

Tópicos de estatística aplicada utilizando R*

Uma abordagem gerando relatórios reprodutíveis usando RStudio, *LaTeX*, RMarkdown e Quarto

Emerson Scheidegger

2023-06-05

Neste documento tento abordar de forma simples como trabalhar com a Regressão Linear Simples utilizando a linguagem R e a suíte do RStudio. Para gerar este documento utilizei o Rstudio + RMarkdown + LaTeX + Quarto.

Sumário

1	Características do R	3
2	Importação de dados	4
2.1	Importando arquivos .csv	4
2.2	Importando arquivos .xls ou .xlsx	4
2.3	Distribuições de Frequência	4
3	Funções estatísticas	5
3.1	Tabelas	5
3.2	Tabelas de proporções	5
3.3	Medidas de resumo	6
3.4	Summary	6
3.4.1	Média	6
3.4.2	Mediana	6
3.4.3	Variância	7
3.4.4	Desvio-padrão	7
4	Testes de Hipótese	9
4.1	Testes para a média populacional e para a comparação de duas médias	9
4.1.1	Teste t para média populacional	9
4.1.2	Teste t para comparação de duas médias com variâncias iguais	10
4.1.3	Teste t pareado	10
4.2	Testes para uma proporção populacional e para comparação de duas proporções	10
4.2.1	Teste para uma proporção populacional	11

*Agradeço ao R CORE TEAM (2023) e a todos que dedicaram seu tempo no desenvolvimento de funções e pacotes para o ambiente RStudio principalmente a Yihui Xie <https://yihui.org> pela milhares de horas dedicadas a melhorar nossas vidas.

4.2.2	Teste para comparação de duas proporções	11
4.3	Testes para Normalidade	12
4.3.1	shapiro.test()	12
4.3.2	ad.test()	13
4.3.3	cvm.test()	14
4.3.4	lillie.test()	14
4.3.5	pearson.test()	15
4.3.6	sf.test()	15
5	Testes para comparação de variâncias	17
5.1	Pacote: stats	17
5.1.1	bartlett.test()	17
5.2	Pacote: car	18
5.2.1	levene.test()	18
6	Funções Matemáticas	19
6.1	Combinatória	19
6.2	Fatorial	19
6.3	Raiz Quadrada	19
7	Gráficos	20
7.1	Gráfico de barras	20
7.2	Histograma	21
7.3	Boxplot	21
7.4	Gráfico de Pizza	21
7.5	Gráfico de Dispersão	21
8	Probabilidade	21
9	Exemplos	21
	Referências	27

Lista de Figuras

1	Teste de independência do exemplo: Satisfação no trabalho, Agresti(2002, p.57)	12
2	Vendas mensais de três lojas	17
3	Disputa de jogos entre dois times	18
4	Vendas mensais de três lojas	18
5	Gráfico de barras.	21
6	Gráfico de distribuição uniforme.	23
7	Gráfico de distribuição binomial.	24
8	Gráfico de distribuição Normal.	25
9	Gráfico de distribuição exponencial.	26

1 Características do R

- Não foi feito para manipulação de dados em larga escala.
- Forma mais fácil e direta de acessar os dados é convertê-los para texto e importar.
- Salva a sessão em um arquivo .RDATA, que armazena todos os objetos R, possibilitando que um projeto seja retomado posteriormente ou intercambiado com colaboradores.
- Acessa bancos de dados e planilhas Microsoft Excel via ODBC e outros bancos de dados por servidor SQL, ampliando a capacidade de trabalhar com dados em larga escala.
- A partir da versão 2.1.1 possui um editor de script, que facilita a execução de comandos diretamente de dentro do R.
- Possui pacotes com funções específicas que podem ser instalados pela Internet, através do próprio programa.
- Conta com inúmeros colaboradores no mundo inteiro que criam, testam e corrigem as funções que podem ser usadas por qualquer pessoa.
- Gera gráficos em diferentes formatos para as mais diversas utilizações.
- O Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

2 Importação de dados

2.1 Importando arquivos .csv

```
#download.file("https://www.ime.usp.br/~pam/dados.RData", "dados.RData")
#load("dados.RData")

tab2_1<-read.table("tabela2_1.csv", dec=",", sep=";", h=T)

names(tab2_1)
```

```
[1] "N"                "estado_civil"    "grau_instrucao"  "n_filhos"
[5] "salario"          "idade_anos"      "idade_meses"     "reg_procedencia"
```

```
summary(tab2_1$salario)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.000	7.553	10.165	11.122	14.060	23.300

2.2 Importando arquivos .xls ou .xlsx

2.3 Distribuições de Frequência

```
# Calcula a tabela de frequências absolutas e armazena o resultado em 'ni'
ni<-table(tab2_1$grau_instrucao)
fi<-prop.table(ni) # Tabela de frequências relativas (f_i)
p-fi<-100*prop.table(ni) # Porcentagem (100 f_i)

