
Introdução à Estatística no *Software R*

Estatística descritiva

www.de.ufpb.br <https://www.youtube.com/estatisticalivre>



ESTATÍSTICA APLICADA
EM SOFTWARE LIVRE

UFPB  **Departamento de
ESTATÍSTICA**



- **Estatística Descritiva** é definida como sendo o conjunto de técnicas que permite descrever, analisar e interpretar os dados numéricos referentes à uma população ou amostra.
- O objetivo é resumir os dados coletados de forma a extrair conhecimento útil acerca do problema motivador da coleta.
- Nessa fase da pesquisa, estamos preocupados em apresentar os dados em forma de tabelas e gráficos e em obter medidas que quantifiquem os resultados do estudo.



Principais elementos:

- **Representação tabular:** a organização dos dados em tabelas proporciona um meio eficaz de estudo do comportamento de características de interesse.
- **Representação gráfica:** proporciona uma interpretação imediata dos resultados devido a sua simplicidade e clareza.
- **Medidas resumo:** possibilitam representar um conjunto de dados relativo a observação de determinado fenômeno de forma resumida. São classificadas em medidas de posição, dispersão, assimetria e curtose.



- Com base nos conhecimentos adquiridos no Módulo 1, primeiramente devemos fazer o *download* do banco de dados e importá-lo para o Rstudio.
- Vamos utilizar os dados hipotéticos de 36 funcionários da companhia “Milsa”, retirados do livro Estatística Básica-W. Bussab e P. Morettin. [▶ Link](#)

```
milsa = read.delim("/home/herminia/Documentos/milsa.txt")
```



Importação e preparação de dados

- Após a importação dos dados vamos criar um *dataframe* chamado *dados* para armazenar nossos dados. Dessa forma, vamos editar apenas o *dataframe*:

```
dados = data.frame(milsa)
```

- Vamos imprimir um resumo sobre as variáveis do banco de dados:

```
str(dados)
```

```
'data.frame':   36 obs. of   8 variables:
 $ funcionario: int   1 2 3 4 5 6 7 8 9 10 ...
 $ civil      : int   1 2 2 1 1 2 1 1 2 1 ...
 $ instrucao  : int   1 1 1 2 1 1 1 1 2 2 ...
 $ filhos     : int   NA 1 2 NA NA 0 NA NA 1 NA ...
 $ salario    : num   4 4.56 5.25 5.73 6.26 6.66 6.86 7.39 7.59 7.44 ...
 $ ano        : int   26 32 36 20 40 28 41 43 34 23 ...
 $ mes        : int   3 10 5 10 7 0 0 4 10 6 ...
 $ regioao    : int   1 2 2 3 3 1 1 2 2 3 ...
```



- Podemos ainda imprimir as primeiras linhas do banco de dados:

```
head(dados)
```

	funcionario	civil	instrucao	filhos	salario	ano	mes	regiao
1	1	1	1	NA	4.00	26	3	1
2	2	2	1	1	4.56	32	10	2
3	3	2	1	2	5.25	36	5	2
4	4	1	2	NA	5.73	20	10	3
5	5	1	1	NA	6.26	40	7	3
6	6	2	1	0	6.66	28	0	1



- As variáveis *civil*, *instrucao* e *regiao* são do tipo **qualitativas**, porém estão representadas por números. Vamos associar cada um desses números à uma categoria:

```
dados$civil <- factor(dados$civil,  
  label = c("solteiro", "casado"), levels = 1:2)  
dados$instrucao <- factor(dados$instrucao,  
  label = c("1° Grau", "2° Grau", "Superior"),  
  lev = 1:3, ord= T)  
dados$regiao <- factor(dados$regiao,  
  label = c("capital", "interior", "outro"),  
  lev = c(2, 1, 3))
```



- Vamos criar agora a variável *idade* como sendo a soma dos anos inteiros e os meses divididos por doze. Dessa forma teremos a idade completa de cada indivíduo:

```
dados$idade <- dados$ano + dados$mes/12
```

- Por fim, vamos utilizar o comando *attach* para que o Rstudio reconheça todas as variáveis dentro do *dataframe dados*:

```
attach(dados)
```




- Vamos imprimir novamente um resumo sobre as variáveis do banco de dados:

```
str(dados)
```

```
'data.frame':  36 obs. of  9 variables:
 $ funcionario: int  1 2 3 4 5 6 7 8 9 10 ...
 $ civil      : Factor w/ 2 levels "solteiro","casado": 1 2 2 1 1 2 1 1
 $ instrucao  : Ord.factor w/ 3 levels "1° Grau"<"2° Grau"<..: 1 1 1 2
 $ filhos     : int  NA 1 2 NA NA 0 NA NA 1 NA ...
 $ salario    : num  4 4.56 5.25 5.73 6.26 6.66 6.86 7.39 7.59 7.44 ...
 $ ano        : int  26 32 36 20 40 28 41 43 34 23 ...
 $ mes        : int  3 10 5 10 7 0 0 4 10 6 ...
 $ regiao     : Factor w/ 3 levels "capital","interior",...: 2 1 1 3 3 2
 $ idade      : num  26.2 32.8 36.4 20.8 40.6 ...
```



- Para representar os dados em tabelas utilizaremos a **distribuição de frequências**.
- Vamos calcular a **frequência simples** da variável qualitativa *civil*:

```
freq = table(civil)
freq
```

```
civil
solteiro    casado
      16         20
```



Representação tabular

- Para calcular a tabela de **frequências relativas** desta mesma variável:

```
freq_rel = prop.table(freq)
freq_rel
```

```
civil
solteiro    casado
0.4444444 0.5555556
```

- Caso queira calcular a **porcentagem** da variável em questão:

```
p_freq_rel = 100*prop.table(freq)
p_freq_rel
```

