

MAT02035 - Modelos para dados correlacionados

Visão geral de modelos lineares para dados longitudinais

Rodrigo Citton P. dos Reis
citton.padilha@ufrgs.br

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DEPARTAMENTO DE ESTATÍSTICA

Porto Alegre, 2019

Breve introdução ao R

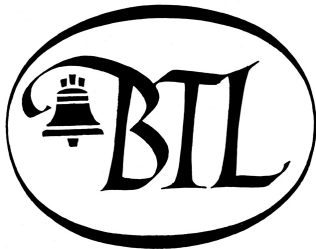
O que é o R?

- ▶ O R é uma linguagem de programação desenvolvida para:
 - ▶ Manipulação de dados;
 - ▶ Análise estatística;
 - ▶ Visualização de dados.
- ▶ O que diferencia o R de outras ferramentas de análise de dados?
 - ▶ Desenvolvido por estatísticos;
 - ▶ É um software livre;
 - ▶ É extensível através de pacotes.



Breve histórico

- ▶ **R** é a versão livre, de código aberto, e gratuita do **S**.
- ▶ Nos anos 1980 o **S** foi desenvolvido nos **Laboratórios Bell**, por **John Chambers**, para análise de dados e geração de gráficos.



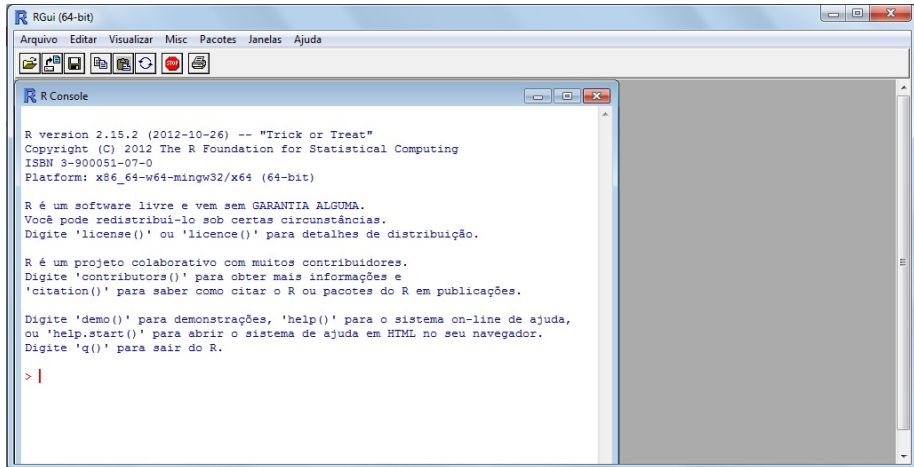
Breve histórico

- ▶ O **R** foi inicialmente escrito no começo dos anos 1990.
 - ▶ **Robert Gentleman** e **Ross Ihaka** no Dep. de Estatística da Universidade de Auckland.
- ▶ O nome **R** se dá em parte por reconhecer a influência do **S** e por ser a inicial dos nomes **Robert** e **Ross**.



- ▶ Desde 1997 possui um grupo de 20 desenvolvedores.
 - ▶ A cada 6 meses uma nova versão é disponibilizada contendo atualizações.

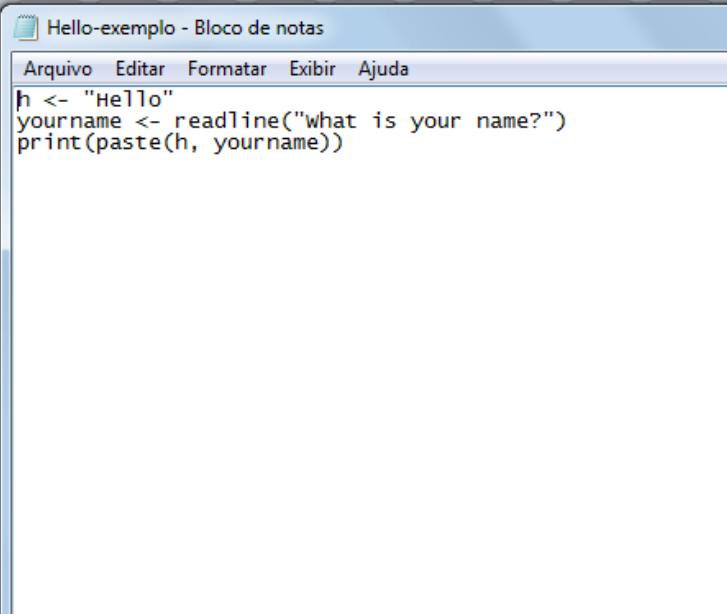
Interface do R



Como trabalhar com o R?

- ▶ Por ser uma linguagem de programação, o **R** realiza suas tarefas através de **funções** e **operadores**.
 - ▶ A criação de **scripts** (rotinas) é **a melhor prática para se trabalhar com o R**.
 - ▶ **OBSERVAÇÃO:** sempre salve seus scripts (em um *pen drive*, dropbox ou e-mail); você pode querer utilizá-los novamente no futuro.
 - ▶ Utilização de editores de texto: **bloco de notas**, **Notepad ++**, **Tinn-R**, etc.
 - ▶ Interfaces de R para usuários: **RStudio**.

