

Estatística Computacional I

Lupércio França Bessegato
Dep. de Estatística/UFJF

Gráficos em R



Roteiro do Módulo



3. Gráficos em R:

- a) Introdução
- b) Gráficos e argumentos padrão
- c) Personalização de gráficos tradicionais
- d) Controle de aparência dos gráficos
- e) Criação de novos gráficos
- f) Gráficos dinâmicos
- g) Referências

Estatística Computacional I - 2020

4

Controle da Aparência dos Gráficos



Controle da Aparência de Gráficos



- Aparência de gráficos:
 - ✓ Cor
 - ✓ Tipo de linha
 - ✓ Fonte
 - ✓ Etc.
- Ajuste dos parâmetros gráficos no R base
 - ✓ Comando `par()`.
 - ✓ Argumento em função *high level*.

Estatística Computacional I - 2020

126



- Exemplo – alteração de cor de gráfico:



```
✓ par(col = "red");  
  - Altera todas as gráficas subsequentes.  
✓ plot(..., col = "red");  
  - Altera apenas os parâmetros do gráfico construído  
  pelo comando em execução da função high level.  
✓ lines(..., col = "red");  
  - Idem, para funções low level.
```

Estatística Computacional I - 2020

127



- Aspectos da aparência dos gráficos:



- ✓ Grupos de argumentos de funções gráficas afetam aspectos específicos.
 - Mais de uma função ajusta parâmetros de cor dos gráficos, por exemplo.

Estatística Computacional I - 2020

128



- Argumentos de ajuste de cores:



```
✓ col,  
✓ col.axis,  
✓ col.lab,  
✓ col.main,  
✓ col.sub,  
✓ bg,  
✓ gamma,  
✓ etc.
```

Estatística Computacional I - 2020

129

✓ Argumento `col`:

- Controla cores de símbolos, linhas, texto, etc.
- Efeito pode variar conforme função:
 - `plot()`: cor de símbolos e linhas da região do gráfico
 - `barplot()`: cor do interior das barras
 - `rect()`: cor do interior retângulo
- Para colorir outras regiões da figura:
 - `col.axis`: cor da anotação dos eixos
 - `col.lab`: cor dos rótulos dos eixos
 - `col.main`: cor da fonte do título
 - `col.sub`: cor da fonte do subtítulo

130

✓ Argumento `bg`:

- Na função `par()`:
 - Cor de preenchimento de todo o fundo da figura.
- Nas funções `plot()` e `points()`:
 - Cor de preenchimento do interior dos símbolos.

✓ Argumento `fg`:

- Na função `par()`:
 - Cor dos eixos e dos limites da região do gráfico.

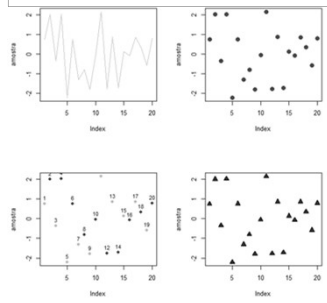
✓ Argumento `gamma`:

- Controla correção gama para o dispositivo.

131

✓ Argumentos `col` e `bg` na função `plot()`:

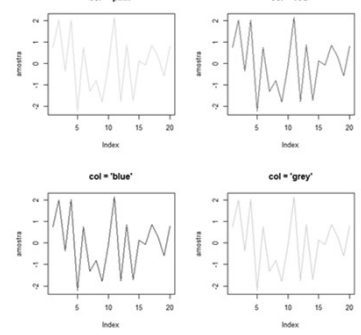
```
> # Argumentos col e bg na função plot
> amostra <- rnorm(20)
> par(mfrow = c(2, 2))
> plot(amostra, type = "l", col = "grey")
> plot(amostra, pch = 19, col = "darkgreen", cex = 1.5)
> plot(amostra, pch = 19, col = c("green", "red"))
> text(1:20, amostra, 1:20, cex = 0.8, pos = 3)
> plot(amostra, pch = 24, cex = 1.2, col = "blue", bg = "red", lwd = 2)
```



132

✓ Alguns ajustes do argumento `col`:

```
> # valores do argumento col
> par(mfrow = c(2, 2))
> for(i in c("pink", "red", "blue", "grey"))
+ plot(amostra, type = "l", col = i, main = paste0("col = ", i, "\n"))
```

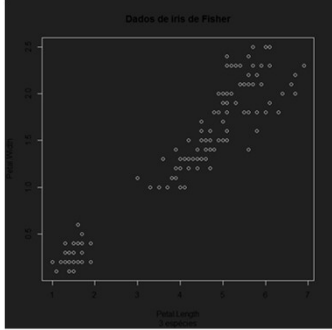


133

UFJF

✓ Aparência das cores da região da figura:
– Fundo de toda a figura e limites região do *plot*:

```
> # Controle de aparência região da figura - função par()
> # fundo da figura e linhas de limite entre regiões da figura
> par(bg = "midnightblue", fg = "yellow")
> plot(Petal.Width ~ Petal.Length, data = iris, main = "Dados de iris de Fisher",
+ sub = "3 espécies")
```



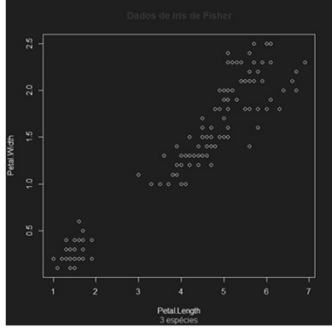
134

Estatística Computacional I - 2020

UFJF

✓ Aparência das cores da região da figura:
– Cores dos textos dos eixos e títulos:

```
> # cores textos de eixos e titulos
> par(bg = "midnightblue", fg = "yellow", col.lab = "white", col.main = "red",
+ col.sub = "green", col.axis = "bisque")
> plot(Petal.Width ~ Petal.Length, data = iris, main = "Dados de iris de Fisher",
+ sub = "3 espécies")
```



135

Estatística Computacional I - 2020

UFJF

• Códigos de cores do R base
✓ Função `colors()`:

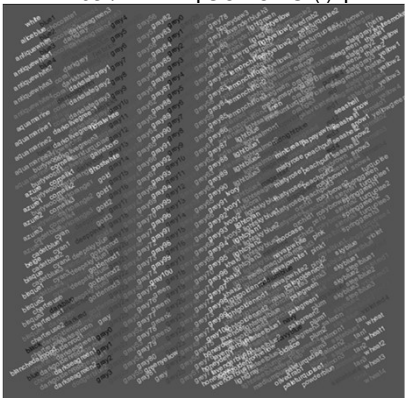
```
> # Códigos de cores - R base
> # Todas as cores do R base (n = 657)
> colors()
[1] "white" "aliceblue" "antiquewhite"
[4] "antiquewhite1" "antiquewhite2" "antiquewhite3"
[7] "antiquewhite4" "aquamarine" "aquamarine1"
[10] "aquamarine2" "aquamarine3" "aquamarine4"
[13] "azure" "azure1" "azure2"
[16] "azure3" "azure4" "beige"
[19] "bisque" "bisque1" "bisque2"
[22] "bisque3" "bisque4" "black"
...
> # Primeiras 60 cores
> colors()[1:60]
[1] "white" "aliceblue" "antiquewhite" "antiquewhite1"
[5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
[9] "aquamarine1" "aquamarine2" "aquamarine3" "aquamarine4"
[13] "azure" "azure1" "azure2" "azure3"
[17] "azure4" "beige" "bisque" "bisque1"
[21] "bisque2" "bisque3" "bisque4" "black"
[25] "blanchedalmond" "blue" "blue1" "blue2"
[29] "blue3" "blue4" "blueviolet" "brown"
[33] "brown1" "brown2" "brown3" "brown4"
...
```

136

Estatística Computacional I - 2020

UFJF

• Códigos de cores do R base:
✓ Todas as 657 cores [`colors()`]



137

Estatística Computacional I - 2020



- Códigos de cores do R base:
✓ Primeiras 60 cores (`colors()[1:60]`).



