Universidade Federal da Bahia
Instituto Multidisciplinar em Saúde
Curso de extensão: Introdução ao uso do programa R
Prof. Leandro Martins de Freitas e Profª Leila Cruz

Introdução ao uso do programa R

Universidade Federal da Bahia
Instituto Multidisciplinar em Saúde
Curso de extensão: Introdução ao uso do programa R
Prof. Leandro Martins de Freitas e Profª Leila Cruz

Introdução ao uso do programa R
Dia 4: Análise de dados I

Nosso objetivo hoje será conhecer as funções que o R possui para análise estatística. Vamos abordar os testes de estatística frequentista mais comumente utilizados, para análise de dados quantitativos e qualitativos.

De novo, nosso propósito aqui é apresentar as ferramentas existentes no R e as possibilidades de ampliação, e não discutir os fundamentos e metodologias dos testes. Estes últimos variam muito e devem ser escolhidos tendo em vista o tipo de dado que se quer analisar.

1. Correlação

Testes de correlação mensuram a associação entre duas variáveis. O coeficiente de correlação é escala-invariante, simétrico e varia de -1 a +1. Os extremos indicam forte associação positiva ou negativa, e coeficientes iguais a zero indicam ausência de relação.

A função que computa correlações no R é "`cor()`", que pode ser aplicada a dois ou mais vetores. É necessário, no entanto, explicitar o procedimento a ser realizado no caso de ausência de valores (`NA`); isto é especificado no argumento "`use`", havendo três opções possíveis:

1. **`use = all.obs`**: assume que não há ocorrêncis "`NA`" e produz erro caso existam.
2. **`use = complete.obs`**: exclui ocorrências de "`NA`" na leitura de vetores individuais.
3. **`use = pairwise.complete.obs`**: exclui ocorrências de "`NA`" quando correlacionando vetores mas mantém os valores não-NA de vetores que estejam também na correlação.

Para entender isso melhor, veja o exemplo[1]:

```
#exemplo de métodos para tratar NAs em correlação de vetores
#cria vetores para correlacionar:
> v1<-c(56, 123, 546, 26, 62, 6, NA, NA, NA, 15)
> v2<-c(21, 231, 5, 5, 32, NA, 1, 231, 5, 200)
> v3<-c(NA, NA, 24, 51, 53, 231, NA, 153, 6, 700)

#cria matriz
> mat<-cbind(v1,v2,v3)

       v1  v2  v3
 [1,]  56  21  NA
 [2,] 123 231  NA
 [3,] 546   5  24
 [4,]  26   5  51
 [5,]  62  32  53
 [6,]   6  NA 231
 [7,]  NA   1  NA
 [8,]  NA 231 153
 [9,]  NA   5   6
[10,]  15 200 700


#correlação entre v1, v2 e v3
#use="all.obs"
> cor(mat,use="all.obs", method="pearson")
Error in cor(mat, use = "all.obs", method = "pearson") :
  observações faltantas em cov/cor

#use="complete.obs"
> cor(mat,use="complete.obs", method="pearson")
           v1         v2         v3
v1  1.0000000 -0.4348000 -0.4190907
v2 -0.4348000  1.0000000  0.9929753
v3 -0.4190907  0.9929753  1.0000000




#use="pairwise.complete.obs"
```

[1] Exemplo encontrado no fórum Stack Overflow: “Complete.obs of cor() function” http://stackoverflow.com/questions/18892051/complete-obs-of-cor-function

```
> cor(mat,use="pairwise.complete.obs", method="pearson")
           v1         v2         v3
v1  1.0000000 -0.2902929 -0.4108040
v2 -0.2902929  1.0000000  0.6964304
v3 -0.4108040  0.6964304  1.0000000
```

Nesse exemplo, o vetor v1 tem valores para todas as variáveis, mas os vetores v2 e v3, não. Por isso, uma correlação usando o método “all.obs” resulta em erro. Na correlação usando “complete.obs”, são descartadas as linhas completas da matriz caso exista um “NA”, de modo que no exemplo acima foram descartadas as linhas 1, 2, 6, 7, 8 e 9. No último exemplo, a correlação usando o método “pairwise.complete.obs” descarta ***apenas*** as ocorrências “NA”, preservando os outros valores presentes nos demais vetores. Nas linhas 1 e 2 são preservados os valores de v1 e v2, na linha 6 é descartado o valor de v2, na linha 7 são descartados os valores de v1 e v3 e nas linhas 8 e 9 são descartados os valores de v1. Observe como o resultado dos coeficientes de correlação mudaram com a alteração dos valores que são descartados.

