



Linguagem R

Aula 9

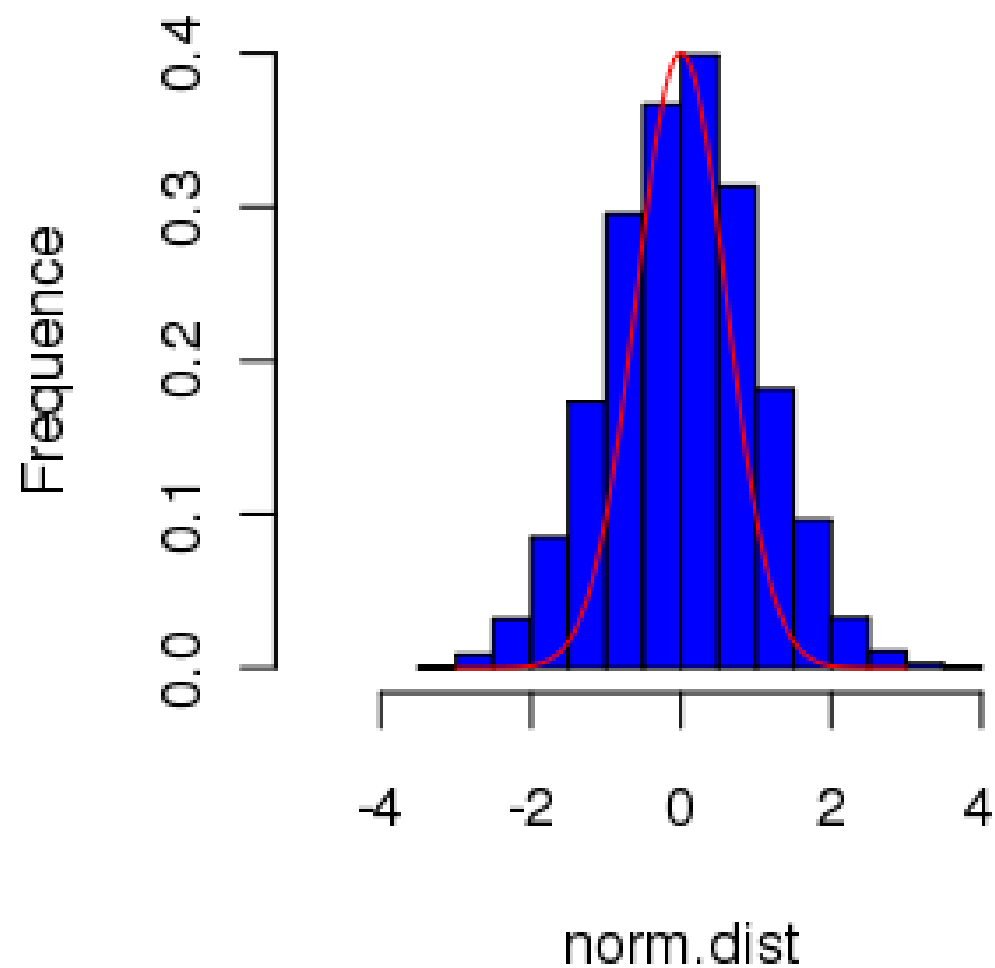
- Objetivo da aula:
- Distribuição Normal;
- Teorema Central do Limite;
- Teste de Normalidade;
- Medidas a longo prazo.

Distribuição Normal:

Ao se pegar dados aleatórios de uma variável aleatória ocorre em algumas situações que a distribuição destes dados obedece uma lei chamada de distribuição normal.

Por exemplo se forem coletadas as alturas de uma população com 1000 pessoas de um condomínio, provavelmente essa distribuição será normal.

Histogram of norm.dist



Distribuição Normal:

Em R utiliza-se a função `dnorm(x, mean, sd)` e a função `rnorm(n, mean, sd)` para se estudar a distribuição normal.

Usando a função `dnorm()`:

Crie uma distribuição uniforme de dados entre 0 e 10, por exemplo:

```
x <- seq(0, 5, by = .1)
```

Escolha um valor para mean e outro para sd, por exemplo 4.5 e 0.2.

