Distribuição Normal

Aula 2

Prof. André Luiz Cunha

21/05/2021

1 Transformação de escala

Um passo importante de qualquer análise de dados é a uniformização do intervalo de dados, de modo que todas as variáveis do banco de dados tenham o mesmo intervalo de variação.

Seja o exemplo do conjunto de dados aleatório abaixo, são apresentados dois tipos de transformação encontrados na literatura.

```
# Conjunto de valores aleatórios com média 50 e desvio 15.
(x <- rnorm(100, 50, 15))
```

```
##   [1] 67.38209 44.36467 53.80939 41.33888 51.60008 54.31530 61.89694 45.33601
##   [9] 60.76003 65.07051 40.53391 35.18354 55.74108 54.07982 36.79692 55.87327
##  [17] 32.93785 37.29865 46.96603 37.63343 53.79591 37.12693 52.32424 43.21072
##  [25] 53.20274 27.54581 52.22474 38.83846 57.11614 68.78142 54.43861 49.09645
##  [33] 19.13876 59.21151 49.28278 50.96855 76.42924 62.61803 65.64291 38.92497
##  [41] 63.61846 53.22992 48.17272 41.52407 42.59956 26.12273 37.13655 36.17765
##  [49] 27.79757 48.58399 19.93381 41.39033 22.66142 65.34623 74.32932 64.87410
##  [57] 61.96386 70.37860 34.95970 49.56880 63.71918 37.05237 60.04204 62.99392
##  [65] 63.36854 26.33910 60.65801 43.19757 44.96063 45.94747 29.23396 42.26117
##  [73] 57.44031 52.76061 69.90471 29.75177 46.80179 60.51939 74.13881 43.52736
##  [81] 80.58713 56.49009 59.40942 44.13878 35.62000 37.41026 44.42762 62.02210
##  [89] 50.20614 27.40463 36.73793 40.95965 35.47438 38.82704 31.44681 27.83082
##  [97] 43.80613 19.71721 24.16069 53.54825
```

```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.14   37.26   47.57   47.84   59.26   80.59
```

1.1 Normalização [0,1]

A normalização transforma os dados no intervalo entre 0 e 1.

```
x_norm <- (x - min(x)) / (max(x) - min(x))
summary(x_norm)
```

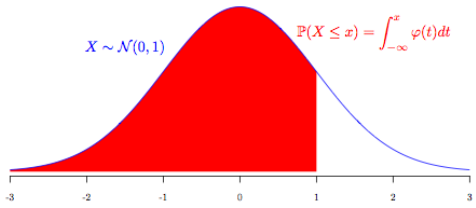
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2949  0.4627  0.4671  0.6529  1.0000
```

1.2 Padronizar [-3,3]

A padronização dos dados nada mais é do que a transformação para a escala da curva normal padrão (z-padrão). Vide Figura 1 a tabela z-padrão.

```
x_pad <- (x - mean(x)) / sd(x)
summary(x_pad)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.04903 -0.75548 -0.01936  0.00000  0.81531  2.33780
```



	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990

Figura 1: Tabela z-padrão