Um exemplo para aplicar: Relação entre as medidas da corola das espécies de *Iris*

Para começar, vamos testar a normalidade das distribuições:

```
###Testa normalidade separadamente
#cria df só para medidas de Iris setosa
> setosa<-which(iris[,5]=="setosa")
> setosa<-iris[setosa,]
> shapiro.test(setosa[,1])    #coluna1: Sepal.Length
> shapiro.test(setosa[,2])    #coluna2: Sepal.Width
> shapiro.test(setosa[,3])    #coluna3: Petal.Length
> shapiro.test(setosa[,4])    #coluna4: Petal.Width

#cria df só para medidas de Iris versicolor
versicolor<-which(iris[,5]=="versicolor")
> versicolor<-iris[versicolor,]
> shapiro.test(versicolor[,1])
> shapiro.test(versicolor[,2])
> shapiro.test(versicolor[,3])
> shapiro.test(versicolor[,4])

#cria df só para medidas de Iris virginica
virginica<-which(iris[,5]=="virginica")
> virginica<-iris[virginica,]
> shapiro.test(virginica[,1])
> shapiro.test(virginica[,2])
> shapiro.test(virginica[,3])
```

```
> shapiro.test(virginica[,4])
```

Os testes de normalidade indicaram que as medidas de comprimento e largura e sépala e comprimento de pétala são normais para as três espécies. A largura da pétala não tem distribuição normal para os dados das espécies *I. setosa* e *I. versicolor*, mas passa no teste de normalidade para *I. virginica*. Sendo assim, iremos computar as correlações entre as medidas das duas primeiras espécies com o teste paramétrico, a correlação de Pearson, para as duas primeiras espécies e o teste não paramétrico, a correlação de Spearman, para a terceira espécie.

```
> cor(setosa[c(1:4)],method="spearman")
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000   0.7553375    0.2788849   0.2994989
Sepal.Width     0.7553375   1.0000000    0.1799110   0.2865359
Petal.Length    0.2788849   0.1799110    1.0000000   0.2711414
Petal.Width     0.2994989   0.2865359    0.2711414   1.0000000

> cor(versicolor[c(1:4)],method="spearman")
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000   0.5176060    0.7366251   0.5486791
Sepal.Width     0.5176060   1.0000000    0.5747272   0.6599826
Petal.Length    0.7366251   0.5747272    1.0000000   0.7870096
Petal.Width     0.5486791   0.6599826    0.7870096   1.0000000

> cor(virginica[c(1:4)],method="pearson")
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000   0.4572278    0.8642247   0.2811077
Sepal.Width     0.4572278   1.0000000    0.4010446   0.5377280
Petal.Length    0.8642247   0.4010446    1.0000000   0.3221082
Petal.Width     0.2811077   0.5377280    0.3221082   1.0000000
```

e então testamos a significância dos coeficientes encontrados:

```
### Testa significância
# Sepal.Length vs Sepal.Width

> cor.test(setosa[,1],setosa[,2],method="pearson")

      Pearson's product-moment correlation

data:  setosa[, 1] and setosa[, 2]
t = 7.6807, df = 48, p-value = 6.71e-10
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5851391 0.8460314
sample estimates:
      cor
0.7425467

> cor.test(versicolor[,1],versicolor[,2],method="pearson")

      Pearson's product-moment correlation

data:  versicolor[, 1] and versicolor[, 2]
t = 4.2839, df = 48, p-value = 8.772e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2900175 0.7015599
sample estimates:
      cor
0.5259107

> cor.test(virginica[,1],virginica[,2],method="pearson")

      Pearson's product-moment correlation

data:  virginica[, 1] and virginica[, 2]
t = 3.5619, df = 48, p-value = 0.0008435
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2049657 0.6525292
sample estimates:
      cor
0.4572278
# Petal.Length vs. Petal.Width
> cor.test(setosa[,3],setosa[,4],method="spearman")
```

```
        Spearman's rank correlation rho

data:  setosa[, 3] and setosa[, 4]
S = 15178.48, p-value = 0.05683
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2711414

> cor.test(versicolor[,3],versicolor[,4],method="spearman")

        Spearman's rank correlation rho

data:  versicolor[, 3] and versicolor[, 4]
S = 4435.525, p-value = 1.229e-11
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7870096

> cor.test(virginica[,3],virginica[,4],method="pearson")

        Pearson's product-moment correlation

data:  virginica[, 3] and virginica[, 4]
t = 2.3573, df = 48, p-value = 0.02254
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.0480704 0.5510499
sample estimates:
      cor
0.3221082
```

Exercício: Sua vez: teste agora a correlação entre as medidas da SÉPALA versus as medidas das PÉTALAS. Atenção para a normalidade das distribuições. O que você encontrou?

