Introdução ao R

Noções Básicas da Linguagem de Programação
Segunda Parte

Cássio Roberto de Andrade Alves
Denise Manfredini
PPGEco/UFSC

30 de setembro de 2018

Análise de Dados

Retomando o assunto do sábado passado...

Pergunta
Como analisar dados no R?

Objetivo
Aprender a carregar, analisar e manipular dados.

Análise de Dados

- Manipulação e exploração de dados

Filtro: Abordagem do R

```
library(dplyr)
# Seleciona as linhas que são "Domestic" e possuem
# preço menor ou igual a 4000
dados[origem=="Domestic" & preço <=4000,]
```

- pacote 'dplyr'

Filtro: abordagem do pacote

```
library(dplyr)
# Seleciona as linhas que são "Domestic" e possuem
# preço menor ou igual a 4000
filter(dados, origem == "Domestic" & preço <= 4000)
```

Análise de dados

- **Exercício 1:**
  Escreva um código que separe os carros de origem "Doméstica" e "Estrangeira" e avalie se, em média, carros importados são mais caros.
- **Exercício 2:**
  Suponha que você é um vendedor e está com um cliente buscando um carro econômico (pelo menos 10 km/l) *e* barato (menos de \$5.000,00). Quais os carros você ofereceria para ele?

Análise de dados

- Análise de correlação
- Coeficiente de correlação de Pearson
  - É uma medida de associação linear entre as variáveis.
  - $-1 \leq \rho \leq 1$

```
# Correlação
peso <- dados[,4]
comp <- dados[,5]
cor(comp, peso)
```

Análise de dados

- Matriz de correlação

```
# Seleciona apenas os dados numéricos
numerico <- select_if(dados, is.numeric)

# Matriz de correlação
cor(numerico)
```

```
# Abordagem do R
numerico <- unlist(lapply(dados, is.numeric))

# Matriz de correlação
cor(numerico)
```

Condições

- **Executar ou não executar? Eis a questão...**
- Nem sempre queremos que todas as linhas sejam executadas.
- if ou "se".

Formato da estrutura condicional if

```
if (condição) { # bloco verdadeiro
}
```

Condições

- Exemplo simples: Escreva um código que leia dois números, a e b, e imprima o maior deles.

Exemplo

```
a <- as.numeric(readline("Digite um número:" ))
b <- as.numeric(readline("Digite outro número:" ))

if (a > b) {
  print("O primeiro número é o maior!")
}

if (b > a){
  print("O segundo número é o maior!")
}
```

Condições

- Exemplo/exercício complicado:
  - Imagine que salários abaixo de R$ 1.000,00 não pagam impostos, isto é, alíquota 0%.
  - Para salários entre R$ 1.000,00 e R$ 3.000,00 a alíquota é de 20 %.
  - Para salários acima de R$ 3.000,00 a alíquota é de 30 %.
  - Escreva um programa que peça ao usuário o valor do salário e retorne o valor do imposto que deve ser pago.

Imposto de renda

```
salario <- as.numeric(readline("Digite o valor do salário:"))
base <- salario
imposto <- 0

if (base>3000){
  imposto <- imposto + ((base - 3000)*0.35)
  base <- 3000
}

if (base>1000){
  imposto <- imposto + ((base - 1000)*0.20)
}

print("O valor de imposto a pagar é de:")
print(imposto)
```

Condições

- **else**: caso contrário.
- Usamos quando a segunda condição é o oposto da primeira

Formato da estrutura condicional if

```
if (condição) {
# bloco verdadeiro
}else{
# bloco falso
}
```

- Como ficaria o exemplo dos números a e b?

Fluxo de controle

- Repetições
- É a base de vários programas
- Utilizados para executar a mesma parte de um programa várias vezes.
- Uma estrutura de repetição muito utilizada é o **for**

Formato da estrutura for

```
for (i in conjuntoDeValores){
# bloco de comandos que serão repetidos
}
```

- Obs.: a variável i é uma variável auxiliar que assume os sucessivos valores do nosso conjunto de valores.