Estatística Computacional I - 2020



138



- Tabela com códigos de cores para o R:
– Códigos de cores sequenciais

R colors

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475
476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650
651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675

Estatística Computacional I - 2020

139



- Código de cores ordenados por valor de saturação (*HSV – hue-saturation-value*)

R colors – Sorted by Hue, Saturation, Value																								
24	153	261	154	262	155	263	156	264	157	265	158	266	159	267	160	268	161	269	162	270	163	271	164	272
165	273	166	274	167	275	168	276	169	277	170	278	171	279	172	280	173	281	174	282	175	283	176	284	177
178	285	179	286	180	287	181	288	182	289	183	290	184	291	185	292	186	293	187	294	188	295	189	296	190
191	297	192	298	193	299	194	300	195	301	196	302	197	303	198	304	199	305	200	306	201	307	202	308	203
204	309	205	310	206	311	207	312	208	313	209	314	210	315	211	316	212	317	213	318	214	319	215	320	216
217	321	218	322	219	323	220	324	221	325	222	326	223	327	224	328	225	329	226	330	227	331	228	332	229
231	333	232	334	233	335	234	336	235	337	236	338	237	339	238	340	239	341	240	342	241	343	242	344	243
245	345	246	346	247	347	248	348	249	349	250	350	251	351	252	352	253	353	254	354	255	355	256	356	257
259	357	260	358	261	359	262	360	263	361	264	362	265	363	266	364	267	365	268	366	269	367	270	368	271
273	369	274	370	275	371	276	372	277	373	278	374	279	375	280	376	281	377	282	378	283	379	284	380	285
287	381	288	382	289	383	290	384	291	385	292	386	293	387	294	388	295	389	296	390	297	391	298	392	299
301	393	302	394	303	395	304	396	305	397	306	398	307	399	310	400	401	402	403	404	405	406	407	408	409
410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434
435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459
460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484
485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509
510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534
535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559
560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584
585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609
610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634
635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659
660	661	662	663	664																				

144

143

144

145

• Pacote RColorBrewer – palette RdBu:

```
> # barplot usando RColorBrewer
> barplot(c(2,5,7), col = brewer.pal(n = 3, name = "RdBu"))
```

146

Estatística Computacional I - 2020

• Pacote wesanderson – palettes:

✓ BottleRocket1	✓ FantasticFox1	
✓ BottleRocket2	✓ Moonrise1	
✓ Rushmore1	✓ Moonrise2	
✓ Royal1	✓ Moonrise3	
✓ Royal2	✓ Cavalcanti1	
✓ Zissou1	✓ GrandBudapest1	
✓ Darjeeling1	✓ GrandBudapest2	
✓ Darjeeling2	✓ IsleofDogs1	
✓ Chevalier1	✓ IsleofDogs2	

147

Estatística Computacional I - 2020

✓ Tipos de cores - pacote wesanderson:

```
> # Tipos de cores - pacote wesanderson
> library(wesanderson)
> help(package = "wesanderson")
> wes_palette("GrandBudapest1")
> wes_palette("BottleRocket1")
```

148

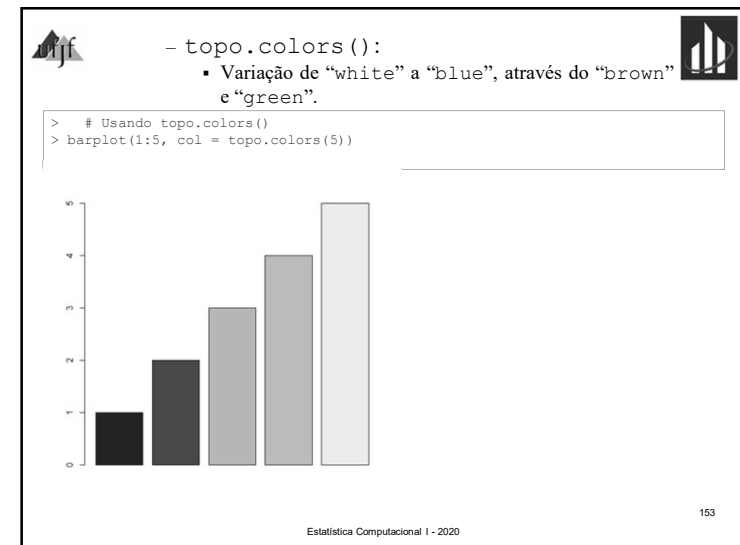
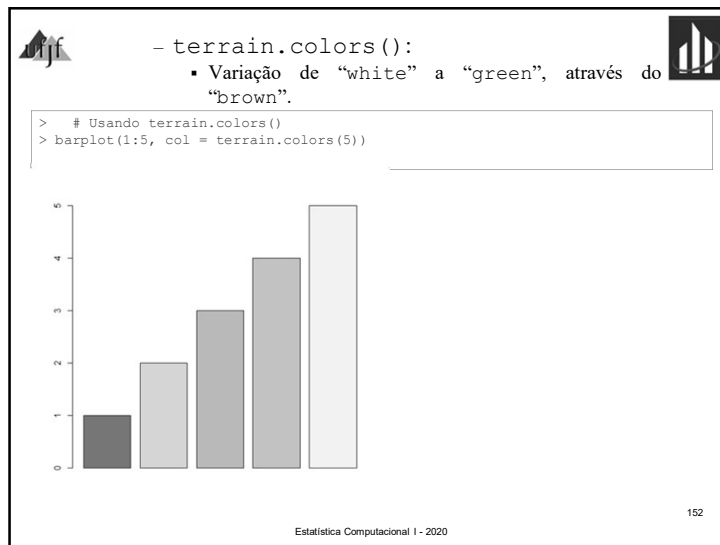
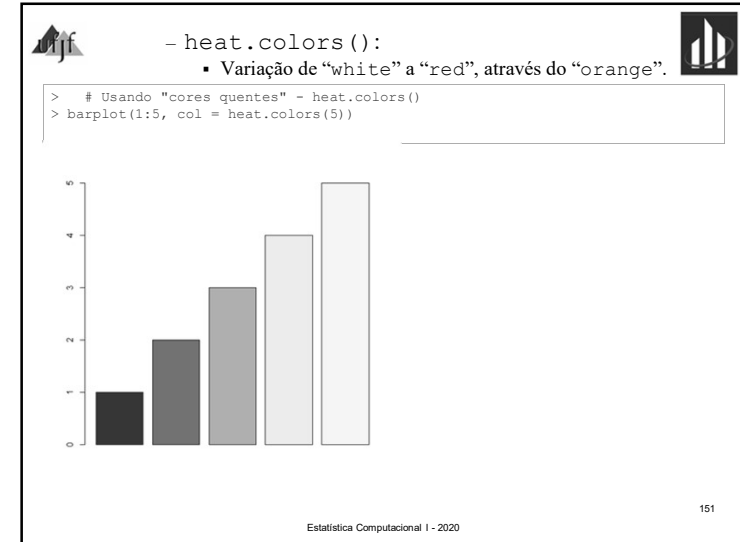
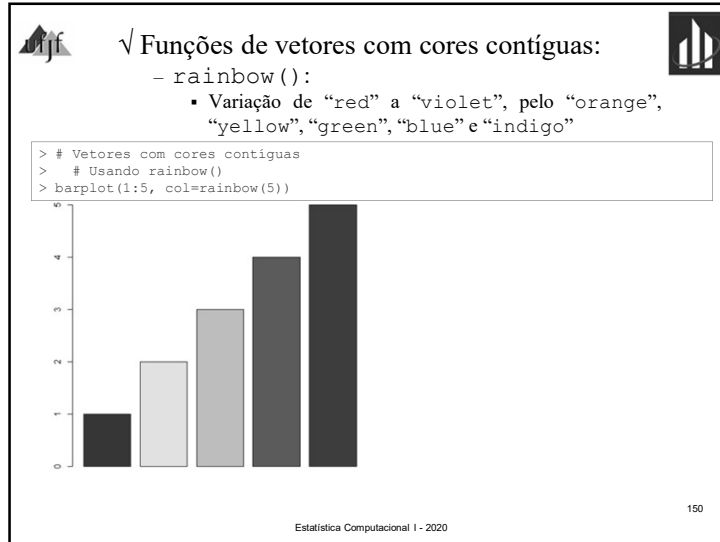
Estatística Computacional I - 2020