Exercício: O **dataset Thuesen** apresenta resultados de pesquisa que mensurou a velocidade de contração muscular em pacientes com diabetes melitus, e suas respectivas taxas de glicose sanguínea.

a. Teste a correlação entre estas características.
b. Teste a significância da correlação com a função cor.test().

2. Testes de comparação de grupos: Teste t, Mann-Whitney, Kruskal-Wallis, X-quadrado

A comparação de grupos é realizada de diversas formas. Medidas de tendência central (média, mediana), comparando as diversidades (desvio padrão).
Para avaliar a médida de tendência central e comparar dois grupos podemos usar a função `mean()` ou `median()`.

A função mean retorna a média dos valores contidos no vetor.
função mean:

x: vetor numérico

trim: porcentagem dos valores extremos que deseja eliminar do calculo. Valores de 0 a 1.

na.rm: valor lógico indicando se os valores NA devem ser removidos.

Vamos considerar esses dois vetores, pop1 e pop2, onde foi medida uma variavel de 30 indivíduos em cada população.

```
#carregar o vetor
>pop1 <- read.csv("pop1.csv", row.names=1)
>pop2 <- read.csv("pop2.csv", row.names=1)

>mean(pop1[,1])
>mean(pop2[,1])
```

Agora vamos fazer a média retirando os valores extremos, usando o parâmentro trim = 0.1

```
#estamos removendo 3 elementos do lado esquerdo e 3 do lado direito do
vetor em ordem crescente
>mean(pop1[,1], trim=0.1)
>mean(pop2[,1], trim=0.1)
```

Agora vamos usar a mediana como medida de tendência central.

x: vetor numérico

na.rm: valor lógico indicando se os valores NA devem ser removidos.

Como a mediana não sofre influência de valores extremos, não tem a necessidade de remover esses elementos do vetor.

```
>median(pop1[,1])
>median(pop2[,1])
```

Como medida de dispersão dos dados podemos calcular o desvio padrão com a função `sd()` ou o desvio da mediana com a função `mad()`. A função `sd()` calcula a variação com base na média, enquanto que a função `mad()` calcula o desvio com base na mediana.

função sd:

x: vetor numérico
na.rm: valor lógico indicando se os valores NA devem ser removidos.

```
>sd(pop1[,1])
>sd(pop2[,1])
```

mad:[2]

x: vetor numérico
na.rm: valor lógico indicando se os valores NA devem ser removidos.

```
>mad(pop1[,1])
>map(pop2[,1])
```

2.1 Teste-t

Vamos comparar as duas amostras e verificar se a diferença das médias é igual a zero.
Para isso vamos usar um teste de hipótese:

H0: as médias das duas populações não diferem
H1: as médias das duas populações diferem

parâmetros do t.test

```
>t.test(pop1[,1], pop2[,1])
```

Foi realizada a contagem de Caranguejo-ferradura em diferentes praias. Esse é um uso de teste t pareado. Nesse teste não é comparado a média das amostras, mas a diferença das observações e se esse diferença é significativa.

H0: o número de caranguejos não alterou do ano de 2011 para 2012
H1: o número de caranguejos alterou do ano de 2011 para 2012 (aumentou ou reduziu)
alfa = 0.05

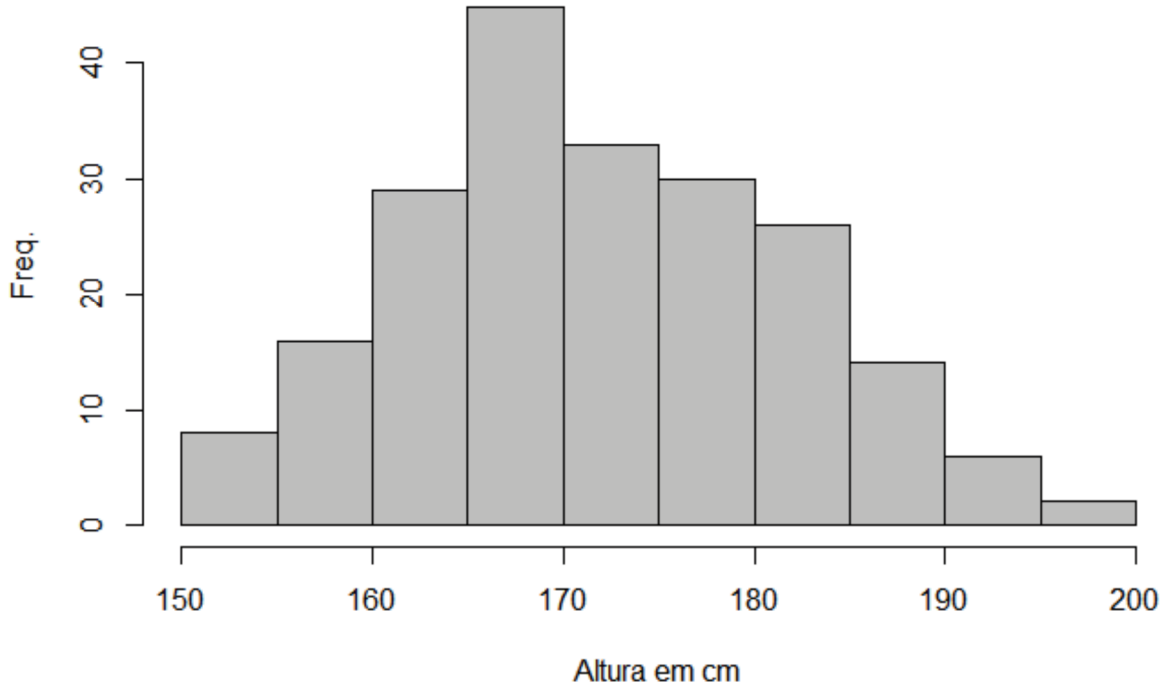
```
>praia<-read.csv("praia.TXT", row.names=1,sep=";")
>dim(praia)
>head(praia)
```

