

Noções Básicas de Análise Exploratória de Dados

Docente: Manuel G. Scotto

Departamento de Matemática
IST, ULisboa

Aquisição e Recolha de Dados

- Folha “Excel” ou ficheiro “txt” (abordagem tradicional)

Date	Open	High	Low	Close	Shares	Trades	Turnover	Vwap
11/03/2019	14,245	14,4	14,08	14,2	1394031	2640	19778649	14,1881
12/03/2019	14,23	14,23	13,945	14,085	1866338	3513	26261444	14,0711
13/03/2019	14,06	14,55	14,03	14,55	2120258	3107	30603388	14,4338
14/03/2019	14,58	14,895	14,575	14,69	1296071	2276	19074602	14,7173
15/03/2019	14,695	14,715	14,31	14,31	2012936	2804	28948840	14,3814
18/03/2019	14,375	14,565	14,36	14,565	1291694	2792	18730617	14,5008
19/03/2019	14,595	14,81	14,595	14,72	954677	1834	14038073	14,7045
20/03/2019	14,56	14,675	14,375	14,41	1194989	2586	17275958	14,457
21/03/2019	14,49	14,535	14,23	14,235	1895477	3594	27166250	14,3321
22/03/2019	14,36	14,4	13,835	13,835	1510088	2701	21107420	13,9776
25/03/2019	13,85	13,9	13,71	13,855	1601845	2936	22122207	13,8105
26/03/2019	13,9	14,12	13,875	14,075	1299265	2326	18245165	14,0427
27/03/2019	14,1	14,23	14,02	14,18	1794435	3190	25377231	14,1422
28/03/2019	14,15	14,315	14,09	14,13	1272433	2634	18030413	14,17

Preço das ações da GALP ENERGIA-NOM desde 11/03/2019 (ficheiro “GALP.txt”)

Noções Básicas de Análise Exploratória de Dados

Aquisição e Recolha de Dados

- Imagens, vídeos, áudio, redes sociais, etc.

2021 *This Is What Happens In An Internet Minute*



<https://ediscoverytoday.com/2021/04/16/here-is-your-2021-internet-minute-infographic-ediscovery-trends/>

Objetivo

- From **data** to **knowledge** and from **knowledge** to **value!!**

Métodos para obter informação a partir dos dados

- Data **Visualization** (tabelas e gráficos)
- Data **Visuanimation**

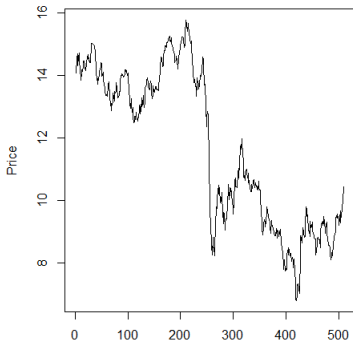
Exemplo

- GALP ENERGIA-NOM (11/03/2019 - 08/03/2021)

Open	High	Low	Close
Min. : 6.790	Min. : 6.986	Min. : 6.554	Min. : 6.820
1st Qu.: 9.252	1st Qu.: 9.384	1st Qu.: 9.103	1st Qu.: 9.264
Median :11.985	Median :12.085	Median :11.637	Median :11.810
Mean :11.684	Mean :11.822	Mean :11.521	Mean :11.667
3rd Qu.:13.995	3rd Qu.:14.114	3rd Qu.:13.871	3rd Qu.:13.986
Max. :15.750	Max. :15.950	Max. :15.630	Max. :15.760

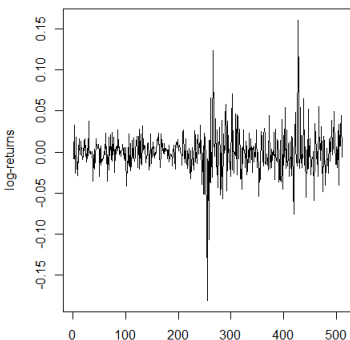
Noções Básicas de Análise Exploratória de Dados

Closing Values Share Prices (GALP ENERGIA-NOM)