```
## OLHANDO A TABELA
#z = 1,0 ----> p(z) = 0,1587
1 - 2 * 0.1587
```

```
## [1] 0.6826
```

```r
#z = 2,0 ----> p(z) = 0,0228
1 - 2 * 0.0228
```

```
## [1] 0.9544
```

```r
# p(z) = 95% -----> z(p = 0,025) = ?
1.96
```

```
## [1] 1.96
```

```r
# p(z) = 99% -----> z =??
2.575
```

```
## [1] 2.575
```

2 Funções do R

2.1 Números aleatórios

```r
## Uniformemente distribuídos
```

Função: `runif(n, min, max)`

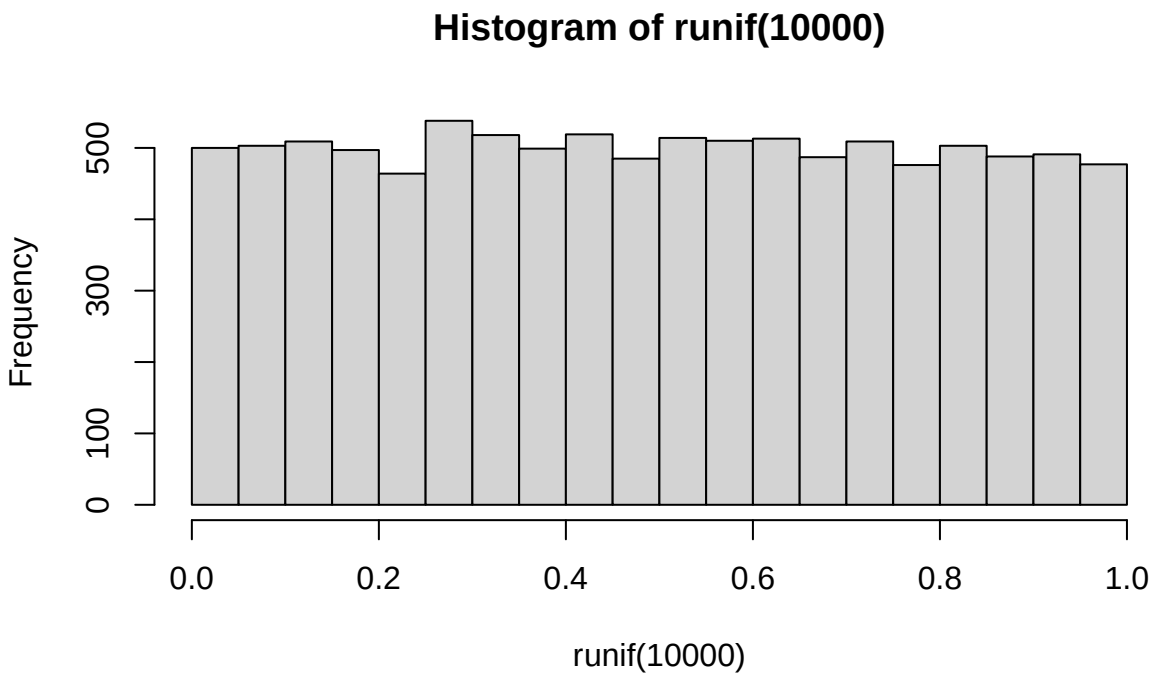
```r
runif(10)
```

```
##  [1] 0.2921710 0.1243811 0.7045777 0.9484180 0.5995654 0.8572610 0.9077824
##  [8] 0.5266294 0.4437137 0.8350196
```

```r
runif(10, 100, 150)
```

```
##  [1] 137.2926 117.7441 118.1983 144.7208 142.3846 146.1939 135.9226 116.5154
##  [9] 117.0558 133.1327
```

```r
hist(runif(10000))
```




```
## Normalmente distribuídos
```

Função: `rnorm(n, mean, sd)`

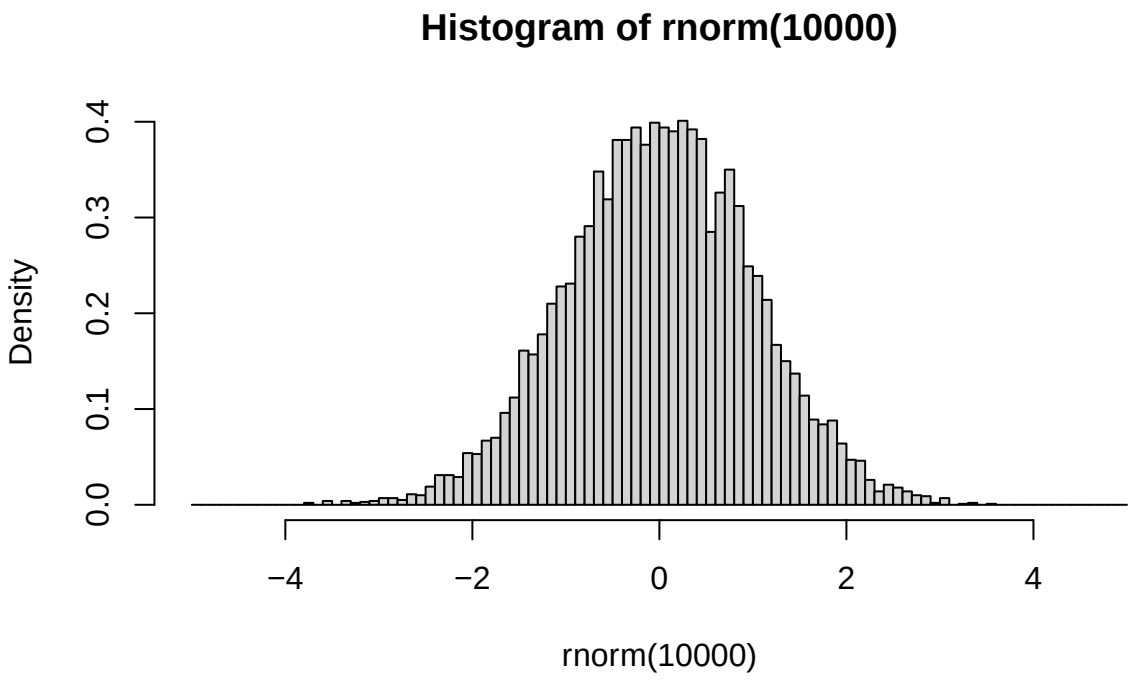
```
rnorm(10)
```