[2] A função mad tem outros parâmetros para decidir quando o vetor tem número par de elementos. Consulte `?mad` para ver quais são esses parâmetros.

```
#variaveis nominais
#Beach
#2011
#2012
#variavel medida, número de caranguejos
>ano2011 <- praia$X2011
>ano2012 <- praia$X2012
>t.test(ano2011, ano2012, paired=T)

      Paired t-test

data:  ano2011 and ano2012
t = 2.1119, df = 24, p-value = 0.04529
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
   642.1576 55808.6424
sample estimates:
mean of the differences
                28225.4
```

Vamos agora usar o dataset pesquisa.csv. Esse data set é uma pesquisa feita com estudantes universitários. Vamos fazer alguns data seting para fazer nossas análises.

```
>pesquisa <- read.csv("pesquisa.csv", row.names=1)
>dim(pesquisa)
>head(pesquisa)
>hist(pesquisa$Height, xlab="Altura em cm",
     ylab="Freq.", main="", col="gray",
     breaks="sturges")
```




```
#teste de normalidae

>shapiro.test(pesquisa$Height)
       Shapiro-Wilk normality test

   data:  pesquisa$Height
   W = 0.98841, p-value = 0.08844

#testando se altura entre M e F é igual
#H0: não existe diferença na média de altura entre M e F
#H1: existe diferença na média de altura entre M e F

>t.test(Height ~ Sex, data=pesquisa, paired=F)

Welch Two Sample t-test

data:  Height by Sex
t = -12.924, df = 192.7, p-value <
2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

```
 -15.14454 -11.13420
sample estimates:
mean in group Female   mean in group Male
            165.6867             178.8260


>shapiro.test(pesquisa$Pulse)
        Shapiro-Wilk normality test

    data:  pesquisa$Pulse
    W = 0.98742, p-value = 0.08631


>t.test(Pulse ~ Sex, data=pesquisa, paired=F)

      Welch Two Sample t-test

data:  Pulse by Sex
t = 1.1384, df = 188.7, p-value = 0.2564
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.413139  5.269937
sample estimates:
mean in group Female   mean in group Male
            75.12632             73.19792
```

2.2 Teste de Mann-Whitney

Compara a distribuição de duas amostras.