Period: 2019-2021

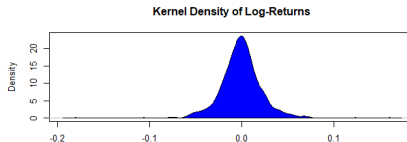
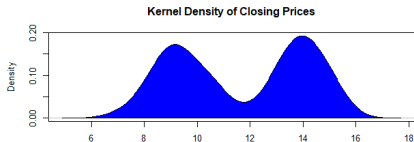
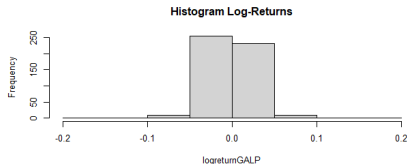
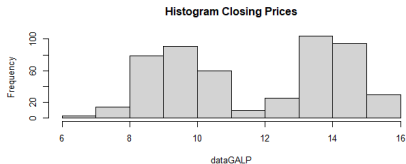
Log-RETURNS (GALP ENERGIA-NOM)



Period: 2019-2021

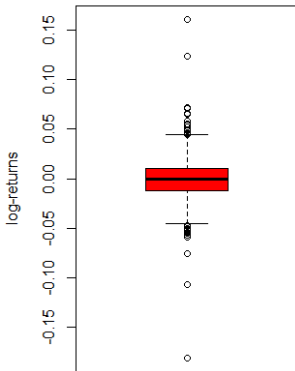
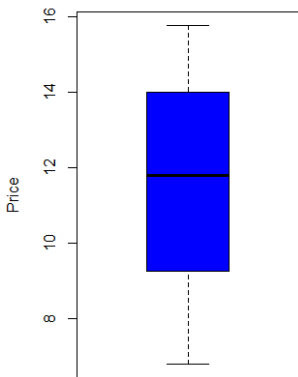
```
dataGALP = read.csv("GALP.txt", header=TRUE, sep="", dec=",")
dataGALP = dataGALP[,5]
dataGALP=ts(dataGALP)
ts.plot(dataGALP, xlab="Period: 2019-2021", ylab="Price", main="Closing Values Share Prices (GALP
ENERGIA-NOM)")
logreturnGALP = as.ts(diff(log(dataGALP)))
ts.plot(logreturnGALP, xlab="Period: 2019-2021", ylab="log-returns", main="Log-RETURNS (GALP
ENERGIA-NOM)")
```

Noções Básicas de Análise Exploratória de Dados



```
par(mfrow=c(2,2))
hist(dataGALP, main="Histogram Closing Prices")
hist(logreturnGALP, main="Histogram Log>Returns")
dGALP=density(dataGALP)
plot(dGALP, xlab="", main="Kernel Density of Closing Prices")
polygon(dGALP, col="blue")
dlogGALP=density(logreturnGALP)
plot(dlogGALP, xlab="", main="Kernel Density of Log>Returns")
polygon(dlogGALP, col="blue")
```

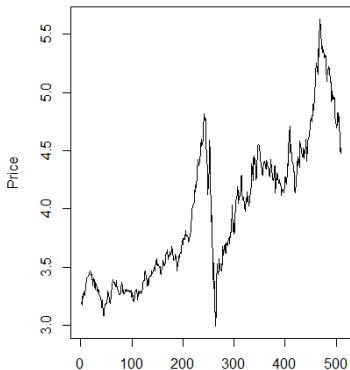
Noções Básicas de Análise Exploratória de Dados



```
par(mfrow=c(1,2))  
boxplot(dataGALP, col="blue", ylab="Price")  
boxplot(logreturnGALP, col="red", ylab="log-returns")
```

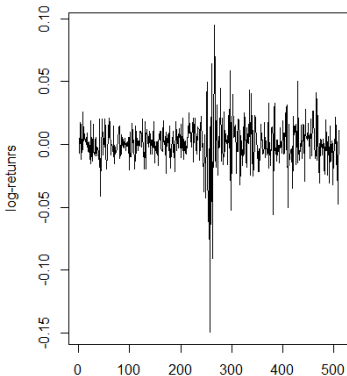

Noções Básicas de Análise Exploratória de Dados

Closing Values Share Prices (EDP)



Period: 2019-2021

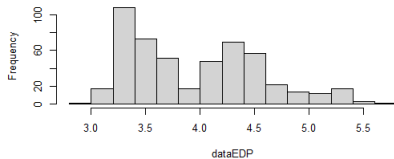
Log-RETURNS (EDP)