Rode no RStudio a função `dnorm()`.

```
y <- dnorm(x, mean = 2.5, sd = 0.2)
```

A seguir `plot(x,y)` para ver o plot da distribuição normal.

```
185  
186 x <- seq(0, 5, by = .1)  
187 y <- dnorm(x, mean = 2.5, sd = 0.2)  
188 plot(x,y)  
189  
190  
191
```

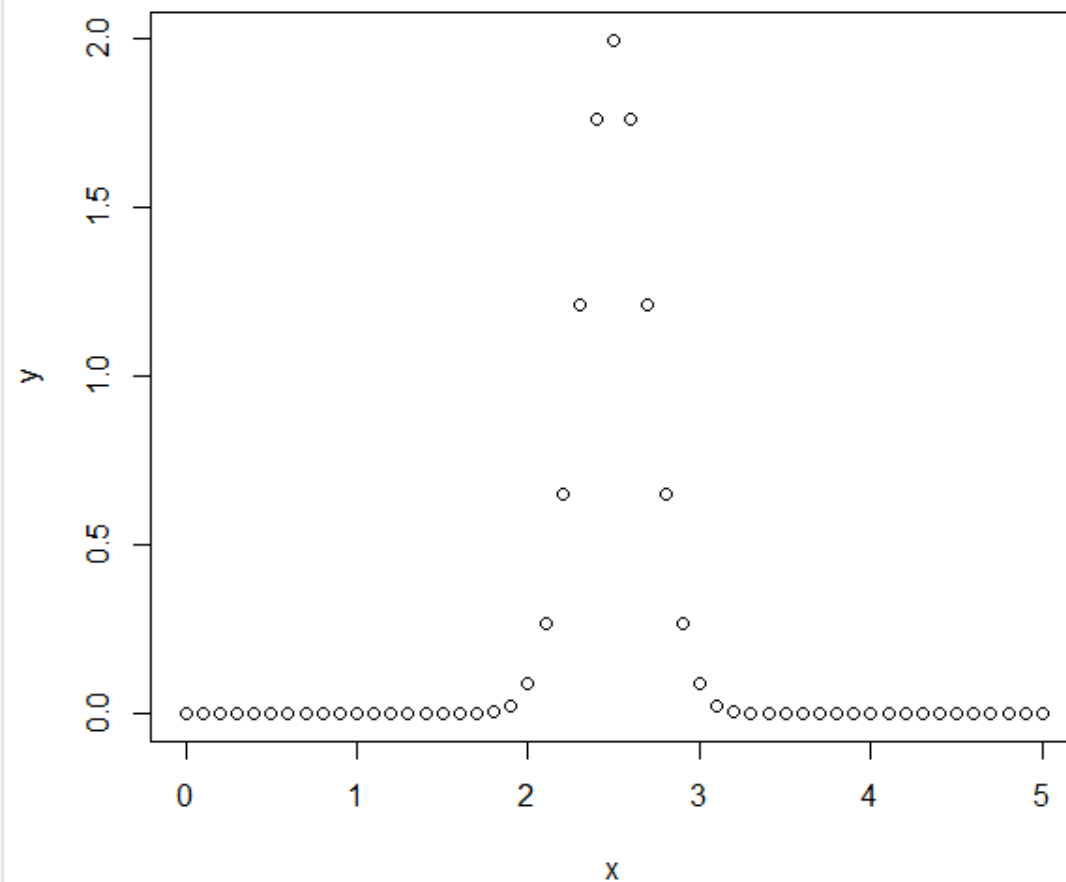
186:1 (Top Level) ↕

R Script ↕

Console Terminal × Background Jobs ×

R 4.2.2 · ~/

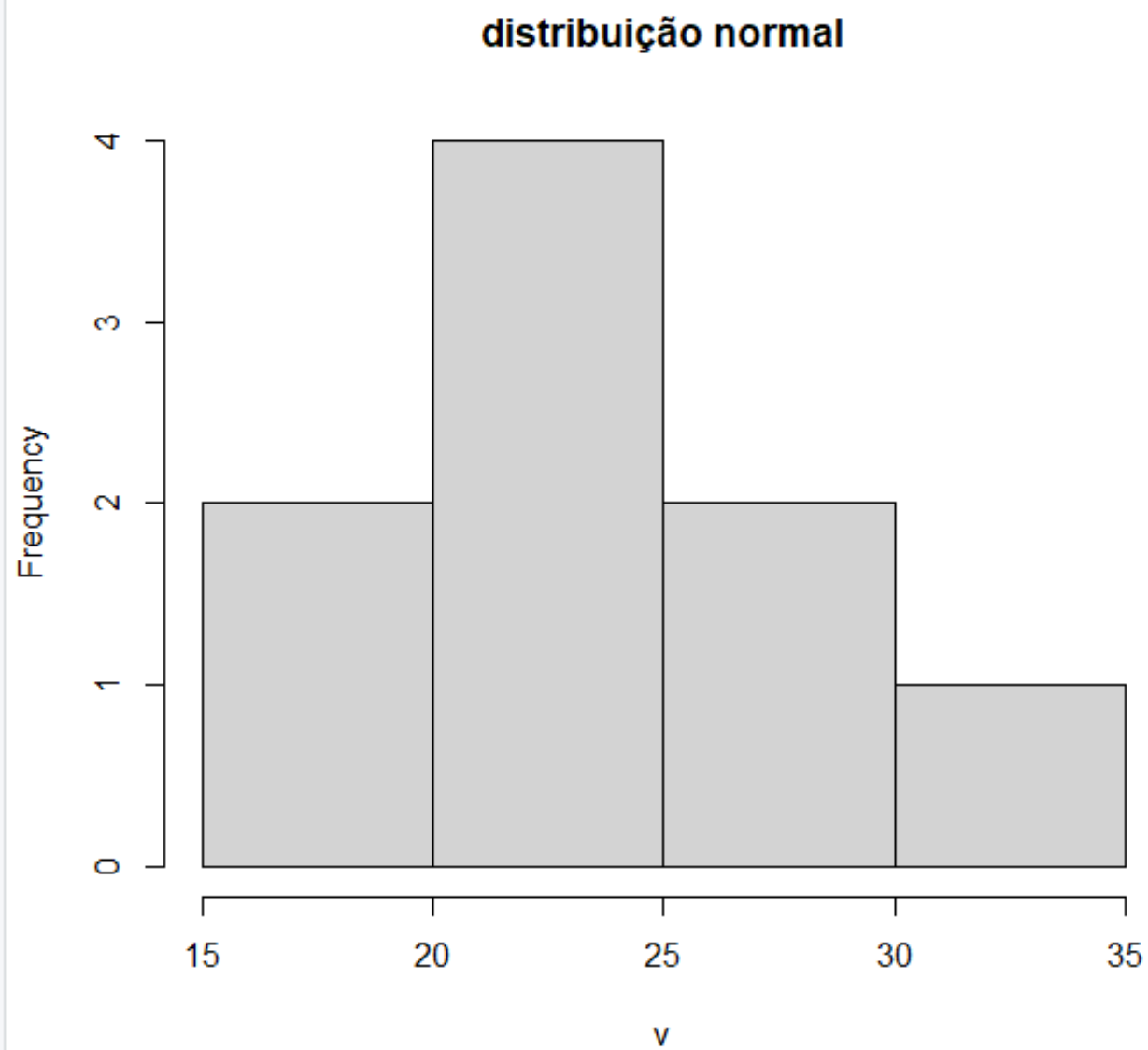
```
· x <- seq(0, 5, by = .1)  
· y <- dnorm(x, mean = 2.5, sd = 0.2)  
· plot(x,y)  
·
```



Histograma:

Construa o histograma da distribuição normal a partir dos dados:

```
v <- c(25,23,21,28,22,15,31,27,19)  
hist(v, main = "distribuição normal")
```

Teorema do Limite Central:

O teorema afirma que quando o tamanho da amostra aumenta, a distribuição amostral da sua média aproxima-se cada vez mais de uma distribuição normal.

O teorema do limite central também afirma que a distribuição amostral terá as seguintes propriedades:

A média da distribuição amostral será igual à média da distribuição populacional:

O desvio padrão da distribuição amostral será igual ao desvio padrão da distribuição populacional dividido pelo tamanho da amostra.

Aplicação do Teorema do Limite Central:

1º Passo: Constrói-se um histograma com dados de uma distribuição normal.

Para isto usa-se a função: `rnorm(valor,mean,sd)` que gera dados aleatórios seguindo uma distribuição normal.

```
data<- rnorm(1000,500,10)
```

```
hist(data, main = "distribuição normal")
```

```
print(data)
```

```

194 data<- rnorm(1000,500,10)
195 hist(data, main = "distribuição normal")
196 print(data)
197 sample1 <- c()
198 n = 100
199 for (i in 1:n){
200   sample1[i] = mean(sample(data, 5, replace=TRUE))
201 }
202 print(sample1)
203 hist(sample1, col = 'steelblue', xlab='Turtle shell width', main='sample size = 5')
204
205