```
>wilcox.test(pop1[,1], pop2[,1], alternative = "g")
```

H0: distribuição dos dois grupos é identica
H1: distribuição dos dois grupos não é identica

2.3 Teste Kruskal–Wallis

Kruskal–Wallis compara uma variável que foi medida e outra nominal. O teste Kruskal–Wallis deve ser usado quando a variável que foi medida não tem distribuição normal.

H0: distribuição das amostras é igual entre os grupos
H1: distribuição das amostras é diferente entre os grupos

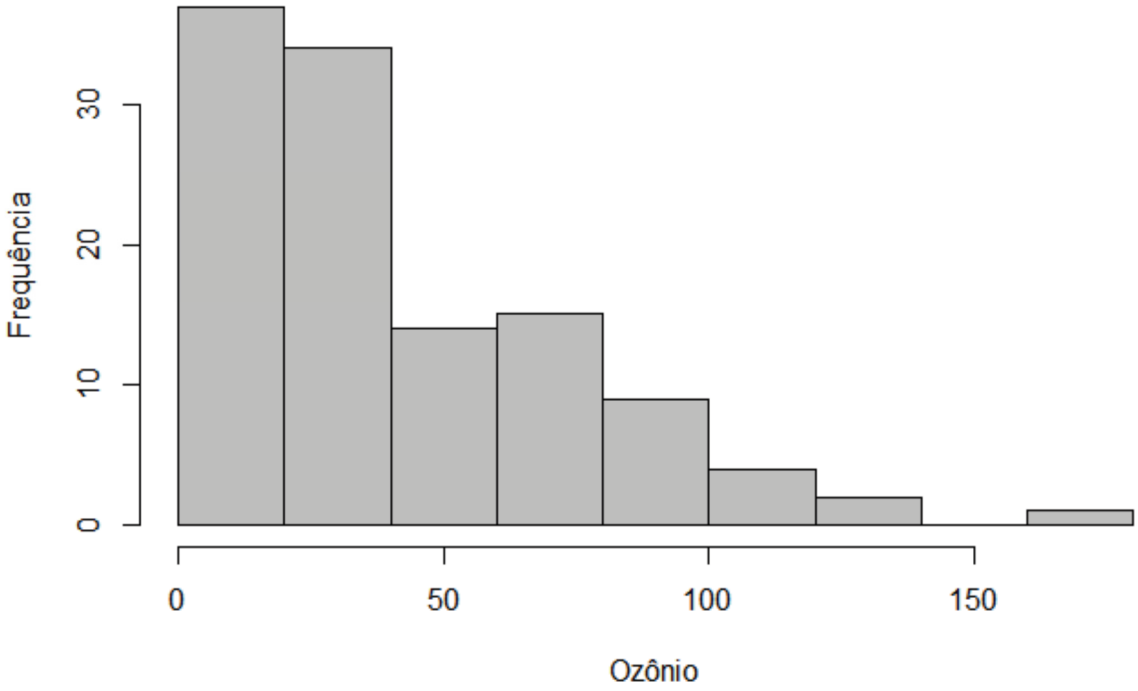
Carregar o arquivo qualidade-ar.csv na variavel qualidadeAr.

```
>qualidadeAr <- read.csv("qualidade_ar.csv", row.names=1)

#variável é ozonio, distribuição
ozonio <- airquality$Ozone
mes <- airquality$Month
#grupos serão os meses
table(mes)
```

Categorias que o teste Kruskal–Wallis vai comparar

Mês	5	6	7	8	9
Contagem	31	30	31	31	30

```
hist(ozonio, breaks="sturges", col="gray",
     xla="Ozônio", ylab="Frequência",
     main="")
```




```
#teste de normalidade
>shapiro.test(ozonio)
      Shapiro-Wilk normality test

   data:  ozonio
   W = 0.87867, p-value = 2.79e-08

#teste se ozonio apresenta a mesma distribuição nos meses
>kruskal.test(ozonio, mes)