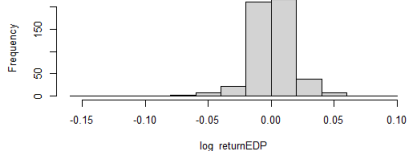
Period: 2019-2021

Noções Básicas de Análise Exploratória de Dados

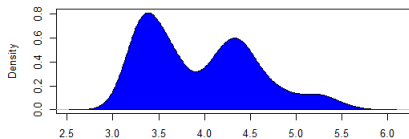
Histogram Closing Prices



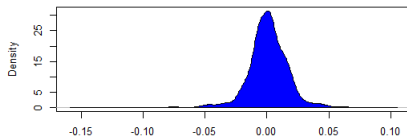
Histogram Log>Returns



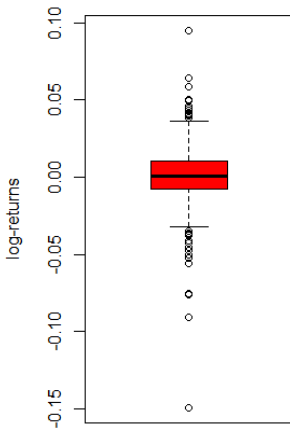
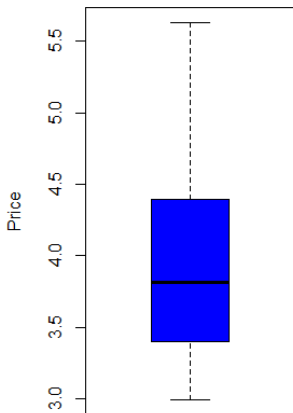
Kernel Density of Closing Prices



Kernel Density of Log>Returns

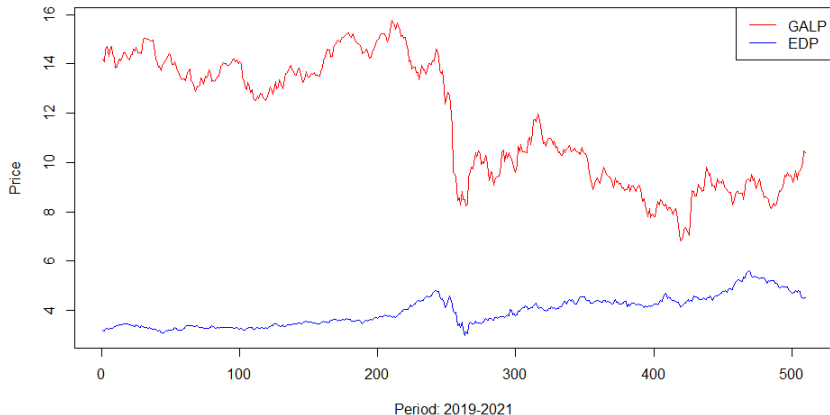


Noções Básicas de Análise Exploratória de Dados



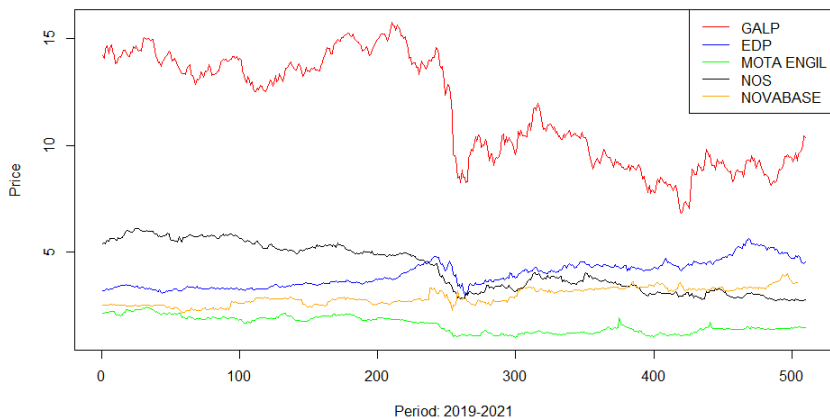
Noções Básicas de Análise Exploratória de Dados