# Adiciona linhas de total
ni<-c(ni,sum(ni))
fi<-c(fi,sum(fi))
p-fi<-c(p-fi,sum(p-fi))
names(ni)[4]<-"Total"
tab2_2<-cbind(ni,fi=round(fi,digits=2),p-fi=round(p-fi,digits=2))
tab2_2
```

	ni	fi	p-fi
ensino fundamental	12	0.33	33.33
ensino médio	18	0.50	50.00
superior	6	0.17	16.67
Total	36	1.00	100.00

3 Funções estatísticas

3.1 Tabelas

```
#Grau de instrução
table(tab2_1$grau_instrucao)
```

ensino fundamental	ensino médio	superior
12	18	6

```
#Grau de instrução x Estado Civil
table(tab2_1$grau_instrucao, tab2_1$estado_civil)
```

	casado	solteiro
ensino fundamental	5	7
ensino médio	12	6
superior	3	3

```
table(tab2_1$n_filhos, tab2_1$grau_instrucao)
```

	ensino fundamental	ensino médio	superior
0	1	2	1
1	1	4	0
2	2	5	0
3	1	0	2
5	0	1	0

3.2 Tabelas de proporções

```
# Grau de instrução x Estado Civil
prop.table(table(tab2_1$grau_instrucao, tab2_1$estado_civil))
```

	casado	solteiro
ensino fundamental	0.13888889	0.19444444
ensino médio	0.33333333	0.16666667
superior	0.08333333	0.08333333

```
# Com duas casas decimais
round(prop.table(table(tab2_1$grau_instrucao, tab2_1$estado_civil)), 2)
```

	casado	solteiro
ensino fundamental	0.14	0.19
ensino médio	0.33	0.17
superior	0.08	0.08

3.3 Medidas de resumo

3.4 Summary

Resume a variável quantitativa em: mínimo, máximo, média, mediana, 1º quartil, 3º quartil e dados não preenchidos. Caso a variável seja qualitativa, é informado o número de observações para cada nível.

Medidas de resumo dos salários:

```
summary(tab2_1$salario)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.000	7.553	10.165	11.122	14.060	23.300

Resumo da variável salário apenas para casados

```
summary(tab2_1$salario[tab2_1$estado_civil=="casado"])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.560	8.742	11.925	12.123	15.030	23.300

! Importante

Observação: Caso a variável desejada seja qualitativa numérica, é possível que o R interprete-a como sendo uma variável quantitativa. Para evitar que isso aconteça, utilize a função `as.factor()`.
Ex: `summary(as.factor(dados$sexo))`

3.4.1 Média

```
# Média dos salários  
cat('A média de salário é:', mean(tab2_1$salario), '\n')
```

A média de salário é: 11.12222

```
# Média de idade  
cat('A média de idade é:', mean(tab2_1$idade_anos), '\n')
```

A média de idade é: 34.58333

3.4.2 Mediana

```
# Mediana dos salários
cat('A mediana de salário é:', median(tab2_1$salario), '\n')
```

A mediana de salário é: 10.165

```
# Mediana de idade
cat('A mediana de idade é:', median(tab2_1$idade_anos), '\n')
```

A mediana de idade é: 34.5

3.4.3 Variância

```
# Variância dos salários
cat('A variância de salário é:', var(tab2_1$salario), '\n')
```

A variância de salário é: 21.04477

3.4.4 Desvio-padrão

sintaxe: `sd(variável)`

opções:

`na.rm`: TRUE, calcula o desvio padrão considerando apenas os dados existentes, ignora os dados faltantes.

FALSE, calcula o desvio padrão apenas se todos os valores estiverem preenchidos, caso contrário retorna NA.

Exemplo:

```
cat('O desvio-padrão dos salários é:', sd(tab2_1$salario), '\n')
```

O desvio-padrão dos salários é: 4.587458

Erro em `var(tab2_1$n_filhos)` : observações faltantes em cov/cor

```
sd(tab2_1$n_filhos)
```

[1] NA

Tratando as observações faltantes

```
sd(tab2_1$n_filhos, na.rm=TRUE)
```

```
[1] 1.268028
```


4 Testes de Hipótese

4.1 Testes para a média populacional e para a comparação de duas médias

`t.test()`

Realiza o teste t-Student para uma ou duas amostras.

sintaxe:

`t.test(amostra1, amostra2, opções)`

parâmetros

`amostra1`: Vetor contendo a amostra da qual se quer testar a média populacional, ou comparar a média populacional com a média populacional da amostra 2.

`amostra2`: Vetor contendo a amostra 2 para comparação da média populacional com a média populacional da amostra 1.

opções

`alternative`: string indicando a hipótese alternativa desejada. Valores possíveis: “two-sided”, “less” ou “greater”.

`mu`: valor indicando o verdadeiro valor da média populacional para o caso de uma amostra, ou a diferença entre as médias para o caso de duas amostras.

`paired`: TRUE – realiza o teste t pareado.

FALSE – realiza o teste t não pareado.

`var.equal`: TRUE – indica que a variância populacional é a igual nas duas amostras.

FALSE – indica que a variância populacional de cada amostra é diferente.

`conf.level`: coeficiente de confiança do intervalo.

Exemplo:

4.1.1 Teste t para média populacional