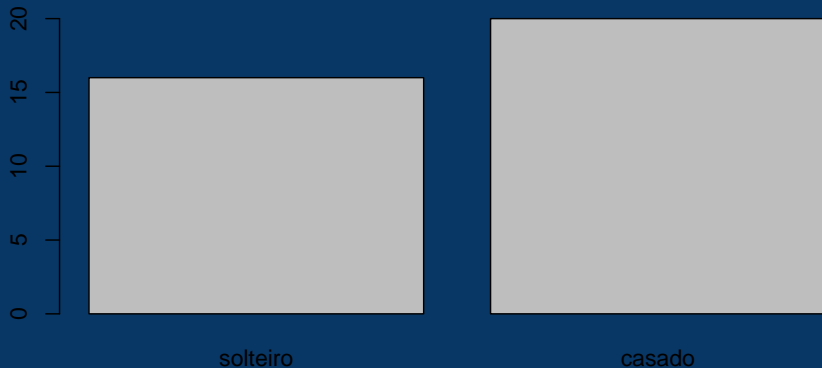
```
civil
solteiro    casado
44.44444 55.55556
```



- Outra opção mais completa está disponível no pacote *descr*.
- Vamos construir uma tabela de frequências para a variável *civil*:

```
library(descr)  
freq(civil)
```

Representação tabular



## civil		
##	Frequência	Percentual
## solteiro	16	44.44
## casado	20	55.56
## Total	36	100.00



Tabela de frequências por classes

- Para construir uma tabela de **frequências por classes**, podemos utilizar o pacote *fdth*.
- Vamos construir uma tabela de frequências por classes para a variável *salario*.

```
library(fdth)
tabClasses= fdt(salario)
tabClasses
```

Class limits	f	rf	rf(%)	cf	cf(%)
[3.96,6.7561)	6	0.17	16.67	6	16.67
[6.7561,9.5523)	10	0.28	27.78	16	44.44
[9.5523,12.348)	7	0.19	19.44	23	63.89
[12.348,15.145)	6	0.17	16.67	29	80.56
[15.145,17.941)	4	0.11	11.11	33	91.67
[17.941,20.737)	2	0.06	5.56	35	97.22
[20.737,23.533)	1	0.03	2.78	36	100.00



Tabela de frequências bivariada

- Podemos também construir uma **tabela bivariada** para as variáveis *civil* e *instrucao*:

```
library(descr)
crosstab(civil, instrucao)
```

Conteúdo das células

```
|-----|
|                Contagem |
|-----|
```

=====				
	instrucao			
civil	1° Grau	2° Grau	Superior	Total

solteiro	7	6	3	16

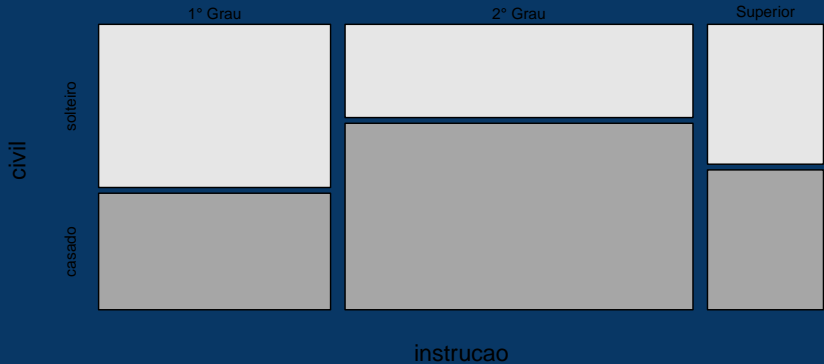
casado	5	12	3	20

Total	12	18	6	36
=====				

Tabela de frequências bivariada



- Gráfico relativo à tabela anterior:





Medidas resumo

- Outra forma de resumir os dados é utilizando as **medidas resumo**.
- Nesse contexto temos as medidas de **posição** ou de **tendência central** e as medidas de **dispersão**.
- Inicialmente vamos obter um resumo de todas as variáveis do banco de dados:

```
summary(dados)
```

funcionario	civil	instrucao	filhos	salario
Min. : 1.00	solteiro:16	1° Grau :12	Min. :0.00	Min. : 4.000
1st Qu.: 9.75	casado :20	2° Grau :18	1st Qu.:1.00	1st Qu.: 7.553
Median :18.50		Superior: 6	Median :2.00	Median :10.165
Mean :18.50			Mean :1.65	Mean :11.122
3rd Qu.:27.25			3rd Qu.:2.00	3rd Qu.:14.060
Max. :36.00			Max. :5.00	Max. :23.300
			NA's :16	

ano	mes	regiao	idade
Min. :20.00	Min. : 0.000	capital :11	Min. :20.83
1st Qu.:30.00	1st Qu.: 3.750	interior:12	1st Qu.:30.67
Median :34.50	Median : 6.000	outro :13	Median :34.92
Mean :34.58	Mean : 5.611		Mean :35.05
3rd Qu.:40.00	3rd Qu.: 8.000		3rd Qu.:40.52
Max. :48.00	Max. :11.000		Max. :48.92



Média aritmética

- A medida de posição mais utilizada é a **média aritmética**.
- É calculada somando-se os valores das observações da amostra ou população e dividindo-se o resultado pelo tamanho da amostra ou população.
- Assim, a média amostral é dada por

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} = \frac{X_1 + \dots + X_n}{n}.$$

- Vamos calcular a média da variável *salario*:

```
mean(salario)
```

```
[1] 11.12222
```



- Para calcular a **mediana** deve-se ordenar os dados de forma crescente.
- Se o número de observações for ímpar, a mediana será a observação central.
- Se o número de observações for par, a mediana será a média aritmética das duas observações centrais.
- Vamos calcular a mediana da variável *salario*:

```
median(salario)
```

```
[1] 10.165
```



- Os **quantis** são valores dados a partir do conjunto de observações ordenado de forma crescente, dividindo a distribuição em partes iguais.
- O mais usual é dividir a distribuição em quatro partes. Neste caso temos os quartis.
- Se dividirmos os dados em dez partes iguais, teremos os decis, em cem partes iguais teremos os centis e assim por diante.
- Vamos calcular os quartis da variável *salario*, usaremos a função *quantile*:

```
quantile(salario)
```