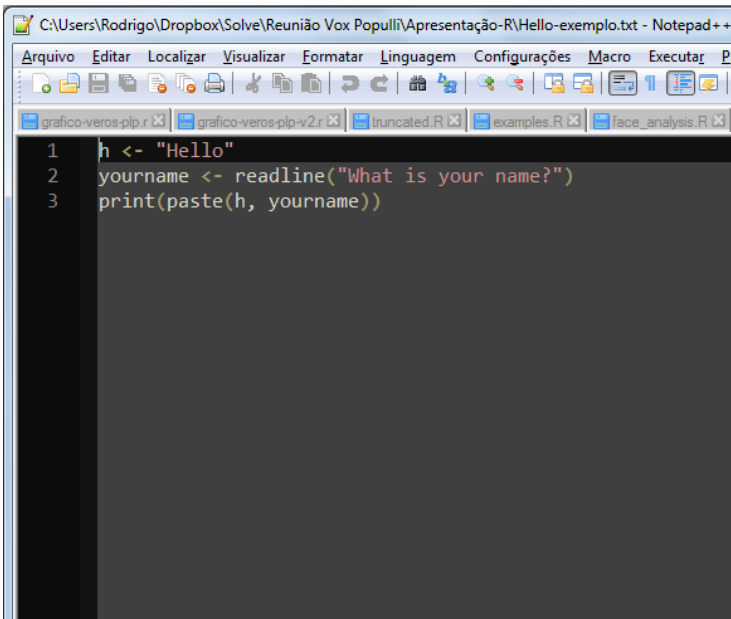
Editores de texto



A screenshot of a text editor window titled "Hello-exemplo - Bloco de notas". The window has a menu bar with the following options: "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The main text area contains the following R code:

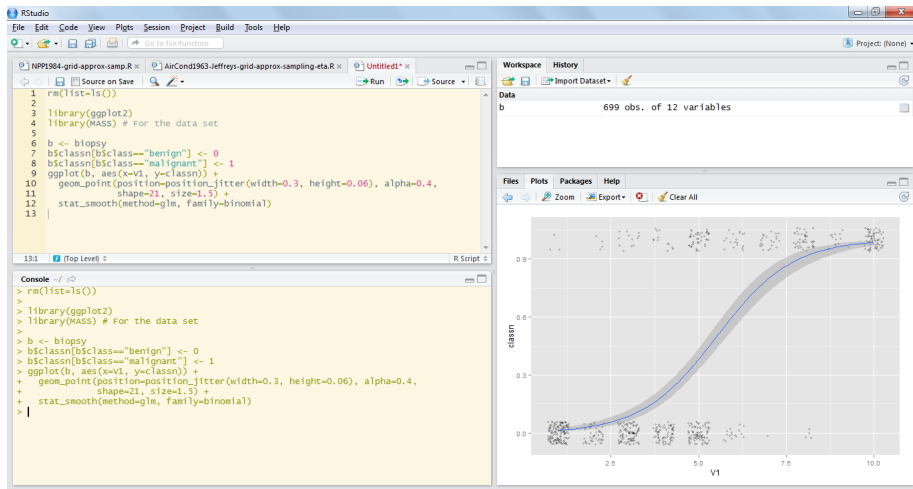
```
h <- "Hello"
yourname <- readline("what is your name?")
print(paste(h, yourname))
```


Editores de texto

A screenshot of a Notepad++ text editor window. The title bar shows the file path: C:\Users\Rodrigo\Dropbox\Solve\Reunião Vox Populi\Apresentação-R\Hello-exemplo.txt - Notepad++. The menu bar includes Arquivo, Editar, Localizar, Visualizar, Formatar, Linguagem, Configurações, Macro, Executar, and Pl. The toolbar contains various icons for file operations and editing. The tab bar shows several open files: grafico-veros-plp.r, grafico-veros-plp-v2.r, truncated.R, examples.R, and face_analysis.R. The main text area contains three lines of R code:

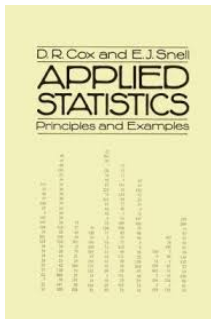
```
1 h <- "Hello"  
2 yourname <- readline("What is your name?")  
3 print(paste(h, yourname))
```

Interface do RStudio



Analisando dados

Fases de análise



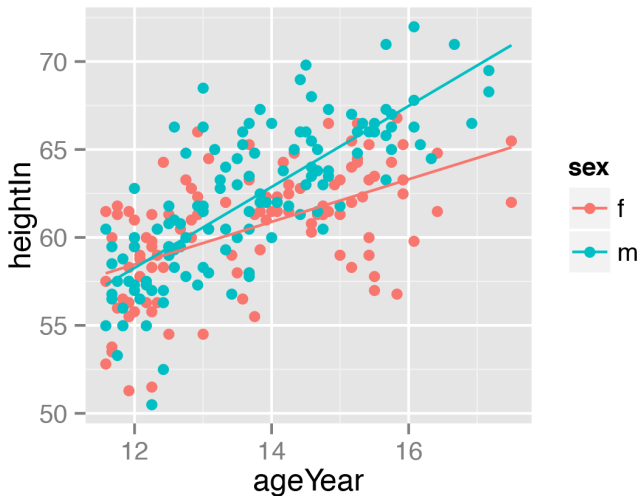
1. **Manipulação inicial** dos dados.
 - ▶ Limpeza dos dados.
 - ▶ Criação, transformação e recodificação de variáveis.
2. **Análise preliminar.**
 - ▶ Conhecimento dos dados, identificação de outliers, investigação preliminar.
3. **Análise definitiva.**
 - ▶ Disponibiliza a base para as conclusões.
4. **Apresentação das conclusões** de forma precisa, concisa e lúcida.