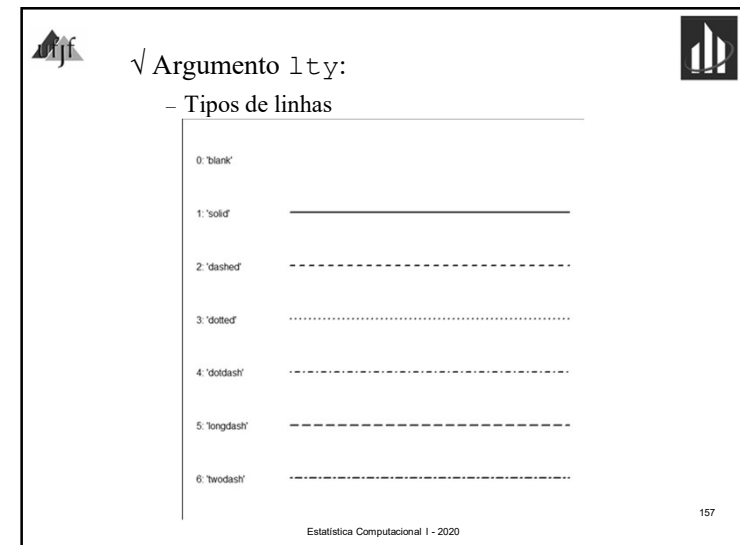
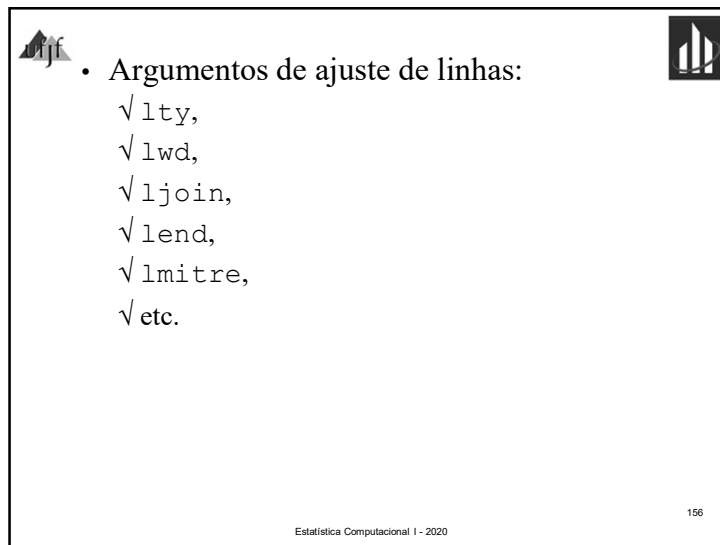
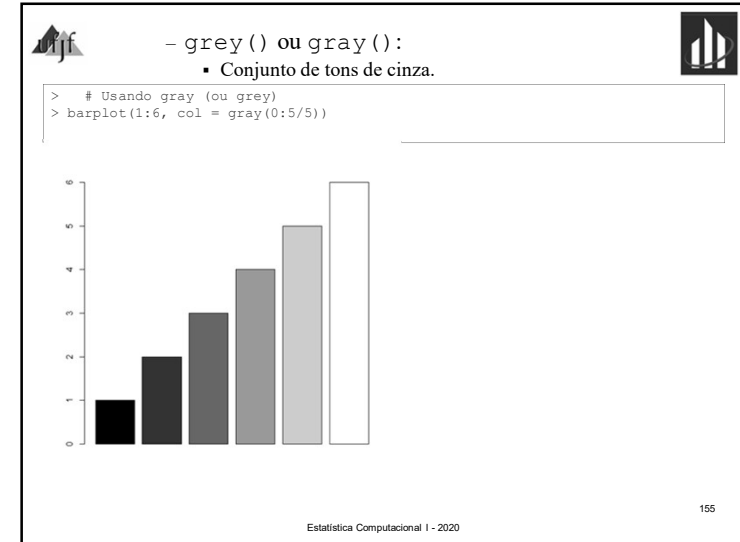
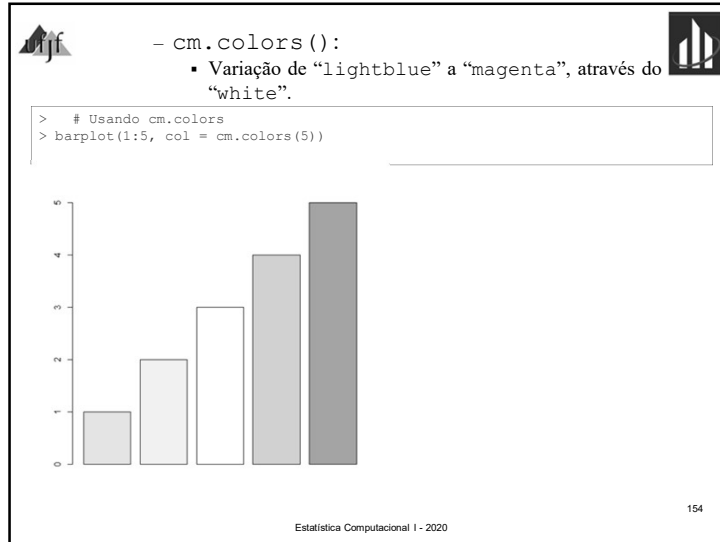
✓ Colorindo com pacote wesanderson:

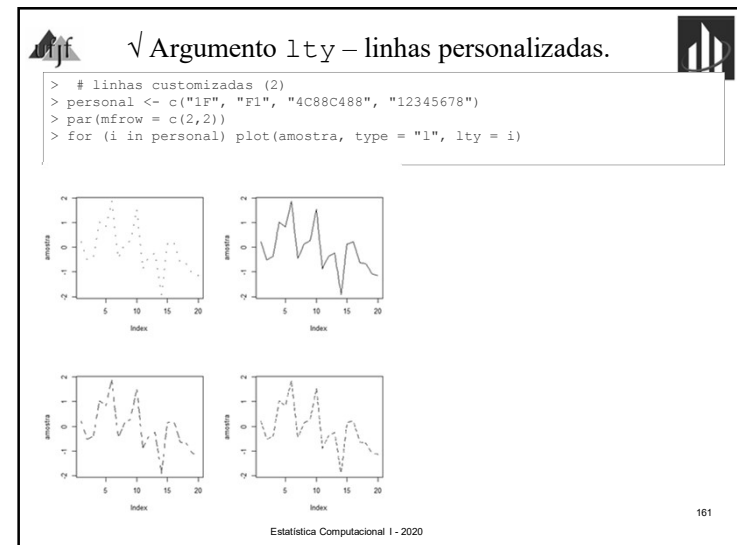
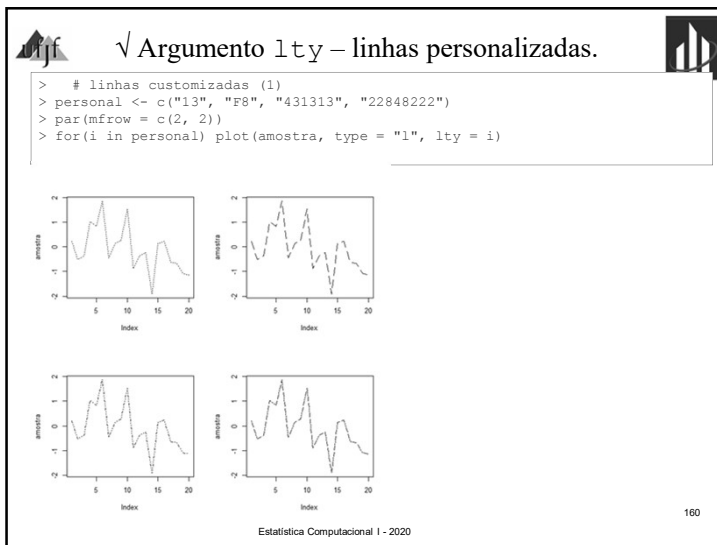
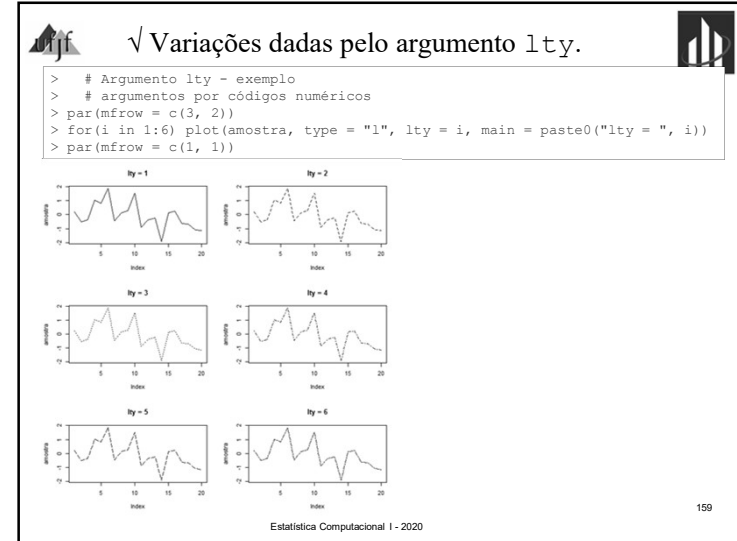
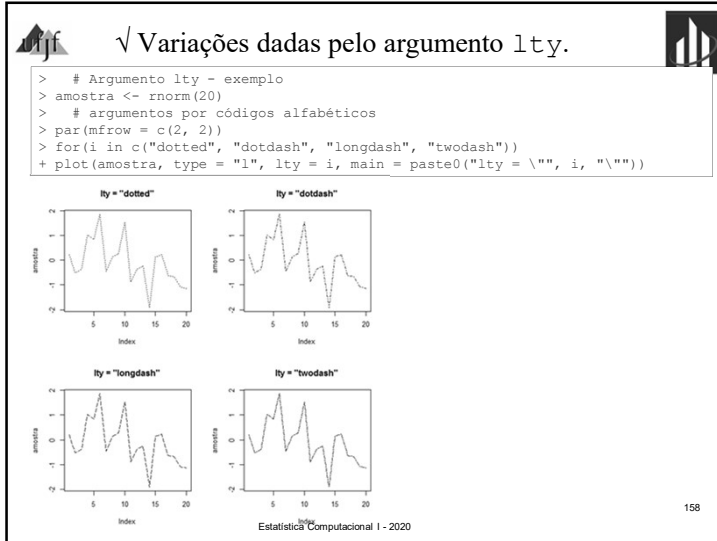
```
> # barplot usando wesanderson
> barplot(c(2,5,7), col = wes_palette(n = 3, name = "GrandBudapest1"))
> # plot usando wesanderson
> plot(Petal.Width ~ Petal.Length, data = iris, pch = 19, cex = 1.2,
+ col = wes_palette(n = 3, name = "GrandBudapest1")[as.factor(Species)])
```