0%	25%	50%	75%	100%
4.0000	7.5525	10.1650	14.0600	23.3000

- Caso queiramos calcular outros quantis, faremos uso do argumento *probs*:

```
quantile(salario,probs = seq(0, 1, 0.1))
```

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
4.000	5.995	7.390	8.290	9.130	10.165	11.590	13.415	14.710	16.935	23.300



- A **moda** de um conjunto de valores é o valor que apresenta a maior frequência.
- Pode ser calculada para variáveis qualitativas.
- Vamos calcular a moda da variável *idade* de duas formas:

```
library(modeest)  
mfv(idade)
```

```
[1] 41
```

```
names(table(idade))[table(idade)==max(table(idade))]
```

```
[1] "41"
```



- A **variância** de uma amostra X_1, \dots, X_n de n elementos é definida como a soma dos desvios quadráticos dos elementos em relação à sua média \bar{x} dividida por $(n - 1)$:

$$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n - 1}$$

- Vamos calcular a variância amostral da variável *salario*:

```
var(salario)
```

```
[1] 21.04477
```



- O **desvio padrão** amostral de um conjunto de dados é a raiz quadrada da variância amostral.
- Vamos calcular o desvio-padrão amostral da variável *salario*:

```
sd(salario)
```

```
[1] 4.587458
```



Outras formas de obter medidas resumo

- Podemos fazer uso da função *tapply* para calcular **medidas de uma variável quantitativa, para cada categoria de uma variável qualitativa** do banco de dados:

```
tapply(salario, instrucao, mean)
```

```
1° Grau  2° Grau  Superior  
7.836667 11.528333 16.475000
```

```
tapply(salario, instrucao, sd)
```

```
1° Grau  2° Grau  Superior  
2.956464 3.715144 4.502438
```

```
tapply(salario, instrucao, quantile)
```

```
$'1° Grau'  
  0%   25%   50%   75%  100%  
4.0000 6.0075 7.1250 9.1625 13.8500
```

```
$'2° Grau'  
  0%   25%   50%   75%  100%  
5.7300 8.8375 10.9100 14.4175 19.4000
```

```
$Superior  
  0%   25%   50%   75%  100%  
10.5300 13.6475 16.7400 18.3775 23.3000
```

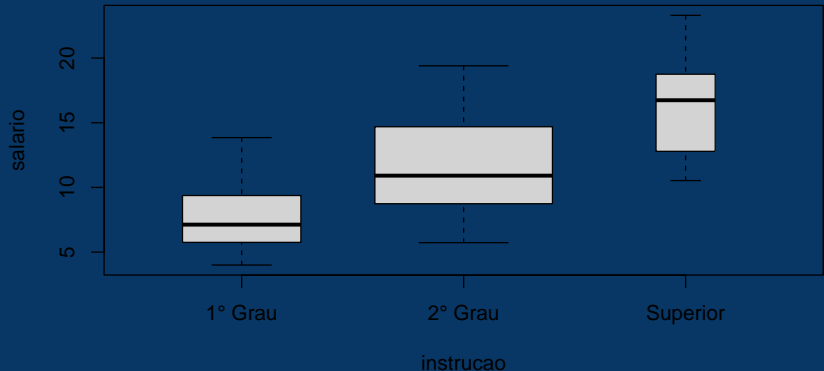



- Podemos usar a função *compmeans* do pacote *descr* para calcular a **média e o desvio padrão** da variável quantitativa para o total da amostra e **para cada categoria da variável qualitativa**.
- Vamos usar esta função para as variáveis *salario* e *instrucao*:

```
library(descr)
compmeans(salario, instrucao)
```



Outras formas de obter medidas resumo



Valor médio de "salario" segundo "instrucao"

	Média	N	Desv.	Pd.
1° Grau	7.836667	12	2.956464	
2° Grau	11.528333	18	3.715144	
Superior	16.475000	6	4.502438	
Total	11.122222	36	4.587458	



- A **representação gráfica** proporciona uma interpretação imediata dos resultados, devido à sua simplicidade e clareza. Vamos a seguir como elaborar alguns gráficos simples.



Gráfico em Barras ou colunas

- É utilizado para comparar grandezas por meio de barras de igual largura e alturas proporcionais às respectivas grandezas. É apropriado para representar variáveis qualitativas e quantitativas discretas. Vamos construir este gráfico para a variável *instrucao*:

```
barplot(table(instrucao), main = "Grau de Instrução")
```

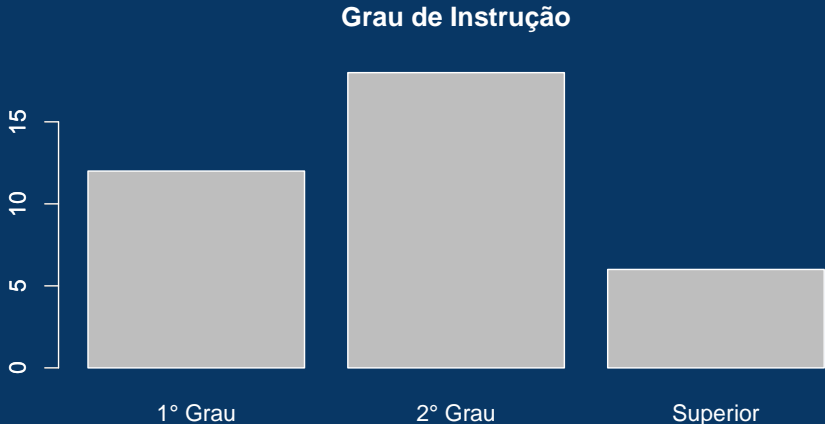




Gráfico em Barras ou colunas

- Podemos gerar este gráfico pra mais de uma variável:

```
barplot(table(civil, instrucao), legend = T,  
        main = "Estado Civil versus Grau de Instrução")
```