Você pode usar o R para

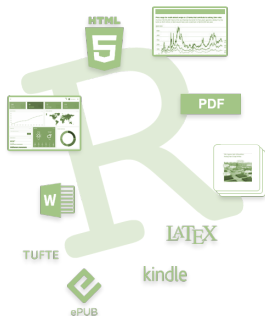
- ▶ **Importação e exportação de dados**
- ▶ **Manipulação de dados:** Transformação e recodificação de variáveis; Aplicação de filtros
- ▶ **Visualização de dados:** Diversos gráficos; Mapas; Gráficos e mapas interativos
- ▶ **Análise de dados:** Análise descritiva; Ajuste de modelos; Técnicas multivariadas; Análise de amostras complexas
- ▶ **Geração de relatórios:** Relatórios nos formatos pdf, HTML, Word, Power Point

Resumindo: você pode usar o R em todas as etapas de uma análise de dados!

Gráficos do R



Comunicação de resultados através do R: R Markdown



1. Produz **documentos dinâmicos** em R.
2. Documentos R Markdown são completamente **reproduzíveis**.
3. R Markdown suporta dezenas de formatos de saída, incluindo **HTML**, **PDF**, **MS Word**, **Beamer**, **dashboards**, **aplicações shiny**, **artigos científicos** e muito mais.

Comunicação de resultados através do R:

CompareGroups

Características dos grupos do estudo

	Total N=6324	Control N=2042	MDN N=2100	MDV N=2182	p-valor
Age	67.0 (6.17)	67.3 (6.28)	66.7 (6.02)	67.0 (6.21)	0.003
Sex: Female	3645 (57.6%)	1230 (60.2%)	1132 (53.9%)	1283 (58.8%)	<0.001
Smoking:					0.444
Never	3892 (61.5%)	1282 (62.8%)	1259 (60.0%)	1351 (61.9%)	
Current	858 (13.6%)	270 (13.2%)	296 (14.1%)	292 (13.4%)	
Former	1574 (24.9%)	490 (24.0%)	545 (26.0%)	539 (24.7%)	
Waist circumference	100 [93.0;107]	101 [94.0;108]	100 [93.0;107]	100 [93.0;107]	0.085
Hormone-replacement therapy	97 (2.80%)	31 (2.64%)	30 (2.81%)	36 (2.95%)	0.898

Comunicação de resultados através do R: stargazer

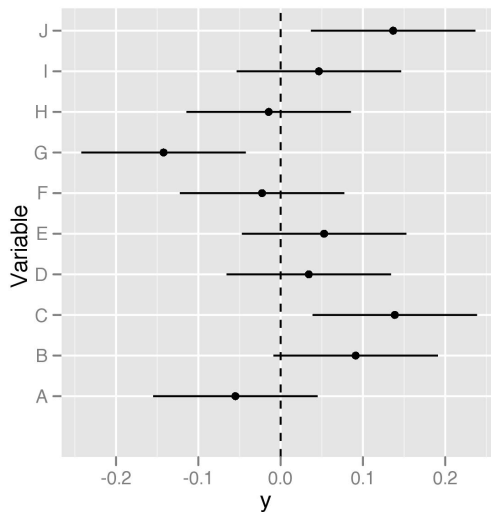
Estimativas dos efeitos fixos dos modelos simples.

	Variável resposta				
	Média de cinza				
	(1)	(2)	(3)	(4)	(5)
time1	4.190** (0.364, 8.016)	4.183** (0.355, 8.011)	4.190** (0.363, 8.017)	4.199** (0.372, 8.026)	4.191** (0.364, 8.019)
time2	9.155*** (4.789, 13.521)	9.138*** (4.768, 13.508)	9.161*** (4.791, 13.532)	9.081*** (4.712, 13.450)	9.178*** (4.808, 13.549)
forca.de.mordida	0.096*** (0.041, 0.150)				
idade		-1.241** (-2.376, -0.105)			
sexoFeminino			-6.492 (-27.707, 14.722)		
provisorioSim				16.420* (-0.556, 33.396)	
archMandibula					9.322 (-6.396, 25.040)
Constant	51.023*** (24.326, 77.721)	172.271*** (101.403, 243.139)	100.214*** (81.940, 118.489)	90.139*** (79.631, 100.646)	90.109*** (76.930, 103.287)
Observations	319	319	319	319	319

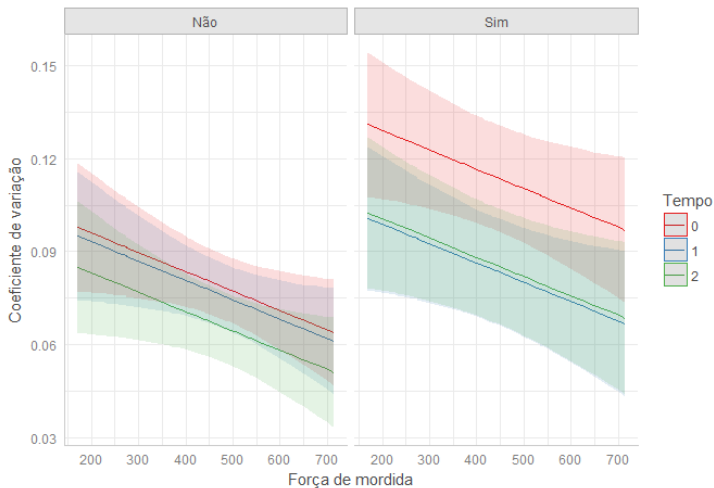
Note:

$p < 0.1$; $p < 0.05$; $p < 0.01$

Comunicação de resultados através do R



Comunicação de resultados através do R



Comunicação de resultados através do R: Shiny

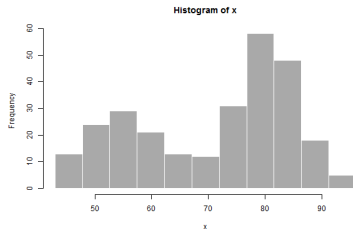
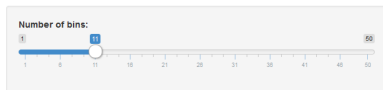
- ▶ Shiny é um pacote do R que torna mais fácil a construção de **aplicações web interativas** (apps) diretamente do R.
 - ▶ Permite a criação e compartilhamento de aplicativos.
 - ▶ Espera **nenhum conhecimento** de tecnologias web como HTML, CSS ou JavaScript (mas você pode aproveitá-las, caso as conheça)
 - ▶ Um aplicativo Shiny consiste em duas partes: uma **interface de usuário** (UI) e um **servidor**.

