Aula 5 - Manipulação de dados, análise exploratória

Vitor Rios

11 de novembro de 2017

#Função subset() Divide um objeto em subconjuntos

`x` = objeto a ser dividido `subset` = condição lógica indicando o que deve ser mantido `select`= indica as colunas a serem selecioandas em um dataframe

#Familia apply() Existem várias funções no R que aplicam funções sobre objetos, e elas variam principalmente no tipo de objeto que devolvem

##Função apply()

Aplica uma funçao a todas as linhas ou todas as colunas de um array (dataframe numérico ou matriz), retorna um vetor ou matriz *ATENÇÃO* : só funciona para dados numéricos, não serve para fatores `apply(X, MARGIN, FUN, ...)` `x` é array que ser quer agrupar `MARGIN` 1 = linhas, 2 = colunas, c(1,2), ambos `FUN` é a função que se quer aplicar nas margens, sem parênteses `...` são os argumentos que serão passados para a função `FUN` Por exemplo, se queremos a soma de cada coluna

#Função lapply() Aplica uma função a cada elemento de uma lista, e retorna uma lista. Pode lidar com qualquer tipo de dado, a depender de `FUN` `lapply(X, FUN, ...)` `X` um lista ou objeto que será convertido em lista (pode ser data.frame ou matriz) `FUN` é a função que se quer aplicar nos elementos de, sem parênteses `...` são os argumentos que serão passados para a função `FUN`

```
## $dragoes_completo.peso
## [1] 10377.68
##
## $dragoes_completo.n_chifres
## [1] 560
##
## $dragoes_completo.tamanho_asa
## [1] 890.0223
##
## $dragoes_completo.idade
## [1] 9069.259
```

```
## $dragoes_completo.peso
##  [1] 4.801752 4.642442 4.868722 4.587953 4.638930 4.629375 4.832305
```

#Função tapply() `tapply(X, INDEX, FUN, ...)` Aplica uma função a subsets do objeto `X` = um objeto, tipicamento um vetor `INDEX` uma lista de fatores, com comprimento igual a `X`, usado para criar subconjuntos nos quais `FUN` será aplicada `FUN` é a função que se quer aplicar nos elementos de `X`, sem parênteses `...` são os argumentos que serão passados para a função `FUN`

```
##  a  b  c  d  e
## 10 26 42 58 74
```

```
## [1] "array"
```

Para facilitar:

- apply: genérica: aplica uma função a linhas ou colunas de uma matriz (ou às dimensões de um array), retorna vetor ou matriz
- lapply: “list apply”. Age em uma lista ou vetor e retorna uma lista
- sapply: “simple lapply”. Igual a lapply, mas retorna um vetor ou matriz sempre que possível
- tapply: “tagged apply”. subconjuntos (tags) identificam os grupos nos quais a função será aplicada. Tipo de retorno depende da função, geralmente array
- aggregate: tapply que converte o resultado para dataframe

Na maioria dos casos você vai usar aggregate ou tapply

Análise exploratória: verificando seus dados

Antes de qualquer análise estatística, é necessário verificar a distribuição dos dados, se há outliers, se a distribuição é normal, assimétrica, se há dados ausentes, erros de digitação, se as premissas dos testes são cumpridas, se é necessário transformar os dados, etc...

Antes de tudo, verifique a estrutura dos dados, NAs e erros de digitação

```
## 'data.frame':    80 obs. of  7 variables:
##  $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dieta      : Factor w/ 4 levels "aventureiros",..: 3 3 3 3 3 3 3 3
##  $ peso       : num  121.7 103.8 130.2 98.3 103.4 ...
##  $ n_chifres  : int  9 8 8 8 4 8 7 12 4 3 ...
##  $ cor        : Factor w/ 10 levels "azul","banco",..: 3 8 6 9 3 9 1 3 9
##  $ tamanho_asa: num  15.95 4.31 10.23 13.3 7.12 ...
##  $ idade      : num  140.84 7.64 66.4 149.28 85.02 ...
```