Repetições

- Exemplo simples: Escreva um programa que imprima os números de 1 até 5.

```
for (i in 1:5){
print(i)
}
```

Repetições

- Exemplo intermediário: Considere a variável:

  Notas <− c(9, 10, 8, 5, 7)

- Escreva um programa que acesse cada uma das notas e imprima seu valor.

```
Notas <− c(9, 10, 8, 5, 7)
for (i in 1:5){
print(Notas[i])
}
```

Repetições

- Exemplo mais complicado: Repetições + if + dados
- Carregue os dados do IPCA mensal.
- Construa um número índice baseado na inflação mensal medida pelo IPCA

Construindo número índice

```
dados <- read.csv('dados_macro.csv',
                  header=TRUE,
                  sep=',',
                  dec = ',')
tx_inf <- ts(dados[,2], start=c(1999,06), frequency = 12)
y <- ts(dados[,3], start=c(1999,06), frequency = 12)
```

Exemplo

Construindo número índice

```
pi <- BETSget(433, from = "1999-05-30",
                        to = "2018-06-30")
N <- length(pi)
indice<- c(rep(0,N))

for (t in 1:N){
      if (t==1){
      indice[t]=1
      }
else {indice[t] <- indice[(t-1)]*(1 + pi[t]/100)}
}
```

Deflacionando uma série de tempo

- Vamos trabalhar com a série do PIB nominal

```
y <- BETSget(4380, from="1999-05-30",
                   to="2018-06-30")
```

- Vamos transformar essa série em valores reais de jun/2018.
- Mudança de base do nosso indice + multiplicação pela série nominal

```
y_real <- y * (tail(indice,n=1)/indice)
```

- Mais sobre deflacionar uma série: ▸ Link

Gráficos no R

Pergunta
Como fazer gráficos no R?

Objetivo
Fazer gráficos para variáveis quantitativas, séries temporais e histogramas.

Bibliografia
Grant V. Farnsworth. Econometrics in R. Technical report, outubro 2008. Link

Marcelo S. Perlin . Processamento e Análise de Dados Financeiros e Econômicos com o R. Self Published; Edição: 2, junho 2018. Link

Gráficos no R base e ggplot2

- R possui algumas funções nativas para criar figuras;
- Essas funções são limitadas;
- Era considerada uma das limitações no uso do R.

ggplot2

O ggplot2 é um pacote de visualização de dados onde os os gráficos são construídos camada por camada.

Guia ggplot2 ▸ Link
Folha de cola do ggplot2 ▸ Link

Gráfico simples ggplot2 vs função nativa

```
#importa os dados
dados_curso <- read_csv("C:/Users/dados_curso.csv")
# abre em uma janela separada
x11()
```

função nativa

```
plot(dados_curso$preço, dados_curso$`km por litro`)
```

ggplot2

```
library(ggplot2)

p <- ggplot(dados_curso, aes(x=preço, y=`km por litro`))
+ geom_point()
plot(p)
```