```
amostra1 = c(14.9,13.4,14.5,13.5,15.0,13.9,14.9,16.4,14.6,15.4)
t.test(amostra1,mu=15)
```

One Sample t-test

data: amostra1

t = -1.2252, df = 9, p-value = 0.2516

alternative hypothesis: true mean is not equal to 15

95 percent confidence interval:

14.00375 15.29625

sample estimates:

mean of x

14.65

4.1.2 Teste t para comparação de duas médias com variâncias iguais

```
amostra1 = c(16.6,13.4,14.6,15.1,12.9,15.2,14.0,16.6,15.4,13.0)
amostra2 = c(15.8,17.9,18.2,20.2,18.1,17.8,18.3,18.6,17.0,18.4)
t.test(amostra1, amostra2, var.equal = TRUE)
```

Two Sample t-test

```
data: amostra1 and amostra2
t = -6.0257, df = 18, p-value = 1.069e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.518003 -2.181997
sample estimates:
mean of x mean of y
 14.68    18.03
```

4.1.3 Teste t pareado

```
antes = c(16.6,13.4,14.6,15.1,12.9,15.2,14.0,16.6,15.4,13.0)
depois = c(15.8,17.9,18.2,20.2,18.1,17.8,18.3,18.6,17.0,18.4)
t.test(antes,depois,paired=TRUE)
```

Paired t-test

```
data: antes and depois
t = -5.3231, df = 9, p-value = 0.000479
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -4.773642 -1.926358
sample estimates:
mean difference
 -3.35
```

4.2 Testes para uma proporção populacional e para comparação de duas proporções

`prop.test()`

Realiza o teste de proporções para uma ou duas amostras.

sintaxe:

`prop.test(x, n, p, opções)`

Parâmetros

x: Vetor contendo o número de sucessos em cada amostra.

n: Vetor contendo o número de realizações de cada amostra.

p: Vetor contendo as probabilidades de sucesso de cada amostra.

Exemplo:

4.2.1 Teste para uma proporção populacional

```
prop.test(104,200,0.6,correct=F)
```

1-sample proportions test without continuity correction

```
data: 104 out of 200, null probability 0.6
X-squared = 5.3333, df = 1, p-value = 0.02092
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
 0.4510379 0.5882083
sample estimates:
      p
0.52
```

4.2.2 Teste para comparação de duas proporções

```
prop.test(c(104,50),c(200,95),correct=F)
```

2-sample test for equality of proportions without continuity correction

```
data: c(104, 50) out of c(200, 95)
X-squared = 0.010297, df = 1, p-value = 0.9192
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.1282799 0.1156483
sample estimates:
  prop 1    prop 2 
0.5200000 0.5263158
```

Teste

Realiza o teste exato de independência de linhas e colunas em uma tabela de contingência com as marginais fixas.

Sintaxe:

`fisher.test(x, opções)`

Parâmetros

x: Matriz (tabela) contendo a frequência de observações em cada casela.

Opções

alternative: string indicando a hipótese alternativa desejada.

Valores possíveis: “two-sided”, “less” ou “greater”.

conf.int: TRUE: calcula o intervalo de confiança para a razão de chances em tabelas de dimensão 2x2.

conf.level: coeficiente de confiança do intervalo.

Exemplo:

Receita	Satisfação			
	Muito Insatisfeito	Pouco Insatisfeito	Moderadamente Satisfeito	Muito Satisfeito
< 15 mil	1	3	10	6
15-25 mil	2	3	10	7
25-40 mil	1	6	14	12
> 40 mil	0	1	9	11

Figura 1: Teste de independência do exemplo: Satisfação no trabalho, Agresti(2002, p.57)

```
Trabalho = matrix(c(1,2,1,0, 3,3,6,1, 10,10,14,9, 6,7,12,11), 4, 4,  
dimnames = list(Receita=c("< 15mil", "15-25mil", "25-40mil", "> 40mil"),  
Satisfação=c("M.Insatisfeito", "P.Insatisfeito", "Mod.Satisfeito", "M.Satisfeito")))  
fisher.test(Trabalho)
```

Fisher's Exact Test for Count Data

