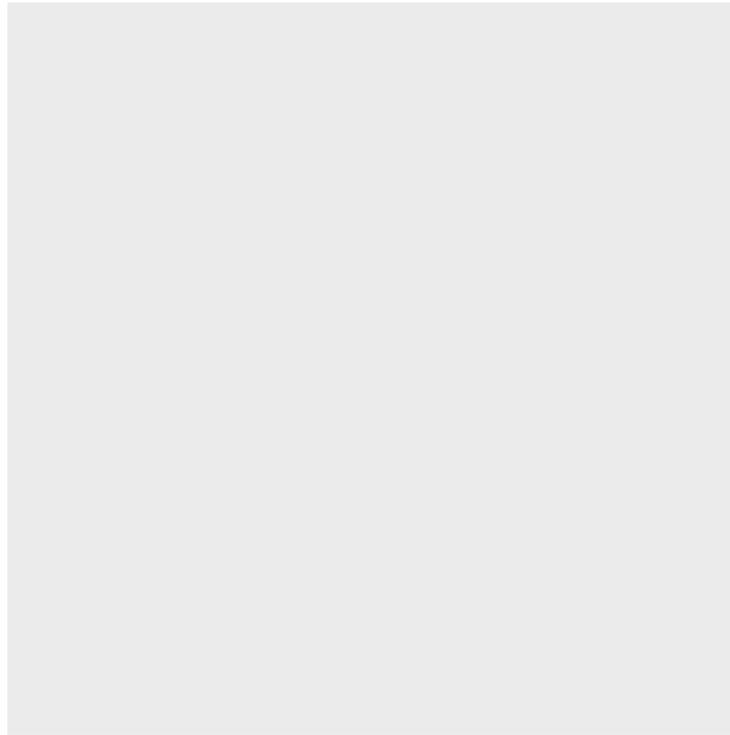
- gdpPercap: **renda per capita** ("PIB/Pop").
- lifeExp: **expectativa de vida** ao nascer (número de anos aproximados que se espera que um grupo de indivíduos nascidos no mesmo ano irá viver).
- year: 1952 a 2007 em incrementos de 5 anos.

```
library(dplyr)
gapminder <- gapminder %>%
  mutate(pop_m = pop/1e6)

gapminder07 <- gapminder %>%
  filter(year == 2007)
```

Camadas: dados

```
p <- ggplot(data = gapminder07)  
p
```



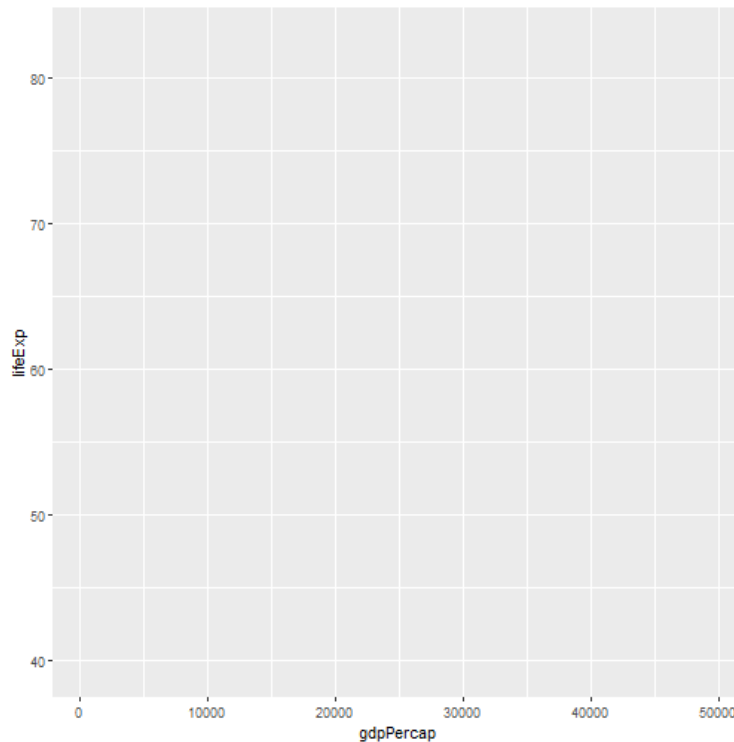
```
# print(p)
```

Camadas: estética



Camadas: elementos estéticos

```
p <- ggplot(data = gapminder07,  
            mapping = aes(x = gdpPercap, y = lifeExp))  
p
```

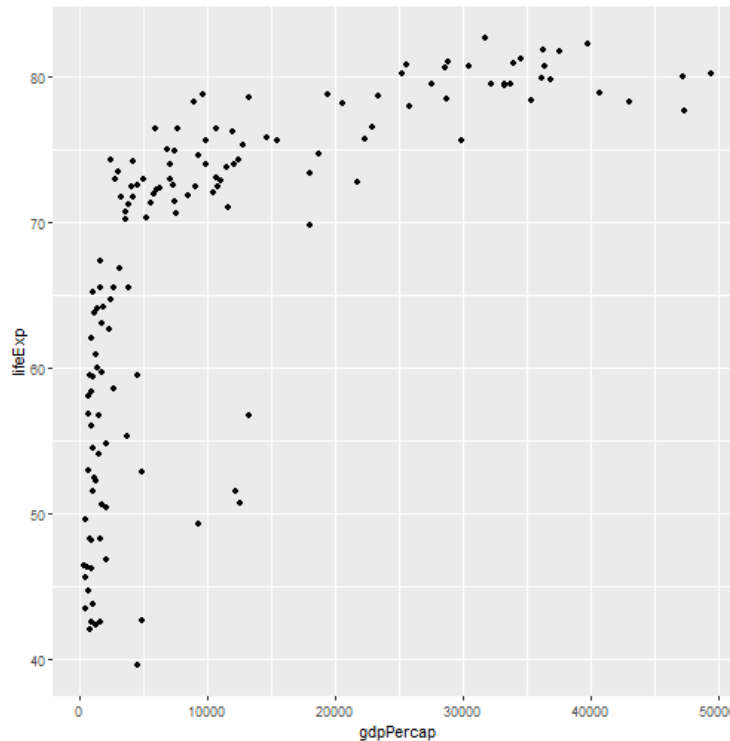


Camadas: elementos estéticos

- Os dois elementos estéticos mais importantes são os eixos x e y do gráfico.
- Existem outros elementos estéticos (**mais variáveis podem ser incorporadas ao gráfico**) que devem ser especificados de acordo com o tipo de gráfico.
 - `color` ou `colour`
 - `fill`
 - `size`
 - `group`
- Veremos a utilização destes elementos na sequência.

Camadas: elementos geométricos

```
p <- ggplot(data = gapminder07,  
            mapping = aes(x = gdpPercap, y = lifeExp)) +  
  geom_point()  
p
```



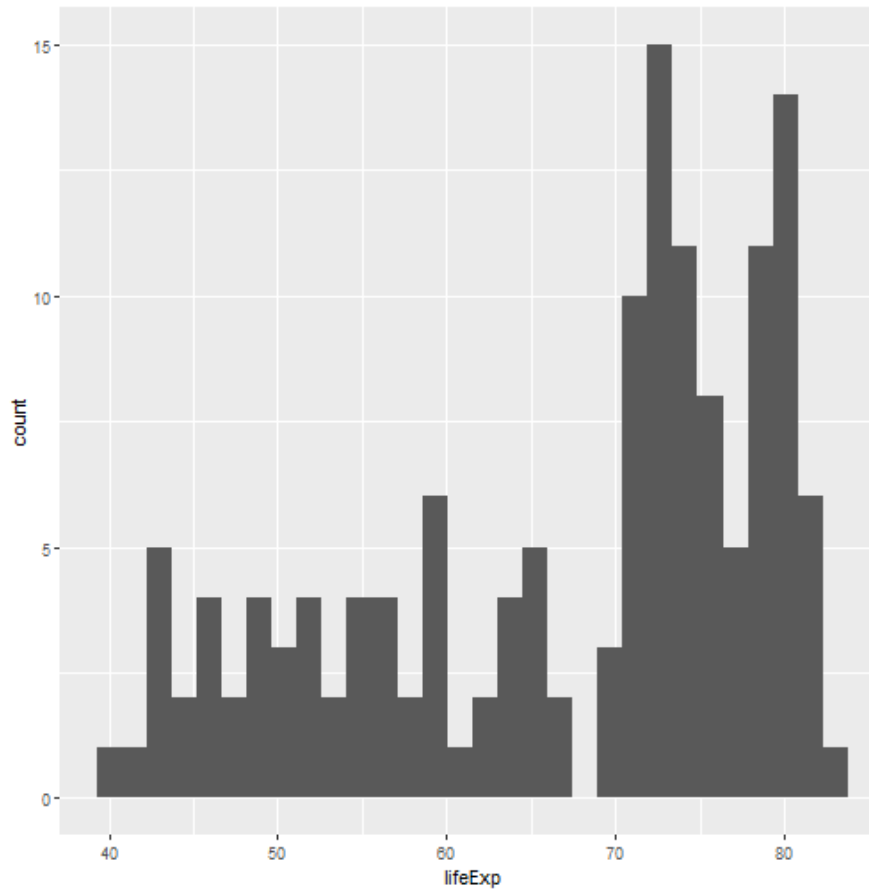
Camadas: elementos geométricos

- O gráfico anterior é um **diagrama de dispersão** (*scatter plot*) e é definido pela função `geom_point()`.
 - Este é acrescentado ao gráfico com o **operador +**.
- Diferentes elementos geométricos definem diferentes tipos de gráficos. Os principais são listadas a seguir:
 - `geom_histogram()`: gera um histograma
 - `geom_line()`: gera um gráfico de linha
 - `geom_boxplot()`: gera um diagrama de caixa (boxplot)
 - `geom_bar()`: gera um gráfico de barras/colunas
 - `geom_density()`: gera um gráfico da densidade estimada (por *kernel*)
 - `geom_violin()`: gera um gráfico de violino (mistura de boxplot com densidade estimada)
- Para uma lista completa de geoms veja:
<https://ggplot2.tidyverse.org/reference/index.html#section-layer-geoms>.
- **Elementos estéticos** (`aes()`) podem ser passados para cada geom.

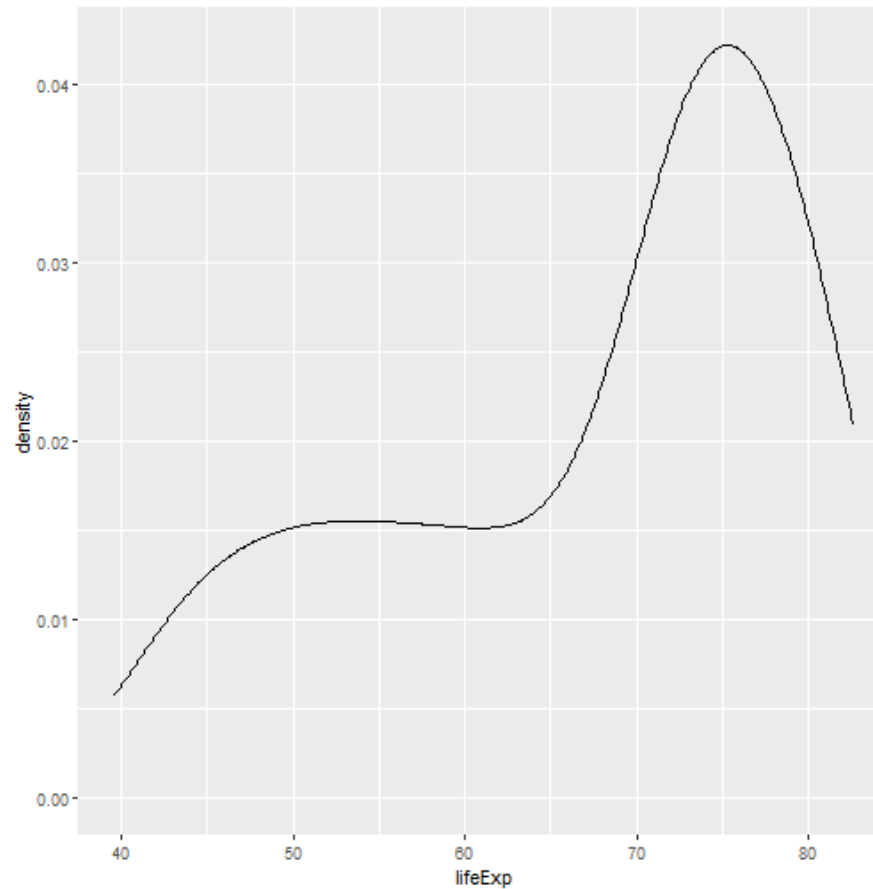
Exemplos!