Gráficos

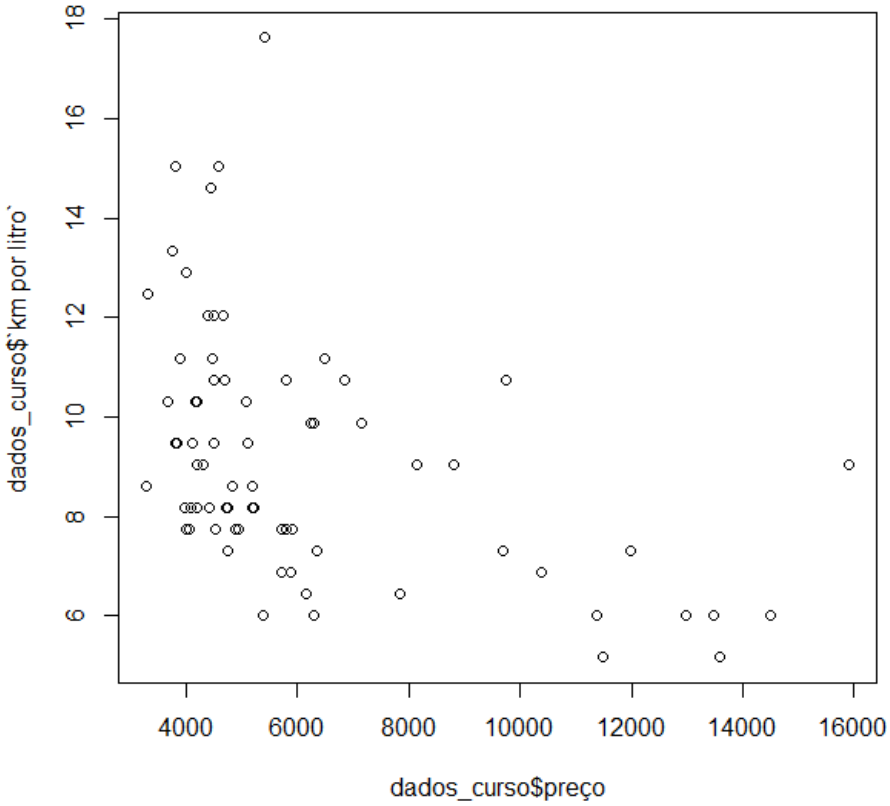


Figura: Função nativa

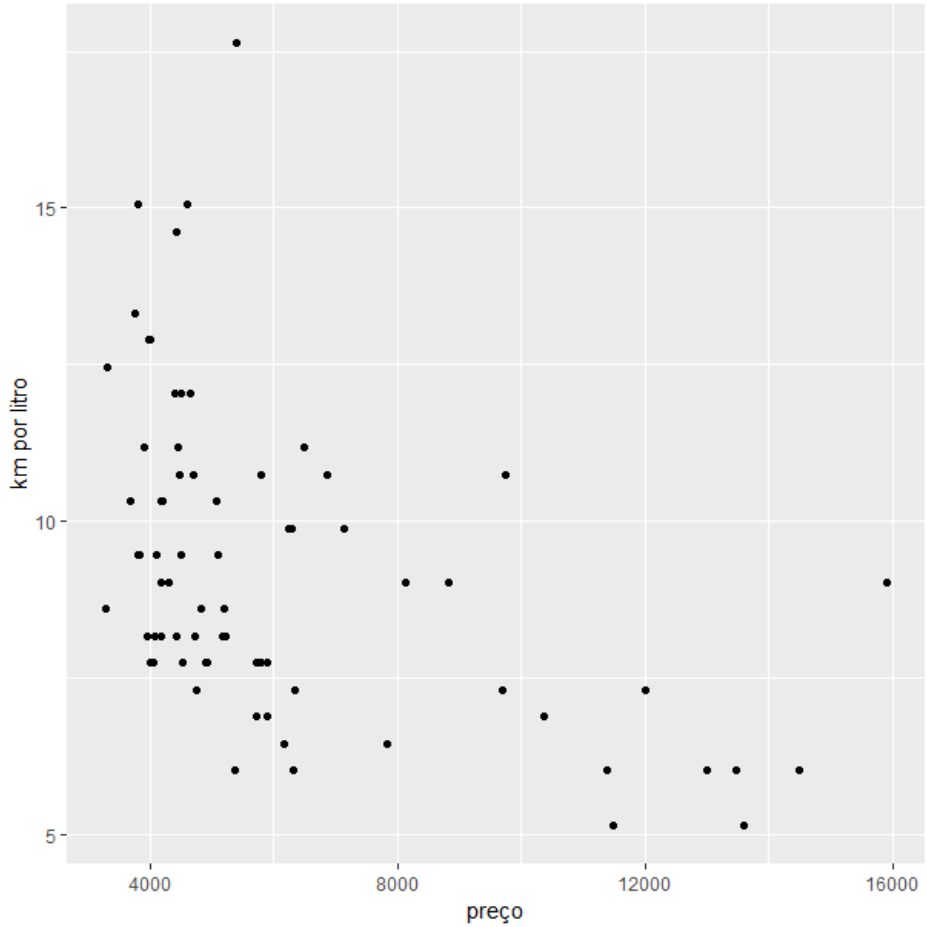


Figura: Função ggplot

ggplot2

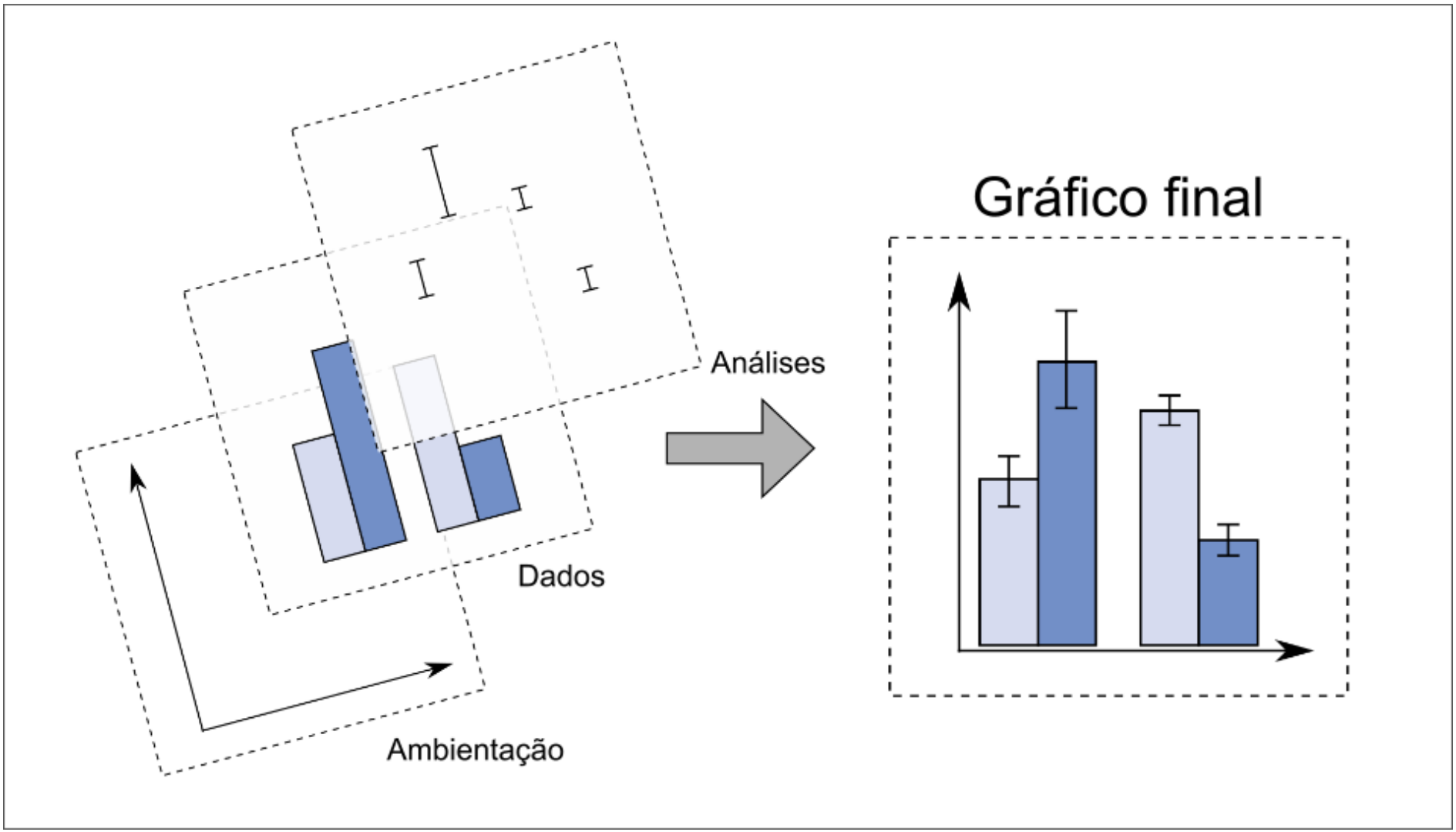


Figura: ▸ Link

Gráficos: Infinitas possibilidade

```
install.packages(c("ggplot2", "plotly", "gapminder"))

library(ggplot2)
library(plotly)
library(gapminder)

p <- gapminder %>%
  filter(year==1977) %>%
  ggplot(aes(gdpPercap, lifeExp, size=pop, color=continent))+
  geom_point()+
  scale_x_log10()+
  theme_bw()


ggplotly(p)
```

Gráfico na Função Nativa

Onde mostrar as figuras?

- No canto direito inferior do RStudio tem uma aba com "plots".
- Mas é melhor criar uma janela externa ao RStudio para visualizar as figuras utilizando o comando: x11()

Comando úteis

- Comando de ajuda: help(plot)
- Desligar o x11(): dev.off()
- Comando para limpar os gráfico: graphics.off()

  É prática comum usar essa função no início de uma rotina de pesquisa.