149

Estatística Computacional I - 2020

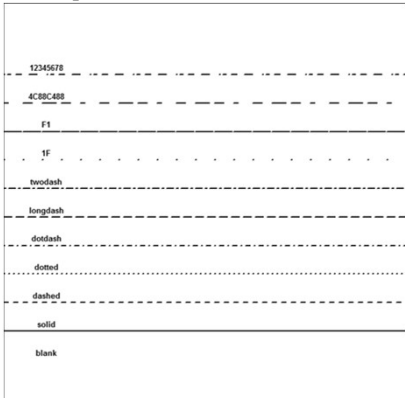






✓ Argumento `lty`:

- Mais tipos de linhas



162

Estatística Computacional I - 2020

✓ Argumento `lwd`:

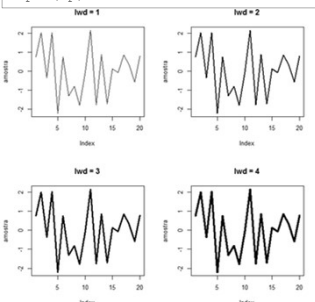
- Largura de linha
- Valor: multiplicador largura padrão (positivo).
- *Default*: 1.
- Interpretação depende do dispositivo
 - Alguns dispositivos não implementam larguras menores que 1.

163

Estatística Computacional I - 2020

✓ Gráficos com vários ajustes de `lwd`:

```
> # argumento lwd
> # geração de dados
> amostra <- rnorm(20)
> # gráficos com larguras diferentes
> op <- par(no.readonly = TRUE)
> par(mfrow = c(2, 2))
> for(i in 1:4) plot(amostra, type = "l", lwd = i, main = paste0("lwd = ", i))
> par(op)
```



164

Estatística Computacional I - 2020

• Argumentos `ljoin`, `lend`, `lmitre`:


✓ Estilos de junções e extremidades de linhas.

- Importante ao desenhar linhas grossas

✓ Controle de estilos fornecidos pelo R:


- `lend`: controle das extremidades da linha
 - 0: "round" (*default*), 1: "square", 2: "butt"

	0	1	2
"round"	<code>plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lwd=5, lend=0)</code>		
"butt"	<code>plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lwd=5, lend=1)</code>		
"square"	<code>plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lwd=5, lend=2)</code>		

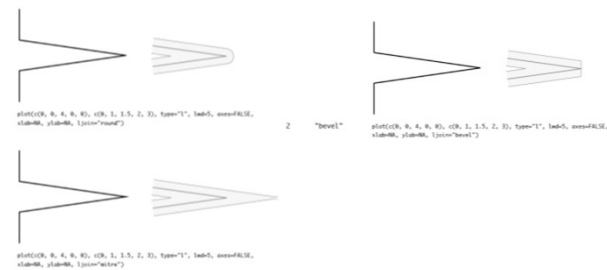


165

Estatística Computacional I - 2020

 - `ljoin`: controle da junção de linhas

- 0: "round" (*default*), 1: "mitre" (pontuda), 2: "bevel" (chanfrado).
- Estilo de junção será convertido de "mitre" para "bevel" se ângulo da junção for muito pequeno.




0 "round" `plot(c(0, 8, 4, 8, 8), c(0, 1, 1.5, 2, 3), type="l", lwd=5, yaxp=10, xlab="x", ylab="y", ljoin="round")`

1 "mitre" `plot(c(0, 8, 4, 8, 8), c(0, 1, 1.5, 2, 3), type="l", lwd=5, yaxp=10, xlab="x", ylab="y", ljoin="mitre")`


2 "bevel" `plot(c(0, 8, 4, 8, 8), c(0, 1, 1.5, 2, 3), type="l", lwd=5, yaxp=10, xlab="x", ylab="y", ljoin="bevel")`

Estatística Computacional I - 2020 166

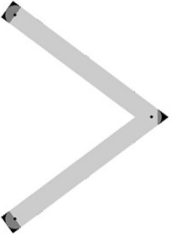
 - `lmitre`: configuração ponto de conversão

- Proporção do comprimento da ponta, dividida por largura da linha
- *Default*: 10 – conversão em ângulos de junção $< 11^\circ$

Estatística Computacional I - 2020 167


 ✓ Estilos de união e de extremidade:

- Especificados pela função `par()`.
- Influenciam vértices de retângulos e polígonos e junções de linha



- ✓ Linha preta:
 - Extremidade "square"
 - Junção: "mitre"
- ✓ Linha cinza escura:
 - Extremidade: "round"
 - Junção: "round"
- ✓ Linha cinza claro:
 - Extremidade: "butt"
 - Junção: "bevel"

Estatística Computacional I - 2020 168

 • Argumentos de ajuste de texto:

- ✓ `ps`,
- ✓ `cex`,
- ✓ `cex.axis`, `cex.main` e `cex.sub`,
- ✓ `font`,
- ✓ `family`,
- ✓ `adj`,
- ✓ `str`,
- ✓ `ann`,
- ✓ `etc`.

Estatística Computacional I - 2020 169



✓ Argumento cex:

- cex: especifica multiplicador tamanho da fonte
 - $cex = tamanho_fonte * ps.$
- ps: especifica tamanho absoluto da fonte

✓ Argumento cex :

- No comando `par()`:
 - Afeta a maioria dos textos
- No comando `plot()`:
 - Afeta tamanho dos símbolos dos dados

✓ Alteração tamanho da fonte em outras regiões

- `cex.axis`: texto dos eixos
- `cex.lab`: texto rótulos dos eixos
- `cex.main` e `cex.sub`: títulos

Estatística Computacional I - 2020

170



• Argumento cex em função `par()`:

```
> # cex em comando par()
> op <- par(no.readonly = TRUE)
> par(cex.main = 4, cex.lab = 3, cex.axis = 2, cex.sub = 0.7)
> plot(x = 1:12, y = c(1:10, 10, 10), type="n", xlab = "X", ylab = "Y",
+ main = "Título", sub = "PEQUETITO")
> for (i in 1:10) text(i, i, "O que rolou no RU hoje?", cex = i/10)
> par(op)
```



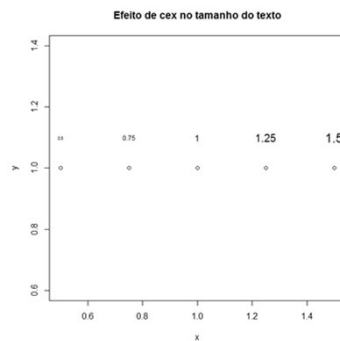
Estatística Computacional I - 2020

171



• Argumento cex em função `text()`:

```
> # cex em comando text()
> x <- seq(0.5, 1.5, 0.25)
> y <- rep(1, length(x))
> plot(y ~ x, main = "Efeito de cex no tamanho do texto")
> text(x, y + 0.1, labels = x, cex = x)
```



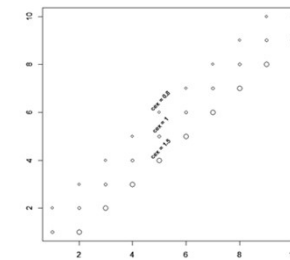
Estatística Computacional I - 2020

172





• Argumento cex em função `plot()`:

```
> # cex em comando plot()
> plot(1:10, cex = 1, xlab = "", ylab = "")
> par(new = T)
> plot(1:10, 2:11, cex = 0.8, ylim = c(1, 10), xlab = "", ylab = "")
> par(new = T)
> plot(1:10, 0:9, cex = 1.5, ylim = c(1, 10), xlab = "", ylab = "")
> text(5, 6 + 0.5, "cex = 0.8", cex = 0.8, srt = 45, font = 2)
> text(5, 5 + 0.5, "cex = 1", cex = 0.8, srt = 45, font = 2)
> text(5, 4 + 0.5, "cex = 1.5", cex = 0.8, srt = 45, font = 2)
```



Estatística Computacional I - 2020

173

✓ Argumento `font`:

- Atua apenas na região do gráfico (*plot region*).