Estado Civil versus Grau de Instrução

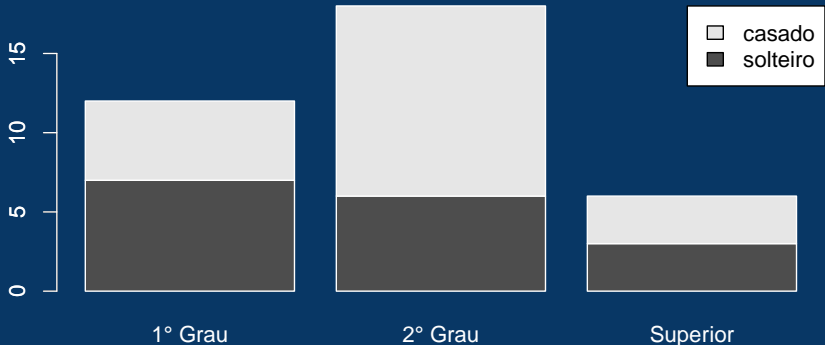




Gráfico em Barras ou colunas

- Ou ainda com as barras lado a lado:

```
barplot(table(civil, instrucao), beside = T, legend = T,  
        main = "Estado Civil versus Grau de Instrução")
```

Estado Civil versus Grau de Instrução

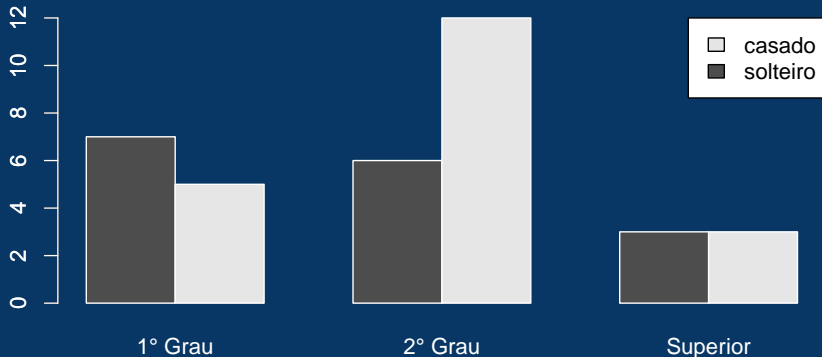




Gráfico em Barras ou colunas

- Podemos gerar este gráfico com as colunas horizontais, adicionando o argumento *horiz = T*:

```
barplot(table(civil, instrucao), beside = T,  
        legend = T, horiz = T,  
        main = "Estado Civil versus Grau de Instrução")
```

Estado Civil versus Grau de Instrução

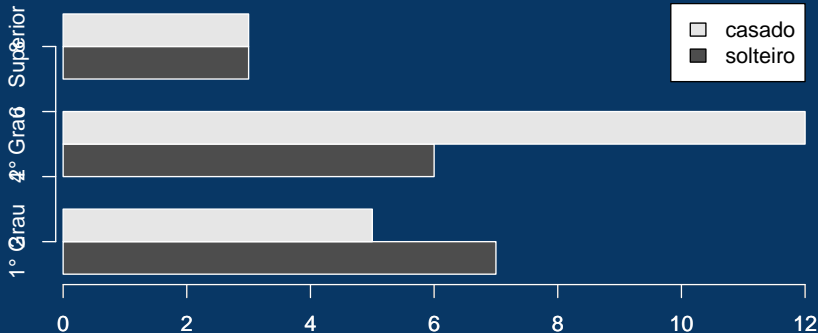




Gráfico de Pizza ou Setores

- É apropriado para representar variáveis qualitativas e quantitativas discretas com poucas categorias. Vamos construir este gráfico para a variável *regiao*:

```
pie(table(regiao))
```

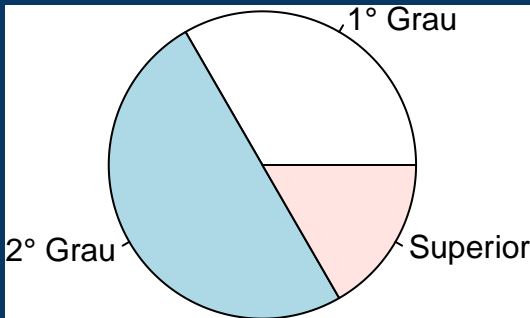




Gráfico de Pizza ou Setores

- Caso queira inserir os percentuais nesse gráfico:

```
porc = round(table(regiao)*100/sum(table(regiao)),2)
rotulos = paste("(",porc,"%)",sep="")
pie(table(regiao),labels = rotulos, col=palette())
legend(1.3,1,levels(regiao),col=palette(),pch = rep(20,6))
```

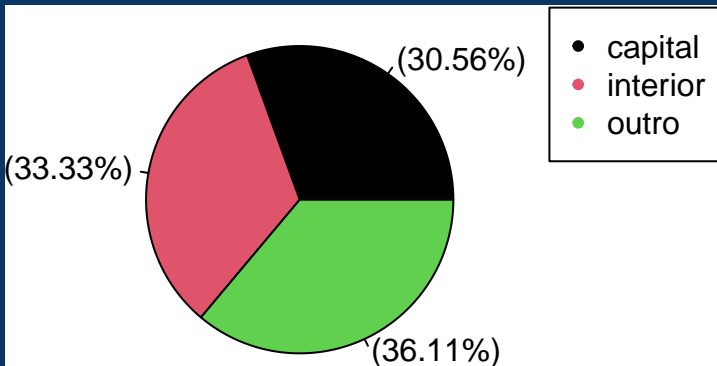
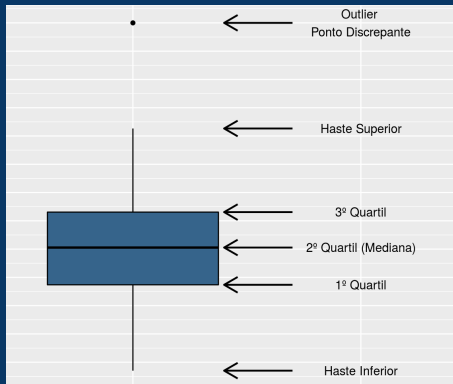


Gráfico Box-Plot



- Podemos utilizar este gráfico para comparar visualmente dois ou mais grupos.
- A diferença entre os quartis ($Q_3 - Q_1$) é uma medida da variabilidade dos dados.