```
##  [1]  0.01682654  0.51782051 -0.52881088  0.31302410 -0.72271552  0.38712121
##  [7]  0.40100879  0.87852581  1.70679692 -2.34912372
```

```
rnorm(10, 100, 15)
```

```
##  [1] 103.13431  94.11965  92.94349 104.12939  88.03196  95.35177 104.54607
##  [8] 102.47912 103.32911  85.33530
```

```
hist(rnorm(10000), breaks = seq(-5,5,.1),
     freq = FALSE)
```



2.2 Distribuição Normal

Encontrando o valor z-padrão com a função `qnorm(area da curva, mean=0, sd=1)`

- Unicaudal a esquerda: `z_alpha = qnorm(alpha)`
- Unicaudal a direita: `z_alpha = qnorm(1 - alpha)`
- Bicaudal: `z_alpha/2 = qnorm(1 - alpha/2)`

```
qnorm(.90)
```

```
## [1] 1.281552
```

```
qnorm(.5)
```

```
## [1] 0
```

Encontrando o p-valor com a função `pnorm(valor z, mean=0, sd=1)`

- Unicaudal a esquerda: `p-value = pnorm(z, lower.tail=TRUE)`
- Unicaudal a direita: `p-value = pnorm(z, lower.tail=FALSE)`
- Bicaudal: `p-value = 2 * pnorm(abs(z), lower.tail=FALSE)`

```
pnorm(1.96)
```

```
## [1] 0.9750021
```

```r
pnorm(1.96, lower.tail = FALSE)
```

```
## [1] 0.0249979
```

```r
pnorm(0)
```

```
## [1] 0.5
```

Encontrando a densidade do valor com a função `dnorm(valor z, mean=0, sd=1)`