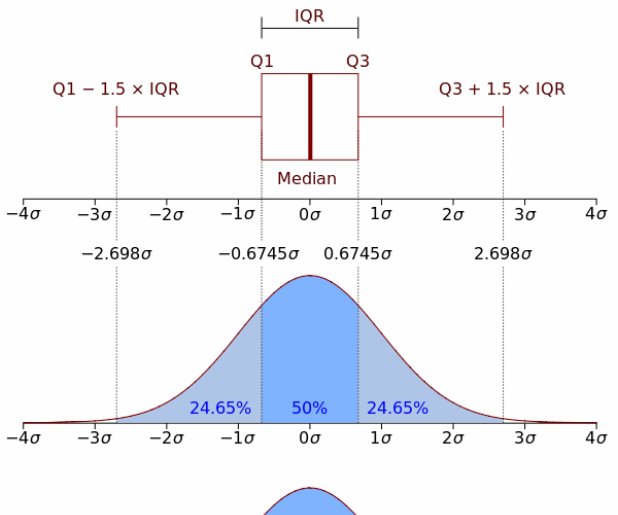
```
##   X dieta      peso n_chifres      cor tamanho_asa      idade
## 1 1 vacas 121.72355         9   branco   15.950930 140.837112
## 2 2 vacas 103.79754         8    verde    4.305912   7.640123
```

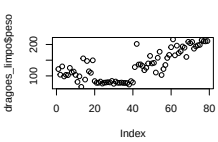
Plote seus dados

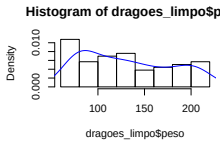
summary() para estatísticas básicas

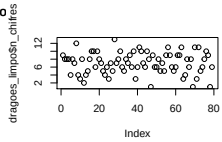
```
##        X                  dieta         peso           n_chifres
##  Min.   : 1.00   aventureiros:20   Min.   : 65.10   Min.   : 1.000
##  1st Qu.:20.50   fazendeiros :20   1st Qu.: 82.74   1st Qu.: 5.000
##  Median :40.00   vacas       :20   Median :125.50   Median : 7.000
##  Mean   :40.37   virgens     :19   Mean   :131.36   Mean   : 6.975
##  3rd Qu.:60.50                     3rd Qu.:169.68   3rd Qu.: 9.000
##  Max.   :80.00                     Max.   :216.17   Max.   :13.000
##      cor              tamanho_asa         idade
##  Length:79          Min.   : 3.111   Min.   :  3.421
##  Class :character   1st Qu.: 9.385   1st Qu.: 77.825
##  Mode  :character   Median :11.059   Median :114.258
##                     Mean   :11.103   Mean   :113.240
##                     3rd Qu.:13.435   3rd Qu.:143.541
##                     Max.   :17.244   Max.   :217.578
```

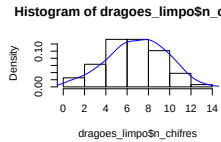
Essas estatísticas são suficientes? Para cada coluna, summary nos dá - Min. : valor mínimo dos dados - 1st Qu. : primeiro quartil - Median : mediana - Mean : média - 3rd Qu. : terceiro quartil - Max. : valor máximo dos dados - NA's : quantidade de NAs nos dados

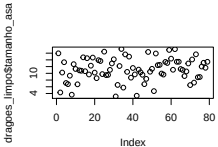


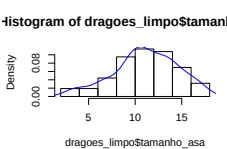

dragoes_limpo$peso
100
200
0
20
40
60
80
Index

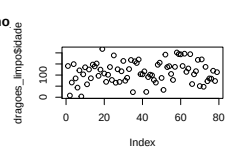

Histogram of dragoes_limpo$peso
Density
0.000
0.010
100
150
200
dragoes_limpo$peso

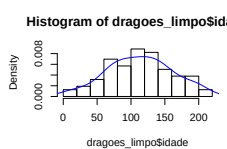

dragoes_limpo$n_chifres
2
6
12
0
20
40
60
80
Index


Histogram of dragoes_limpo$n_chifr
Density
0.00
0.10
0
2
4
6
8
10
12
14
dragoes_limpo$n_chifres


dragoes_limpo$tamanho_asa
4
10
0
20
40
60
80
Index


Histogram of dragoes_limpo$tamanho
Density
0.00
0.08
5
10
15
dragoes_limpo$tamanho_asa


dragoes_limpo$idade
0
100
0
20
40
60
80
Index


Histogram of dragoes_limpo$idade
Density
0.000
0.008
0
50
100
150
200
dragoes_limpo$idade