Gráfico Box-Plot

Vamos construir um box-plot para *salario* por categoria de *instrucao*:

```
boxplot(salario ~ instrucao,col= rainbow(7),  
xlab = "Grau de Instrução",ylab = "Salário")
```

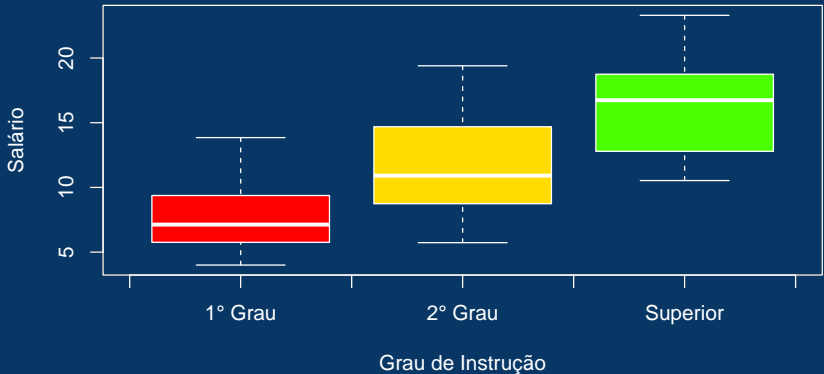
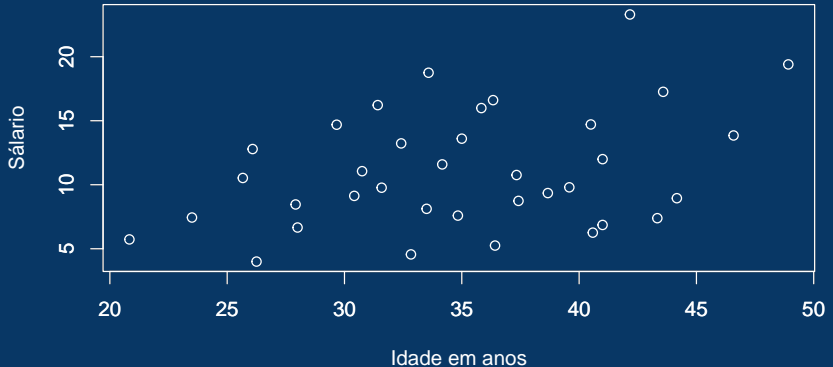




Gráfico de dispersão

- É uma representação de dados de duas ou mais variáveis. Vamos construir um gráfico de dispersão para a variável *idade*:

```
plot(idade, salario, xlab = "Idade em anos",  
     ylab = "Sálario")
```

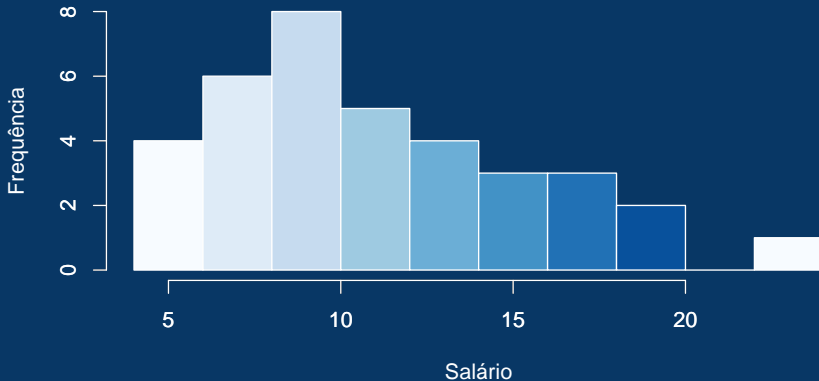


Histograma



- Representação gráfica de uma distribuição de frequências por meio de retângulos justapostos, cujas áreas são proporcionais às frequências das classes. Vamos construir um histograma para a variável *salario*:

```
hist(salario,xlab = "Salário",ylab = "Frequência",  
     col = blues9,main = "")
```





Histograma

- O histograma é frequentemente utilizado para verificar a simetria dos dados. Para visualizar melhor a simetria dos dados, podemos incluir a curva gaussiana no gráfico:

```
hist(salario,xlab = "Sálario",ylab = "Densidade",col = blues9,freq = F,  
     xlim = c(-5,30),main = "")  
curve(dnorm(x,mean = mean(salario),sd = sd(salario)), add = T)
```

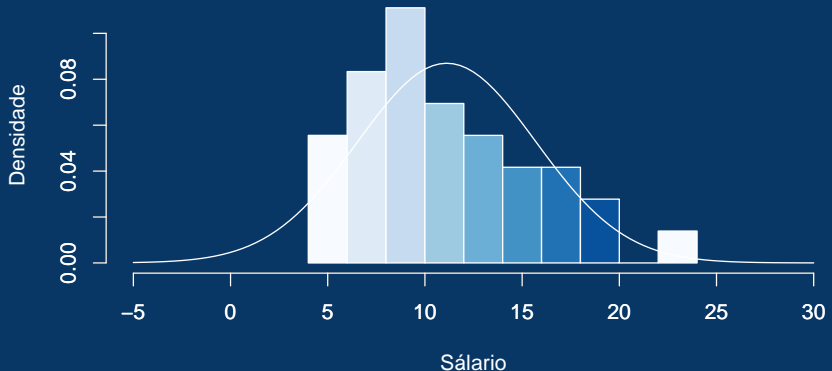


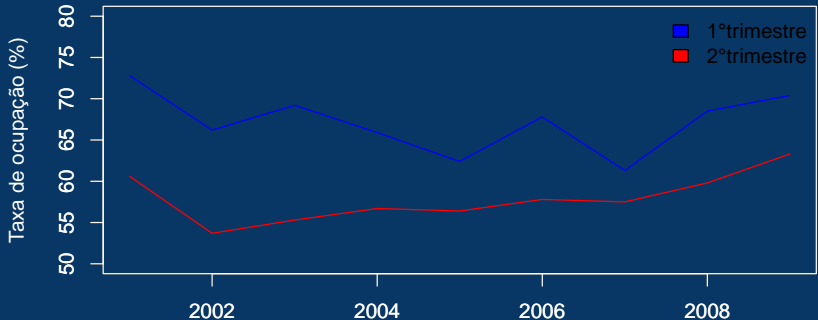


Gráfico de linhas

- Geralmente é utilizado na representação de séries de tempo.

```
ano<-2001:2009
tri1<-c(72.8,66.2,69.2,65.9,62.4,67.8,61.3,68.5,70.4)
tri2<-c(60.6,53.7,55.3,56.7,56.4,57.8,57.5,59.8,63.3)
plot(ano, tri1,type="l",main="Taxa de ocupação dos hotéis-RJ",xlab="ano",ylab="Taxa de ocupação (%)",
col="blue",ylim=c(50,80))
lines(ano,tri2,col="red")
legenda1 = c("1ºtrimestre","2ºtrimestre")
legend(x="topright", legend=legenda1,fill=c("blue","red"), bty="n")
```