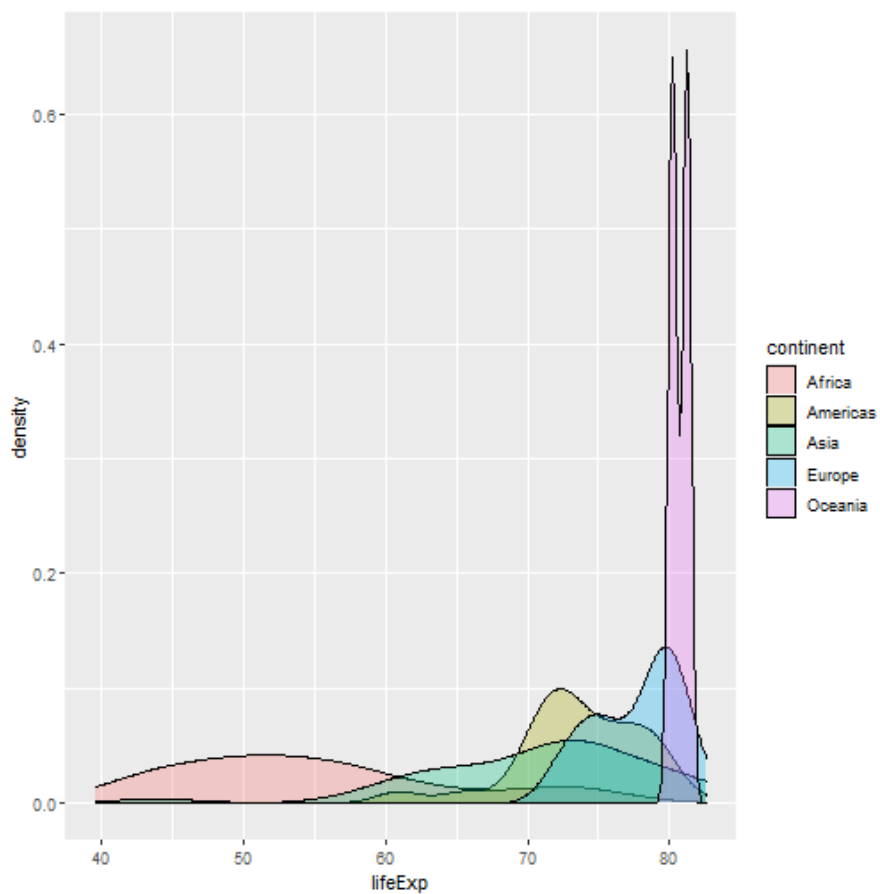
```
# Histograma  
p <- ggplot(data = gapminder07,  
            mapping = aes(x = lifeExp)) +  
  geom_histogram()  
p
```



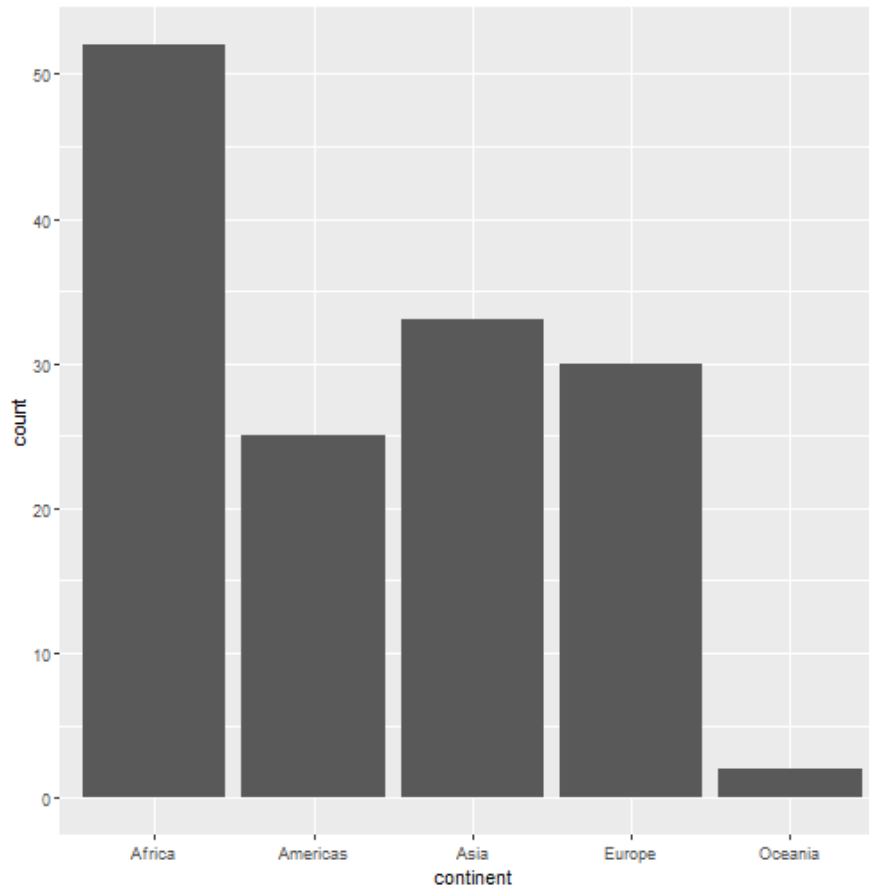
```
# Densidade estimada  
p <- ggplot(data = gapminder07,  
            mapping = aes(x = lifeExp)) +  
  geom_density()  
p
```



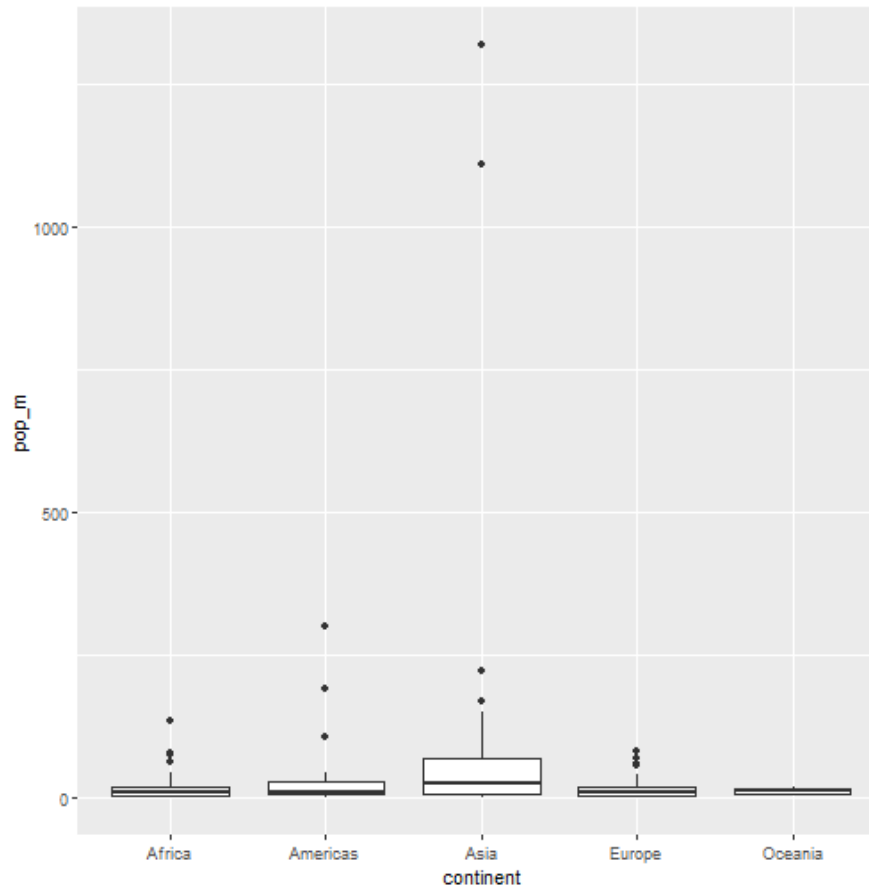

```
# Densidade estimada (grupos)
p <- ggplot(data = gapminder07,
            mapping = aes(x = lifeExp, fill = continent)) +
  geom_density(alpha = 0.3)
p
```



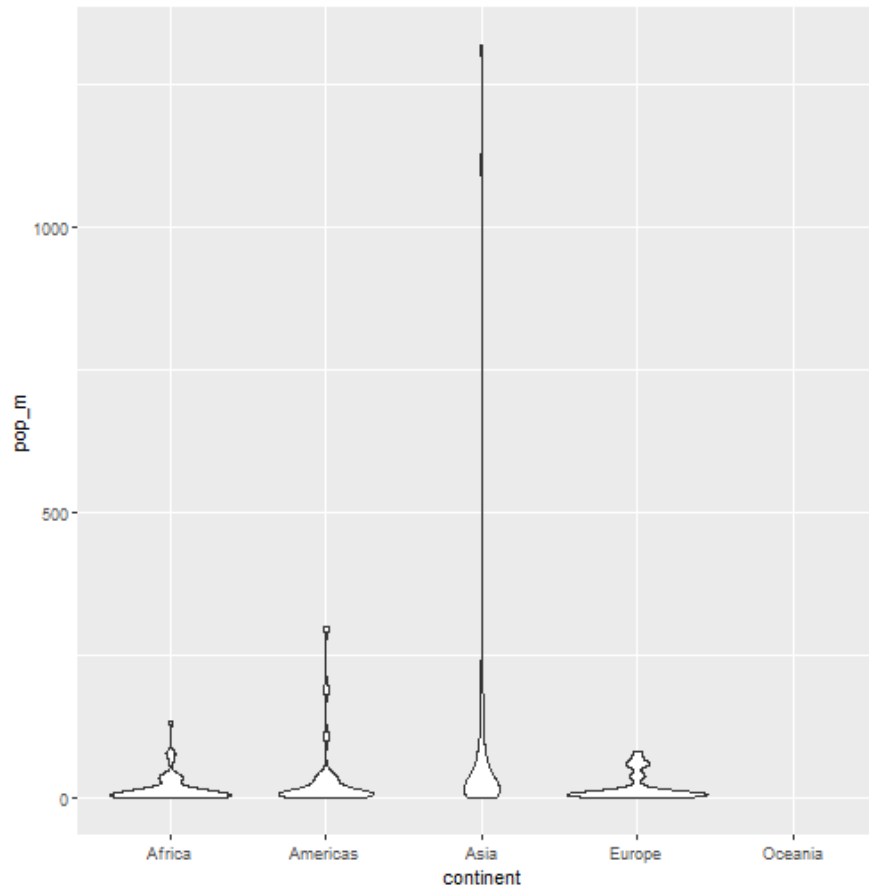
```
# Gráfico de barras  
p <- ggplot(data = gapminder07,  
            mapping = aes(x = continent)) +  
  geom_bar()  
p
```



```
# Boxplot
p <- ggplot(data = gapminder07,
            mapping = aes(x = continent, y = pop_m)) +
  geom_boxplot()
p
```