```
data: Trabalho  
p-value = 0.7827  
alternative hypothesis: two.sided
```

4.3 Testes para Normalidade

Estes pacotes contém diversos testes que verificam se os dados amostrais contém evidências de serem oriundos de uma população com distribuição Normal.

Pacote: base (Este pacote já está instalado)

4.3.1 shapiro.test()

Realiza o teste de Shapiro-Wilk para normalidade.

sintaxe:

```
shapiro.test(amostra)
```

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade. Deve conter uma amostra de tamanho entre 3 e 5000.

São permitidos missing values.

Exemplo:

```
shapiro.test(rnorm(10, mean=10, sd=4))
```

Shapiro-Wilk normality test

```
data:  rnorm(10, mean = 10, sd = 4)
W = 0.88376, p-value = 0.1441
```

Pacote opcional: nortest

Este pacote precisa ser instalado.

4.3.2 ad.test()

Realiza o teste de Anderson-Darling para normalidade.

sintaxe:

ad.test(*amostra*)

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade. Deve conter uma amostra de tamanho maior ou igual a 7.

São permitidos missing values.

Exemplo:

```
ad.test(rnorm(10, mean=10, sd=4))
```

Anderson-Darling normality test

```
data:  rnorm(10, mean = 10, sd = 4)
A = 0.53112, p-value = 0.1293
```

4.3.3 `cvm.test()`

Realiza o teste de Cramer-von Mises para normalidade.

sintaxe:

`cvm.test(amostra)`

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade. Deve conter uma amostra de tamanho maior ou igual a 7.

São permitidos missing values.

Exemplo:

```
cvm.test(rnorm(10, mean=10, sd=4))
```

Cramer-von Mises normality test

data: rnorm(10, mean = 10, sd = 4)

W = 0.039915, p-value = 0.6464

4.3.4 `lillie.test()`

Realiza o teste de Lilliefors (*Kolmogorov-Smirnov*) para normalidade.

sintaxe:

`lillie.test(amostra)`

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade. Deve conter uma amostra de tamanho maior ou igual a 4.

São permitidos missing values.

Exemplo:

```
lillie.test(rnorm(10, mean=10, sd=4))
```

Lilliefors (Kolmogorov-Smirnov) normality test

data: rnorm(10, mean = 10, sd = 4)

D = 0.20514, p-value = 0.2726

4.3.5 `pearson.test()`

Realiza o teste Qui-quadrado de Pearson para normalidade.

sintaxe:

`pearson.test(amostra)`

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade.

São permitidos missing values.

Opções

n.classes: Número de classes. São permitidos missing values.

adjust: TRUE: o valor p é calculado de uma distribuição Qui-quadrado com o número de graus de liberdade igual ao número de classes – 3.

FALSE: o valor p é calculado de uma distribuição Qui-quadrado com o número de graus de liberdade igual ao número de classes – 1.

Exemplo:

```
pearson.test(rnorm(10, mean=10, sd=4))
```

```
Pearson chi-square normality test
```

```
data:  rnorm(10, mean = 10, sd = 4)
```

```
P = 2, p-value = 0.5724
```

4.3.6 `sf.test()`

Realiza o teste de Shapiro-Francia para normalidade.

sintaxe:

`sf.test(amostra)`

Parâmetros

amostra: Vetor contendo a amostra da qual se quer testar normalidade.

Deve conter uma amostra de tamanho entre 5 e 5000. São permitidos missing values.

Exemplo:

```
sf.test(rnorm(10, mean=10, sd=4))
```

Shapiro-Francia normality test

data: rnorm(10, mean = 10, sd = 4)

W = 0.9238, p-value = 0.3362

5 Testes para comparação de variâncias

5.1 Pacote: stats

5.1.1 bartlett.test()

Realiza o teste de Bartlett com a hipótese nula de que as variâncias dos grupos são iguais (R CORE TEAM, 2023).

sintaxe:

```
bartlett.test(formula, dados)
```

Parâmetros

formula: Relação entre a variável dependente e o fator. Ex: “Vendas ~ Mês”.

dados: Conjunto de dados onde será aplicada a formula.

Exemplo: Queremos comparar a variabilidade das vendas entre os meses

Loja	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
A	10	14	16	19	12	15	11	10	12	13	18	23
B	12	13	13	14	13	11	16	09	11	11	16	25
C	15	17	12	16	10	16	12	12	08	14	21	24

Figura 2: Vendas mensais de três lojas