Shiny

```
# Run the application
shinyApp(ui = ui, server = server)
```

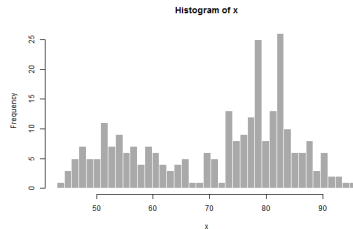
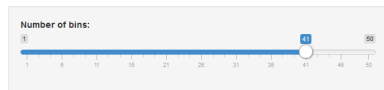
~/Stats4Good/shinyAppEx1/AppEx1 - Shiny
http://127.0.0.1:3589 | Open in Browser | Publish

Old Faithful Geyser Data



~/Stats4Good/shinyAppEx1/AppEx1 - Shiny
http://127.0.0.1:3589 | Open in Browser | Publish

Old Faithful Geyser Data



Baixando e instalando o R

Para instalação do R acesse o site <https://www.r-project.org/>:

1. Em **Download** clique em CRAN.
 - ▶ O **CRAN** (*The Comprehensive R Archive Network*) é uma rede de servidores ftp e web em todo o mundo que armazena versões de código e documentação idênticas e atualizadas para o R.
2. Escolha um repositório de sua preferência, por exemplo, Universidade Federal do Paraná (<http://cran-r.c3sl.ufpr.br/>).
3. Em **Download and Install R** clique no link adequado para o seu sistema operacional (no caso de Windows, clique no link **Download R for Windows**).
4. Clique no link **base** (no caso do sistema operacional ser Windows).
5. Finalmente clique no link para baixar o arquivo executável (a versão mais atual **Download R 3.6.1 for Windows**).

Após baixar o arquivo executável, abra-o e siga as etapas de instalação conforme as configurações padrões.

Baixando e instalando o RStudio

Para instalação do RStudio acesse o site

<https://www.rstudio.com/products/rstudio/download/>.

- ▶ Em **Installers for Supported Platforms** baixe a versão mais recente do instalador do RStudio de acordo com o seu sistema operacional (no caso de Windows clique no link **RStudio 1.2.1335 - Windows Vista/7/8/10**).

Pacotes

- ▶ Assim como a maioria dos softwares estatísticos, o R possui os seus “módulos”, mais conhecidos como **pacotes** do R.
- ▶ **Pacote:** é uma coleção de funções do R; os pacotes também são gratuitos e disponibilizados no **CRAN**.
 - ▶ Outros repositórios também são utilizados, como por exemplo: Github, Bitbucket, Bioconductor, entre outros.
 - ▶ Você também pode fazer o seu pacote do R (com as funções utilizadas na disciplina, por exemplo).
- ▶ Um pacote inclui: **funções** do R, **conjuntos de dados** (utilizados em exemplos das funções), arquivo com **ajuda (*help*)**, e uma **descrição** do pacote.
- ▶ Atualmente, o repositório oficial do R possui mais de 14.900 pacotes disponíveis.
- ▶ As funcionalidades do R, podem ser ampliadas carregando estes pacotes, tornando-o um software muito poderoso, capaz de realizar inúmeras tarefas.

Pacotes

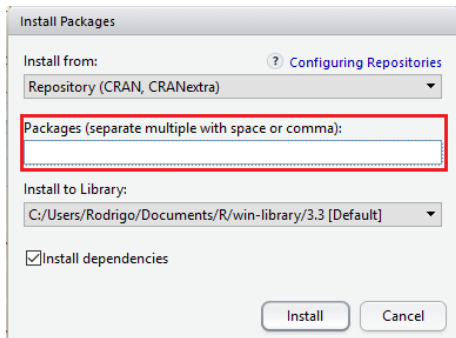
- ▶ Alguns exemplos destas tarefas e alguns destes pacotes são listados abaixo:
 - ▶ **Importação e exportação de dados:** readr, readxl, haven, foreign
 - ▶ **Manipulação de dados:** tidyr, dplyr, stringr
 - ▶ **Descrição e visualização de dados:** compareGroups, ggplot2, GGally
 - ▶ **Modelagem de dados:** lme4, nlme, geepack, geeglm
 - ▶ **Comunicação estatística:** stargazer, ggeffects, knitr, rmarkdown, officer

Instalando pacotes

- ▶ Para **instalação de um pacote**, basta um simples comando.