✓ Argumento `font.axis`:

- Ajuste da fonte de texto dos eixos



✓ Argumento `font.label`:

- Ajuste da fonte de texto dos rótulos dos eixos.

✓ Argumento `font.main` e `font.sub`:

- Ajuste da fonte de texto dos títulos.



174

✓ Estilos de fontes *default*:

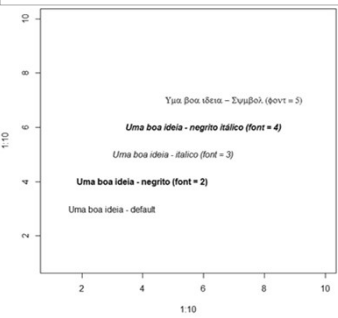
Inteiro	Descrição
<i>Fontes independentes de dispositivo</i>	
1	Estilo romano
2	Estilo negrito
3	Estilo itálico
4	Estilo negrito e itálico
5	Estilo símbolo
<i>Para a família HersheySerif</i>	
5	Fonte cirílico
6	Fonte cirílico itálico
7	Caracteres japoneses

175






✓ Argumento `font` – texto na região do gráfico:

```
> # argumento font
> plot(1:10, 1:10, type = "n")
> text(3, 3, font = 1, "Uma boa ideia - default")
> text(4, 4, font = 2, "Uma boa ideia - negrito (font = 2)")
> text(5, 5, font = 3, "Uma boa ideia - italico (font = 3)")
> text(6, 6, font = 4, "Uma boa ideia - negrito itálico (font = 4)")
> text(7, 7, font = 5, "Uma boa ideia - Symbol (font = 5)")
```



176

✓ Argumento `family`:

- Especificado apenas pela função `par()`.
- Tabela de famílias de fontes (depende dispositivo)

Nome	Descrição
<i>Fontes independentes de dispositivo</i>	
"serif"	Fonte de largura variável <i>Serif</i>
"sans"	Fonte de largura variável <i>Sans-serif</i> .
"mono"	Fonte de largura única <i>typewriter</i>
<i>Fontes Hershey</i>	
"HersheySerif"	Fonte de largura variável <i>Serif</i>
"HersheySans"	Fonte de largura variável <i>Sans-serif</i>
"HersheyScript"	Fonte caligráfica <i>Serif</i>
"HersheyGothicEnglish"	Fonte manuscrita gótica
"HersheyGothicGerman"	Fonte manuscrita gótica
"HersheyGothicItalian"	Fonte manuscrita gótica
"HersheySymbol"	Fonte símbolo <i>Serif</i>
"HersheySansSymbol"	Fonte símbolo <i>Sans-serif</i>

177

✓ Aparência combinações de famílias e fontes:

family = "mono" font = 1 family = "mono" font = 2 family = "mono" font = 3 family = "mono" font = 4

family = "serif" font = 1 family = "serif" font = 2 family = "serif" font = 3 family = "serif" font = 4

family = "sans" font = 1 family = "sans" font = 2 family = "sans" font = 3 family = "sans" font = 4

Estatística Computacional I - 2020 178

✓ Alguns tipos de famílias – Windows:
– Mapeamento depende do dispositivo

```
> # Exemplo de tipos de famílias - novo mapeamento
> plot(1:10, 1:10, type = "n")
> windowsFonts(A = windowsFont("Arial Black"), B = windowsFont("Bookman Old
+ Style"), C = windowsFont("Comic Sans MS"), D = windowsFont("Symbol"))
> text(3, 3, "Uma boa ideia - default")
> text(4, 4, family = "A", "Uma boa ideia - Arial Black")
> text(5, 5, family = "B", "Uma boa ideia - Bookman Old Style")
> text(6, 6, family = "C", "Uma boa ideia - Comic Sans MS")
> text(7, 7, family = "D", "Uma boa ideia - Symbol")
```

Estatística Computacional I - 2020 179

✓ Argumento adj:

- Controla justificação do texto
- Valores
 - 0: justificação à esquerda
 - 1: justificação à direita
 - 0.5: centralizado
- Na região do gráfico (*plot region*):
 - Justificação em referência à locação pontos (x , y).
- Possível especificar dois valores
 - (vertical, horizontal).
- Podem ser especificados valores não finitos (NA, NaN, Inf)
 - Consideradas centralizações exatas

Estatística Computacional I - 2020 180

– Justificação vertical:

- 0.5: centralização vertical baseada na altura total abaixo da linha de base do texto (ignora descendentes com cauda do “y”)
- Não-finito: centralização baseada na altura completa do texto

Estatística Computacional I - 2020 181

✓ Justificação na região do gráfico (a_x , a_y):

✓ a_x : justificação horizontal:

- 0: à esquerda
- 1: à direita
- 0.5: centralizado

✓ a_y : justificação vertical:

- 0: abaixo
- 1: acima
- 0.5: centralizado

Estadística Computacional I - 2020

182

✓ Argumento `str`:

- Ajusta rotação do texto, especificando ângulo de rotação anti-horário, em graus
- Atua apenas na região do gráfico

```
> # argumento str
> par(mar = rep(0, 4), xaxt = "n", yaxt = "n", font = 2, cex = 0.9)
> plot(NA, xlim = c(0, 1), ylim = c(0, 1))
> text(0.1, 0.1, "Rotação = 0", srt = 0)
> text(0.3, 0.3, "Rotação = 45", srt = 45)
> text(0.5, 0.5, "Rotação = 90", srt = 90)
> text(0.7, 0.7, "Rotação = 180", srt = 180)
> text(0.9, 0.9, "Rotação = 270", srt = 270)
```

Estadística Computacional I - 2020

183

✓ Argumentos `ps`:

- Especifica tamanho absoluto da fonte
- Valor *default*: 12 pontos
- Resultado depende do dispositivo gráfico

✓ Argumento `cex`:

- Não é definido em unidades do tamanho da fonte
- *Default* é 1 na abertura de dispositivo gráfico.

Estadística Computacional I - 2020

184

✓ Tamanho de fontes de elementos gráficos:

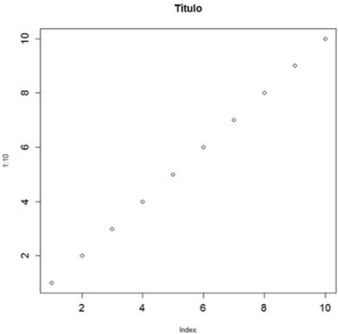
```
> # argumentos ps
> # default do tamanho da fonte
> par("ps")
[1] 12
> # tamanho da fonte dos rótulos dos eixos em 18.52 pontos
> par(mfrow = c(1,2))
> # tamanho da fonte dos rótulos dos eixos em 10 pontos
> plot(1:10, cex.lab = 10/12)
> # tamanho da fonte dos rótulos dos eixos em 18.52 pontos
> plot(1:10, cex.lab = 18.52/12)
```

Estadística Computacional I - 2020

185

✓ Modificação tamanho de fonte `type`.

```
> # tamanho da fonte dos rótulos dos eixos em 10 pts e título 14 pts
> par(ps = 14, cex = 1, cex.main = 1, cex.lab = 10/14)
> plot(1:10, main = "Titulo")
```



Estadística Computacional I - 2020

186

✓ Especificação tamanho da fonte depende do dispositivo gráfico:

- `pdfFonts()`.
- `pdf()`.
- `embedFonts()`.
- etc.

```
> # Tamanho de fonte em dispositivo gráfico
> caminho <- getwd()
> pdf(file.path(caminho, "outputs", "minhaFig.pdf"), pointsize = 10)
> plot(Petal.Width ~ Petal.Length, data = iris, main = "Minha Figura")
> dev.off()
windows
2
```

Estadística Computacional I - 2020

187

✓ Argumento `ann`:

- Especifica a impressão de título em gráfico
- Valores: `TRUE` (*default*); `FALSE`.
- Não funciona em todas as funções gráficas *high-level*
 - Especialmente em funções de pacotes gráficos

Estadística Computacional I - 2020

188

• Argumentos de ajuste de símbolos dos dados:

- ✓ `type`,
- ✓ `pch`,
- ✓ etc.