Closing Values Share Prices (GALP, EDP)



```
jointGALPEDP=cbind(dataGALP, dataEDP)
jointGALPEDP.ts=ts(jointGALPEDP)
ts.plot(jointGALPEDP.ts, xlab="Period: 2019-2021", ylab="Price", main="Closing Values Share Prices (GALP, EDP)")
lines(jointGALPEDP.ts[,1], col="red")
lines(jointGALPEDP.ts[,2], col="blue")
legend("topright", legend=c("GALP", "EDP"), col=c("red", "blue"), lty=1)
```

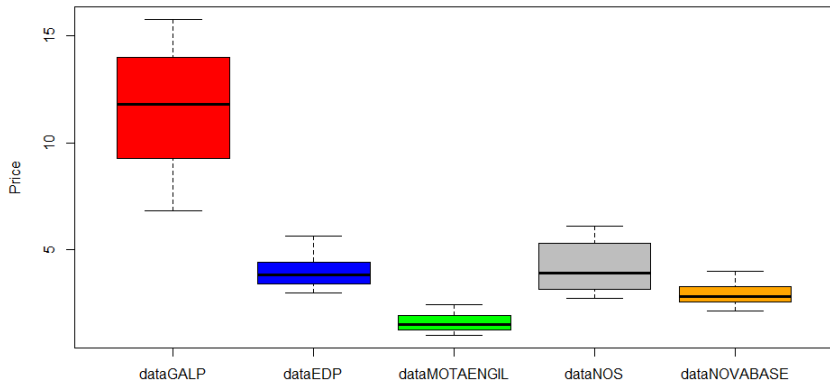
Closing Values Share Prices (GALP, EDP, MOTA, NOS, NOVABASE)



Noções Básicas de Análise Exploratória de Dados

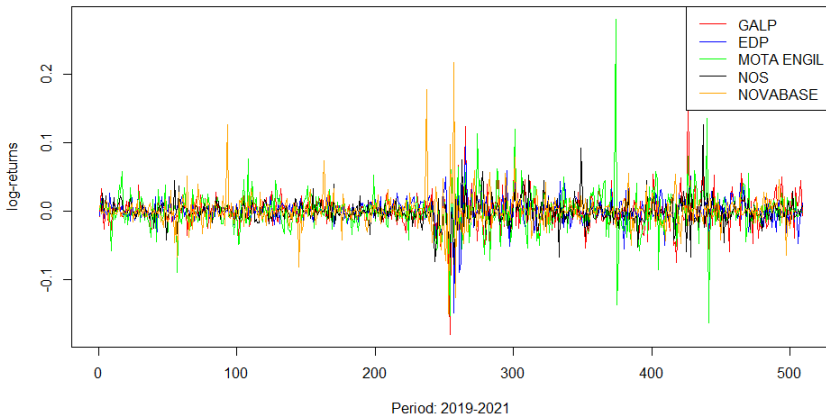
```
dataGALP = read.csv("GALP.txt", header=TRUE, sep=";", dec=";")
dataEDP = read.csv("EDP.txt", header=TRUE, sep=";", dec=";")
dataMOTAENGIL =read.csv("MOTA ENGIL.txt", header=TRUE, sep=";", dec=";")
dataNOS = read.csv("NOS.txt", header=TRUE, sep=";", dec=";")
dataNOVABASE = read.csv("NOVABASE.txt", header=TRUE, sep=";", dec=";")
dataGALP = dataGALP[,5]
dataEDP = dataEDP[,5]
dataMOTAENGIL = dataMOTAENGIL[,5]
dataNOS = dataNOS[,5]
dataNOVABASE = dataNOVABASE[,5]
dataGALP=ts(dataGALP)
dataEDP=ts(dataEDP)
dataMOTAENGIL=ts(dataMOTAENGIL)
dataNOS=ts(dataNOS)
dataNOVABASE=ts(dataNOVABASE)
jointGEMNN=cbind(dataGALP, dataEDP, dataMOTAENGIL, dataNOS, dataNOVABASE)
jointGEMNN.ts=ts(jointGEMNN)
ts.plot(jointGEMNN.ts, xlab="Period: 2019-2021", ylab="Price", main="Closing Values Share Prices (GALP,
EDP, MOTA, NOS, NOVABASE)")
lines(jointGEMNN.ts[,1], col="red")
lines(jointGEMNN.ts[,2], col="blue")
lines(jointGEMNN.ts[,3], col="green")
lines(jointGEMNN.ts[,4], col="black")
lines(jointGEMNN.ts[,5], col="orange")
legend("topright", legend=c("GALP", "EDP", "MOTA ENGIL", "NOS", "NOVABASE"), col=c("red", "blue",
"green", "black", "orange"), lty=1)
```