Alguns elementos da função *plot*

```
# Gráfico padrão
plot(dados_curso$preço, dados_curso$`km por litro`)

# Gráfico com os pontos azuis
plot(dados_curso$preço, dados_curso$`km por litro`, col="blue")

# Gráfico com o intervalo do x limitado entre 0 e 12000
plot(dados_curso$preço, dados_curso$`km por litro`,
     col="blue", xlim=c(0,12000))
```

Alguns elementos da função *plot*

```
# Gráfico com novo nome de x e y
plot(dados_curso$preço, dados_curso$`km por litro`,col="blue", xlab="preço
      ylab="Km/l", col.lab=rgb(0,0.5,0))

# Colocando título
title(main="Preço vs Km/l", col.main="black", font.main=4)

# Colocando legenda
legend(12000, 14, legend=c("Pontos"),
        col=c( "blue"),pch = c(1))
```

Como salvar as imagens?

1. Pela interface do RStudio.
2. Colocando o comando no script:

```
graphics.off()
png(filename="your/file/location/name.png")
plot(<variavel>)
dev.off()
```

3. Copiando o que aparece na tela:

```
graphics.off()
plot(<variavel>)
dev.print(pdf, 'filename.pdf')
```

Histograma

```
# Histograma padrão
hist(dados_curso$`km por litro`)

# Hitograma modificado
hist(dados_curso$`km por litro`,
     main="Histograma",
     xlab="km␣por␣litro",
     col="darkmagenta",
     ylab="Frequncia",
     freq=TRUE)
```

Gráfico de Série Temporal

```
# Importa dados de macro
dados_macro <- read_csv("C:/Users/dados_macro.csv")

# Declara dados de série temporal
ts_macro <- ts(dados_macro$`PIB nominal`,
               start = c(1999, 6), frequency = 12)

# Conferindo o número de observações
length(ts_macro)

# Faz gráfico de série temporal
plot.ts(dados_macro$`PIB nominal`)
```