```
Vendas = c(10,12,15,14,13,17,16,13,12,19,14,16,12,
13,10,15,11,16,11,16,12,10,9,12,12,
11,8,13,11,14,18,16,21,23,25,24)
Mes = c("Jan","Jan","Jan","Fev","Fev","Fev","Mar","Mar","Mar","Abr","Abr","Abr",
"Mai","Mai","Mai","Jun","Jun","Jun","Jul","Jul","Jul","Ago","Ago","Ago",
"Set","Set","Set","Out","Out","Out","Nov","Nov","Nov","Dez","Dez","Dez")
dados = data.frame(Vendas=Vendas, Mes=Mes)
bartlett.test(Vendas ~ Mes, data=dados)
```

Bartlett test of homogeneity of variances

data: Vendas by Mes

Bartlett's K-squared = 2.844, df = 11, p-value = 0.9926

i Nota

Também é possível especificar os grupos da seguinte forma: `bartlett.test(list(GRUPO1, GRUPO2))`, onde GRUPO1 e GRUPO2 são vetores contendo os valores das observações de cada amostra.

Exemplo:

	Jogo 1	Jogo 2	Jogo 3	Jogo 4	Jogo 5	Jogo 6
Time A	30	25	32	22	19	26
Time B	18	24	31	28	29	30

Figura 3: Disputa de jogos entre dois times

```
TimeA = c(30,25,32,22,19,26)
TimeB = c(18,24,31,28,29,30)
bartlett.test(list(TimeA, TimeB))
```

Bartlett test of homogeneity of variances

```
data: list(TimeA, TimeB)
Bartlett's K-squared = 0.00032462, df = 1, p-value = 0.9856
```

5.2 Pacote: car

Este pacote precisa ser instalado (FOX; WEISBERG; PRICE, 2023).

5.2.1 levene.test()

Realiza o teste de Bartlett com a hipótese nula de que as variâncias dos grupos são iguais.

sintaxe:

```
leveneTest(formula, dados)
```

Parâmetros

formula: Relação entre a variável dependente e o fator. Ex: “Vendas ~ Mês”.

dados: Conjunto de dados onde será aplicada a formula.

Exemplo: Utilizando o mesmo exemplo visto no teste de Bartlett:

Loja	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
A	10	14	16	19	12	15	11	10	12	13	18	23
B	12	13	13	14	13	11	16	09	11	11	16	25
C	15	17	12	16	10	16	12	12	08	14	21	24

Figura 4: Vendas mensais de três lojas

```
leveneTest(dados$Vendas, dados$Mes)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group 11  0.1678 0.9981
      24
```

6 Funções Matemáticas

6.1 Combinatória

Calcula o número de combinações de n elementos em grupos de tamanho k.

sintaxe:

choose(n,k)

exemplo:

```
choose(8,5)
```

```
[1] 56
```

6.2 Fatorial

Calcula o fatorial de x.

sintaxe:

factorial(x)

exemplo:

```
factorial(5)
```

```
[1] 120
```

6.3 Raiz Quadrada

Calcula a raiz quadrada de x.

sintaxe:

sqrt(x)

exemplo:

```
sqrt(81)
```

```
[1] 9
```

```
sqrt(128)
```

```
[1] 11.31371
```

7 Gráficos

Os gráficos nos permitem analisar uma grande quantidade de informações de forma rápida, sem que seja necessário olhar tabelas e medidas de resumo. O R possui uma enorme capacidade para gerar diversos tipos de gráficos de alta qualidade totalmente configuráveis, desde cores e tipos de linhas, até legendas e textos adicionais.

A grande maioria das funções gráficas faz uso de opções comuns, ou seja, é extremamente fácil personalizar qualquer tipo de gráfico pois muitas das opções são iguais. As opções comuns a todos os gráficos serão abordadas aqui, e em cada seção seguinte as opções específicas àquele determinado tipo de gráfico serão apresentadas.

7.1 Gráfico de barras