Taxa de ocupação dos hotéis-RJ

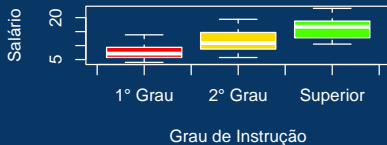




Mais recursos gráficos

- Para melhor visualização dos dados, podemos plotar mais de um gráfico:

```
par(mfrow = c(2,2));hist(salario, col = "lightblue",main = "",xlab = "Sálario",ylab = "Frequência")
boxplot(salario ~ instrucao,col=rainbow(7),xlab = "Grau de Instrução",ylab = "Salário")
hist(salario,xlab = "Sálario",main = "",col = blues9,freq = F,xlim = c(-5,30),ylab = "Densidade")
curve(dnorm(x,mean = mean(salario),sd = sd(salario)), add = T);boxplot(salario,col="darkblue")
```





Mais recursos gráficos

- Para ter acesso às 657 cores disponíveis, podemos utilizar a função *colors*:

```
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"
## [25] "blanchedalmond"  "blue"            "blue1"
## [28] "blue2"           "blue3"           "blue4"
## [31] "blueviolet"      "brown"           "brown1"
## [34] "brown2"          "brown3"          "brown4"
## [37] "burlywood"       "burlywood1"      "burlywood2"
## [40] "burlywood3"      "burlywood4"      "cadetblue"
## [43] "cadetblue1"      "cadetblue2"      "cadetblue3"
## [46] "cadetblue4"      "chartreuse"       "chartreuse1"
## [49] "chartreuse2"     "chartreuse3"     "chartreuse4"
## [52] "chocolate"       "chocolate1"      "chocolate2"
## [55] "chocolate3"     "chocolate4"      "coral"
## [58] "coral1"          "coral2"          "coral3"
## [61] "coral4"          "cornflowerblue"  "cornsilk"
## [64] "cornsilk1"       "cornsilk2"       "cornsilk3"
## [67] "cornsilk4"       "cyan"            "cyan1"
## [70] "cyan2"           "cyan3"           "cyan4"
## [73] "darkblue"        "darkcyan"        "darkgoldenrod"
## [76] "darkgoldenrod1"  "darkgoldenrod2"  "darkgoldenrod3"
## [79] "darkgoldenrod4"  "darkgray"        "darkgreen"
## [82] "darkgrey"        "darkkhaki"       "darkmagenta"
```



- Para saber mais sobre parâmetros gráficos, como outras cores, eixos, títulos, símbolos, acesse o [▶ Link](#).



Gráficos 3D

- Por vezes, surge a necessidade de analisar a relação de mais de duas variáveis. Podemos então confeccionar gráficos em terceira dimensão, utilizando o pacote *scatterplot3d*:

```
library("scatterplot3d")
```

- Para tanto, vamos utilizar um banco de dados sobre as propriedades físicas da água, retirado do livro “Fundamentos da Engenharia Hidráulica”. [▶ Link](#)

```
agua = read.delim("/home/herminia/Documentos/agua.txt")
attach(agua)
head(agua)
```

	temp	massa	peso	pressao	elasticidade	visc_dinamica	visc_cinematica
1	0	101.9	999.9	62	2.08	1.83	1.79
2	5	101.9	1000.0	89	2.10	1.55	1.52
3	10	101.9	999.0	129	2.15	1.33	1.31
4	15	101.8	999.7	174	2.18	1.16	1.14
5	20	101.8	999.1	238	2.24	1.03	1.01
6	25	101.6	998.2	323	2.26	0.91	0.90

- Observe que todas as variáveis são quantitativas.

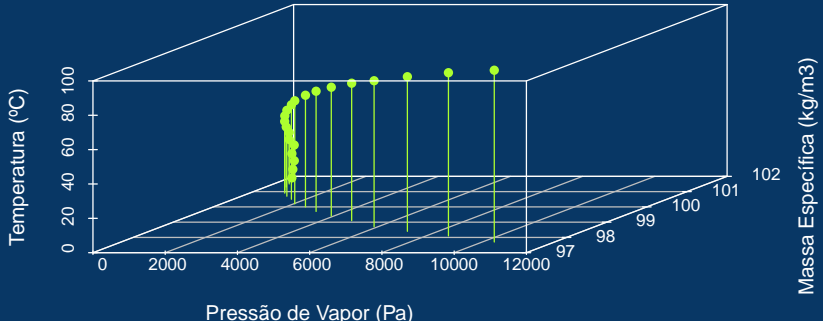


Gráficos 3D

- Vamos fazer um gráfico tridimensional com as variáveis *pressão*, *massa* e *temp*:

```
scatterplot3d(pressao, massa, temp, xlab = "Pressão de Vapor (Pa)",  
              ylab = "Massa Específica (kg/m3)", zlab = "Temperatura (°C)",  
              main = "Propriedades Físicas da Água", pch = 16,  
              color = "greenyellow", type = "h")
```