```r
dnorm(1.96)
```

```
## [1] 0.05844094
```

```r
dnorm(-1.96)
```

```
## [1] 0.05844094
```

```r
dnorm(0)
```

```
## [1] 0.3989423
```

EXEMPLO 1

```r
## P(z > 1,65)
pnorm(1.65, lower.tail = FALSE)
```

```
## [1] 0.04947147
```

```r
## P(z < 1,65)
pnorm(1.65)
```

```
## [1] 0.9505285
```

```r
## P(1,40 < z < 1,70)
pnorm(1.7) - pnorm(1.4)
```

```
## [1] 0.0361912
```

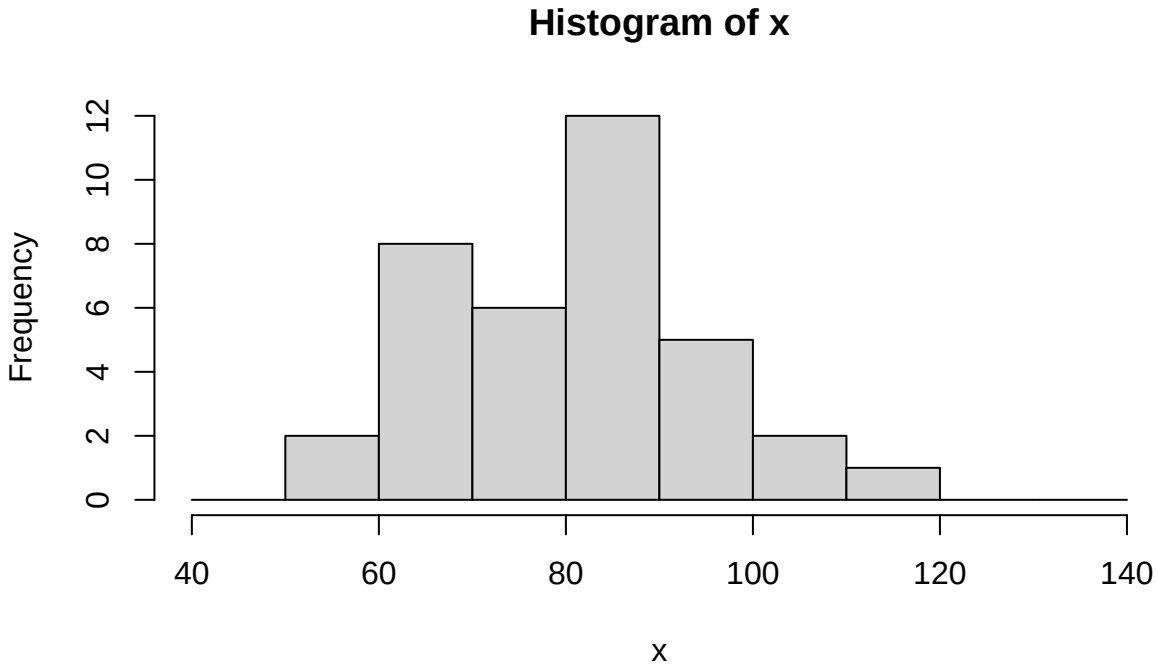
EXEMPLO 2

```r
x = c(58,78,84,90,97,70,
      90,86,82,59,90,70,
      74,83,90,75,88,84,
      68,96,70,94,70,110,
      67,68,75,80,68,82,
      104,92,112,84,98,80)


## Análise descritiva
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   58.00   70.00   82.50   82.39   90.00  112.00
```

```
hdados <- hist(x,
               breaks = seq(40,140,10))
```




```
hdados$breaks
```

```
##  [1]  40  50  60  70  80  90 100 110 120 130 140
```

```
hdados$counts
```

```
##  [1]  0  2  8  6 12  5  2  1  0  0
```

```
hdados$density
```

```
##  [1] 0.000000000 0.005555556 0.022222222 0.016666667 0.033333333 0.013888889
##  [7] 0.005555556 0.002777778 0.000000000 0.000000000
```

O parâmetro `density` traz a razão entre a porcentagem de elementos e o intervalo de bins, tanto que a soma das porcentagens `density` é igual a
`0.1`

```
#.

sum(hdados$density)
```

```
## [1] 0.1
```

Ao multiplicar cada densidade pelo intervalo do bin, a porcentagem total será de 100%.