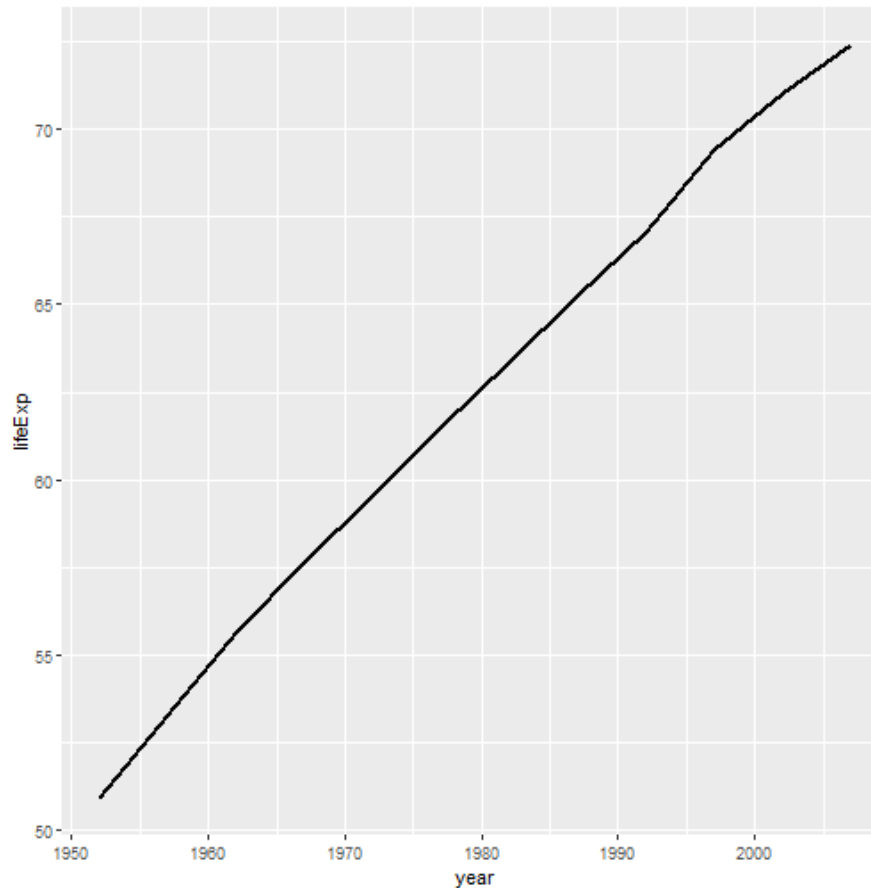


```
# Gráfico de violino
p <- ggplot(data = gapminder07,
            mapping = aes(x = continent, y = pop_m)) +
  geom_violin()
p
```



```
# Gráfico de linha
```

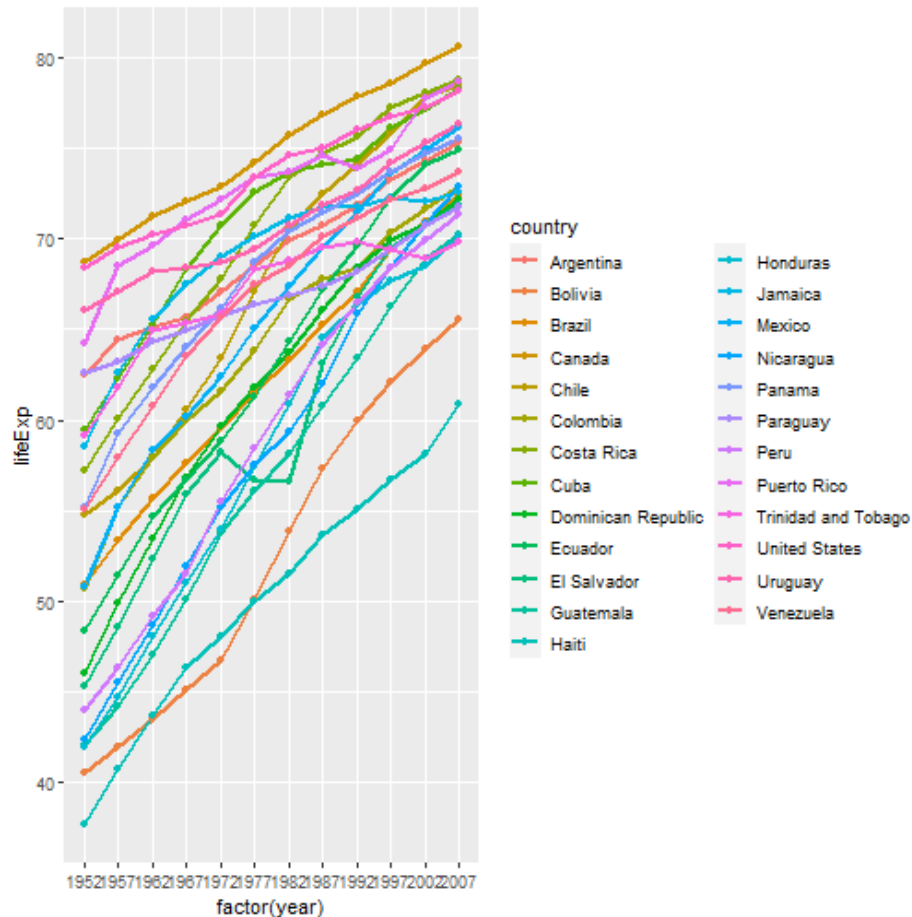
```
p <- ggplot(data = gapminder[which(gapminder$country == "Brazil"),],  
           mapping = aes(x = year, y = lifeExp)) +  
  geom_line(size = 1)  
p
```



```
# Gráfico de linha
```

```
p <- ggplot(data = gapminder[which(gapminder$continent == "Americas"),  
      mapping = aes(x = factor(year), y = lifeExp, group = country),  
      geom_line(size = 1) + geom_point()
```

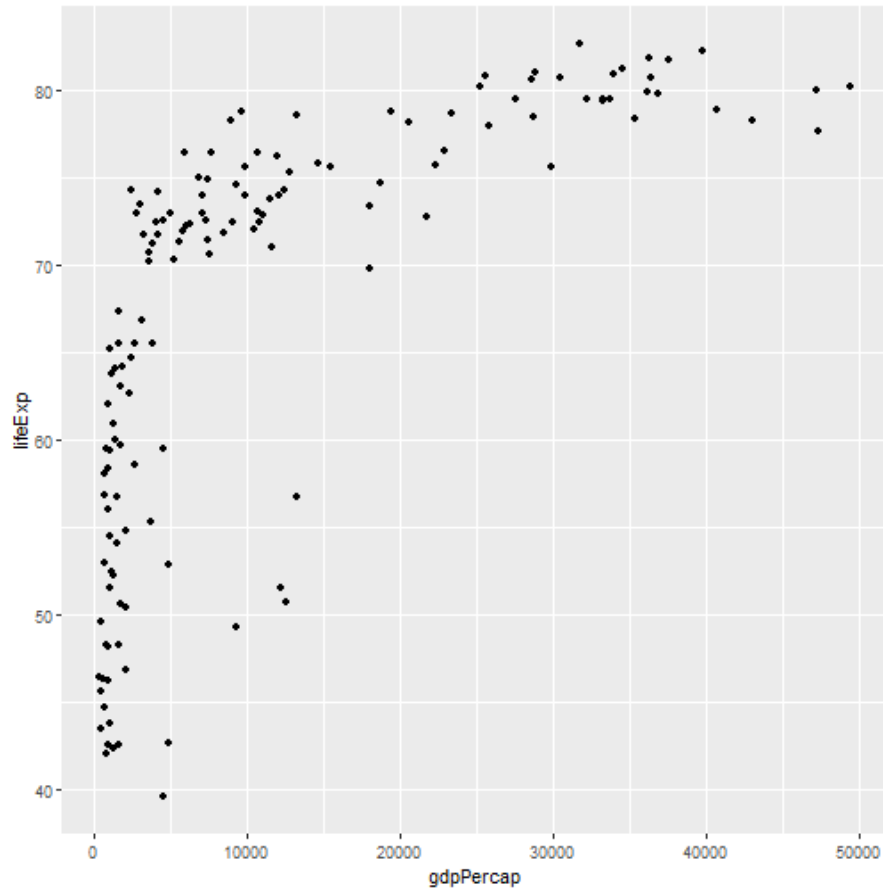
```
p
```



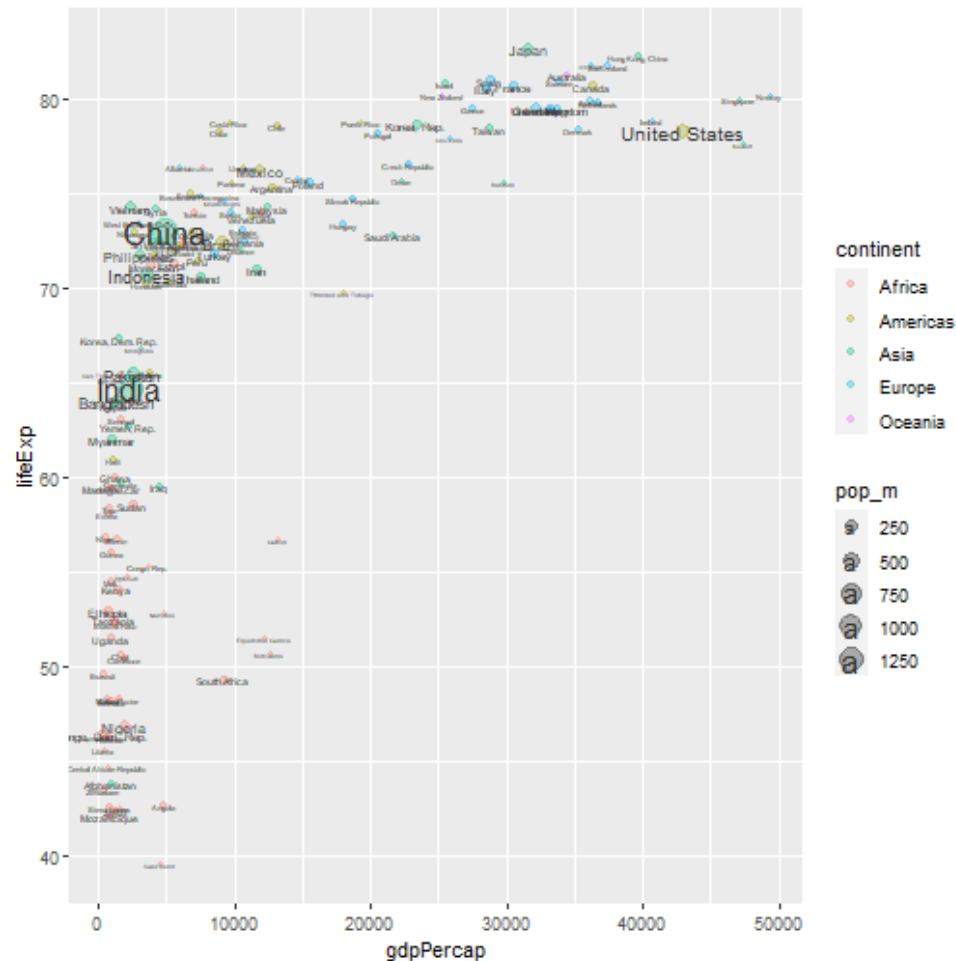
Exercício 1



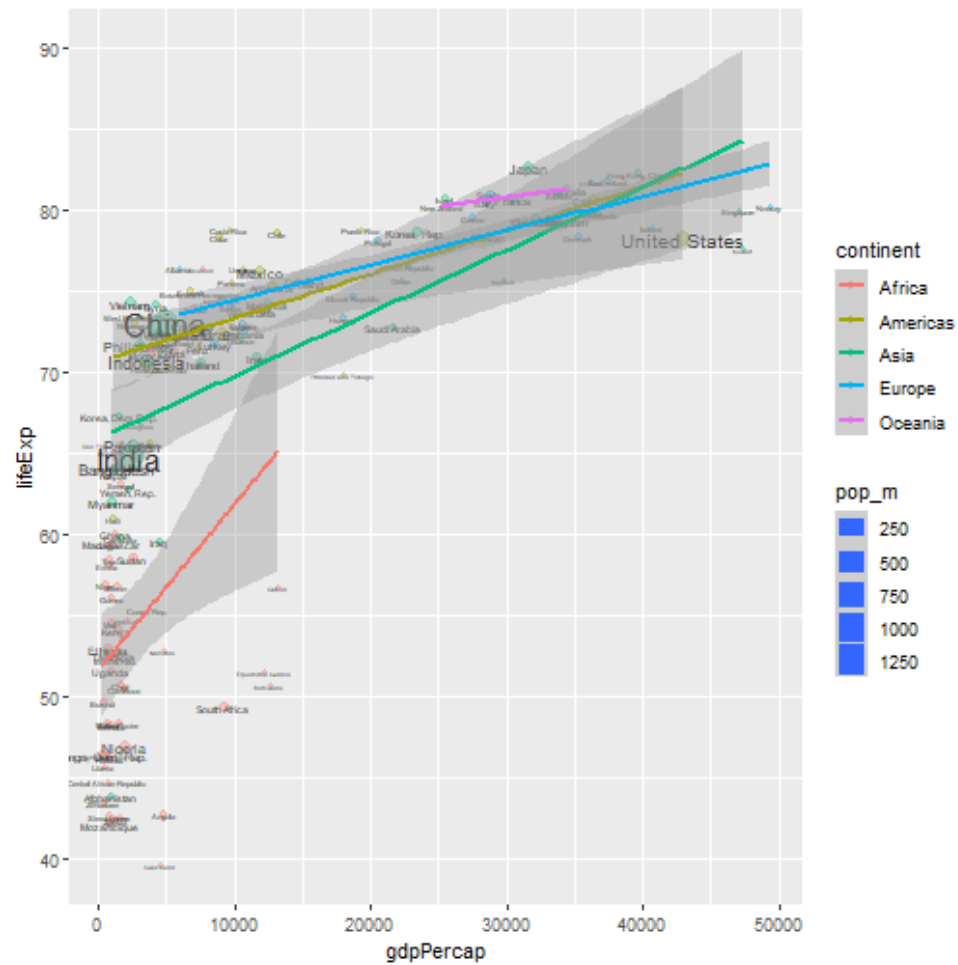
- Acrescente as variáveis continente (continent) e população (pop ou pop_m) ao gráfico abaixo.