```
barplot(table(tab2_1$estado_civil), col=c("blue","red"),
ylim=c(0,25),
space=.8, width=c(.2,.2),
main="Número de filhos por estado civil",
xlab="Estado Civil", ylab="Número de Filhos")
```

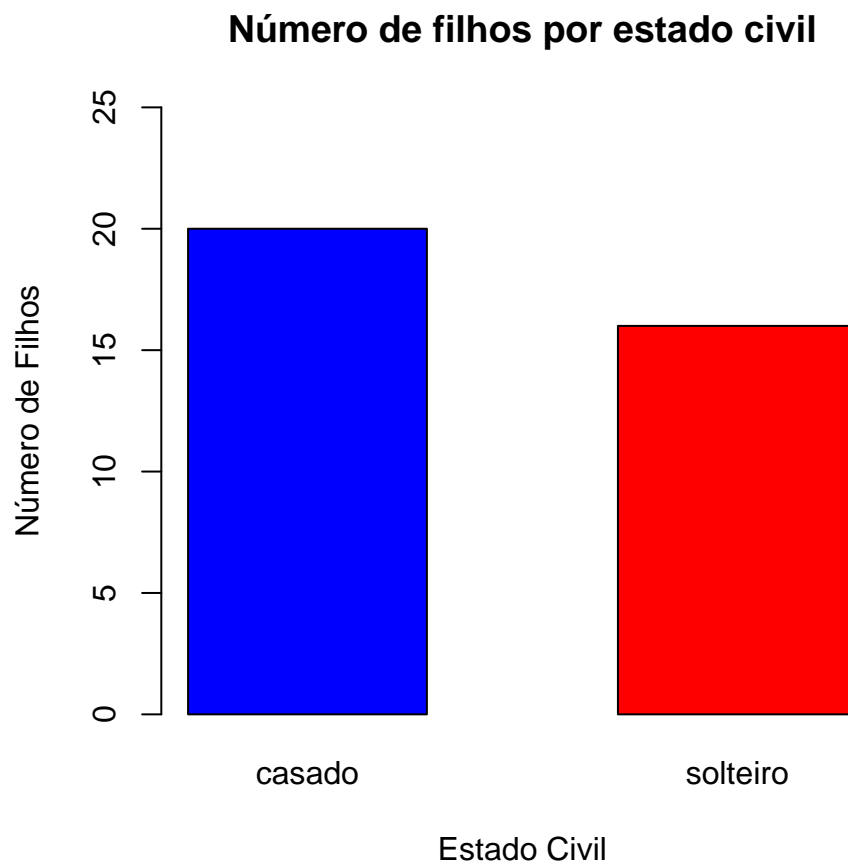


Figura 5: Gráfico de barras.

7.2 Histograma

7.3 Boxplot

7.4 Gráfico de Pizza

7.5 Gráfico de Dispersão

8 Probabilidade

9 Exemplos

Dividindo as observações em intervalos

```
salario <- c(12, .4, 5, 2, 50, 8, 3, 1, 4, .25)
intervalo <- cut(salario,breaks=c(0,1,5,max(salario)))
```

```
intervalo
```

```
[1] (5,50] (0,1] (1,5] (1,5] (5,50] (5,50] (1,5] (0,1] (1,5] (0,1]  
Levels: (0,1] (1,5] (5,50]
```

Frequência de observações em cada intervalo

```
table(intervalo)
```

```
intervalo  
(0,1] (1,5] (5,50]  
      3      4      3
```

```
intervalo
```

```
[1] (5,50] (0,1] (1,5] (1,5] (5,50] (5,50] (1,5] (0,1] (1,5] (0,1]  
Levels: (0,1] (1,5] (5,50]
```

Rotulando os intervalos

```
levels(intervalo) <-c("pobre","rico","rolando na grana")  
table(intervalo)
```

```
intervalo  
      pobre      rico  rolando na grana  
         3         4         3
```

Criando variáveis e exibindo-as em tabelas

```
Peso <- c(60, 75, 55, 68)  
Altura <- c(65, 61, 70, 65)  
Genero <- c("Fe","Fe","M","Fe")  
estudo <- data.frame(Peso,Altura,Genero)  
estudo
```

```
  Peso Altura Genero  
1   60    65    Fe  
2   75    61    Fe  
3   55    70     M  
4   68    65    Fe
```

Alterando os rótulos das variáveis

```
estudo = data.frame(P=Peso,H=Altura,G=Genero)
row.names(estudo)<-c("Maria","Alice","André","Julia")
estudo
```

```
      P  H  G
Maria 60 65 Fe
Alice 75 61 Fe
André 55 70  M
Julia 68 65 Fe
```

Histogramas com curvas teóricas

- Uniforme

```
x=runif(100)
hist(x,probability=TRUE,main="Uniforme[0,1]",
     ylab="Densidade",col="yellow")
curve(dunif(x,0,1),add=T)
```

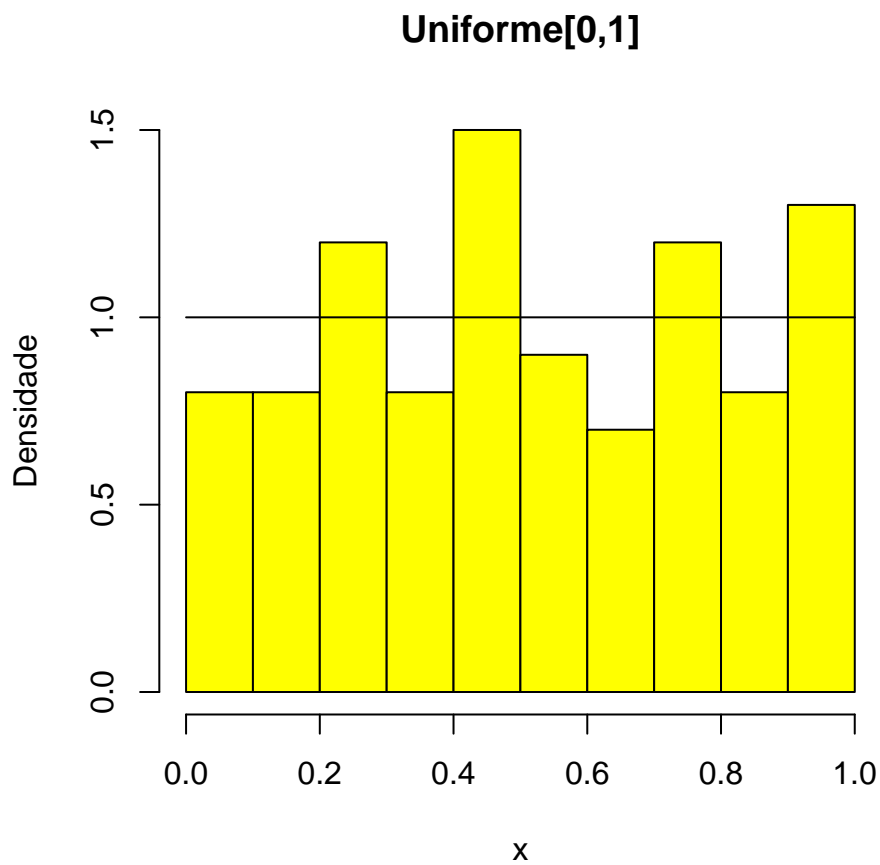


Figura 6: Gráfico de distribuição uniforme.

- Binomial