```

194:1 (Top Level) ↕

R Script ↕

Console Terminal × Background Jobs ×

R 4.2.2 · ~/

```

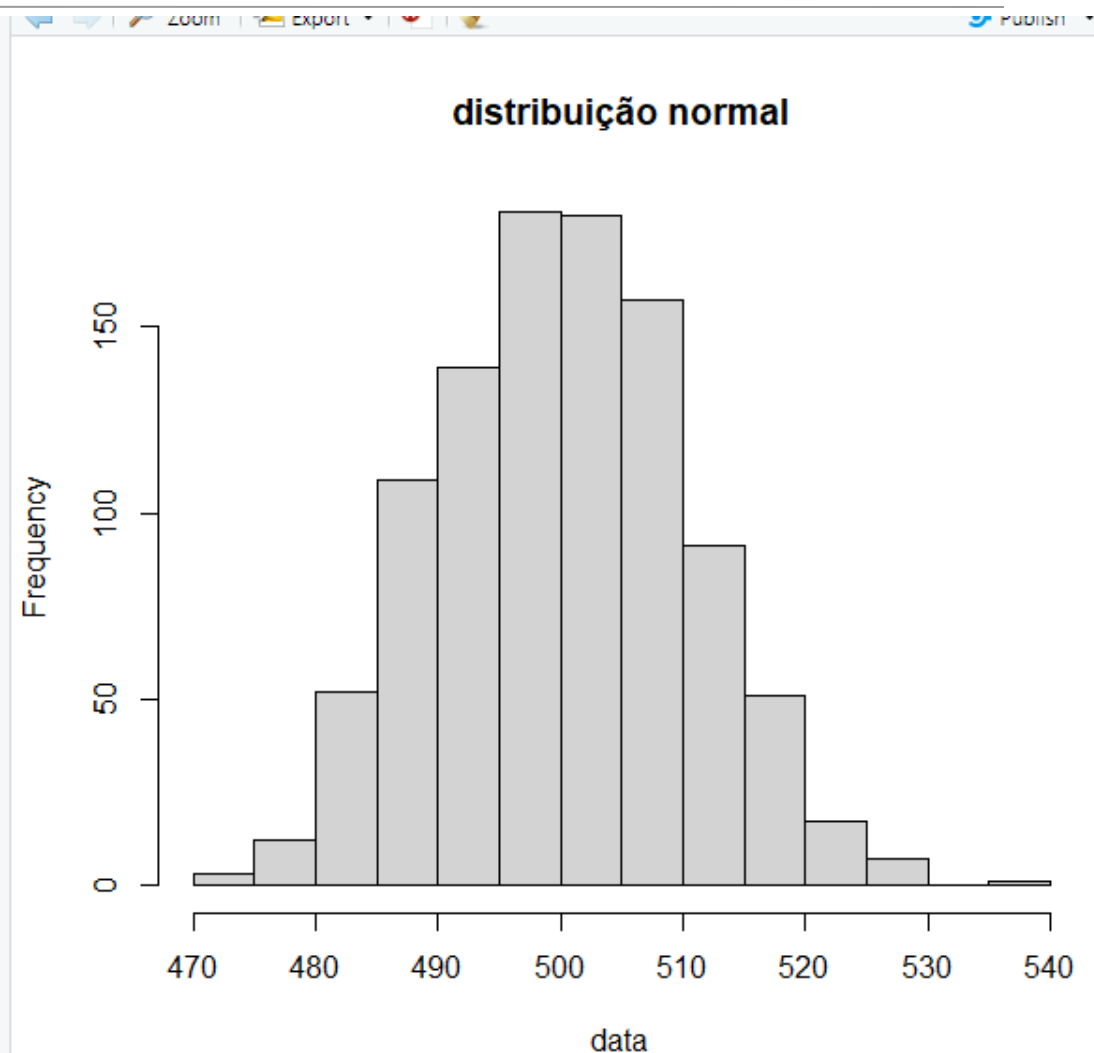
[865] 490.4198 506.1347 507.7428 498.8287 503.2428 514.6199 490.5721 504.8633
[873] 473.1106 488.8803 513.7662 504.3498 501.6836 478.3306 507.4567 490.6548
[881] 495.1306 508.6868 499.3628 515.7836 502.3905 501.5593 488.1636 492.7914
[889] 489.7250 498.6131 507.5182 493.1422 518.2536 521.9070 508.9145 496.8944
[897] 488.2736 497.4648 500.0665 504.7115 492.5447 503.1446 482.3344 493.7637
[905] 481.0503 483.2076 493.3678 500.9274 496.7802 501.4530 497.0713 508.9015
[913] 497.3081 516.2666 504.9917 489.9207 494.4819 488.6784 500.4332 474.0452
[921] 494.8237 500.7002 514.5683 502.4793 503.8692 494.8962 491.8152 499.5105
[929] 496.4557 520.8240 497.4984 507.8591 483.1766 502.9102 498.3227 494.7478
[937] 505.4103 496.1294 501.4012 505.6456 511.1312 506.2277 489.1746 496.7467
[945] 512.2083 502.1847 506.2125 507.6418 511.4386 498.3548 484.6627 498.0400
[953] 497.3359 502.2895 503.7970 495.6071 484.8433 496.0370 495.2682 494.5925
[961] 503.8511 521.1077 504.0968 493.7871 500.3176 518.6806 502.8365 491.2755
[969] 501.9449 522.4462 505.1002 515.8067 479.7057 496.1123 491.8197 503.2780
[977] 512.4372 485.6955 507.1676 499.4168 486.9425 507.7697 499.4074 483.0313
[985] 489.3051 523.2644 495.1049 504.8646 507.1884 492.1783 490.1657 500.5430
[993] 513.0106 491.4723 499.9184 493.8808 484.8472 510.9066 488.5181 510.8846