```
sum(hdados$density) * 10
```

```
## [1] 1
```

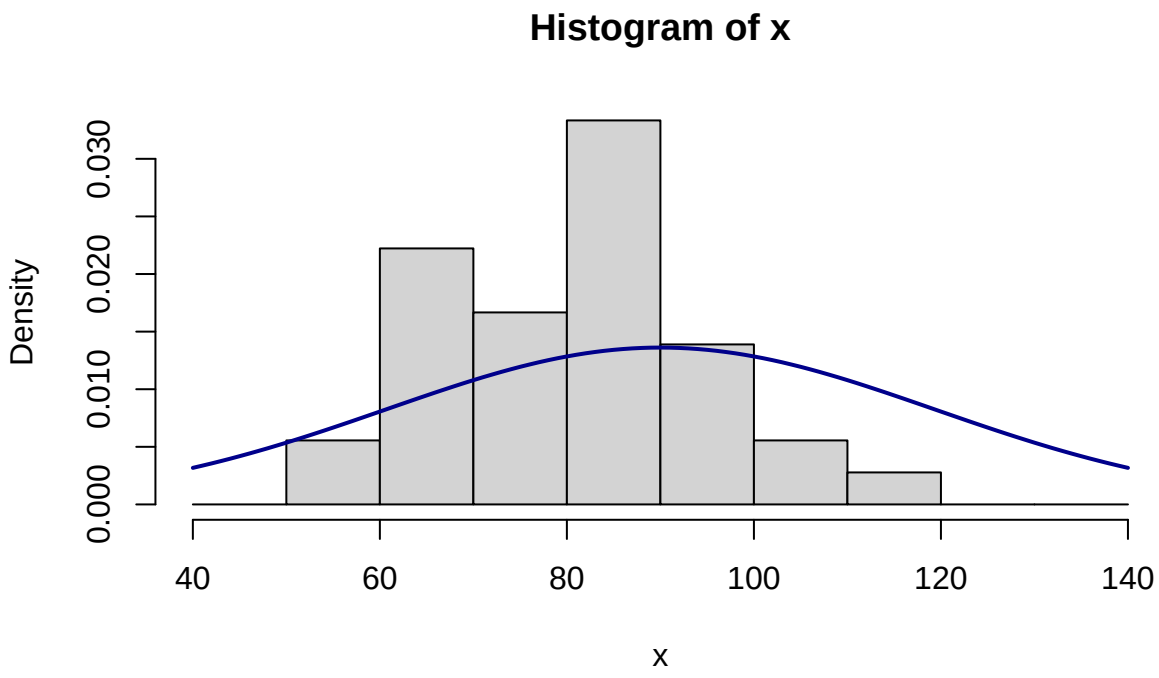
3 Testes do R

```
x_pad <- (x - mean(x))/sd(x)
```

3.1 Qui-quadrado

Teste de aderência de Qui-quadrado é usado para compara distribuições observadas com distribuições esperadas em dados discretos (histogramas de frequências).

```
## Densidade dos valores X com a curva normal teórica
bin <- 10
hist.real <- hist(x, breaks = seq(40,140,bin), freq=FALSE)
curve(dnorm(x, mean(x), sd(x)), col='darkblue', lw=2, add=TRUE)
```



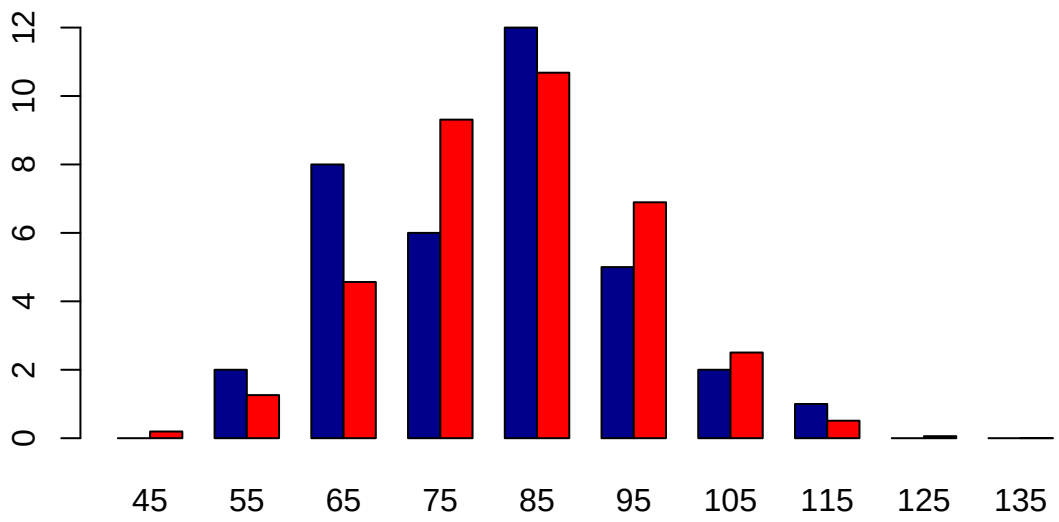
```
hist.real$density
```

```
##  [1] 0.000000000 0.005555556 0.022222222 0.016666667 0.033333333 0.013888889
##  [7] 0.005555556 0.002777778 0.000000000 0.000000000
```

```
hist.real$counts
```

```
##  [1]  0  2  8  6 12  5  2  1  0  0
```

```
## Frequências das distribuições observadas e esperadas
barplot(rbind(hist.real$counts, dnorm(hist.real$mids, mean(x), sd(x))*bin*length(x) ),
        names.arg = hist.real$mids,
        col = c("darkblue", "red"),
        beside = TRUE
)
```




```
chisq.test(hist.real$counts, # Frequência observada - dados originais
           p = dnorm(hist.real$mids, mean(x), sd(x))*bin, # Frequência teórica - distribuição normal
           rescale.p = TRUE )
```

```
## Warning in chisq.test(hist.real$counts, p = dnorm(hist.real$mids, mean(x), :
## Chi-squared approximation may be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  hist.real$counts
## X-squared = 5.7035, df = 9, p-value = 0.7692
```

3.2 Kolmogorv-Smirnov (KS)

```
ks.test(x, "pnorm", mean(x), sd(x))
```

```
## Warning in ks.test(x, "pnorm", mean(x), sd(x)): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.10407, p-value = 0.8304
## alternative hypothesis: two-sided
```

```
ks.test(x_pad, "pnorm")
```

```
## Warning in ks.test(x_pad, "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x_pad
## D = 0.10407, p-value = 0.8304
## alternative hypothesis: two-sided
```

3.3 Shapiro-Wilk

```
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.97612, p-value = 0.6139
```