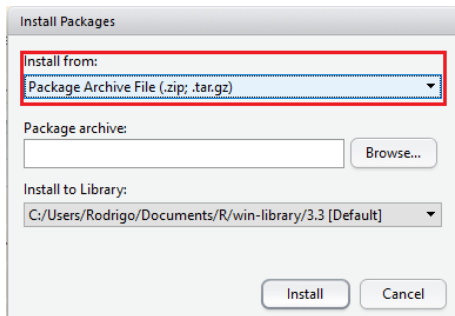
```
install.packages("tidyverse")
```

- ▶ Além da opção de comando, também podemos instalar pacotes utilizando o menu **Tools** do RStudio, opção **Install packages ...** e preenchendo com o(s) nome(s) do(s) pacote(s):



Instalando pacotes

- ▶ Outra opção é instalar o pacote a partir de seu arquivos fonte (**.zip** ou **.tar.gz**):
 - ▶ Para isso, obtenha o arquivo fonte do pacote (geralmente através do **CRAN**) e no menu **Tools** do RStudio, opção **Install packages ...** em **Install from** escolha a seguinte opção:



Instalando pacotes

- ▶ Após a instalação do pacote, temos que **carregar o pacote** para nossa área de trabalho para podermos usufruir de suas funções.

```
library("tidyverse")  
require("tidyverse")
```

Obtendo ajuda no R

- ▶ Para conhecer quais as funções disponíveis no pacote, faça:

```
help(package = "tidyverse")
```

- ▶ Para pedir ajuda de uma determinada função:

```
?glm  
help("glm")
```

- ▶ Obtendo ajuda na internet:

```
help.search("t.test")
```

Obtendo ajuda no R

- ▶ Procurando por alguma função, mas esqueci o nome:

```
apropos("lm")
```

- ▶ Para todas as outras dúvidas existe o **Google!**
- ▶ Ver também <http://www.r-bloggers.com/> e <https://rstudio.cloud/>
- ▶ Para algumas demonstrações da capacidade gráfica do R:

```
demo(graphics)  
demo(persp)  
demo(Hershey)  
demo(plotmath)
```

Métodos de análise descritiva

Carregando os dados

```
# -----  
# Carregando pacotes do R  
library(here)  
library(haven)  
library(tidyr)  
library(ggplot2)  
# -----  
# Carregando o arquivo de dados  
here::here("data", "tlc.dta")
```

```
## [1] "C:/Users/Rodrigo/Documents/UFRGS/Disciplinas/2019-02/I
```

```
chumbo <- read_dta(  
  file = here::here("data", "tlc.dta"))
```

Carregando os dados

```
chumbo
```

```
## # A tibble: 100 x 6
##       id      trt    y0    y1    y4    y6
##   <dbl> <dbl+lbl> <dbl> <dbl> <dbl> <dbl>
## 1     1  1 0 [Placebo]  30.8 26.9  25.8  23.8
## 2     2  2 1 [Succimer]  26.5 14.8  19.5   21
## 3     3  3 1 [Succimer]  25.8 23    19.1  23.2
## 4     4  4 0 [Placebo]  24.7 24.5  22    22.5
## 5     5  5 1 [Succimer]  20.4  2.80  3.20  9.40
## 6     6  6 1 [Succimer]  20.4  5.40  4.5   11.9
## 7     7  7 0 [Placebo]  28.6 20.8  19.2  18.4
## 8     8  8 0 [Placebo]  33.7 31.6  28.5  25.1
## 9     9  9 0 [Placebo]  19.7 14.9  15.3  14.7
## 10    10 0 [Placebo]  31.1 31.2  29.2  30.1
## # ... with 90 more rows
```


Transformando os dados

De “largo” para “longo”

```
chumbo.longo <- gather(data = chumbo,  
                        key = "tempo",  
                        value = "chumbo", -id, -trt)
```

```
chumbo.longo
```

```
## # A tibble: 400 x 4
```

```
##       id          trt tempo chumbo  
##   <dbl>    <dbl+lbl> <chr>  <dbl>  
## 1     1 1 0 [Placebo] y0      30.8  
## 2     2 2 1 [Succimer] y0      26.5  
## 3     3 3 1 [Succimer] y0      25.8  
## 4     4 4 0 [Placebo] y0      24.7  
## 5     5 5 1 [Succimer] y0      20.4
```

Transformando os dados

```
##      6      6 1 [Succimer] y0      20.4
##      7      7 0 [Placebo]  y0      28.6
##      8      8 0 [Placebo]  y0      33.7
##      9      9 0 [Placebo]  y0      19.7
##     10     10 0 [Placebo]  y0      31.1
## # ... with 390 more rows
```

```
chumbo.longo$tempo <- as.numeric(
  as.character(
    factor(chumbo.longo$tempo,
           labels = c(1, 2, 4, 6))))
```



```
chumbo.longo$strt <- factor(chumbo.longo$strt,
                             labels = c("Placebo",
                                         "Succimer"))
```



```
chumbo.longo
```

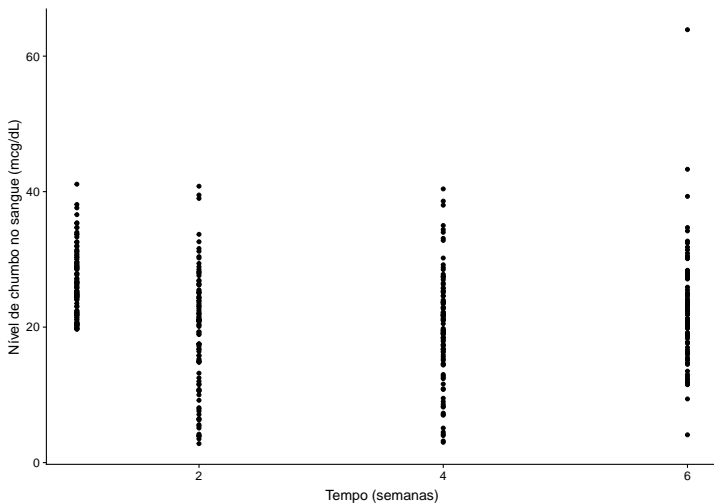
Transformando os dados

```
## # A tibble: 400 x 4
##       id trt      tempo chumbo
##   <dbl> <fct>    <dbl>   <dbl>
## 1     1   1 Placebo      1    30.8
## 2     2   2 Succimer      1    26.5
## 3     3   3 Succimer      1    25.8
## 4     4   4 Placebo      1    24.7
## 5     5   5 Succimer      1    20.4
## 6     6   6 Succimer      1    20.4
## 7     7   7 Placebo      1    28.6
## 8     8   8 Placebo      1    33.7
## 9     9   9 Placebo      1    19.7
## 10    10  10 Placebo      1    31.1
## # ... with 390 more rows
```