```

```

> data<- rnorm(1000,500,10)
> hist(data, main = "distribuição normal")
> print(data)
+

```



Aplicação do Teorema do Limite Central:

2º Passo: Gera-se uma amostra aleatória de dados da população.

```
sample <- c()
```

```
n = 100
```

```
for (i in 1:n){
```

```
  sample[i] = mean(sample(data, 5, replace=TRUE))
```

```
}
```

```
print(sample)
```

```
hist(sample, col ='steelblue', xlab='x', main='Sample size = 5')
```

```

194 data<- rnorm(1000,500,10)
195 hist(data, main = "distribuição normal")
196 print(data)
197 sample <- c()
198 n = 100
199 for (i in 1:n){
200   sample[i] = mean(sample(data, 5, replace=TRUE))
201 }
202 print(sample)
203 hist(sample, col = 'steelblue', xlab='x', main='sample size = 5')
204
205

```

197:1 (Top Level) ↕

R Script ↕

Console

Terminal ×

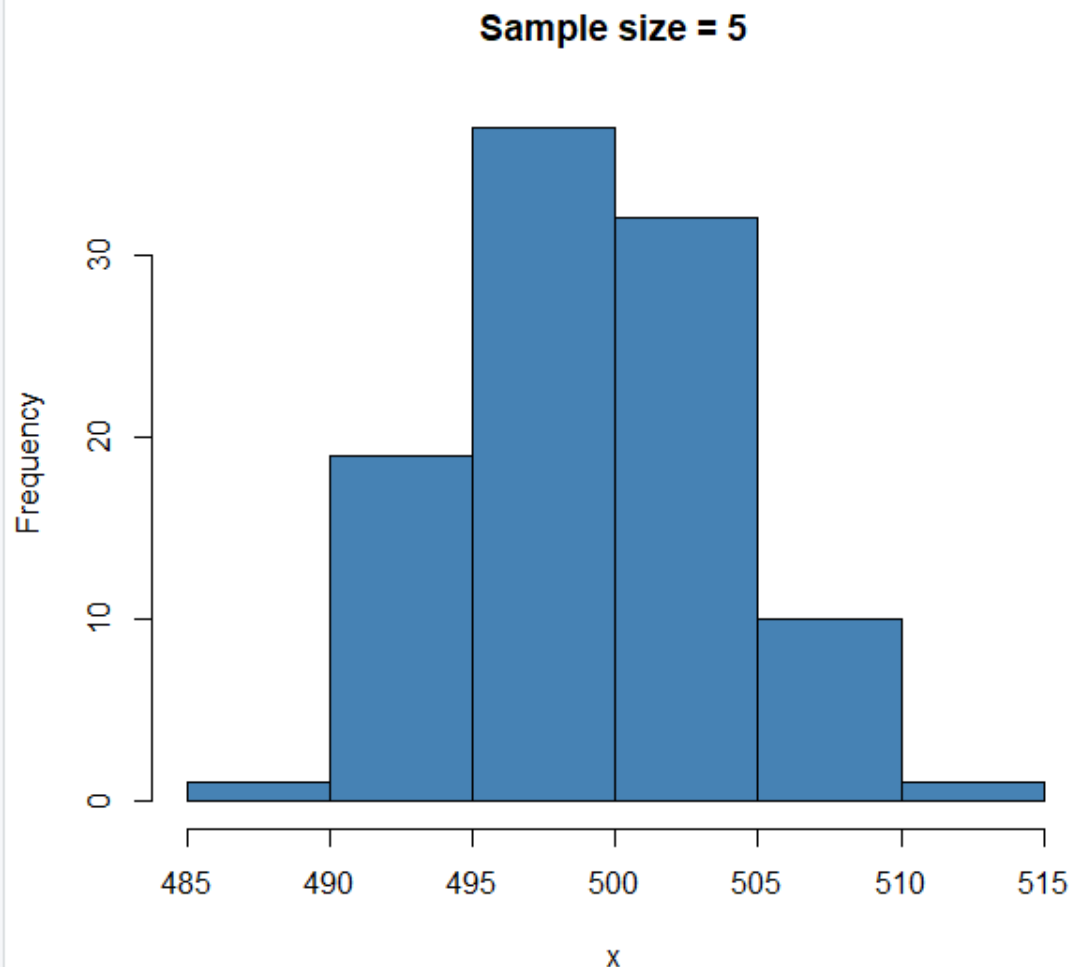
Background Jobs ×

R 4.2.2 · ~/

```

> sample <- c()
> n = 100
> for (i in 1:n){
+   sample[i] = mean(sample(data, 5, replace=TRUE))
+ }
> print(sample)
 [1] 508.2297 499.5633 493.6163 507.1279 496.4855 493.1121 493.2687 496.5737
 [9] 501.8190 503.8312 507.2441 499.5176 499.4596 500.9993 495.7602 497.0363
[17] 499.3316 501.8225 493.2678 501.4151 509.2878 495.9333 499.3737 497.4824
[25] 494.4853 501.3777 501.2901 502.0456 496.3738 500.3675 505.3328 494.8295
[33] 499.7385 494.6413 504.9080 498.9818 499.6709 494.5761 493.6830 494.2429
[41] 494.1312 502.4542 503.1816 492.4319 495.0324 503.3660 500.9128 494.7194
[49] 496.0895 505.7186 492.8713 503.8879 497.0952 504.8865 499.2170 498.1178
[57] 494.8485 498.9991 508.2911 502.8871 504.6890 498.5118 497.1553 501.4675