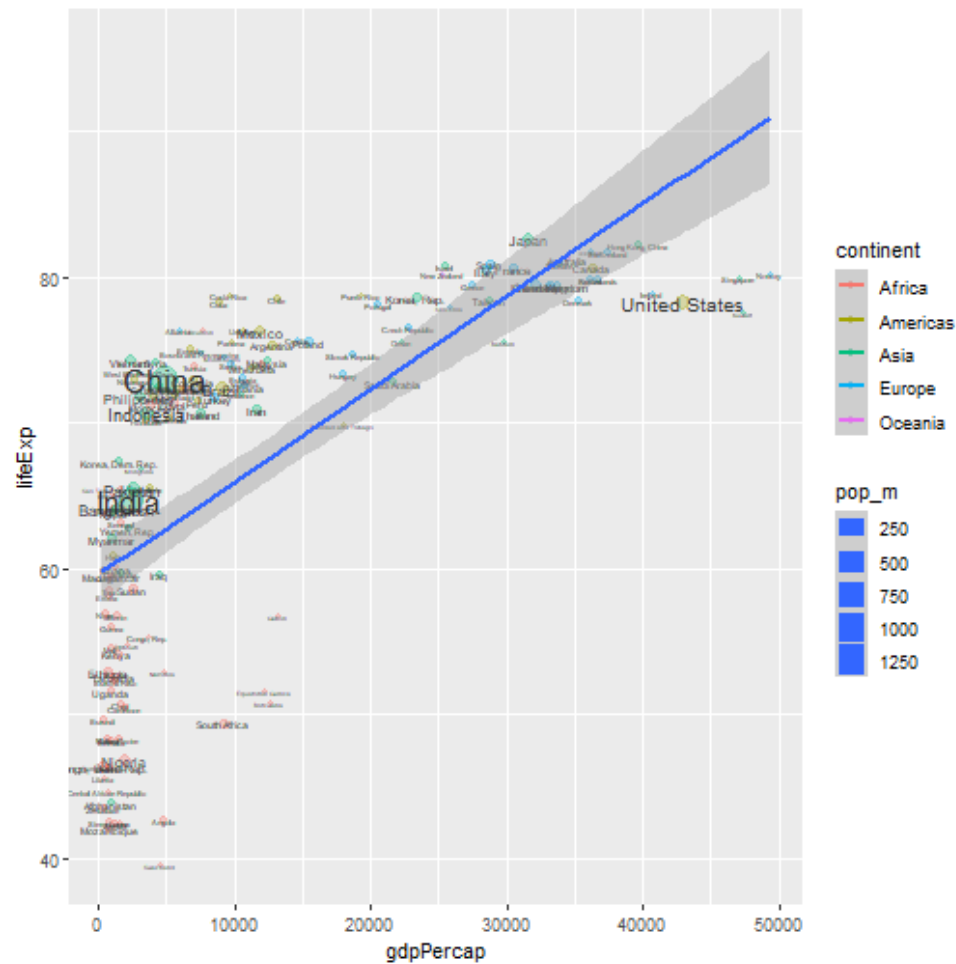

```
p <- ggplot(data = gapminder07,
            mapping = aes(x = gdpPerCap, y = lifeExp, color = continent),
            geom_point(alpha = 0.3) + geom_text(aes(label = country), color = 'black'))
```



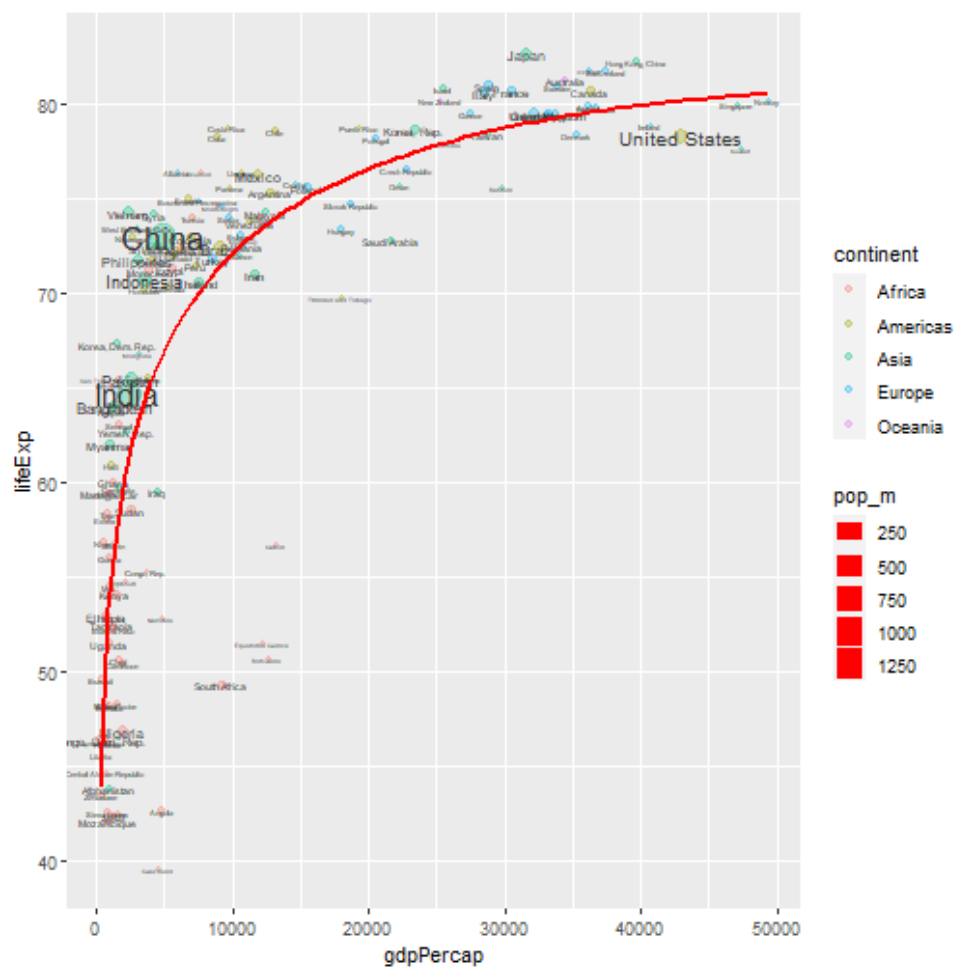
```
p + geom_smooth(method = "lm")
```



```
p + geom_smooth(mapping = aes(x = gdpPercap, y = lifeExp, color = NUI
```



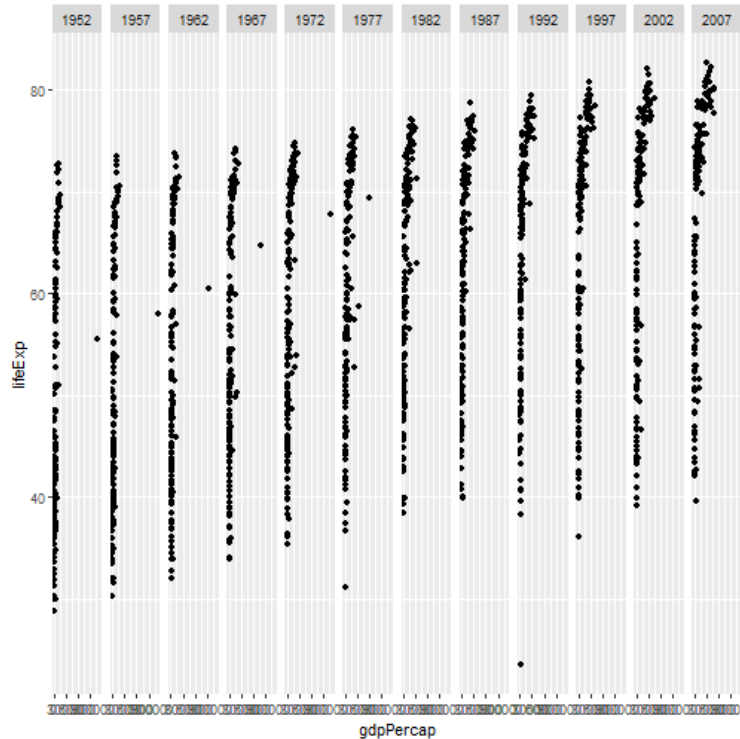
```
p + geom_smooth(mapping = aes(x = gdpPercap, y = lifeExp, color = NUI
```



Camadas: facetas

```
p <- ggplot(data = gapminder,  
            mapping = aes(x = gdpPercap, y = lifeExp)) +  
  geom_point() +  
  facet_grid(. ~ year)
```

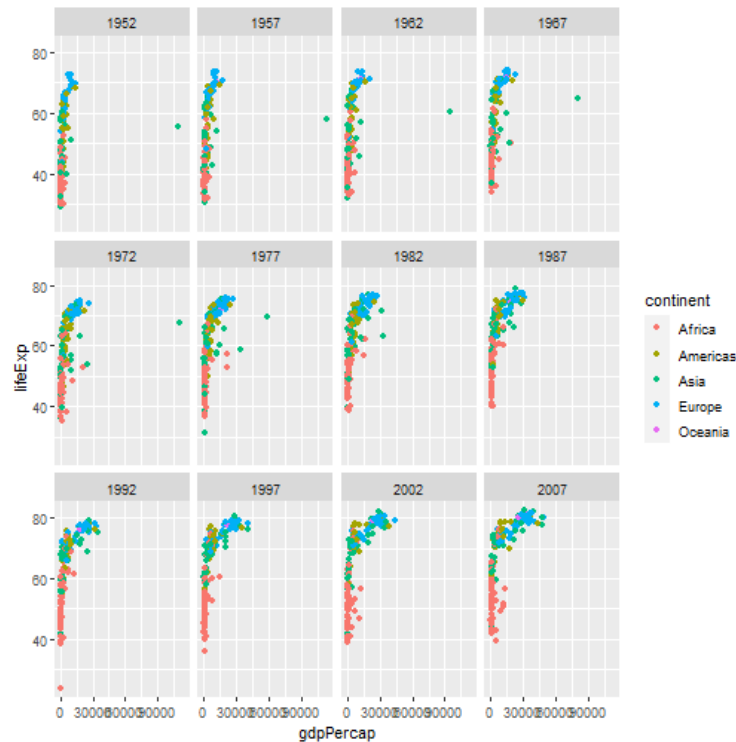
p



Camadas: facetas

```
p <- ggplot(data = gapminder,  
            mapping = aes(x = gdpPercap, y = lifeExp, color = continent),  
            geom_point() +  
            facet_wrap(. ~ year))
```