Noções Básicas de Análise Exploratória de Dados

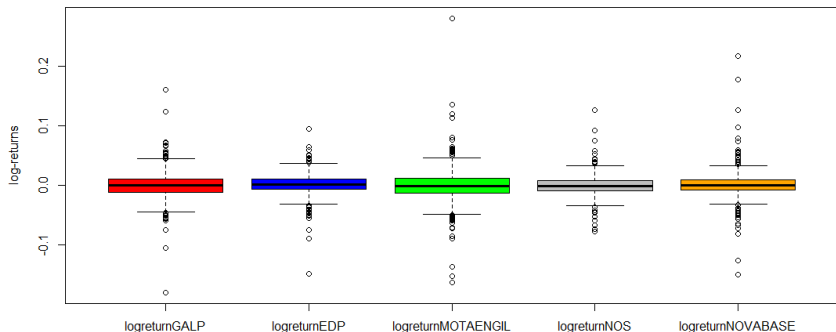


```
boxplot(jointGEMNN[,1:5], col=c('red', 'blue', 'green', 'grey', 'orange'), ylab="Price")
```

Closing Values log-returns(GALP, EDP, MOTA, NOS, NOVABASE)

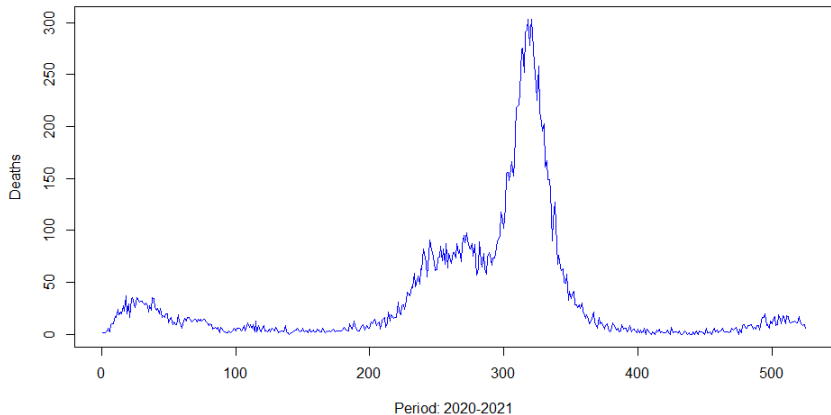


Noções Básicas de Análise Exploratória de Dados



COVID-19 em PORTUGAL (17/03/2020 - 23/08/2021)

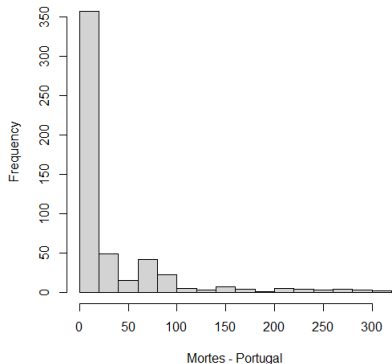
Daily Deaths (17/03/2020 - 23/08/2021)



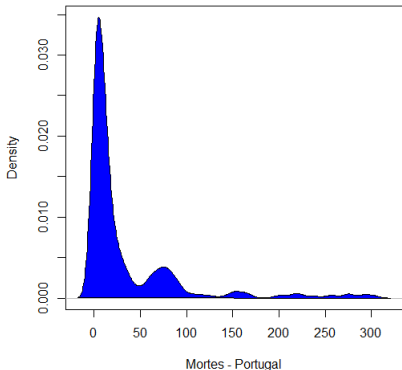
```
MortesPortugal=ts(MortesPortugal[,2])  
ts.plot(MortesPortugal, xlab="Time: 2020-2021", ylab="Deaths", main="Daily Deaths (17/03/2020 -  
23/08/2021)", col="blue")
```

COVID-19 em PORTUGAL (17/03/2020 - 23/08/2021)