[65] 496.5636 503.0702 495.0897 501.6680 500.1803 487.0119 505.9479 502.8428
[73] 501.0566 501.8650 495.6753 497.0336 498.4576 506.0379 497.8512 502.1584
[81] 499.7886 498.0335 498.5362 494.6322 504.1784 500.8331 497.1472 496.8542
[89] 510.7682 501.9478 505.3453 503.9015 503.5944 494.3857 490.7127 497.6303
[97] 496.6225 494.9677 501.8397 498.9956
> hist(sample, col = 'steelblue', xlab='x', main='sample size = 5')
>

```



Teste de Normalidade:

Em estatística os testes de normalidade são usados para determinar se um conjunto de dados, de uma dada variável aleatória, obedece um lei de distribuição normal ou não. A suposição de normalidade dos dados amostrais é uma condição exigida para a realização de inferências sobre parâmetros populacionais.

Como regra para testes de normalidade, usa-se:

Se P-Value for maior que o nível de significância, os dados apresentam distribuição normal.

O nível de significância é adotado como 5%.

Teste de Normalidade:

Os pacotes do R básico, que já são instalados automaticamente, contam com dois únicos testes de normalidade pré-implementados: `ks.test` para o teste de Kolmogorov-Smirnov e `shapiro.test` para o teste de Shapiro-Wilk.

Teste de Normalidade:

Vamos usar o banco de dados interno do R chamado cars.
Para acessar o banco de dados cars deve-se proceder da seguinte maneira:

```
install.packages('lattice')
```

```
library(lattice)
```

A seguir digite:

```
data(cars)
```

```
head(cars)
```

para ver a tabela.

Teste de Normalidade:

A seguir gere o histograma com `hist(cars$speed)` para a variável `speed`.