Estadística Computacional I - 2020

189

✓ Argumento `type`:
– Estilos de linhas

Tipo	Descrição
"p"	Símbolo em cada par ordenado (x , y) (<i>default</i>)
"l"	Pares ordenados (x , y) conectados por linhas.
"b"	Símbolos e linhas conectando os pares ordenados (x , y)
"o"	Linhas sobrepostas nos símbolos conectando os pares ordenados (x , y)
"h"	Linhas verticais ligando ao eixo x cada par ordenado (x , y)
"s"	Pares ordenados (x , y) ligados por linhas horizontais e então verticais (<i>city-block</i>)
"S"	Pares ordenados (x , y) ligados por linhas verticais e então horizontais
"n"	Nada é plotado

Estadística Computacional I - 2020 190

✓ Variações dadas pelo argumento `type`.

```
> # comando plot() - argumento type
> par(mfrow = c(4, 2), mar = rep(0.7, 4), xaxt = "n", yaxt = "n", cex = 0.8,
+ cex.main = 0.9, bty = "l")
> y <- rnorm(20)
> for(i in c("p", "l", "b", "o", "h", "s", "S", "n")){
+ plot(y, type = i)
+ title(main = paste0("type = \"", i, "\""))
+ }
```

Estadística Computacional I - 2020 191

✓ Argumento `pch`:
– Tipos de símbolos

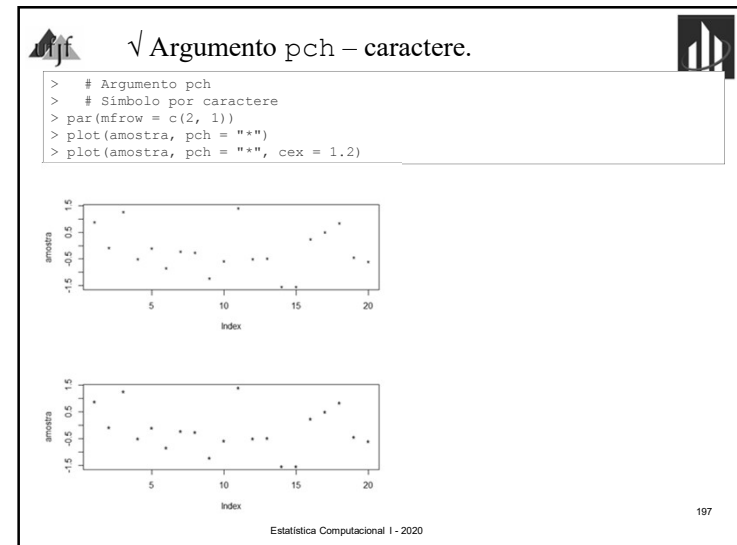
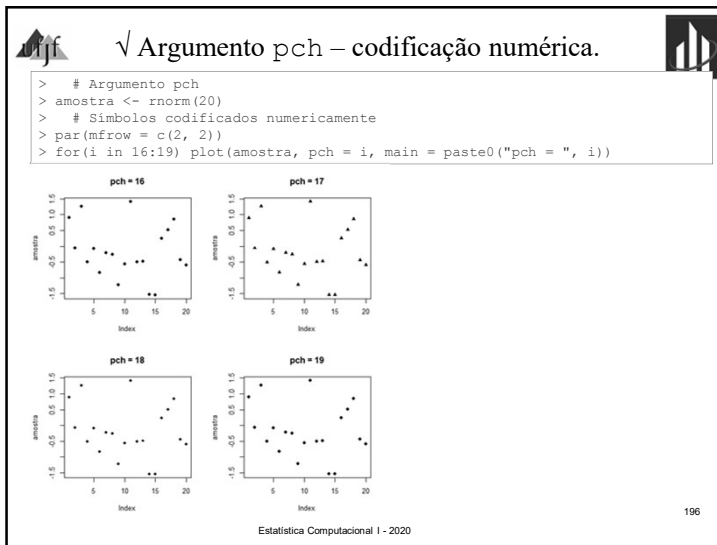
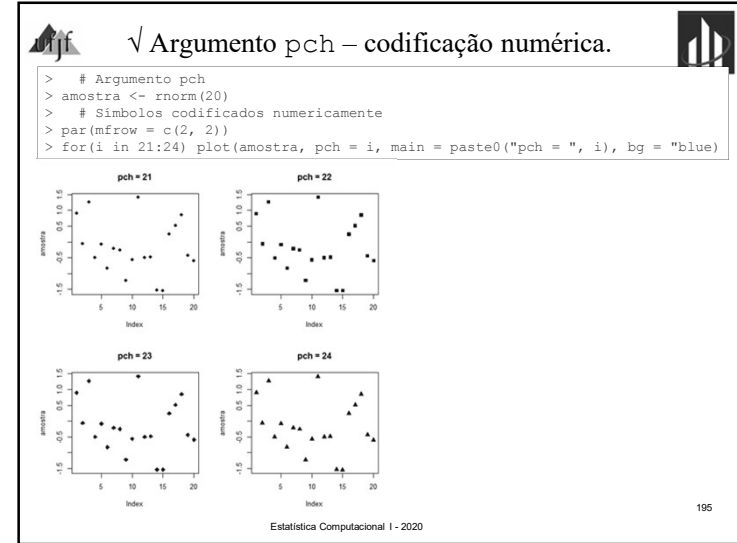
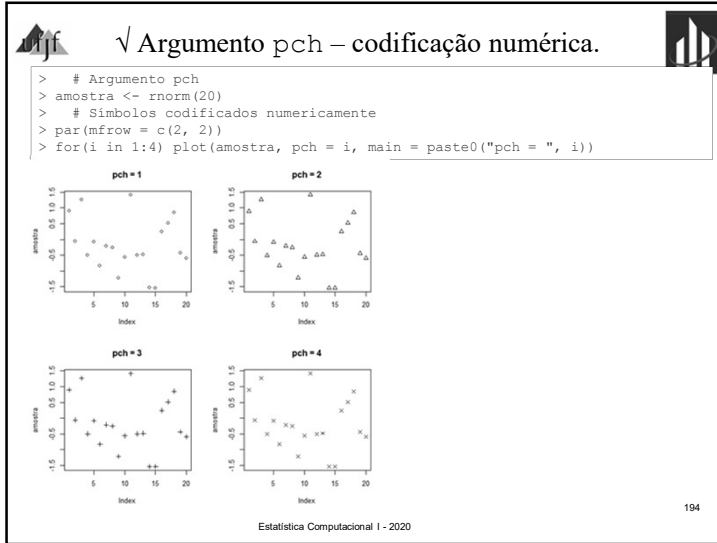
0 □	1 ○	2 △	3 +	4 ×
5 ◇	6 ▽	7 □	8 *	9 ⊕
10 ⊗	11 ⊠	12 ⊞	13 ⊞	14 ⊞
15 ■	16 ●	17 ▲	18 ◆	19 ●
20 *	21 *	22 *	23 *	24 *
				25 ▼

Estadística Computacional I - 2020 192

• Comentários:

- ✓ 26 símbolos para *plot* dos dados
 - Valor pode ser número inteiro
 - Símbolos de 21 a 25 permitem preenchimento
 - Argumento `bg`.
- ✓ Símbolo pode ser um caractere (único)
- ✓ Caractere ". ":
 - Tratado como caso especial
 - Dispositivo tenta desenhar ponto bem pequeno
- ✓ Tamanho do símbolo é afetado pelo ajuste `cex`.

Estadística Computacional I - 2020 193



UFJF

✓ Argumento `pch` – caractere ".".

```
> # Argumento pch
> amostra <- rnorm(1E5)
> # Símbolo do caractere "."
> par(mfrow = c(2, 1))
> plot(amostra)
> plot(amostra, pch = ".")
```

198

Estatística Computacional I - 2020

UFJF

• Argumentos de ajuste dos eixos:

- ✓ `lab`,
- ✓ `las`,
- ✓ `xaxp` e `yaxp`,
- ✓ `xaxs` e `yaxs`,
- ✓ `mgp`,
- ✓ `tck` e `tcl`,
- ✓ `xaxt` e `yaxt`
- ✓ `xlog` e `ylog`,
- ✓ `bty`,
- ✓ `etc`.

199

Estatística Computacional I - 2020

UFJF

• Função `axis()`:

- ✓ Função para construção dos eixos

• Argumentos da função `axis()`:

✓ `lab = c(nx, ny, len)`:

- Controle da quantidade de marcas de escala
- `nx` e `ny`: quantidade nos eixos `x` e `y`.
- `len`: comprimento dos rótulos (não implementado)
- `c(5, 5, 7)`: valores *default*.
- Afeta apenas a maneira que os parâmetros `xaxp` e `yaxp` são definidos quando é ajustado sistema de coordenadas personalizado
 - Não é executado quando os eixos estão desenhados.