            Kruskal-Wallis rank sum test

data:  ozonio and mes
Kruskal-Wallis chi-squared = 29.267, df = 4, p-value = 6.901e-06
```

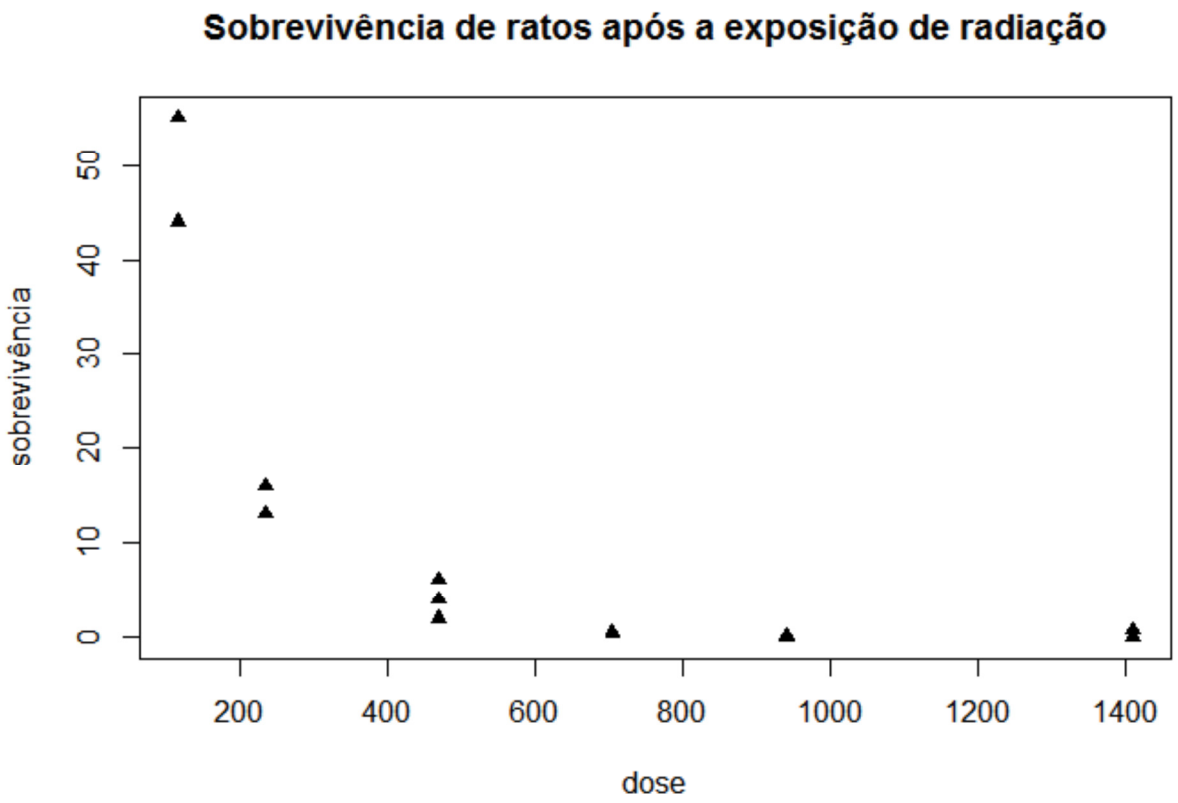
Usando valor alfa = 0.05, observamos que existe diferença na variável concentração de ozônio e entre os diferentes meses.

Exercício: O dataset **qualidadeAr** pode ser usado para fazer análise teste de hipótese com cada uma das variáveis usando os meses como grupos.

Vamos analisar um estudo feito medindo a sobrevivência de ratos após exposição a diferentes doses de radiação. Os valores das doses vamos usar como grupos. Os valores das doses foram 117.5 235 470 705 940 1410 rads.

Como existem vários grupos e a variável dependente não segue uma distribuição normal, vamos usar o teste Kruskal-Wallis para comparar os grupos e saber se existe diferença na sobrevivência dos ratos.

```
#carregando o dataset sobrevivencia
>sobrevivencia <- read.csv("sobrevivencia.csv", row.names=1)
>dim(sobrevivencia)
>head(sobrevivencia)

#distribuição da variável sobrevivência
>hist(sobrevivencia$surv, breaks="Sturges",
     col = "gray", ylab="Frequência",
     xlab="sobrevivência",
     main="Distribuição da sobrevivência\n a radiação")

#teste de normalidade
>shapiro.test(sobrevivencia$surv)

      Shapiro-Wilk normality test

data:  sobrevivencia$surv
W = 0.64198, p-value = 9.851e-05


#quais grupos existem
>table(sobrevivencia$dose)
117.5   235   470   705   940  1410
    2     2     3     2     3     2

> plot(sobrevivencia, pch=17, xlab="dose", ylab="sobrevivência",
main="Sobrevivência de ratos após a exposição de radiação")
```




```
#Usando valor alfa = 0.05
>kruskal.test(surv ~ dose, data= survival)