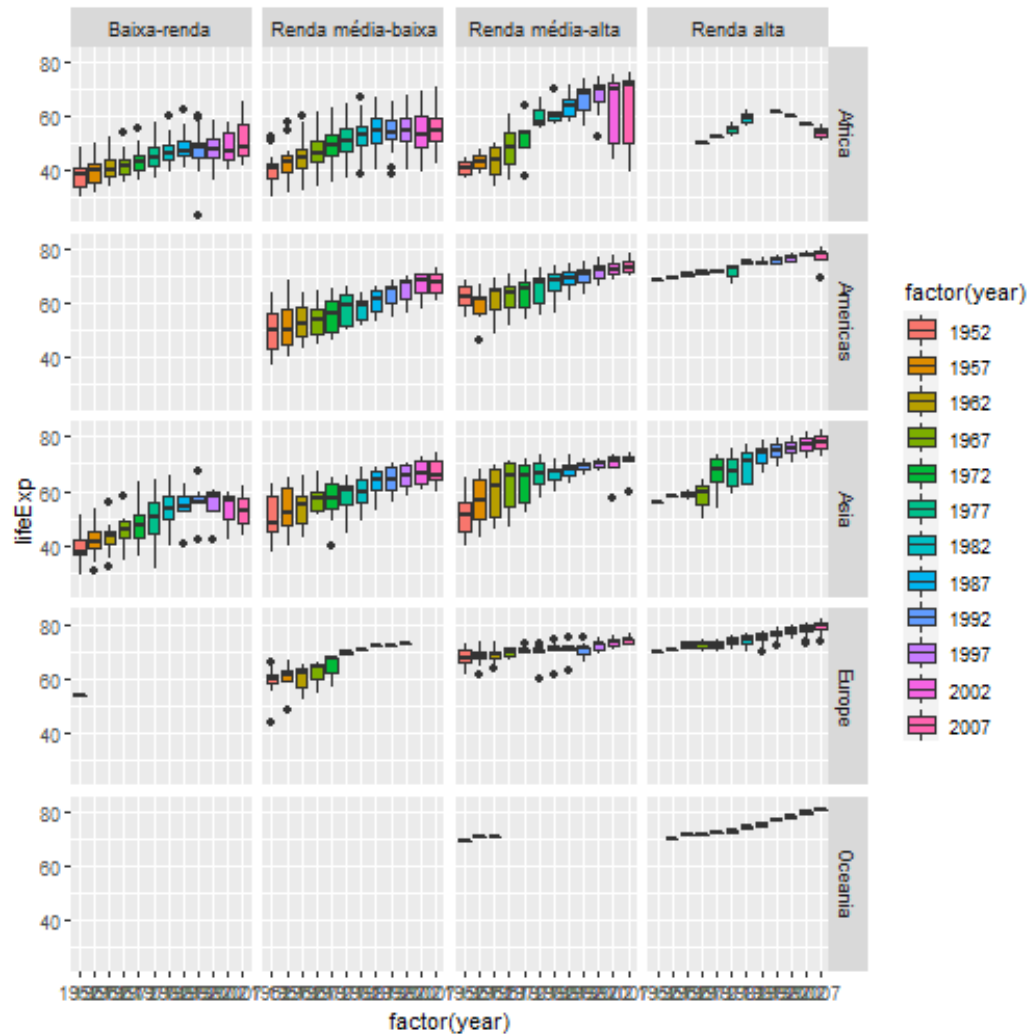
p



Camadas: facetas

```
gapminder$gdpPercap.cat <- cut(gapminder$gdpPercap,  
                               breaks = c(0, 1005, 3955,  
                                           12235, Inf),  
                               labels = c("Baixa-renda",  
                                           "Renda média-baixa",  
                                           "Renda média-alta",  
                                           "Renda alta"))  
  
p <- ggplot(data = gapminder,  
            mapping = aes(x = factor(year),  
                          y = lifeExp, fill = factor(year))) +  
  geom_boxplot() +  
  facet_grid(continent ~ gdpPercap.cat)  
p
```

Camadas: facetas



Eixos e rótulos

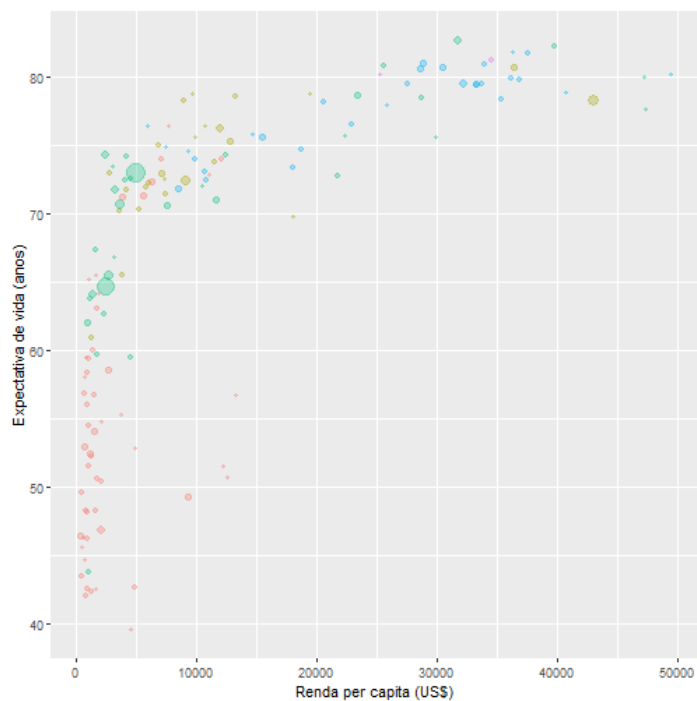
```
p <- ggplot(data = gapminder07,  
            mapping = aes(x = gdpPercap, y = lifeExp,  
                           color = continent, size = pop_m)) +  
  geom_point(alpha = 0.3)  
p
```

Eixos e rótulos

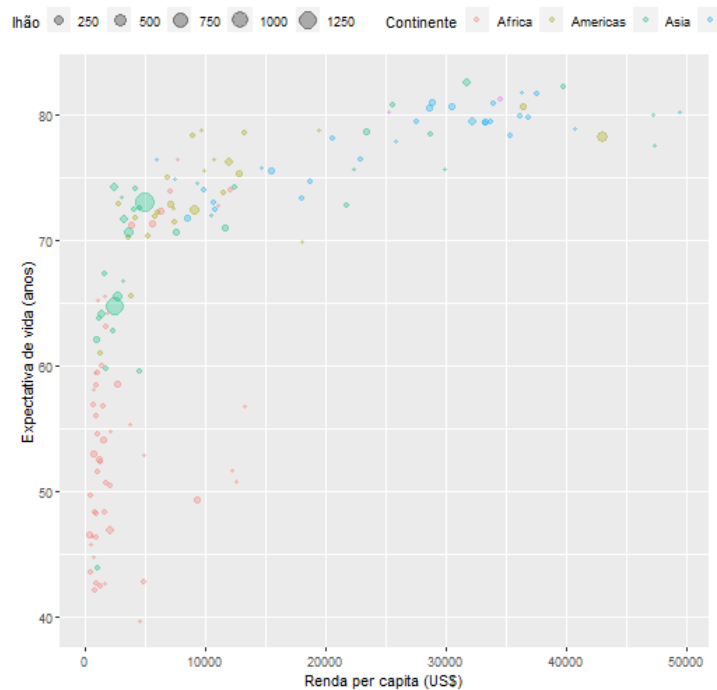
```
p + labs(x = "Renda per capita (US$)",  
        y = "Expectativa de vida (anos)",  
        color = "Continente", size = "População/1 milhão")
```

Legendas e temas

```
p + theme(legend.position = "non
```



```
p + theme(legend.position = "top
```

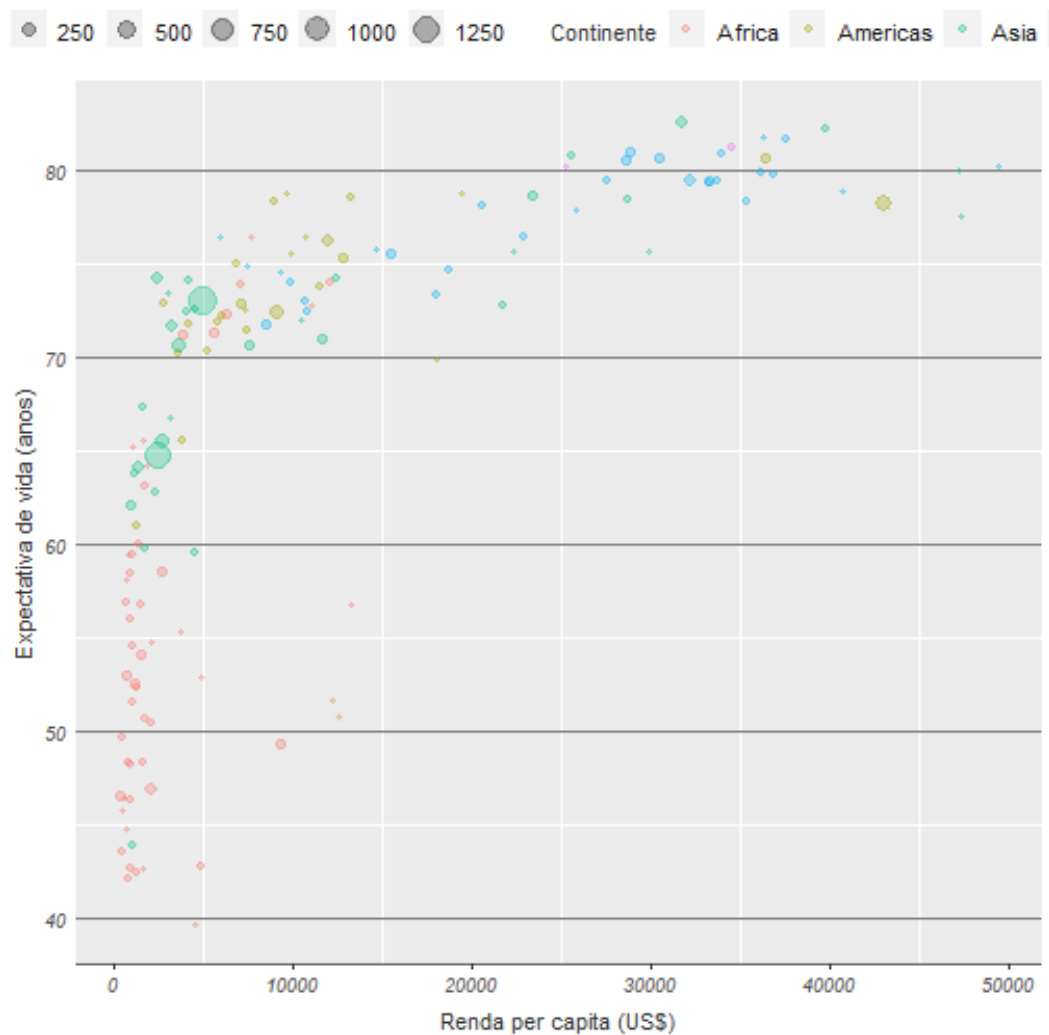


Temas

- Existem diversos outros elementos do tema de um gráfico que podem ser controlados por argumentos da função `theme()`.

```
p + theme(text = element_text(color = "gray20"),
  legend.position = c("top"), # posição da legenda
  legend.direction = "horizontal",
  legend.justification = 0.1, # ponto de ancora para legend.position
  legend.text = element_text(size = 11, color = "gray10"),
  axis.text = element_text(face = "italic"),
  axis.title.x = element_text(vjust = -1),
  axis.title.y = element_text(vjust = 2),
  axis.ticks.y = element_blank(), # element_blank() é como remove
  axis.line = element_line(color = "gray40", size = 0.5),
  axis.line.y = element_blank(),
  panel.grid.major = element_line(color = "gray50", size = 0.5),
  panel.grid.major.x = element_blank())
```