Histogram Daily Deaths

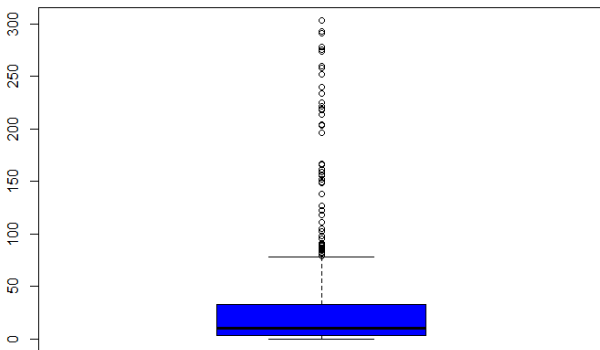


Kernel Density of Daily Deaths



```
par(mfrow=c(1,2))  
hist(MortesPortugal, xlab=" Mortes - Portugal", main="Histogram Daily Deaths")  
dMortesPortugal=density(MortesPortugal)  
plot(dMortesPortugal, xlab=" Mortes - Portugal", main="Kernel Density of Daily Deaths")  
polygon(dMortesPortugal, col="blue")
```

Boxplot Daily Deaths



Análise Exploratória de Dados em R/RStudio

- Instalar o programa R (<http://www.r-project.org>).
- Na secção download escolher qual o seu sistema operativo (**Mac**, **Linux** ou **Windows**) e qual o seu processador **32-** ou **64-**bits.
- Na página [r-project.org](http://www.r-project.org), no lado esquerdo debaixo do texto **download**, aparece a opção CRAN e na nova página deve escolher um dos servidores internacionais.
- O programa R dispõe de uma interface gráfica própria. Porém, utilizaremos um interface gráfico avançado (**IDE-Integrated Development Environment**) que se chama RStudio.



[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Conferences](#)

[Search](#)

[Get Involved: Mailing Lists](#)

[Developer Pages](#)

[R Blog](#)

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- [R version 4.1.1 \(Kick Things\)](#) has been released on 2021-08-10.
- [R version 4.0.5 \(Shake and Throw\)](#) was released on 2021-03-31.
- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the conference are available on the [R Consortium YouTube channel](#).
- You can support the R Foundation with a renewable subscription as a [supporting member](#)

Instalação do RStudio

- Instalar o RStudio (<http://www.rstudio.com>).
- Na página de descarga escolha a opção **RStudio Desktop** (gratuita).

RStudio R&R Holiday | Thur, Sep 2 - Mon, Sep 6, 2021 [LEARN MORE](#)

RStudio

DOWNLOAD SUPPORT DOCS COMMUNITY

Products Solutions Customers Resources About Pricing

Announcing

RStudio Connect 1.9.0

Curate content, and tailor the viewer experience, making communication with stakeholders easier than ever.

[READ THE BLOG POST](#)

Alguns comandos em R

- Linha de comandos (*Console*).
- Se o resultado for uma variável ou um gráfico, estes vão aparecer nas subjanelas: *Environment* ou *Plot*.
- **Concatenate:**
 - **Exemplo 1:** `x=c(1:10) x=c(1,10)`
 - **Exemplo 2:** `y = c(65.2, 73.2, 66.3, 56.7), y[1:3] y[y>60]`
- As **matrizes** correspondem a uma coleção de **elementos do mesmo tipo** definida através de linhas e colunas.
 - **Exemplo 1:** `y = matrix(1:6, nrow=3, ncol=2)`
 - **Exemplo 2:** `mat = c(10, -3, 42, -10)`
`namesL = c("l1", "l2") namesC = c("C1", "C2")`
`matfinal = matrix(mat, nrow=2, ncol=2, byrow=TRUE, dimnames=list(namesL, namesC))`

Alguns comandos em R

- As **arrays** apresentam as mesmas características que as matrizes, mas apresentam a possibilidade de terem mais de duas dimensões.
- As **listas** são conjuntos de dados que podem ser de qualquer tipo.
- Uma **data frame** corresponde a um conjunto de vetores de igual tamanho. Esses vetores não têm de ser necessariamente do mesmo tipo de dados. As data frames são utilizadas para armazenar tabelas de dados.