Histograma e Gráfico de Série Temporal

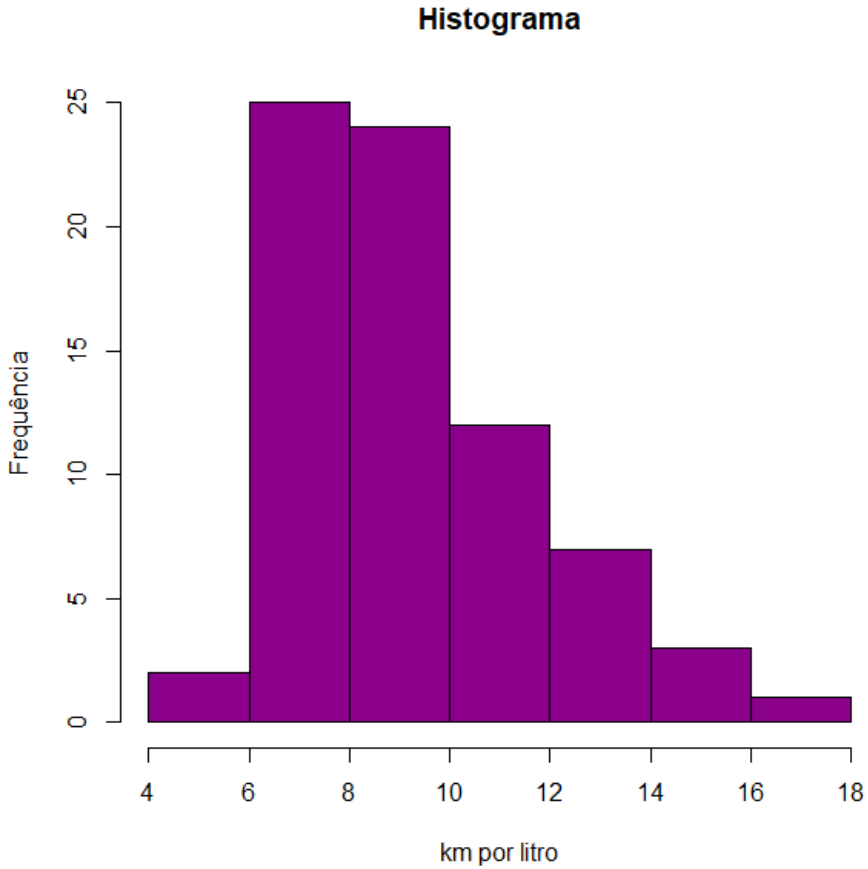


Figura: Histograma

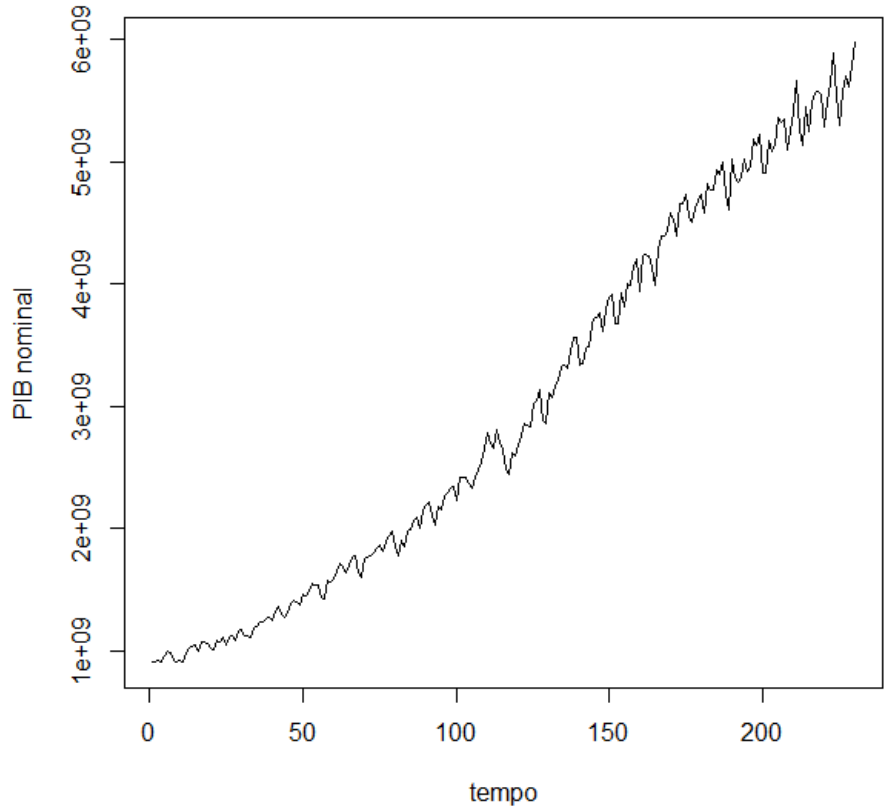


Figura: Gráfico de Série Temporal

Regressão Linear no R

Pergunta

Como fazer ma regressão linear no R?

Objetivo

Fazer uma regressão linear simples (máximo de duas variáveis explicativa).

Bibliografia

Grant V. Farnsworth. Econometrics in R. Technical report, outubro 2008.

▸ Link

Regressão Linear no R

- Vamos ver o método de OLS (ordinary least squares - mínimos quadrados ordinários);
- Relação linear dos dados com forma simples e rápida de estimação;
- Essa estimação minimiza a soma dos erros quadráticos - encontra a forma linear que melhor ser ajusta aos dados históricos.

Função no R para OLS

A principal função para estimar um modelo linear no R é lm.