Temas

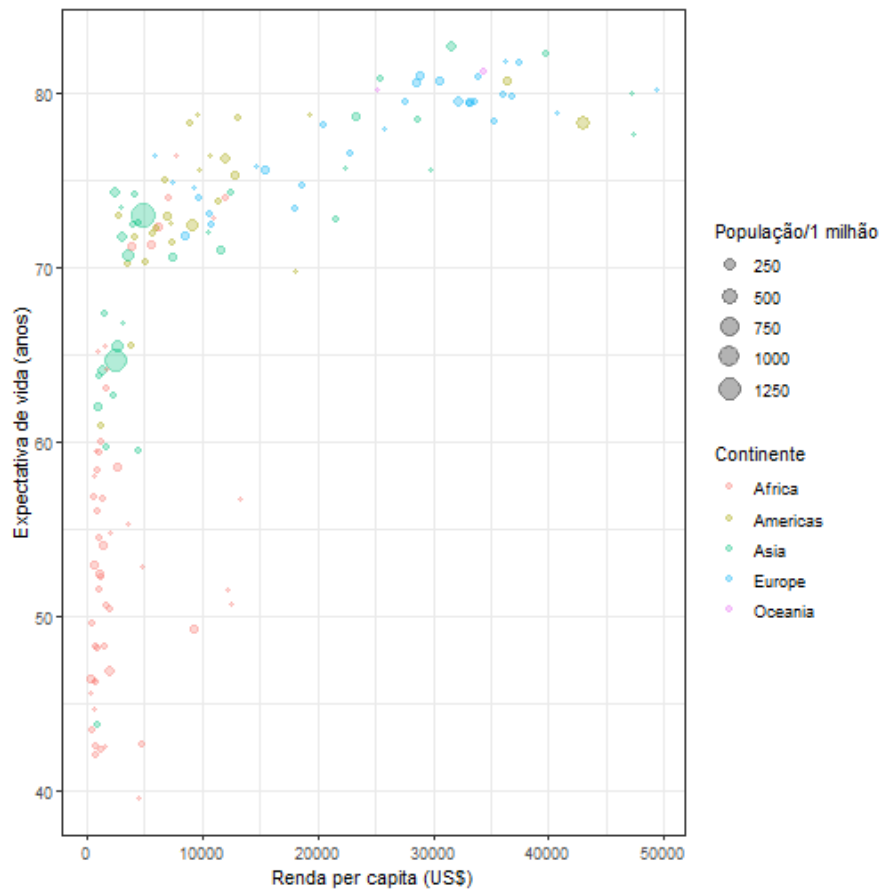


Temas pré-definidos

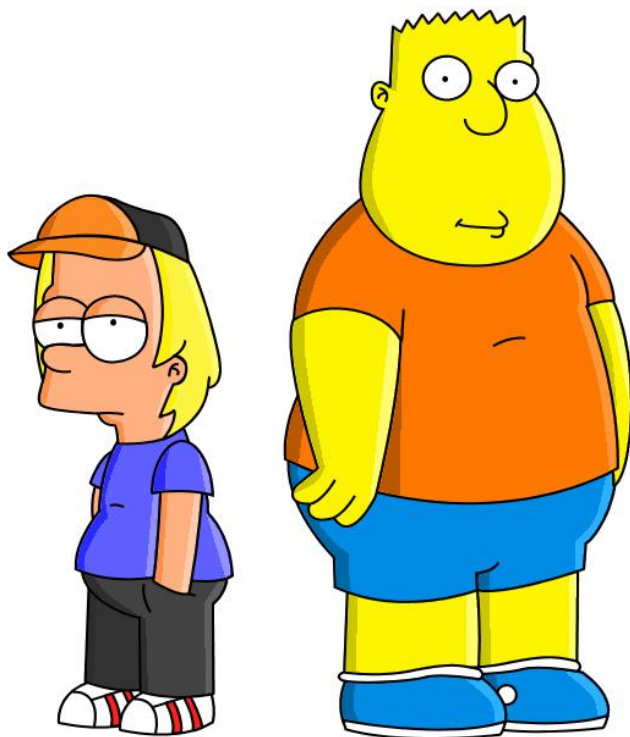
- O pacote **ggplot2** possui alguns temas pré-definidos! Alguns exemplos são:
 - `theme_bw()`: *black and white*
 - `theme_minimal()`: mínimo
 - `theme_classic()`: semelhante aos gráficos do pacote **graphics**
 - `theme_dark()`: escuro

Temas pré-definidos

```
p + theme_bw()
```



Temas pré-definidos II: o pacote ggthemes

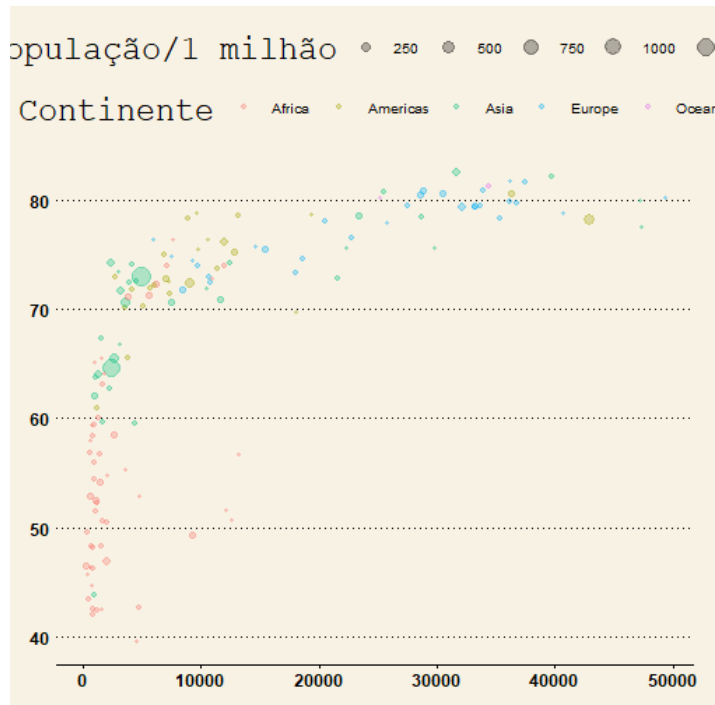


- O pacote **ggthemes** possui uma série de outros temas pré-definidos!

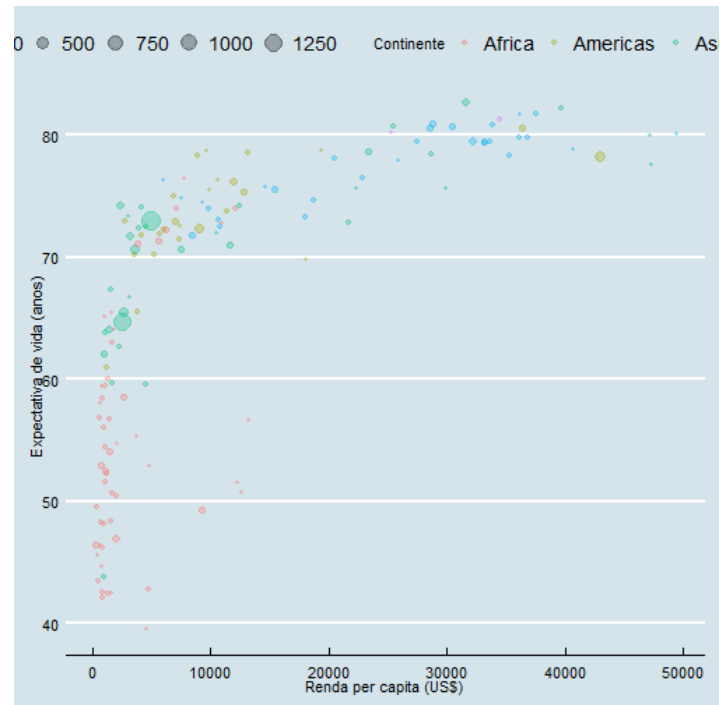
```
# install.packages("ggthemes")  
library(ggthemes)
```


Temas pré-definidos II: ggthemes

```
# Wall Street Journal  
p + theme_wsj()
```



```
# The Economist  
p + theme_economist()
```



ggplot: um resumo esquemático

```
p ← ggplot(data= <data> ,  
           mapping= aes(<aesthetic> = <variable> ,  
                        <aesthetic> = <variable> ,  
                        <...> = <...> )
```

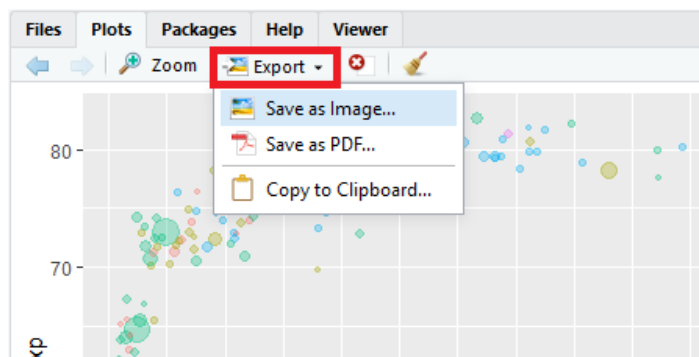
```
p + geom_<type>(<...> ) +  
    scale_<mapping>_<type>(<...> ) +  
    coord_<type>(<...> ) +  
    labs(<...> )
```

Salvando um gráfico

- Você pode salvar o último gráfico realizado com a função `ggsave()`

```
ggsave("MeuPrimeiroGGPLOT.pdf")  
ggsave("MeuPrimeiroGGPLOT.png")  
ggsave("MeuPrimeiroGGPLOT.jpg",  
       width = 4, height = 4)
```