Exemplo

- Exemplo:

```
Alunos = c("Pedro", "Maria", "João", "Ana")
```

```
Idade = c(15, 18, 22, 17)
```

```
Estudos = c("FIS", "MAT", "AMB", "INF")
```

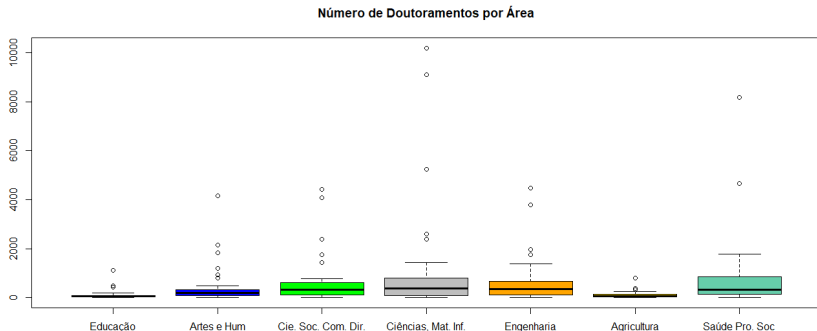
```
frame = data.frame(Alunos, Idade, Estudos)
```

	Alunos	Idade	Estudos
1	Pedro	15	FIS
2	Maria	18	MAT
3	João	22	AMB
4	Ana	17	INF

Número de Doutoramentos (2019) por Área e por País

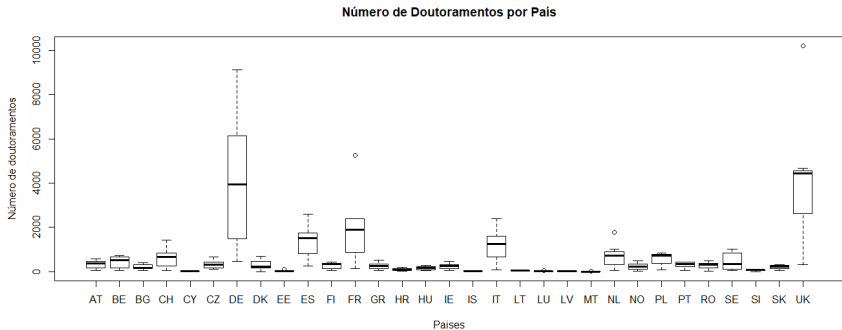
UE27 (2020)	Total (2019)	Educação	Artes e Hum	Cie. Soc. Com. Dir.	Ciências, Mat. Inf.	Engenharia	Agricultura	Saúde Pro. Soc
DE	28 690	471	2 152	4 079	9 110	3 790	798	8 169
AT	2 215	40	253	409	562	495	61	314
BE	3 014	52	283	524	711	675	74	675
BG	1 285	124	206	388	173	153	38	169
CY	126	13	23	26	32	20	2	12
HR	680	35	77	123	99	122	28	183
DK	2 095	0	176	209	357	532	125	696
SK	1 432	84	229	305	271	246	48	190
SI	477	28	86	75	90	104	3	83
ES	9 340	427	1 207	1 753	2 584	1 381	248	1 614
EE	235	3	37	30	93	38	6	25
FI	1 794	81	184	337	387	328	53	415
FR	13 405	179	1 830	2 382	5 245	1 955	143	1 550
GR	1 774	83	195	255	315	337	41	512
HU	1 271	44	178	178	285	123	59	250
IE	1 555	80	182	315	455	198	41	282
IT	7 991	67	930	1 433	2 378	1 743	372	1 059
LV	134	7	8	30	39	34	8	8
LT	325	13	37	50	90	69	16	50
LU	108	3	9	20	57	19	0	0
MT	40	2	4	2	5	8	0	19
NL	4 956	61	340	782	1 012	671	309	1 780
PL	4 039	81	802	759	836	671	145	572
PT	2 103	177	275	427	441	399	34	249
CZ	2 346	115	276	353	649	526	110	224
RO	1 920	31	478	325	223	392	86	328
SE	3 329	85	144	334	769	905	41	1 018
IS	96	7	9	15	25	9	0	31
NO	1 595	59	125	236	463	189	21	482
UK	29 340	1 120	4 144	4 423	10 197	4 463	318	4 668
CH	4 303	47	329	681	1 425	671	151	994