Minimos Quadrados Ordinários - 1 variável explicativa

Representação do modelo linear

$$y = \alpha + \beta_1 x$$

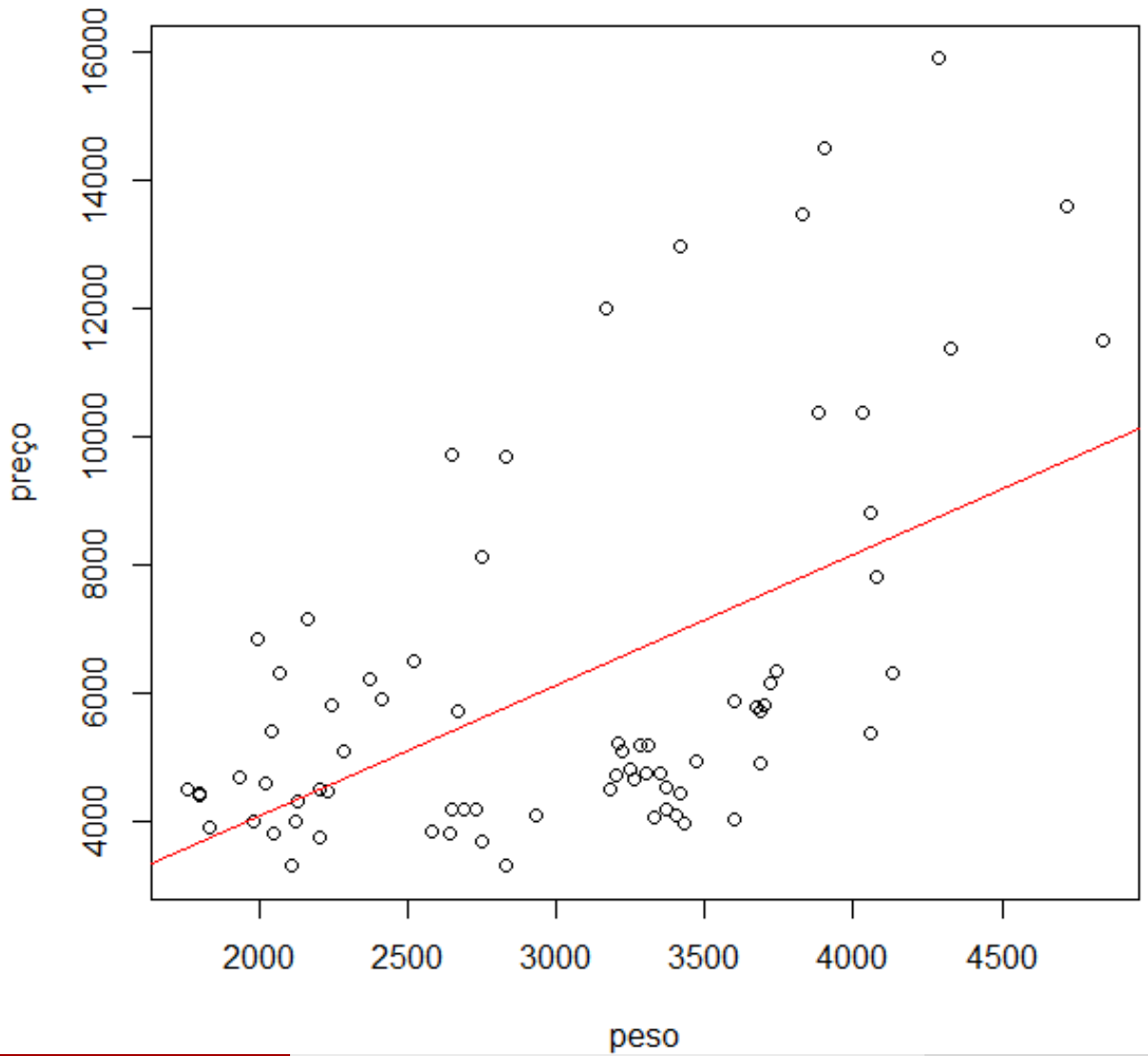
y é preço;e x é peso

```
# Regressão com uma variável explicativa
regressao_simp <- lm(data = dados_curso,
                     formula = preço ~ peso)

# Sumário dos ressultados da regressão
 summary(regressao_simp)

# Gráfico do Ajuste
plot(preço ~ peso, data = dados_curso)
abline(regressao_simp)
```

Gráfico da Regressão



Minimos Quadrados Ordinários - 2 variáveis explicativas

Representação do modelo linear

$$y = \alpha + \beta_1 x + \beta_2 z$$

y é preço; x é peso; e z é km/l

```
# Regressão com duas variáveis explicativas
regressao <- lm(data = dados_curso,
                formula = preço ~ peso +'km por litro')

# Sumário dos ressultados da regressão
 summary(regressao)

# Resultado dos coeficientes da Regressão
print(regressao)
```

Cássio Roberto de Andrade Alves
Mestrando em Economia
Universidade Federal de Santa Catarina

aalves.cassio@gmail.com

Denise Manfredini
Doutoranda em Economia
Universidade Federal de Santa Catarina

manfredini.denise@gmail.com