```
n=10;p=0.25
x=rbinom(100,n,p)
hist(x,probability=TRUE,ylab="Densidade",
col="pink", main="Binomial",ylim=c(0,0.30))
xvalores=0:n
points(xvalores,dbinom(xvalores,n,p),type="h",
lwd=3)
points(xvalores,dbinom(xvalores,n,p),type="p",
lwd=3)
```

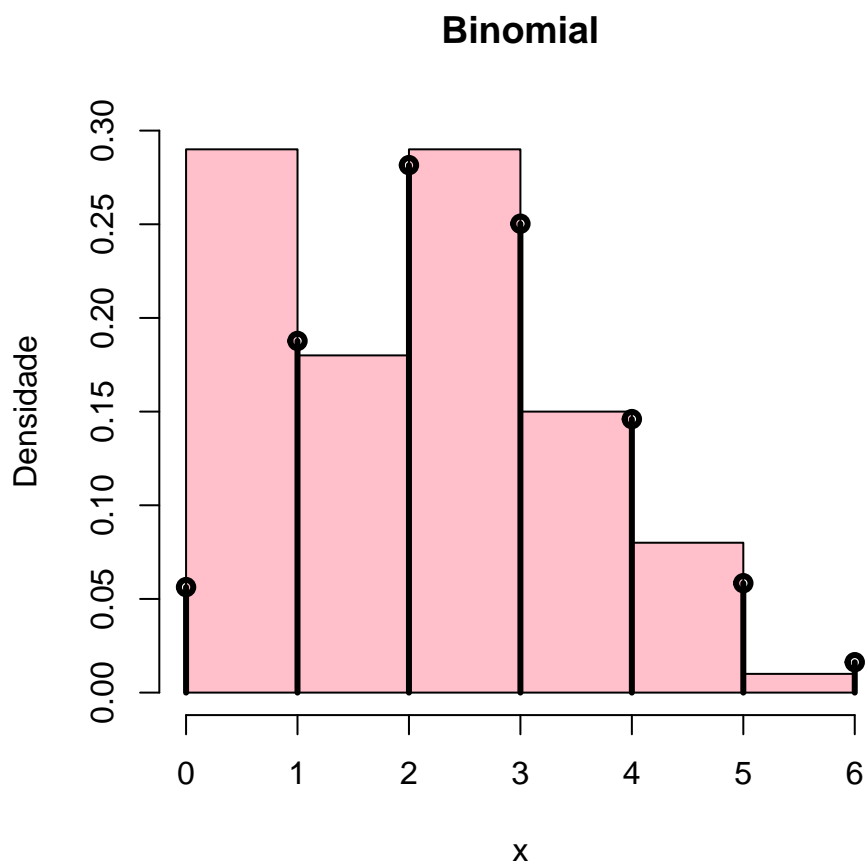


Figura 7: Gráfico de distribuição binomial.

- Normal

```
x=rnorm(100)
hist(x,probability=TRUE,col="lightblue",
main="Normal(0,1)",ylab="Densidade",
ylim=c(0,0.5),xlim=c(-3,3))
```