Número de Doutoramentos (2019) por Área



```
boxplot(PORDATADou2019[,3:9], col=c('red', 'blue', 'green', 'grey', 'orange', 'gold', 'aquamarine3'),  
main="Número de Doutoramentos por Área")
```

Número de Doutoramentos (2019) por País



Links

- <https://www.r-graph-gallery.com/index.html>
- <https://ggplot2-book.org/index.html>
- <https://www.gapminder.org/>

GGPLOT

- A função `ggplot` permite definir os parâmetros iniciais do gráfico.

```
ggplot(data=dados, aes(x=..., y=... ))
```

A função

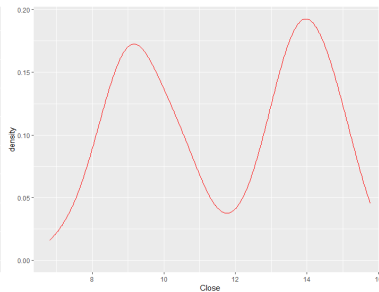
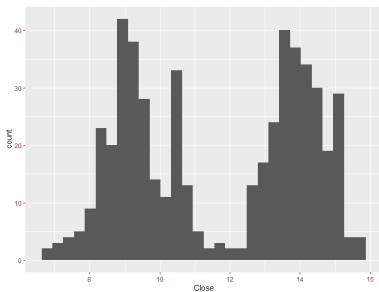
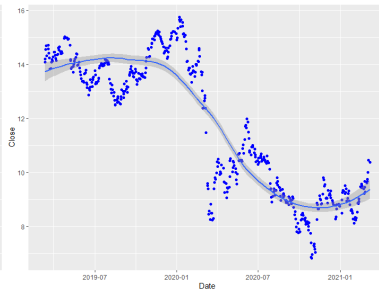
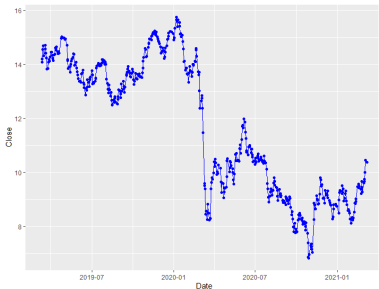
```
geom_XXX()
```

define o tipo de gráfico.

- Exemplos:

`xxx = density, dotplot, point, histogram, bar, dotplot, violin, line, freqpoly, ...`

Gráficos com a função GGLOT



Gráficos com a função GGPLOT

```
> ggplot(data= GALPexcel, aes(x = Date, y = Close)) + geom_line(colour="blue") +  
geom_point(colour="blue")
```

```
> ggplot(data= GALPexcel, aes(x = Date, y = Close)) + geom_point(colour="blue") +  
geom_smooth()
```

```
> ggplot(data= GALPexcel, aes(x = Close)) + geom_histogram(bins=30)
```

```
> ggplot(data= GALPexcel, aes(x = Close)) + geom_density(colour="red")
```

Animated line chart transition with R: GALP dataset

```
library(gganimate)
```

```
library(ggplot2)
```

```
ggplot(data= GALPexcel, aes(x = Date, y = Close)) + geom_line ()+ geom_point(colour="blue")  
+ transition_reveal(along= Date)
```

Animated line chart transition with Gapminder

```
> library(plotly)
> library(gapminder)
> p <- ggplot(gapminder, aes(gdpPercap, lifeExp, color = continent)) +
  geom_point(aes(size = pop, frame = year, ids = country)) +
  scale_x_log10()
> fig <- ggplotly(p)
> fig
```

Funções em R

- É possível criar-mos as nossas **próprias funções** em R.

```
nome_da_função<-function(x) {transformação de x}
```

Exemplo

```
n<-sample(1:30, 50, replace=TRUE)
fff<-function(x){y=x+1
return(y)
}
fff(n)
```

Exemplo

```
fff1<-function(x){  
  print("média")  
  print(mean(x))  
  print("desvio padrão")  
  print(sd(x))  
  hist(x)  
  boxplot(x)  
}  
fff1(n)
```

Exemplo

```
toFahrenheit<-function(celsius) { f = (9/5) * celsius + 32  
return(f)  
}  
toFahrenheit(30)  
86
```