Time plot (diagrama de dispersão)

```
p <- ggplot(data = chumbo.longo,  
            mapping = aes(x = tempo, y = chumbo)) +  
  geom_point() +  
  labs(x = "Tempo (semanas)",  
       y = "Nível de chumbo no sangue (mcg/dL)")  
p
```

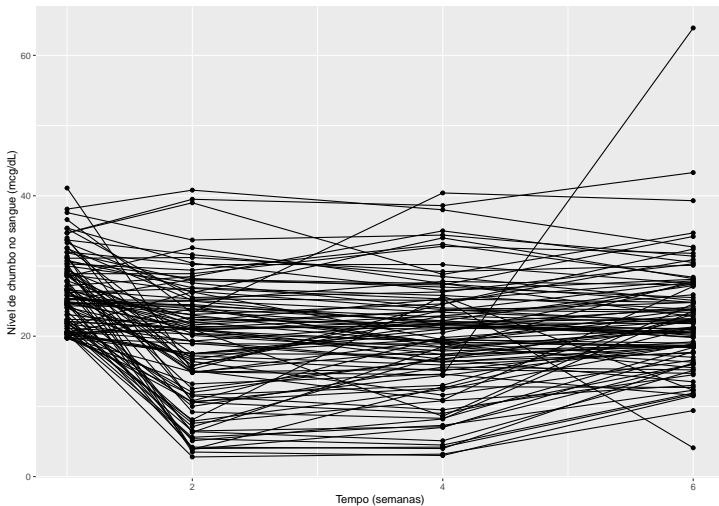
Time plot (diagrama de dispersão)



Time plot (linhas)

```
p <- ggplot(data = chumbo.longo,  
            mapping = aes(x = tempo, y = chumbo,  
                          group = id)) +  
  
  geom_point() +  
  geom_line() +  
  labs(x = "Tempo (semanas)",  
        y = "Nível de chumbo no sangue (mcg/dL)")  
p + theme_gray()
```

Time plot (linhas)



Time plot (perfis médios)

“Pré-processamento”

```
library(dplyr)

chumbo.resumo <- chumbo.longo %>%
  group_by(trt, tempo) %>%
  summarise(chumbo.m = mean(chumbo))

chumbo.resumo
```

```
## # A tibble: 8 x 3
## # Groups:   trt [2]
##   trt      tempo chumbo.m
##   <fct>    <dbl>    <dbl>
## 1 Placebo      1      26.3
## 2 Placebo      2      24.7
```

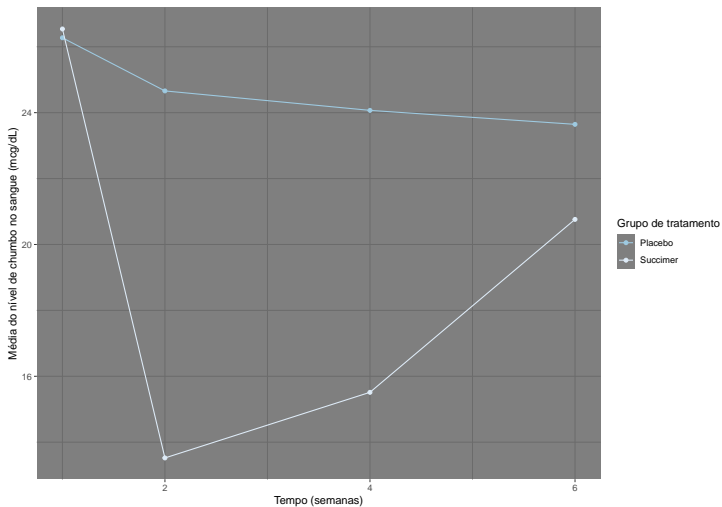
Time plot (perfis médios)

## 3	Placebo	4	24.1
## 4	Placebo	6	23.6
## 5	Succimer	1	26.5
## 6	Succimer	2	13.5
## 7	Succimer	4	15.5
## 8	Succimer	6	20.8

Time plot (perfis médios)

```
p <- ggplot(data = chumbo.resumo,  
            mapping = aes(x = tempo,  
                          y = chumbo.m,  
                          colour = trt)) +  
  
  geom_point() +  
  geom_line() +  
  scale_color_brewer(direction = -1) +  
  labs(x = "Tempo (semanas)",  
       y = "Média do nível de chumbo no sangue (mcg/dL)",  
       colour = "Grupo de tratamento")  
  
p + theme_dark()
```

Time plot (perfis médios)



Time plot (perfis médios com barras de erros)

“Pré-processamento”

```
chumbo.resumo <- chumbo.longo %>%  
  group_by(trt, tempo) %>%  
  summarise(chumbo.m = mean(chumbo),  
            dp = sd(chumbo), n = n()) %>%  
  mutate(ep = dp/sqrt(n))
```

```
chumbo.resumo
```

```
## # A tibble: 8 x 6
```

```
## # Groups:   trt [2]
```

##	trt	tempo	chumbo.m	dp	n	ep
##	<fct>	<dbl>	<dbl>	<dbl>	<int>	<dbl>
## 1	Placebo	1	26.3	5.02	50	0.711
## 2	Placebo	2	24.7	5.46	50	0.772