- Ou ainda, utilizando as funcionalidades do RStudio



Considerações finais

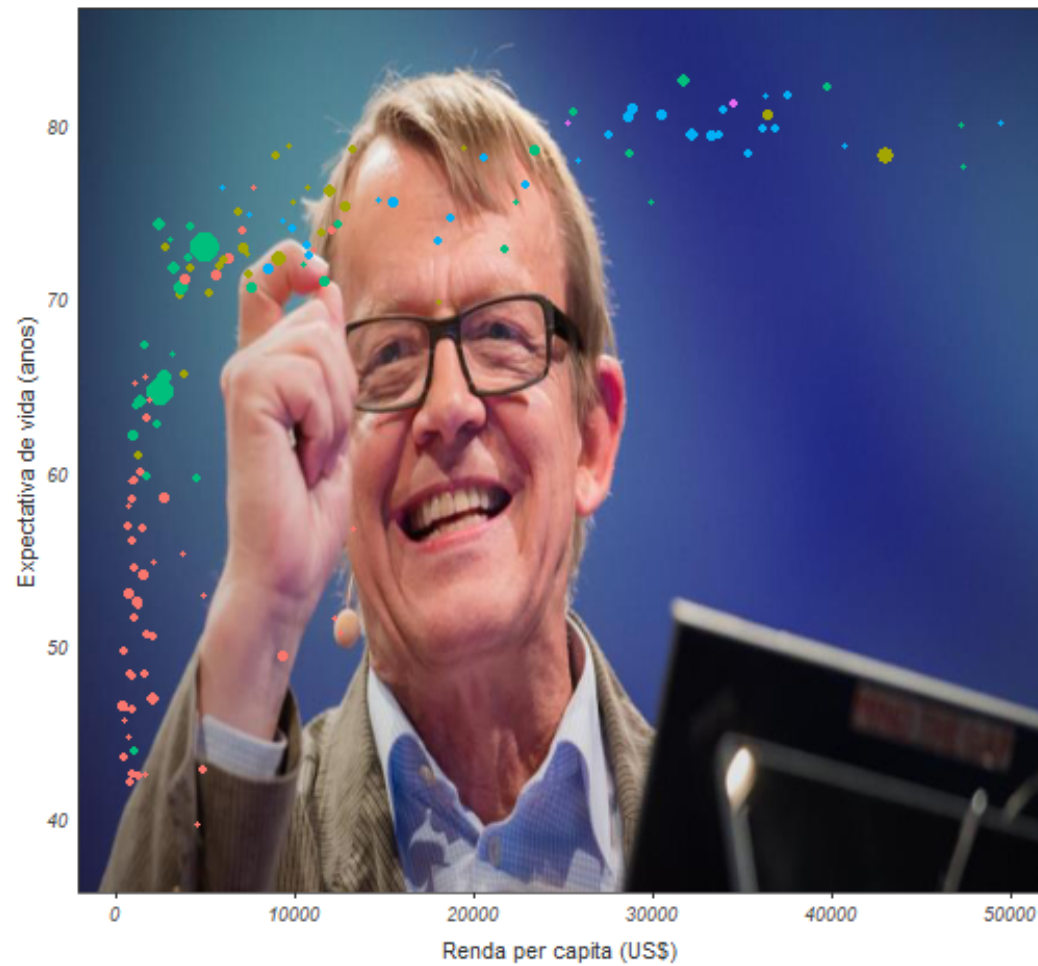
- Existem diversos pacotes que foram criados para complementar o **ggplot2**.
 - **ggthemes**
 - **grid**
 - **magick**
 - **plotly**
 - **jcolors**
- Juntos, estes pacotes tem um potencial quase ilimitado para a geração de gráficos no **R**.

```

# Considerações finais: exemplos
# install.packages("jpeg")
# install.packages("grid")
library(jpeg)
library(grid)
img <- readJPEG("~/PintandoEBordando/ArquivosR/images/hans_rosling.jpg")
# start plotting
p <- ggplot(data = gapminder07,
            mapping = aes(x = gdpPercap, y = lifeExp,
                          color = continent, size = pop_m)) +
  annotation_custom(rasterGrob(img, width = unit(1, "npc"),
                                height = unit(1, "npc")),
                    -Inf, Inf, -Inf, Inf) +
  scale_y_continuous(expand = c(0,0),
                    limits = c(min(gapminder07$lifeExp) * 0.9, max(gapminder07$lifeExp) * 1.1)) +
  geom_point() +
  labs(x = "Renda per capita (US$)",
       y = "Expectativa de vida (anos)",
       color = "Continente", size = "População/1 milhão") +
  theme_bw() +
  theme(text = element_text(color = "gray20"),
        legend.position = c("top"), # posição da legenda
        legend.direction = "horizontal",
        legend.justification = "left", # ponto de ancora para legend.position
        legend.text = element_text(size = 11, color = "gray10"),
        axis.text = element_text(face = "italic"),
        axis.title.x = element_text(vjust = -1),

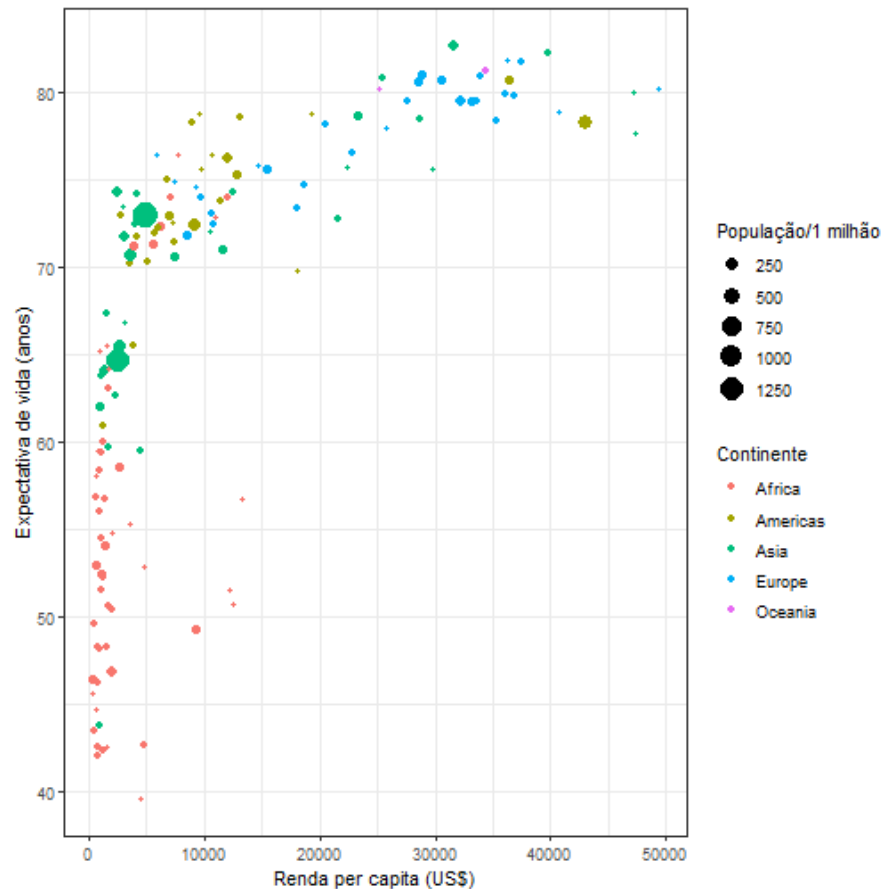
```

● 250 ● 500 ● 750 ● 1000 ● 1250 Continente ● Africa ● Americas ● Asia

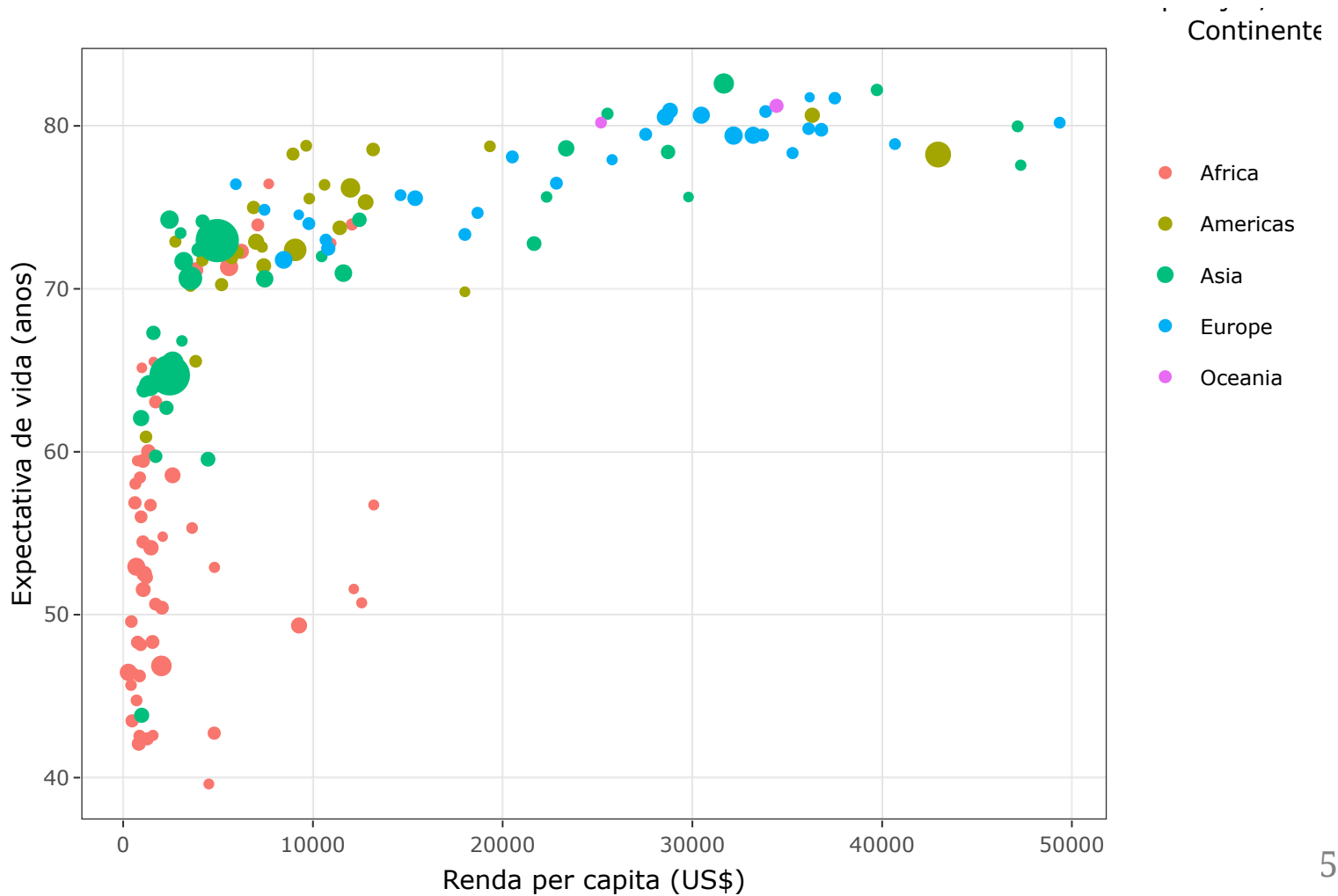


```
# Considerações finais: exemplos - interatividade
```

```
p <- ggplot(data = gapminder07, mapping = aes(x = gdpPercap, y = lifeExp,
                                              color = continent, size = pop_m)) +
  geom_point() + labs(x = "Renda per capita (US$)", y = "Expectativa de vida (anos)",
                    color = "Continente", size = "População/1 milhão") + theme_bw()
p
```



```
# Considerações finais: exemplos - interatividade  
# install.packages("plotly")  
library(plotly)  
ggplotly(p)
```





Perguntas!?!



O que é o rmarkdown¹?



- **R** (códigos) + **Markdown** (linguagem **simples** de marcação para geração de **texto**)
 - **Pandoc** (conversor universal de documentos)

[1] Allaire, J.J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J. e Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.

O fluxo do rmarkdown?



- Quando compilado, o **R Markdown** alimenta o arquivo **.Rmd** para **knitr**, que executa todos os fragmentos de código e cria um novo documento markdown (**.md**) que inclui o código e sua saída.
- O arquivo markdown gerado pelo **knitr** é então processado pelo **pandoc** que é responsável pela criação do formato final.
- Isso pode parecer complicado, mas o **R Markdown** torna extremamente simples encapsulando todo o processamento acima em uma única função de renderização.

Por que rmarkdown?



- Vamos a um exemplo!