Propriedades Físicas da Água



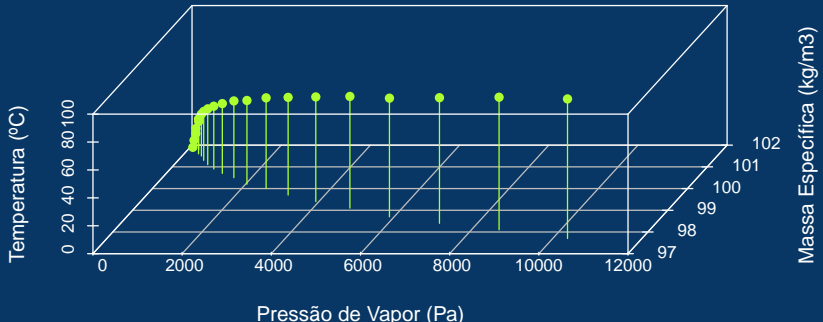


Gráficos 3D

- É possível ainda modificar o ângulo de visualização, utilizando o argumento *angle*:

```
scatterplot3d(pressao, massa, temp, xlab = "Pressão de Vapor (Pa)",  
              ylab = "Massa Específica (kg/m3)", zlab = "Temperatura (°C)",  
              main = "Propriedades Físicas da Água", pch = 16,  
              color = "greenyellow", type = "h", angle = 70)
```

Propriedades Físicas da Água





- Podemos gerar uma versão interativa do gráfico anterior, utilizando os pacotes *rgl* e *car*.

```
library("car")  
library("rgl")  
scatter3d(pressao, massa, temp, point.col = "greenyellow",  
          surface = FALSE, xlab = "Pressão Vapor (Pa)",  
          ylab = "Massa Especifica (kg/m3)",  
          zlab = "Temperatura (C)")
```



- Utilizaremos agora um conjunto de dados que contém informações acerca das dimensões em centímetros da pétala e sépala de 50 flores de três diferentes espécies de íris:

```
data(iris)
head(iris)
```

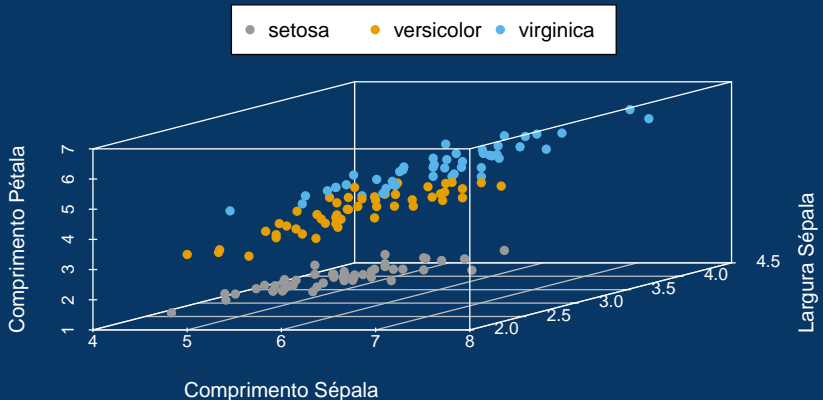
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa



Gráficos 3D

- Vamos construir o gráfico 3D para as variáveis *Sepal.Length*, *Sepal.Width* e *Petal.Length*, indicando pela cor a espécie:

```
cores = c("#999999", "#E69F00", "#56B4E9");cores = cores[as.numeric(iris$Species)]
scatterplot3d(iris[,1:3], pch = 16, color=cores,xlab = "Comprimento Sépala",
              ylab = "Largura Sépala", zlab = "Comprimento Pétala")
legend("top", legend = levels(iris$Species),col = c("#999999", "#E69F00", "#56B4E9"),
      pch = 16,inset = -0.25, xpd = TRUE, horiz = TRUE)
```





- Criada por Leland Wilkinson, nos possibilita construir gráficos de uma forma diferente, utilizando camadas.
- A gramática dos gráficos é composta por sete elementos:
 - Dados
 - Estética
 - Geometria
 - Facets
 - Estatística
 - Coordenadas
 - Temas



Gramática dos gráficos

- Para exemplificar essa forma de construção de gráficos, vamos utilizar o pacote *ggplot2*:

```
library(ggplot2)
```

- Iremos utilizar a base de dados desse pacote chamada *midwest*, que contém informações demográficas acerca dos estados do centro-oeste dos EUA:

```
data("midwest")  
head(midwest)
```

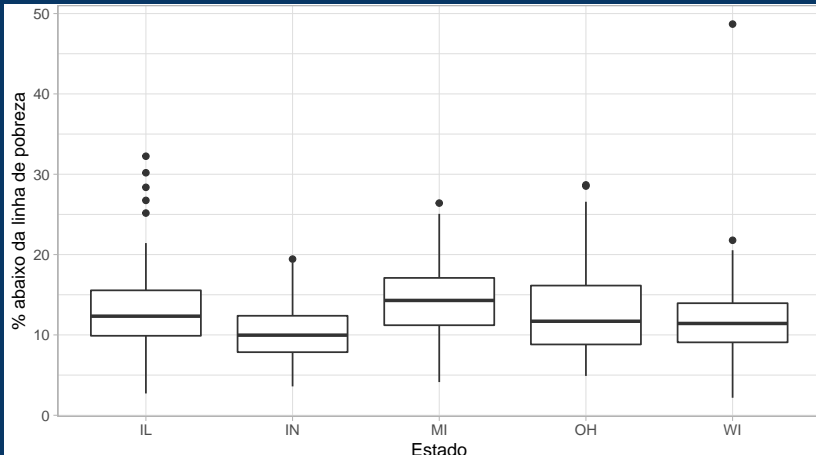
```
# A tibble: 6 x 28  
  PID county      state  area poptotal popdensity popwhite popblack popamerindian  
  <int> <chr>    <chr> <dbl>   <int>    <dbl>    <int>   <int>        <int>  
1  561 ADAMS    IL    0.052  66090   1271.   63917   1702         98  
2  562 ALEXANDER IL    0.014  10626    759    7054   3496        19  
3  563 BOND     IL    0.022  14991    681.   14477   429         35  
4  564 BOONE    IL    0.017  30806   1812.   29344   127         46  
5  565 BROWN    IL    0.018   5836    324.   5264   547         14  
6  566 BUREAU   IL    0.05   35688   714.   35157   50          65  
# ... with 19 more variables: popasian <int>, popother <int>, percwhite <dbl>,  
# perblack <dbl>, percamerindian <dbl>, percasian <dbl>, percother <dbl>,  
# popadults <int>, perchsds <dbl>, percollege <dbl>, percprof <dbl>,  
# poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,  
# perchildbelowpovert <dbl>, percadultpoverty <dbl>,  
# percelderlypoverty <dbl>, inmetro <int>, category <chr>
```