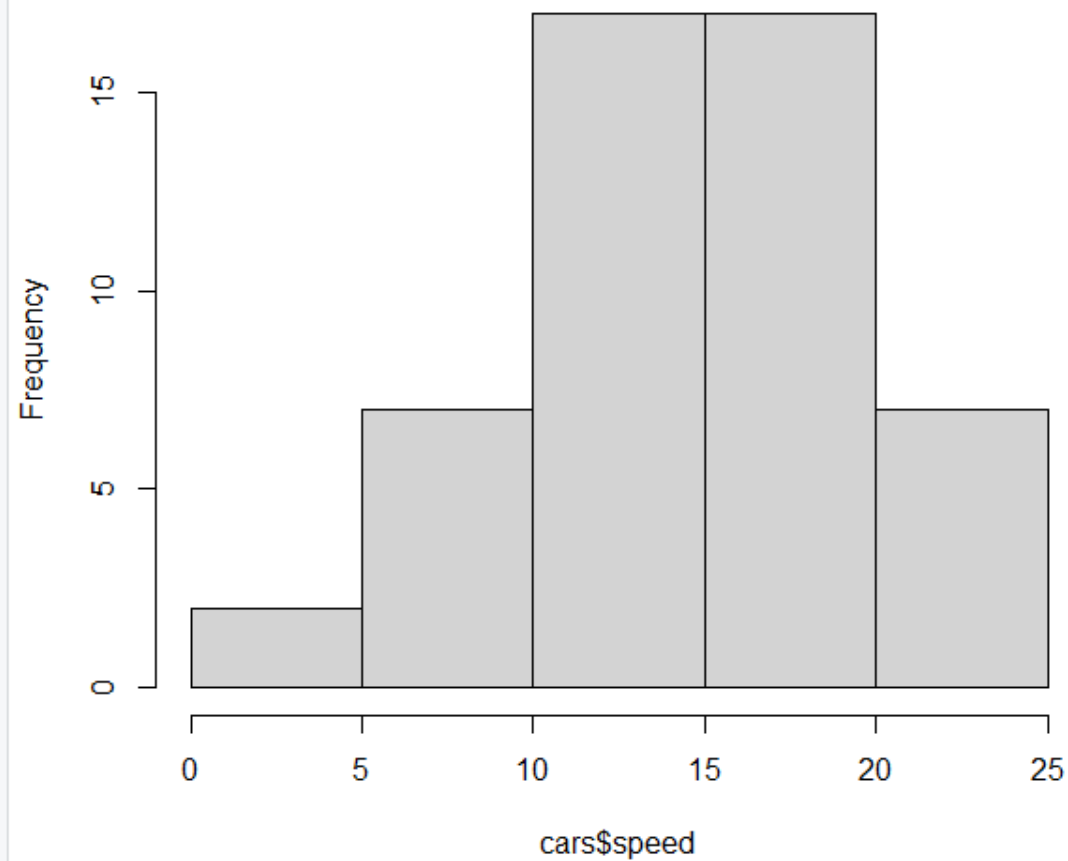
```
207  
208 data(cars)  
209 head(cars)  
210 hist(cars$speed)
```

Console Terminal Background Jobs

R 4.2.2 · ~/

```
> data(cars)  
> head(cars)  
  speed dist  
1     4    2  
2     4   10  
3     7    4  
4     7   22  
5     8   16  
6     9   10  
> hist(cars$speed)  
>
```

Histogram of cars\$speed



Teste de Normalidade:

O histograma da variável speed nos sugere uma normalidade dos dados, mas é necessário realizar os testes para confirmar este fato.

Teste Shapiro-Wilk:

Para o teste de s-w utiliza-se o comando "shapiro.test".
Use o comando: `shapiro.test(cars$speed)`
O resultado do teste é mostrado no print a seguir:

```
shapiro-wilk normality test  
  
data:  cars$speed  
W = 0.97765, p-value = 0.4576  
.
```

Teste Shapiro-Wilk:

Neste teste se o p-valor for < 0.05 indica que os dados não apresentam normalidade.

O p-valor foi de 0.45 e isto quer dizer que os dados estão seguindo uma distribuição normal.

Teste de Kolmogorov-Smirnov:

Outro teste muito utilizado é o k-s. Para realizar este teste é necessário instalar o pacote “dgof”.

Para isto usa-se:

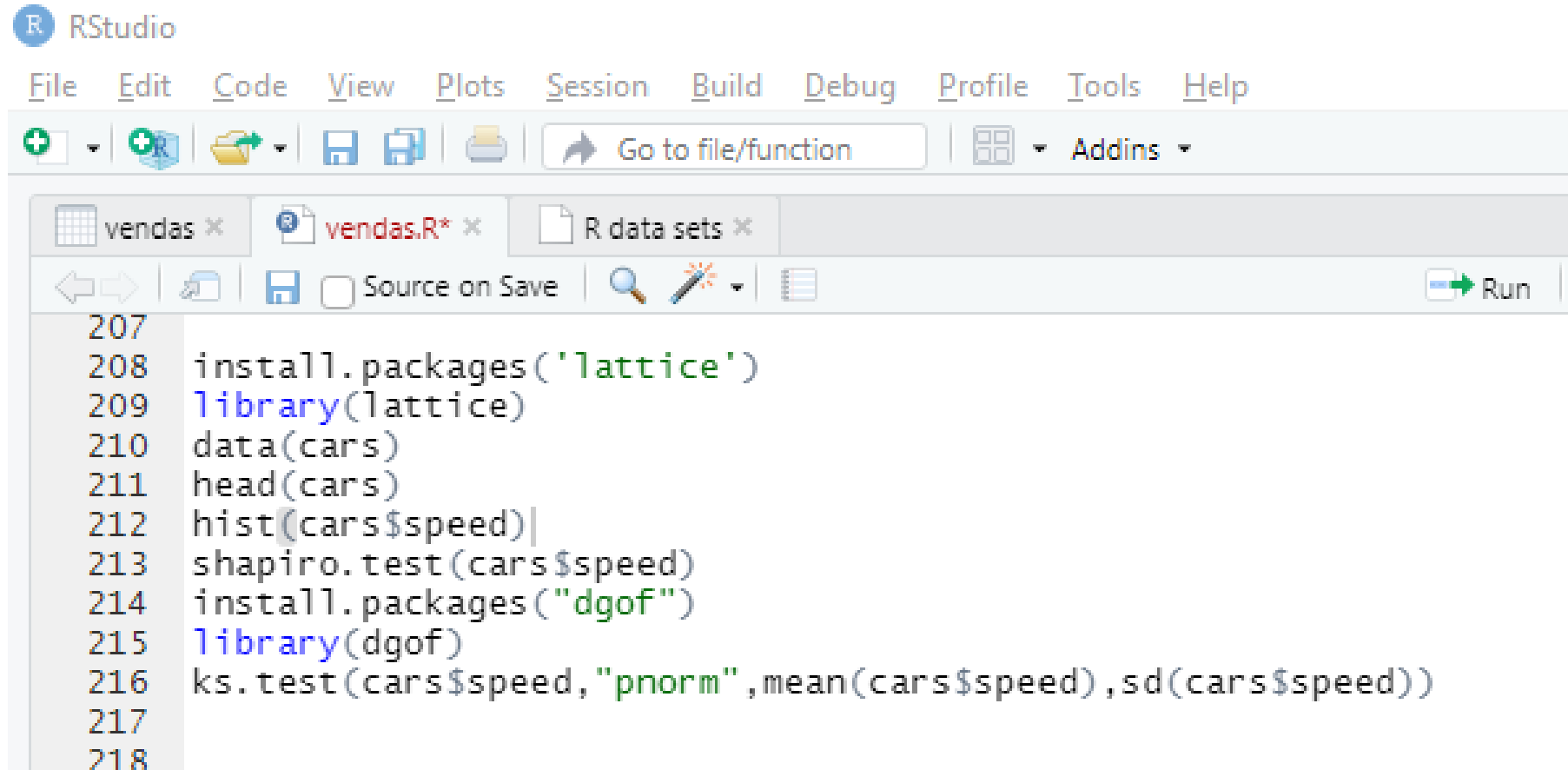
```
install.packages("dgof")  
library(dgof)
```

Teste de Kolmogorov-Smirnov:

A seguir, execute o teste k-s. Para isto use o comando:
`ks.test(cars$speed,"pnorm",mean(cars$speed),sd(cars$speed))`
O resultado é mostrado no print a seguir:

```
one-sample Kolmogorov-Smirnov test

data:  cars$speed
D = 0.068539, p-value = 0.9729
alternative hypothesis: two-sided
```

The screenshot shows the RStudio application window. The title bar reads "RStudio". The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations (new, open, save, print), a search bar labeled "Go to file/function", and a "Addins" dropdown. The tab bar shows three open tabs: "vendas" (spreadsheet icon), "vendas.R*" (script icon), and "R data sets" (data icon). Below the tabs is a toolbar with navigation arrows, a "Source on Save" checkbox, a search icon, a "Run" button with a green arrow, and a console icon. The main editor area displays the following R code:

```
207  
208 install.packages('lattice')  
209 library(lattice)  
210 data(cars)  
211 head(cars)  
212 hist(cars$speed)|  
213 shapiro.test(cars$speed)  
214 install.packages("dgof")  
215 library(dgof)  
216 ks.test(cars$speed, "pnorm", mean(cars$speed), sd(cars$speed))  
217  
218
```

Conclusão:

Ambos os testes mostraram que a distribuição dos dados segue um modelo de distribuição normal.

Muitos trabalhos realizados tem mostrado a eficiência dos testes de normalidade e o teste de Shapiro-Wilk tem dado um resultado melhor.

Exercício:

Crie um dataframe com os dados da tabela a seguir, gere 3 histogramas, realize 3 testes para verificar se as amostras obedecem uma distribuição normal.

Dados:

Concentrações de poluentes na água de uma lagoa em mg/L.

Data 25/08/2012	Data 13/10/2012	Data 15/12/2012
100,1	89,2	90,8
69,3	29,5	67
28	85	104,5
128	51,2	120,2
41	58,6	75
36,8	60,1	67,8
51,5	62,2	55,8