Instalando e carregando o rmarkdown

- Instalando o pacote **rmarkdown**

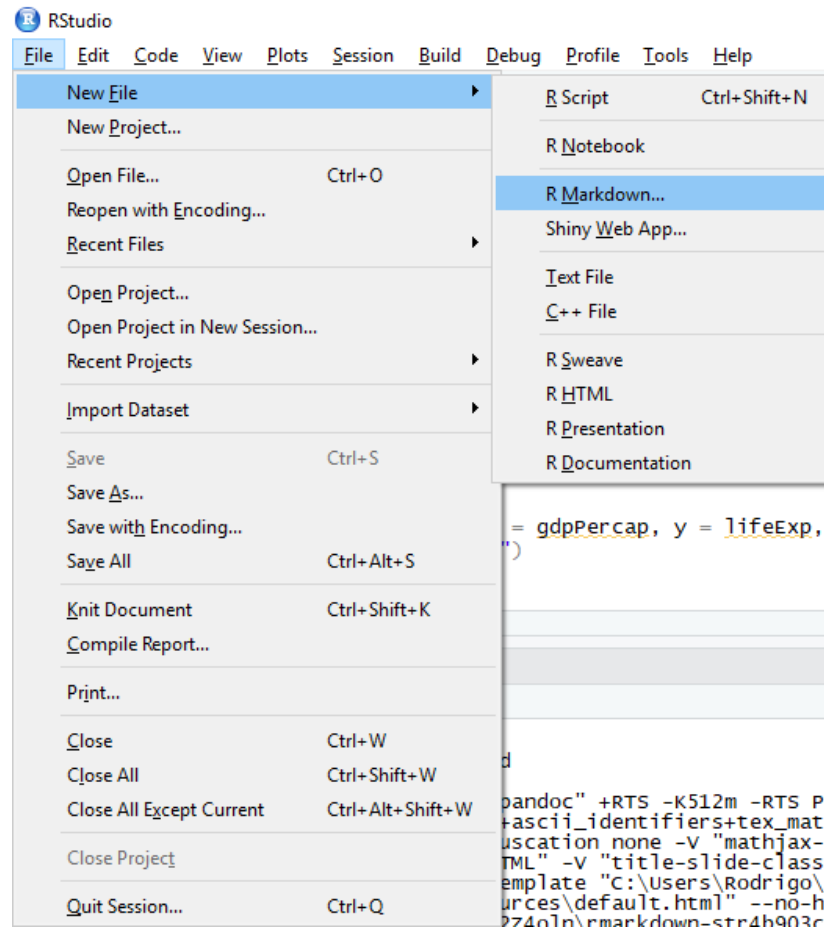
```
install.packages("rmarkdown")
```

- Carregando o pacote **rmarkdown**

```
library(rmarkdown)
```

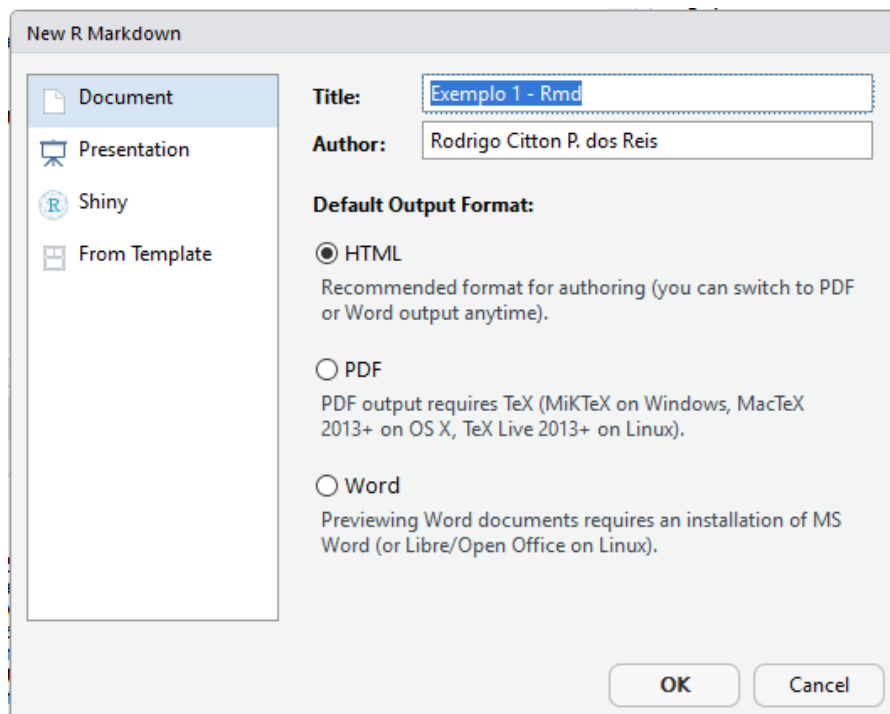
Criando um documento R markdown

- Clique em File -> New File -> R Markdown



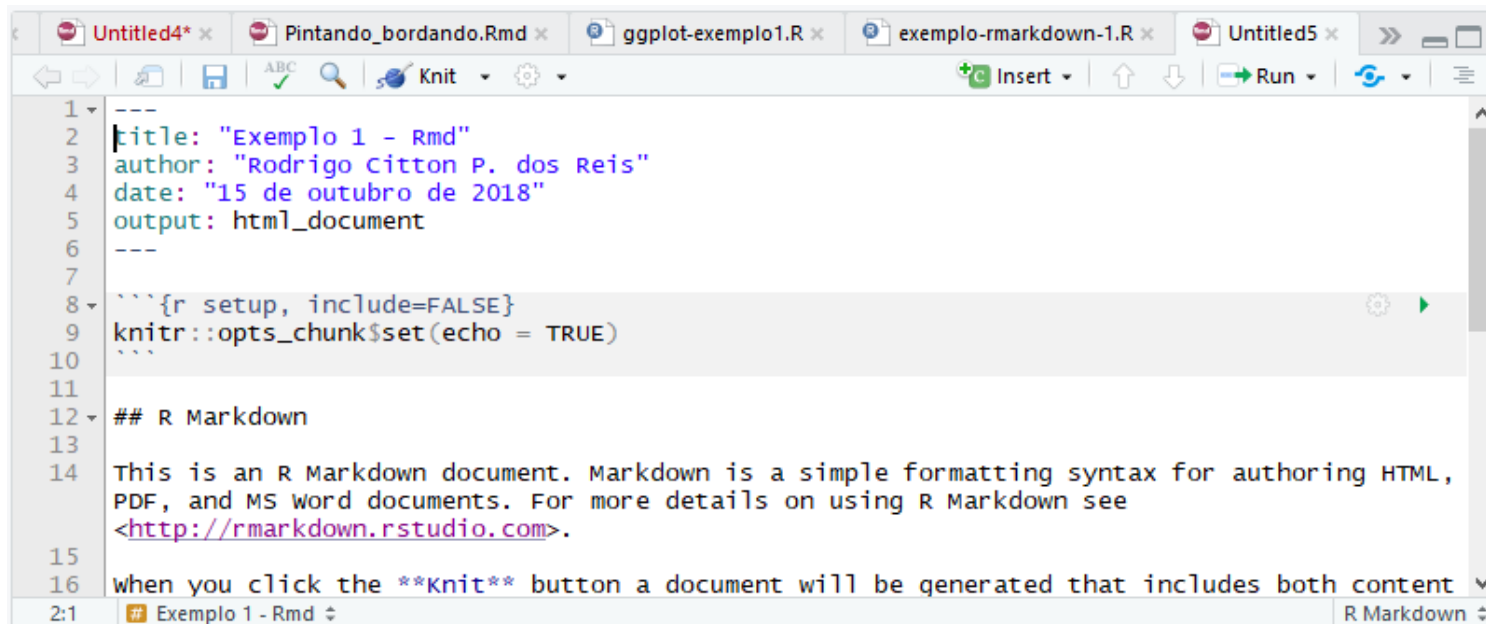
Criando um documento R markdown

- Agora você deve ver uma caixa de diálogo como mostrado abaixo.
- Selecione "Document" no painel à esquerda e preencha o campo de título e autor e clique em "OK".



Criando um documento R markdown

- Agora você deve ter um documento que parece com isso



The screenshot shows the RStudio interface with a file explorer at the top containing several files: 'Untitled4*', 'Pintando_bordando.Rmd', 'ggplot-exemplo1.R', 'exemplo-rmarkdown-1.R', and 'Untitled5'. The main editor window displays an R Markdown document with the following content:

```
1 ---
2 |title: "Exemplo 1 - Rmd"
3 |author: "Rodrigo Citton P. dos Reis"
4 |date: "15 de outubro de 2018"
5 |output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,
15 PDF, and MS word documents. For more details on using R Markdown see
16 <http://rmarkdown.rstudio.com>.
17
18 When you click the **knit** button a document will be generated that includes both content
```

The status bar at the bottom indicates the current position is '2:1' and the file is 'Exemplo 1 - Rmd'.

YAML

- Um conjunto de opções que definem o arquivo de saída
- Este documento gera um arquivo html
- Este documento gera um arquivo word

```
---  
title: "Exemplo 1 - Rmd"  
author: "Rodrigo Citton P. dos R  
date: "15 de outubro de 2018"  
output: html_document  
---
```

```
---  
title: "Exemplo 1 - Rmd"  
author: "Rodrigo Citton P. dos R  
date: "15 de outubro de 2018"  
output: word_document  
---
```

Markdown: sintaxe básica

texto simples

italico

__negrito__

[Datahon](https://www.ufrgs.br/datathon)

Título 1

Título 2

Título 3

texto simples

italico

negrito

Datahon

Título 1

Título 2

Título 3

Markdown: sintaxe básica

- Lista não ordenada
- item 2
 - + sub-item 1
 - + sub-item 2

1. Lista ordenada
2. item 2
 - + sub-item 1
 - + sub-item 2

Cabeçalho tabela	Segundo cabeçalho
Célula tabela	Célula 2
Célula 3	Célula 4

- Lista não ordenada
- item 2
 - sub-item 1
 - sub-item 2

1. Lista ordenada
2. item 2
 - sub-item 1
 - sub-item 2

Cabeçalho tabela	Segundo cabeçalho
Célula tabela	Célula 2
Célula 3	Célula 4

Chunk!



Chunk

- Os códigos em **R** são passados para o arquivo .Rmd por meio de fragmentos de código (*chunk codes*).
- Um exemplo de chunk:

```
```${r}  
summary(gapminder07$pop_m)
```
```

- Saída:

```
summary(gapminder07$pop_m)
```

| ## | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|----|--------|---------|---------|---------|---------|-----------|
| ## | 0.1996 | 4.5080 | 10.5175 | 44.0212 | 31.2100 | 1318.6831 |

Chunk

```
```{r, echo=FALSE}  
summary(gapminder07$pop_m)
```
```

- Saída:

| | | | | | | |
|----|--------|---------|---------|---------|---------|-----------|
| ## | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| ## | 0.1996 | 4.5080 | 10.5175 | 44.0212 | 31.2100 | 1318.6831 |

Chunk

```
```{r, eval=FALSE}  
summary(gapminder07$pop_m)
```
```

- Saída:

```
summary(gapminder07$pop_m)
```

Chunk

```
```{r, echo=FALSE, results='asis'}  
library(knitr)
mod1 <- lm(lifeExp ~ gdpPercap, data = gapminder07)
kable(summary(mod1)$coef, format = "html")
```
```

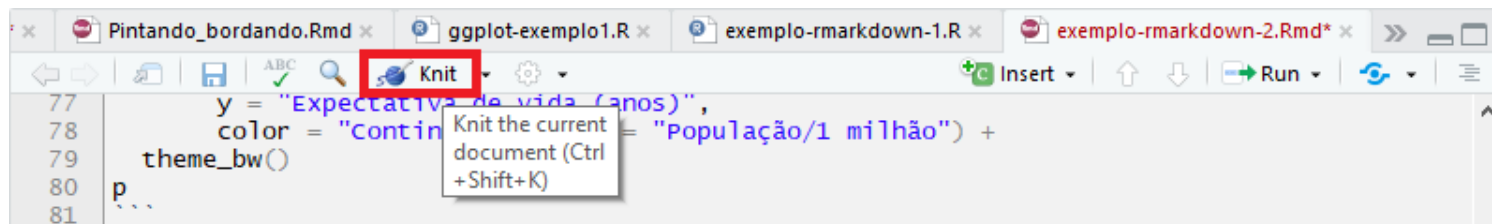
- Saída:

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|-----------------|-------------------|----------------|---------------------|
| (Intercept) | 59.5656501 | 1.0104086 | 58.95204 | 0 |
| gdpPercap | 0.0006371 | 0.0000583 | 10.93340 | 0 |

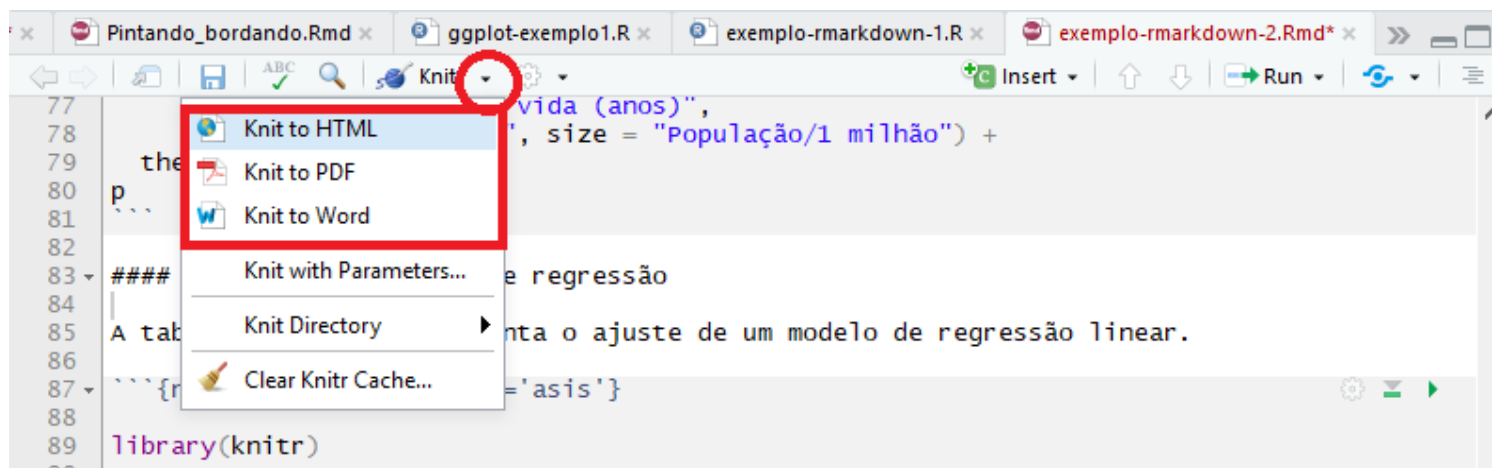
Exercício 1!



1. Crie um arquivo .Rmd.
2. Acrescente texto e a análise do arquivo **exemplo-rmarkdown-1.R**
3. Clique em knit para gerar o arquivo de saída.



- Experimente gerar diferentes formatos de saída.



Por que rmarkdown?



- Reprodutibilidade
- Dinamismo
- Eficiência
- Velocidade

Um último exemplo!





Perguntas!?!

YAY STATISTICS

Muito obrigado!

<https://www.ufrgs.br/datathon>

https://github.com/datathon-ufrgs/Pintando_e_Bordando_no_R

rodrigocpdosreis@gmail.com