```
curve(dnorm(x),add=T)
```

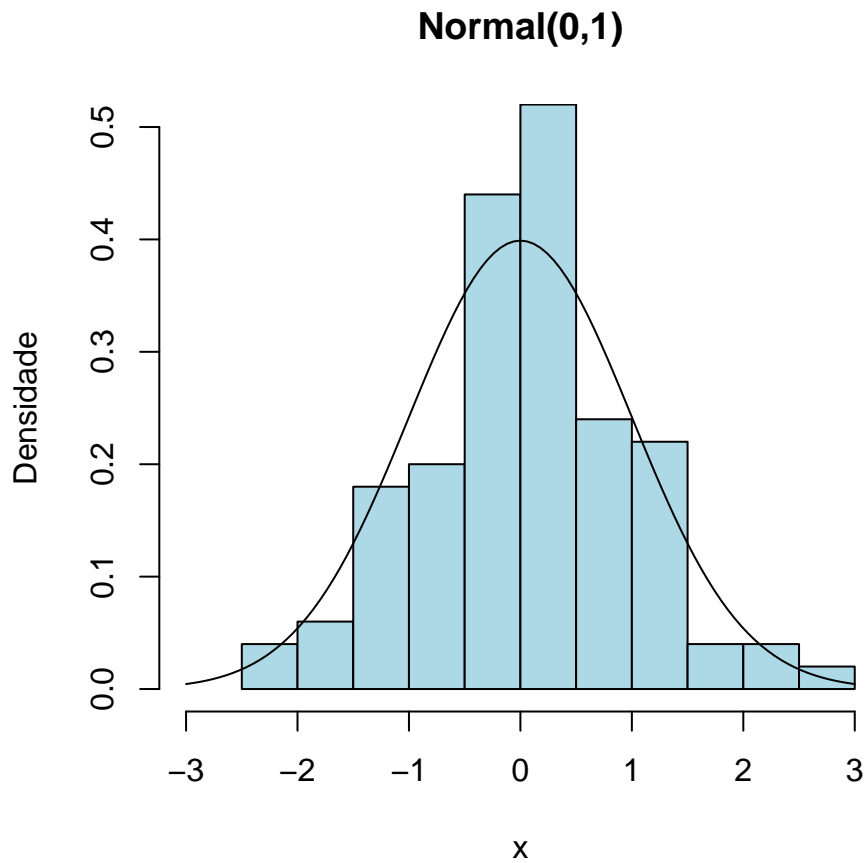


Figura 8: Gráfico de distribuição Normal.

- Exponencial

```
x=rexp(100,1/2500)
hist(x,probability=TRUE,
col="lightgreen",main="Exponencial com média=2500",ylab="Densidade")
curve(dexp(x,1/2500),add=T)
```

Exponencial com média=2500

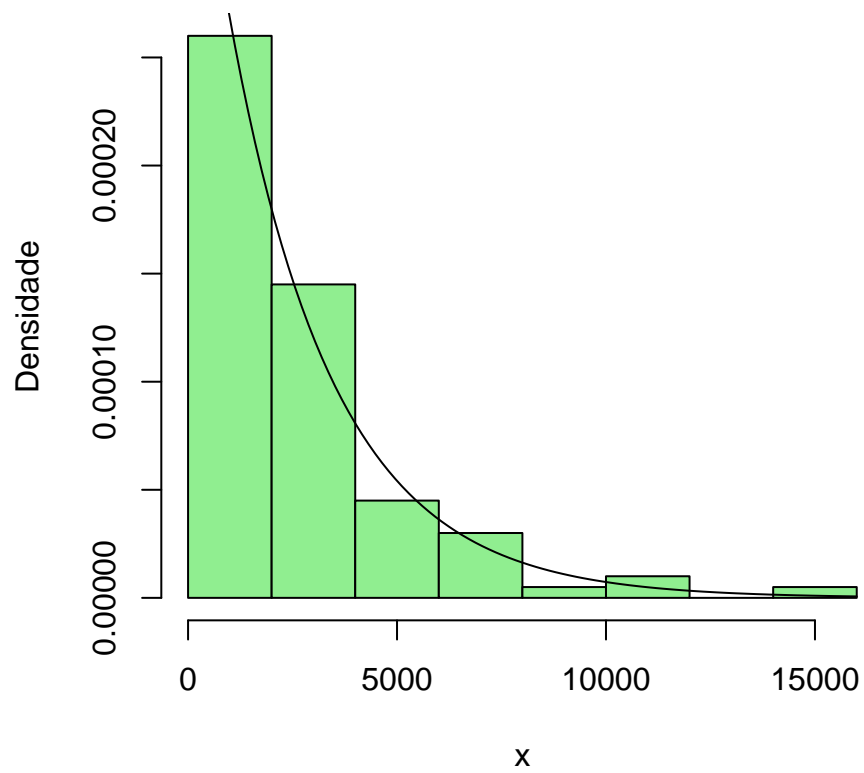


Figura 9: Gráfico de distribuição exponencial.

Referências

FOX, J.; WEISBERG, S.; PRICE, B. **car: Companion to Applied Regression**. [s.l: s.n.].
R CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna,
Austria: R Foundation for Statistical Computing, 2023.