      Kruskal-Wallis rank sum test

data:  surv by dose
Kruskal-Wallis chi-squared = 11.657, df = 5, p-value = 0.0398
```

Usando valor alfa = 0.05, observamos que existe diferença na sobrevivência entre as diferentes doses de radiação.

2.4 Teste X^2

Para fazer o teste Chi-quadrado precisamos construir uma tabela de contingência. Vamos usar novamente o dataset pesquisa que possui dados sobre pesquisa com jovens universitários. Vamos comparar as variáveis Sexo, Exercício, e Tabagismo.

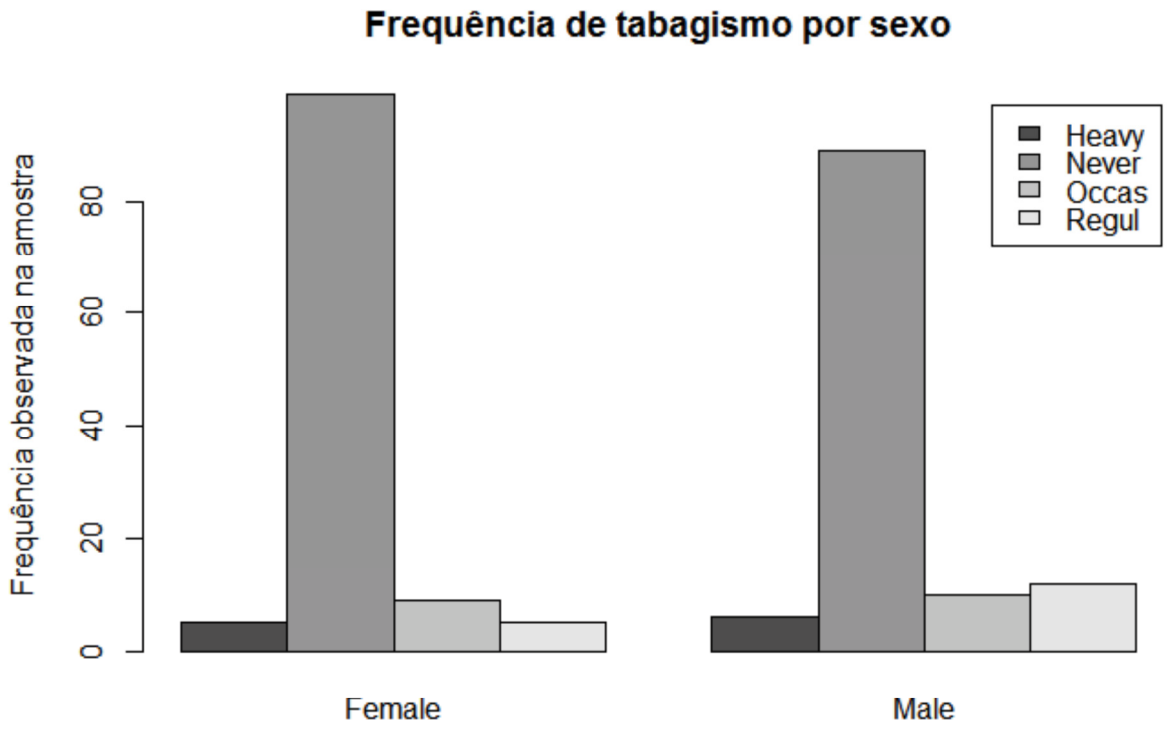
```
#tabela de contingência variaveis exercicio  e tabagismo
```

```
>Exer_Smoke <- table(pesquisa$Exer, pesquisa$Smoke)
#tabela de contingência variaveis sexo e tabagismo
>Sex_Smoke <- table(pesquisa$Sex, pesquisa$Smoke)
#tabela de contingência variaveis exercicio  e sexo
>Exer_Sex <- table(pesquisa$Exer, pesquisa$Sex)


#teste X2 variaveis exercicio  e sexo
>chisq.test(Exer_Sex)

    Pearson's Chi-squared test

 data:  Exer_Sex
 X-squared = 5.7184, df = 2, p-value =
    0.05731



#teste X2 variaveis exercicio  e tabagismo
>chisq.test(Exer_Smoke)

  Pearson's Chi-squared test

data:  Exer_Smoke
X-squared = 5.4885, df = 6, p-value =
  0.4828



#teste X2 variaveis sexo  e tabagismo
>chisq.test(Sex_Smoke)

  Pearson's Chi-squared test

data:  Sex_Smoke
X-squared = 3.5536, df = 3, p-value =
  0.3139
```

Vamos explorar mais os dados usado para fazer o X2 e fazer os gráficos de barras das variavéis nas tabelas de contingência.

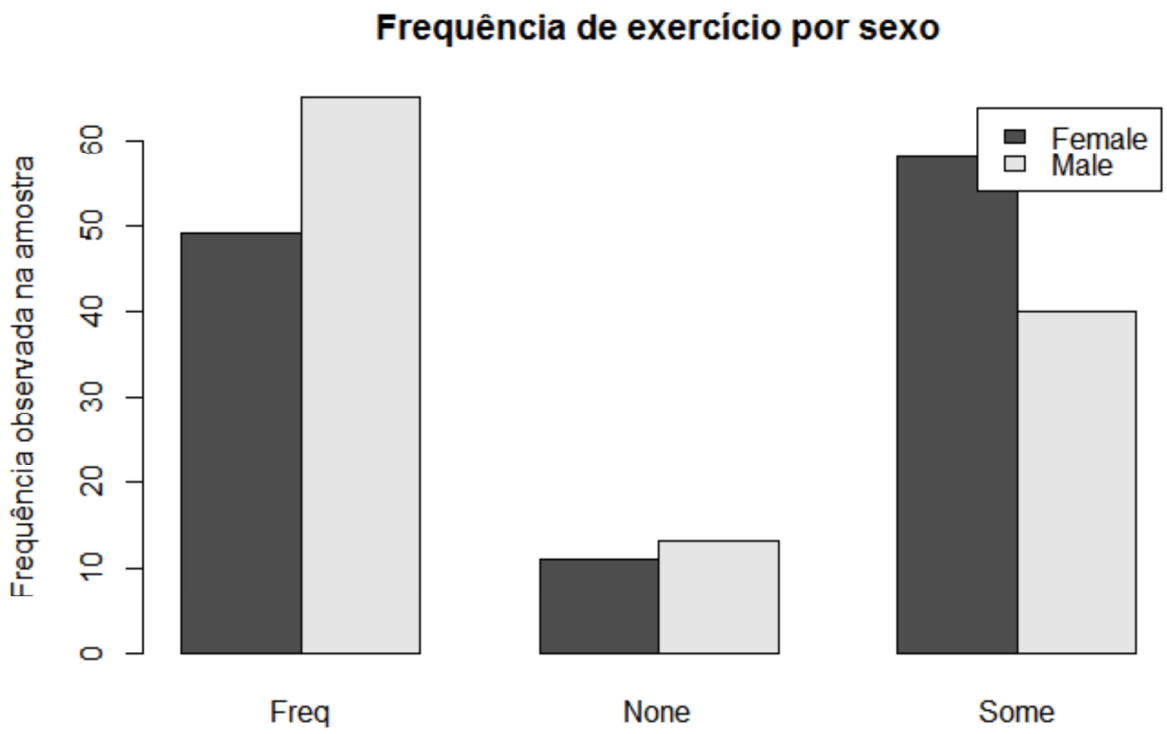
```
>barplot(t(Sex_Smoke), beside=T, legend=T,
```