200

Estatística Computacional I - 2020

UFJF

✓ Argumento `las`:

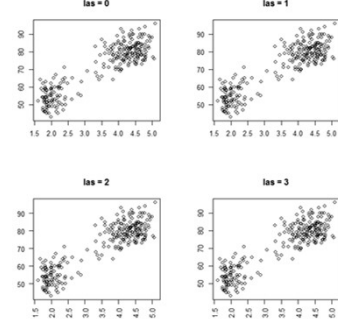
- Controla estilo dos rótulos do eixo, alterando o ângulo de orientação em relação aos eixos
- Valores possíveis:
 - 0: paralelo ao eixo (*default*).
 - 1: sempre horizontal.
 - 2: perpendicular ao eixo.
 - 3: sempre vertical

201

Estatística Computacional I - 2020

✓ Posicionamento rótulos – argumento `las`:

```
> # posicionamento rótulos eixos - argumento las
> op <- par(no.readonly = TRUE)
> par(mfrow = c(2, 2), cex.lab = 0.5)
> for(i in 0:3) plot(faithful, las = i, xlab = "", ylab = "",
+   main = paste0("las = ", i))
> par(op)
```



202

Estatística Computacional I - 2020

✓ Argumentos `xaxp` e `yaxp`:

- Relacionados com quantidade e posição das marcas de escala
- Valores do tipo `c(x1, x2, n)`:
 - Coordenadas das extremidades das marcas de escala.
 - Número de intervalos entre marcas de escala, quando `xlog = FALSE`.
 - `c(0, 1, 5)`: valor *default*.

203

Estatística Computacional I - 2020

✓ Argumentos `xaxs` e `yaxs`:

- Controle do estilo dos eixos
- Valores possíveis:
 - “r”: intervalo de valores no eixo é maior que intervalo dos dados sendo plotados (*default*).
(Símbolos dos dados não colidem com as fronteiras da região do *plot*).
 - “i”: intervalo de valores no eixo coincide com intervalo dos dados.
(Útil se intervalo de valores no eixo estiver sendo controlado explicitamente por `xlim` ou `ylim`).

204



Estatística Computacional I - 2020

✓ Argumento `mgp`:

- Controla número linhas nas margens para nome, rótulos e linhas do eixo (em unidades de `mex`).
- Valores do tipo `c(m1, m2, m3)`:
 - `m1`: controla posição nome dos eixos nas margens
 - `m2` e `m3`: controla eixo
- `c(3, 1, 0)`: valores *default*.
- `mex = 1`: multiplicador para coordenadas margens

205



Estatística Computacional I - 2020

✓ Argumento `tc1`:

- Comprimento das marcas de escala como altura de linhas de texto.
- Valores numéricos:
 - positivos: marcas de escala dentro da região do gráfico
 - negativos: marcas de escala fora da região do gráfico
- `-0.5`: valor *default*.



Estatística Computacional I - 2020 206

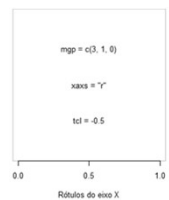
✓ Argumento `tc1`:

- Especifica comprimento da marcas de escala
- Fração da menor dos valores da largura ou do comprimento da região do gráfico
 - `tc1 ≥ 0.5`: Interpretado como fração do lado relevante
 - `tc1 = 1`: são desenhadas linhas de grade.
- `tc1 = NA`: valor *default*
 - Usado para `tc1 = -0.5`.

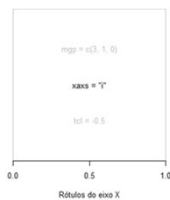
Estatística Computacional I - 2020 207

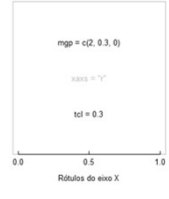
✓ Alguns estilos de eixos:



`mpg = c(3, 1, 0)`
`xaxs = "i"`
`tc1 = -0.5`
Rótulos do eixo X





`mpg = c(3, 1, 0)`
`xaxs = "n"`
`tc1 = -0.5`
Rótulos do eixo X



`mpg = c(2, 0.3, 0)`
`xaxs = "i"`
`tc1 = 0.3`
Rótulos do eixo X

Estatística Computacional I - 2020 208

✓ Argumentos `xaxt` e `yaxt`:

- Especifica tipo dos eixos
- Valores:
 - `"s"`: eixo é traçado (default para ambos os argumentos)
 - `"n"`: eixo não é traçado

✓ Argumentos `xlog` e `ylog`:

- Controla transformação logarítmica dos eixos
- Valores lógicos:
 - `FALSE`: eixo linear e valores não são transformados (Valor *default*)
 - `TRUE`: transformação log
Afeta cálculo das posições das marcas de escala

Estatística Computacional I - 2020 209

✓ Argumento `bty()`:

- Controla saída da função `box()`.
- Valores possíveis:
 - “n”: quadro não é desenhado
 - “o”, “l”, “7”, “c”, “j”: quadro desenhado de maneira semelhante ao caractere maiúsculo correspondente.
 - “o”: valor *default*.

210

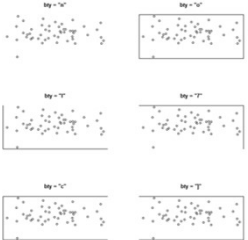
✓ Função `box()`:

- Desenha retângulo em torno dos limites do gráfico
- `box(which="plot", lty="solid", ...)`
- Valores de `which`:
 - “plot”, “figure”, “inner” ou “outer”

211

✓ Efeitos do argumento `bty`:

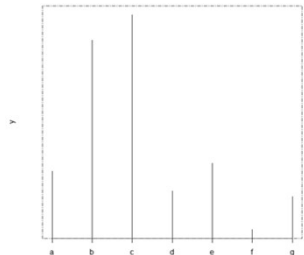
```
> # efeitos do argumento bty - função plot()
> # geração dos dados
> x <- rnorm(50)
> y <- rnorm(50)
> # tipos de quadros
> tipos <- c("n", "o", "l", "7", "c", "j")
> # figuras com tipos de quadro
> par(mfrow = c(3, 2))
> for(i in seq_along(tipos)){
+ plot(x, y, bty = tipos[i], xlab = "", ylab = "", xaxt = "n", yaxt = "n",
+ main = paste0("bty = \"", tipos[i], "\""))
+ }
```



212

✓ Modificação de características *default* de quadro em torno do gráfico:

```
> # Alteração de características default de quadro
> # plot() sem eixos
> plot(1:7, abs(rnorm(7)), type = "h", axes = FALSE, ylab = "y", xlab = "")
> # eixo x
> axis(1, at = 1:7, labels = letters[1:7])
> # quadro com tipo de linha e cor diferentes
> box(lty = 'l373', col = "red")
```



213

✓ Função `plot.window()`:

- Especifica sistema coordenado para janela gráfica
 - Atua após comando `plot.new()`.

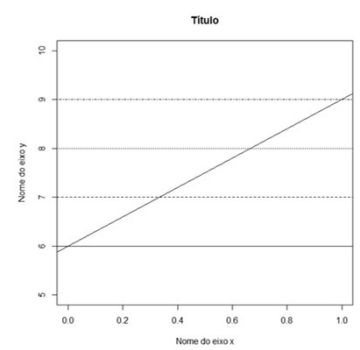
```
> # Plot de gráfico por partes - definição escalas
>
> # abre novo janela gráfica
> plot.new()
> # janela gráfica com escalas personalizadas
> plot.window(xlim = c(0, 1), ylim = c(5, 10))
> # linha a + b*x
> abline(a = 6, b = 3)
> # linhas horizontais
> abline(h = 6:9, lty = 1:4)
> # eixo x
> axis(1)
> # eixo y
> axis(2)
> # título principal
> title(main = "Título")
> # nome eixo x
> title(xlab = "Nome do eixo x")
> # nome eixo y
> title(ylab = "Nome do eixo y")
> # Limites margens região do gráfico
> box()
```

214

Estatística Computacional I - 2020

✓ Observe o efeito da execução da função de cada elemento gráfico.

✓ Gráfico concluído:



215

Estatística Computacional I - 2020

• Construção de *plot* por partes:

- ✓ Funções *high level* são compostas por conjunto de elementos:
 - Entretanto, cada elemento gráfico pode ser desenhado separadamente
- ✓ Funções *low level* permitem controle mais fino para construção de cada elemento gráfico
- ✓ Abordagem padrão para personalizar gráfico:
 - Plotar gráfico sem efeito desejado em elemento
 - Desenhar em separado elemento c/ efeito desejado

216

Estatística Computacional I - 2020

✓ Construção de gráfico por partes:

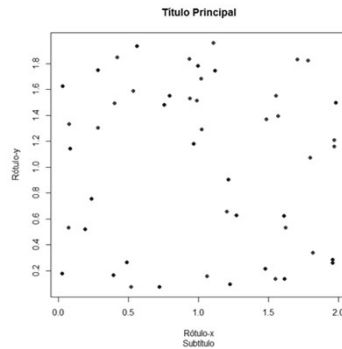
```
> # plot de gráfico por partes
> # geração dos dados
> set.seed(666)
> x <- runif(50, 0, 2)
> y <- runif(50, 0, 2)
> cat <- as.factor(sample(1:2, size = 50, replace = TRUE))
>
> # janela gráfica vazia (nenhum elemento)
> plot(x, y, type = "n", xlab = "", ylab = "", axes = F)
> # eixo x (lado inferior)
> axis(1)
> # eixo y (lado superior) - customização valores
> axis(2, at = seq(0.2, 1.8, 0.2))
> # margens nos limites entre plot e figura
> box()
> # Título e rótulos eixos
> title(main = "Título Principal", sub = "Subtítulo", xlab = "Rótulo-x",
+ ylab = "Rótulo-y")
> # símbolos dos dados, colorido por categoria
> points(x, y, col = c("red", "blue")[cat], pch = 19)
```