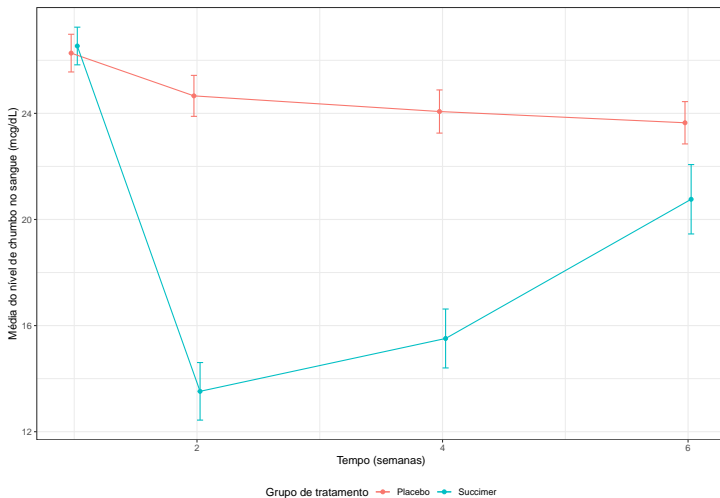
Time plot (perfis médios com barras de erros)

## 3 Placebo	4	24.1	5.75	50 0.814
## 4 Placebo	6	23.6	5.64	50 0.798
## 5 Succimer	1	26.5	5.02	50 0.710
## 6 Succimer	2	13.5	7.67	50 1.09
## 7 Succimer	4	15.5	7.85	50 1.11
## 8 Succimer	6	20.8	9.25	50 1.31

Time plot (perfis médios com barras de erros)

```
p <- ggplot(data = chumbo.resumo,
            mapping = aes(x = tempo,
                          y = chumbo.m,
                          colour = trt)) +
  geom_errorbar(aes(ymin = chumbo.m - ep,
                    ymax = chumbo.m + ep),
               width = .1,
               position = position_dodge(0.1)) +
  geom_point(position = position_dodge(0.1)) +
  geom_line(position = position_dodge(0.1)) +
  labs(x = "Tempo (semanas)",
       y = "Média do nível de chumbo no sangue (mcg/dL)",
       colour = "Grupo de tratamento")
p + theme_bw() + theme(legend.position = "bottom")
```

Time plot (perfis médios com barras de erros)



Dados desbalanceados

```
fev <- read_dta(  
  file = here::here("data", "fev1.dta")  
fev
```

```
## # A tibble: 1,994 x 6
```

```
##       id    ht   age baseht baseage logfev1  
##    <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>  
## 1      1  1.20  9.34    1.20    9.34    0.215  
## 2      1  1.28 10.4     1.20    9.34    0.372  
## 3      1  1.33 11.5     1.20    9.34    0.489  
## 4      1  1.42 12.5     1.20    9.34    0.751  
## 5      1  1.48 13.4     1.20    9.34    0.833  
## 6      1  1.5   15.5     1.20    9.34    0.892  
## 7      1  1.52 16.4     1.20    9.34    0.871  
## 8      2  1.13  6.59     1.13    6.59    0.307  
## 9      2  1.19  7.65     1.13    6.59    0.351
```

Dados desbalanceados

```
## 10      2  1.49 12.7    1.13    6.59    0.756
```

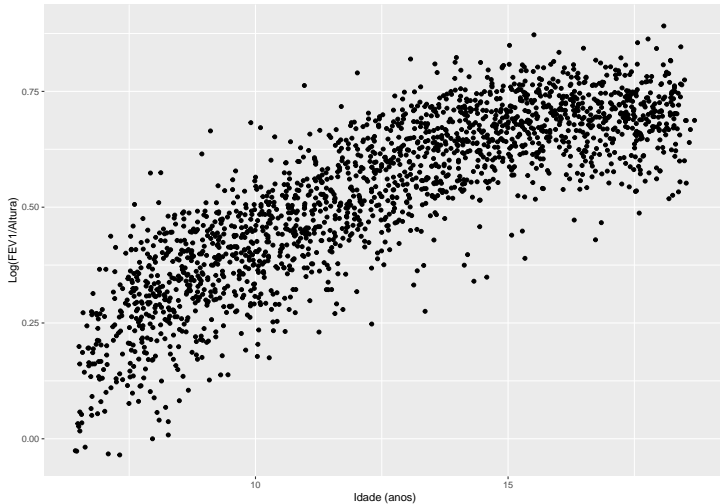
```
## # ... with 1,984 more rows
```

```
fev <- fev[- which(fev$logfev1/fev$ht < -0.5), ]
```

Dados desbalanceados

```
p <- ggplot(data = fev,  
            mapping = aes(x = age, y = logfev1/ht)) +  
  geom_point() +  
  labs(x = "Idade (anos)",  
       y = "Log(FEV1/Altura)")  
p + theme_gray()
```