Alguns gráficos utilizando o pacote *ggplot2*



- Vamos construir um box-plot para a variável *percbelowpoverty* por estado:

```
ggplot(midwest,aes(state,percbelowpoverty)) + geom_boxplot() +  
  theme_light() + xlab("Estado") + ylab("% abaixo da linha de pobreza")
```

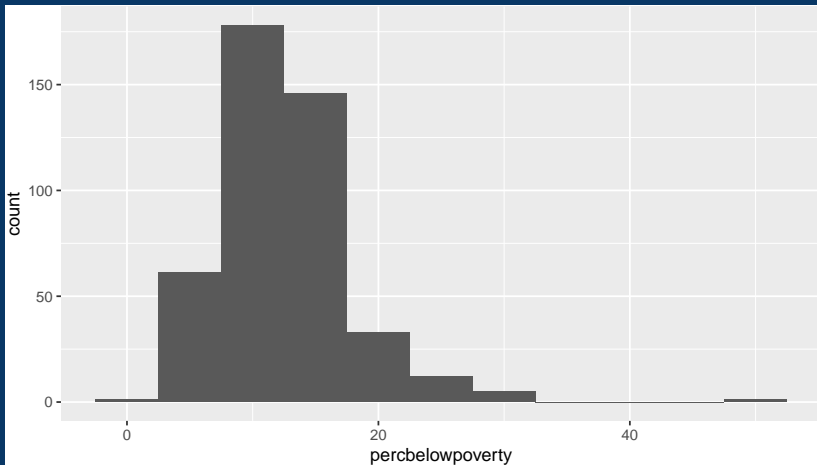


Alguns gráficos utilizando o pacote *ggplot2*



- Vamos construir um histograma para a variável *percbelowpoverty*:

```
ggplot(midwest, aes(percbelowpoverty)) +  
geom_histogram(binwidth = 5)
```

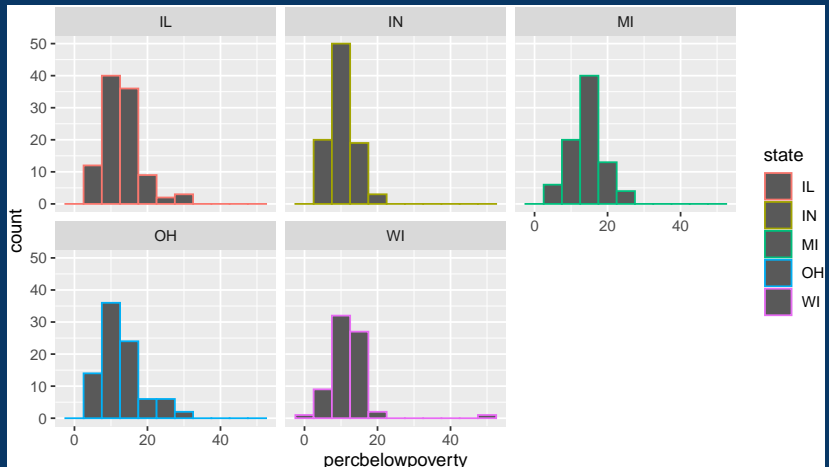




Alguns gráficos utilizando o pacote *ggplot2*

- Podemos ainda fazer mais de um gráfico por vez, por exemplo, um histograma para a variável *percbelowpoverty* por estado:

```
ggplot(midwest,aes(percbelowpoverty, color = state)) +  
geom_histogram(binwidth = 5) + facet_wrap(~state,nrow = 3,ncol = 3)
```

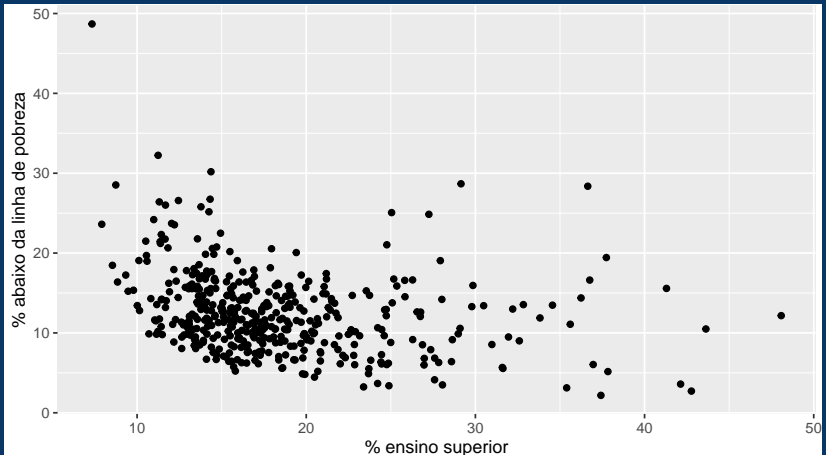




Alguns gráficos utilizando o pacote *ggplot2*

- Vamos construir um gráfico de dispersão para as variáveis *percollege* e *percbelowpoverty* por estado:

```
ggplot(midwest,aes(percollege,percbelowpoverty)) + geom_point() +  
xlab("% ensino superior") + ylab("% abaixo da linha de pobreza")
```





Alguns gráficos utilizando o pacote *ggplot2*

- Podemos utilizar alguns recursos avançados para construir gráficos. Vamos construir um gráfico de dispersão para as variáveis *area* e *poptotal*:

```
ggplot(midwest, aes(area, poptotal)) +  
  geom_point(aes(col = state, size = popdensity)) + xlim(c(0,0.1)) +  
  ylim(c(0,500000)) + geom_smooth(method = "lm", se = F)
```