217

Estatística Computacional I - 2020



- ✓ Observe o efeito da execução da função de cada elemento.
- ✓ Gráfico concluído:



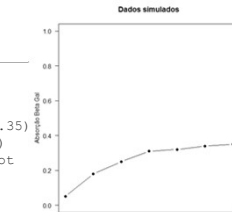
Estatística Computacional I - 2020

218



- ✓ Dois eixos com escalas diferentes no mesmo gráfico:
- Definição área da figura e 1º plot

```
> # Dois eixos no mesmo gráfico
> # geração dos dados
> tempo <- seq(0,72,12)
> betagal.abs <- c(0.05,0.18,0.25,0.31,0.32,0.34,0.35)
> celula.dens <- c(0,1000,2000,3000,4000,5000,6000)
> # adiciona espaço extra na margem direita do plot
> op <- par(no.readonly = TRUE)
> par(mar = c(5, 4, 4, 6) + 0.1)
>
> # Plot do 1º conjunto de dados e desenha seu eixo
> plot(tempo, betagal.abs, pch = 16, axes = FALSE, ylim = c(0,1),
+ xlab = "", ylab = "", type = "b", col = "black",
+ main = "Dados simulados")
> axis(2, ylim = c(0,1), col = "black", las = 1)
> mtext("Absorção Beta Gal", side = 2, line = 2.5)
> box()
```



Estatística Computacional I - 2020

219



- Construção 2º plot e finalização

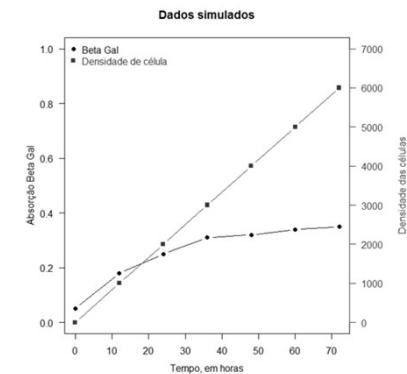
```
> # Permite um segundo plot no mesmo gráfico
> par(new = TRUE)
> # Plot do segundo plot, traçando a escala do segundo eixo à direita
> plot(tempo, celula.dens, pch = 15, xlab = "", ylab = "", ylim = c(0, 7000),
+ axes = FALSE, type = "b", col = "red")
> # posição do nome do eixo um pouco mais distante (line = 4)
> # para permitir o posicionamento dos valores da escala
> mtext("Densidade das células", side = 4, col = "red", line = 4)
> axis(4, ylim = c(0, 7000), col = "red", col.axis = "red", las = 1)
> # Traça o eixo do tempo (eixo x)
> axis(1, pretty(range(tempo), 10))
> mtext("Tempo, em horas", side = 1, col = "black", line = 2.5)
> # Adiciona legenda
> legend("topleft", legend = c("Beta Gal", "Densidade de célula"),
+ text.col = c("black", "red"), pch = c(16, 15), col = c("black", "red"),
+ bty = "n")
> par(op)
```

Estatística Computacional I - 2020

220



- Gráfico final



Estatística Computacional I - 2020

221



• Comentários:

✓ São poucas as situações em que é adequado o uso de duas escalas diferentes no mesmo gráfico

✓ Exemplos e discussões em:

– *Illusions of success*, Kaiser Fung (blog Junk Charts)

https://junkcharts.typepad.com/junk_charts/2006/06/illusion_of_suc.html

– *The Crossover law of Petropolitics*, Junk Charts

https://junkcharts.typepad.com/junk_charts/2006/05/the_crossover_l.html

– *Dual-scaled axes in graphs: are they the best solution?*, Stephen Few

http://www.perceptualedge.com/articles/visual_business_intelligence/dual-scaled_axes.pdf

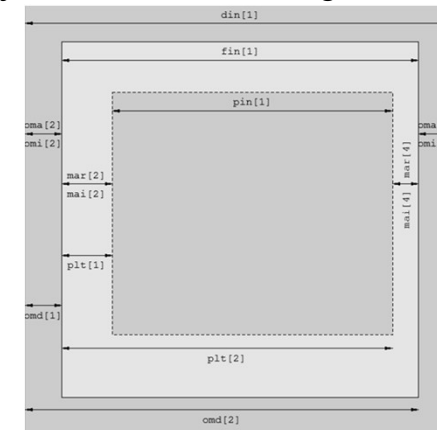


Estatística Computacional I - 2020

222



• Ajustes das dimensões do gráfico:



Estatística Computacional I - 2020

223



• Argumentos de ajuste das regiões do gráfico:

✓ oma, omd e omi,

✓ fig e fin,

✓ mex,

✓ mar, mai e plt,

✓ pty e pin.



Estatística Computacional I - 2020

224





✓ Ajuste das margens externas:

- $c(x_i, x_e, x_s, x_d)$:
 - valores margens: inferior, esquerda, superior e direita.
- *Default*: não há margens externas.
- oma:
 - Margens externas, em linhas de texto
 - *Default*: $c(0, 0, 0, 0)$.
- omi:
 - Margens externas, em polegadas
 - *Default*: $c(0, 0, 0, 0)$.
- omd:
 - Margens externas em coordenadas normalizadas do dispositivo (proporção da região do dispositivo)
 - *Default*: $c(0, 1, 0, 1)$.

Estatística Computacional I - 2020



225

✓ Ajuste da região da figura:

- *Default*: região da figura é calculada a partir dos ajustes das margens externas.
- *fig*:
 - Posição da região da figura em proporção da região interna
 - *Default*: `c(0, 1, 0, 1)`.
- *fin*:
 - Posição da região da figura, em polegadas
 - Valores do argumento: `c(width, height)`.
 - *Default*: `c(6.999999, 6.989582)`.
- Figura é centralizada na região interna



Estatística Computacional I - 2020 226

✓ Ajuste das margens da figura:

- *mar*:
 - Margens da figura em número de linhas de texto
 - *Default*: `c(5, 4, 4, 2) + 0.1`.
- *mai*:
 - Margens da figura, em polegadas
 - *Default*: `c(1.02, 0.82, 0.82, 0.42)`.
- *mex*:
 - Especificação altura de linha de texto nas margens (Multiplicador do tamanho de texto)
 - Não afeta tamanho do texto escrito nas margens.
 - *Default*: 1.



Estatística Computacional I - 2020 227

✓ Ajuste da região do gráfico:

- *Default*:
 - `região plot = região figura - margens figura`.
- *plt*:
 - Posição da região do plot, em proporção da região de figura ativa
 - *Default*: `c(0.1171, 0.9400, 0.1459, 0.8827)`.
- *pin*:
 - Tamanho da região do plot, em polegadas
 - Valores do argumento: `c(width, height)`.
 - *Default*: `c(5.759999, 5.149582)`.
- *pty*:
 - Especificação quantidade de espaço disponível que é ocupado pela região do gráfico
 - "m": região do gráfico ocupa todo o espaço disponível (*def.*)
 - "s": região do gráfico ocupa o possível de espaço disponível, mas ele deve ser quadrado.

Estatística Computacional I - 2020 228






✓ Dimensões do dispositivo gráfico:

- *din*:
 - Valores: `c(width, height)`.
 - Não permite ajuste (apenas leitura).



```
> par("din")
[1] 6.999999 6.989582
```

Estatística Computacional I - 2020 229



- Corte de saídas gráficas:
 - ✓ Em geral as saídas gráficas são cortadas na região de gráfico (*plot region*).
 - Dados (x , y) fora da região do gráfico não são plotados
 - ✓ Pode ser necessário modificar a região de recorte
 - Para a legenda, por exemplo
- Argumentos para ajuste de corte:
 - ✓ `xpd`.



Estatística Computacional I - 2020 230



- Argumentos para ajuste de corte:
 - ✓ `xpd`.
 - Valores:
 - FALSE: Todo elemento gráfico é cortado na região de gráfico (*default*).
 - TRUE: Todo elemento gráfico é cortado na região da figura
 - NA: Todo elemento gráfico é cortado na região do dispositivo.

Estatística Computacional I - 2020 231

Referências



Bibliografia Recomendada

- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHRISTIAN, N. *Basic Programming*, Lecture Notes
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.
- MURRELL, P. *R Graphics*. Chapman & Hall, 2006.

Estatística Computacional I - 2020 245