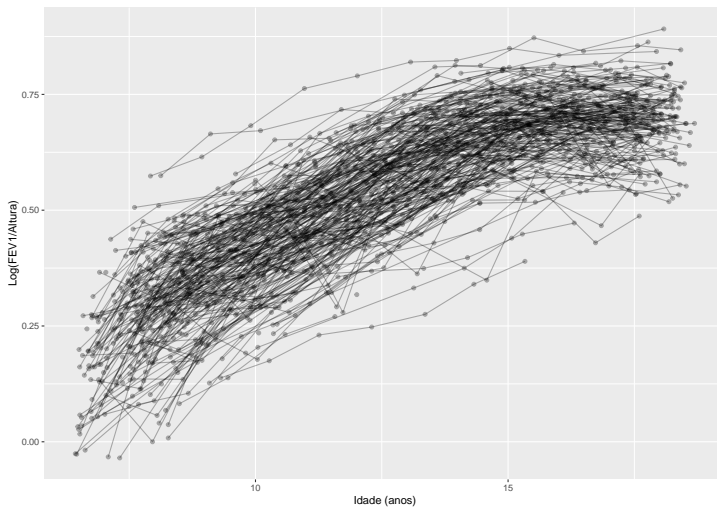
Dados desbalanceados



Dados desbalanceados

```
p <- ggplot(data = fev,  
            mapping = aes(x = age, y = logfev1/ht,  
                          group = id)) +  
  geom_point(alpha = 0.3) +  
  geom_line(alpha = 0.3) +  
  labs(x = "Idade (anos)",  
       y = "Log(FEV1/Altura)")  
p + theme_gray()
```

Dados desbalanceados

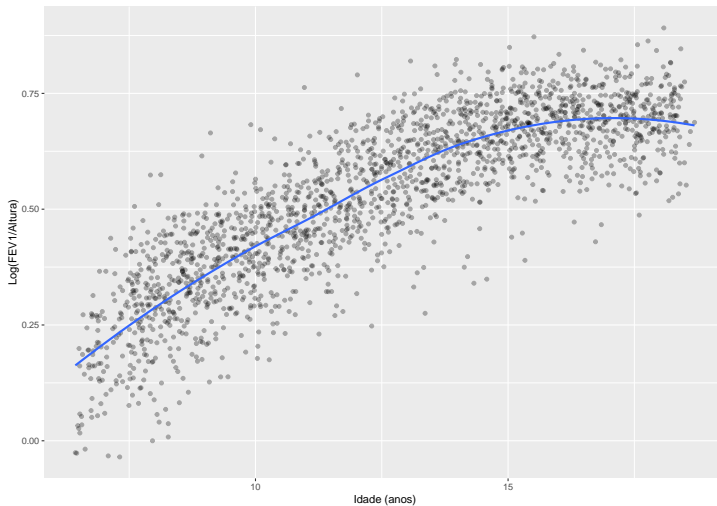


Dados desbalanceados

Regressão local

```
p <- ggplot(data = fev,  
            mapping = aes(x = age, y = logfev1/ht)) +  
  geom_point(alpha = 0.3) +  
  geom_smooth(method = "loess", se = FALSE) +  
  labs(x = "Idade (anos)",  
       y = "Log(FEV1/Altura)")  
p + theme_gray()
```

Dados desbalanceados



Estrutura de correlação

```
chumbo.succimer <- chumbo %>%  
  filter(trt == 1) %>%  
  select(y0, y1, y4, y6)
```

```
chumbo.succimer
```

```
## # A tibble: 50 x 4  
##       y0      y1      y4      y6  
##   <dbl> <dbl> <dbl> <dbl>  
## 1  26.5  14.8  19.5  21  
## 2  25.8  23    19.1  23.2  
## 3  20.4   2.80   3.20   9.40  
## 4  20.4   5.40   4.5    11.9  
## 5  24.8  23.1   24.6   30.9  
## 6  27.9   6.30  18.5   16.3  
## 7  35.3  25.5   26.3   30.3
```

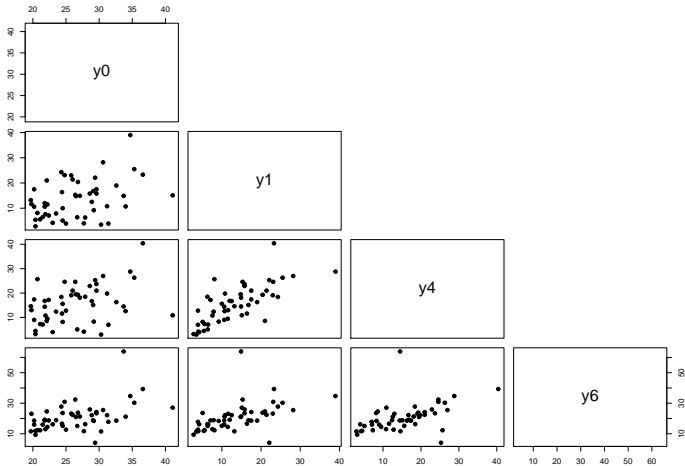
Estrutura de correlação

```
## 8 28.6 15.8 22.9 25.9
## 9 29.6 15.8 23.7 23.4
## 10 21.5 6.5 7.10 16
## # ... with 40 more rows
```

Estrutura de correlação

```
# library(GGally)
#
# p <- ggpairs(chumbo.succimer,
#             columnLabels = paste("Semana", c(0, 1, 4, 6)))
# p
pairs(chumbo.succimer, , pch = 19, upper.panel = NULL)
```

Estrutura de correlação



Exercícios

Exercícios

- ▶ Com o auxílio do computador, faça os exercícios do Capítulo 2 do livro “**Applied Longitudinal Analysis**” (páginas 44 e 45).
- ▶ **Enviar soluções** pelo Moodle através do fórum (será aberto hoje!).

Avisos

- ▶ **Para casa:** ler o Capítulo 3 do livro “**Applied Longitudinal Analysis**”. Caso ainda não tenha lido, leia também os Caps. 1 e 2.
 - ▶ Ver https://datathon-ufrgs.github.io/Pintando_e_Bordando_no_R/
- ▶ **Próxima aula:** Considerações a respeito da modelagem da média e da covariância, e abordagem histórica dos métodos de análise de medidas repetidas.

Bons estudos!