```
        ylab="Frequência observada na amostra",
        main="Frequência de tabagismo por sexo")
```

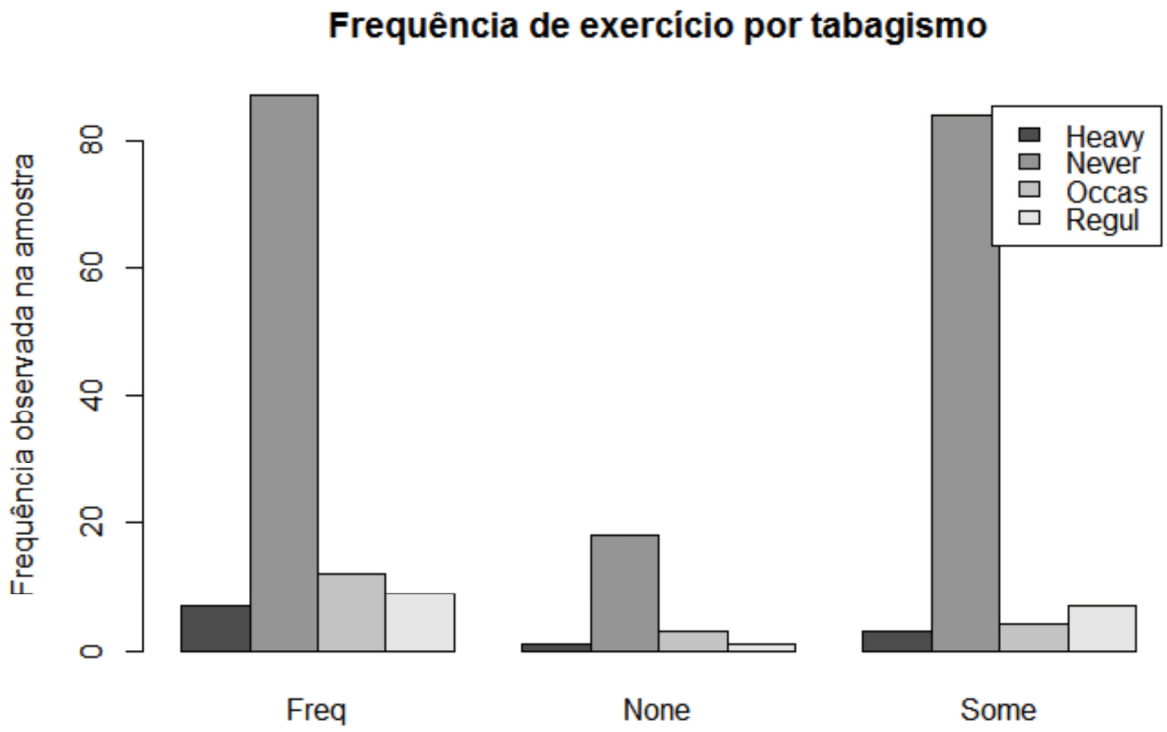


Tabagismo:Heavy, Never, Occas, Regul

```
>barplot(t(Exer_Sex), beside=T, legend=T,
        ylab="Frequência observada na amostra",
        main="Frequência de exercício por sexo")
```



Exercício: Freq, None, e Some



Ecercício: Freq, None, e Some
Tabagismo:Heavy, Never, Occas, Regul