Lição 6: Variáveis Numéricas: Medidas de Tendências Centrais

N.B.: Rode as linhas de comando a seguir antes de iniciar esta lição.

```
notas_turmaA <- c(7.0, 10.0, 10.0, 0.5, 10.0, 8.2, 9.5, 8.1, 5.0, 8.9,
                  8.2, 7.0, 1.5, 5.5, 9.3, 9.3, 9.3, 1.5, 7.0, 9.5, 6.
0,
                  7.5, 9.9, 8.0, 8.1, 8.8, 2.1, 7.0, 9.0, 0.0, 7.2)
notas_turmaB <- c(6.5, 8.5, 9.4, 7.5, 9.3, 9.9, 9.5, 9.8, 0.0, 0.0)

Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
  }
```

Nas duas últimas lições, lidamos com variáveis nominais/categóricas, e as medidas estatísticas apropriadas para esse tipo de dado: frequências e proporções. Também vimos dois tipos de gráfico bastante usados para representar visualmente os dados de variáveis nominais – o gráfico de barras e o gráfico de linhas (este último, para variáveis ordinais).

Nesta e na próxima lição, vamos tratar de variáveis *numéricas*. Primeiro, vamos revisar alguns conceitos básicos do Ensino Médio: *medidas de tendência central* e *medidas de dispersão*. Você certamente se lembra de uma das medidas de tendência central – a média – e se lembra como calculá-la.

Para refrescá-la na memória, tomemos dois conjuntos de dados, que deixei disponíveis para você nesta sessão: `notas_turmaA` e `notas_turmaB`. Cheque a aba Environment. Imagine que essas são as notas dadas por um professor a duas turmas diferentes ao final do semestre. Primeiro inspecione as notas da turma A. Digite `notas_turmaA` e veja o resultado.

```
notas_turmaA
```

```
##  [1]  7.0 10.0 10.0  0.5 10.0  8.2  9.5  8.1  5.0  8.9  8.2  7.0  1
.5
## [14]  5.5  9.3  9.3  9.3  1.5  7.0  9.5  6.0  7.5  9.9  8.0  8.1  8
.8
## [27]  2.1  7.0  9.0  0.0  7.2
```

Agora inspecione as notas da turma B.

```
notas_turmaB
```

```
##  [1] 6.5 8.5 9.4 7.5 9.3 9.9 9.5 9.8 0.0 0.0
```

Você deve ter notado que a turma A tem mais alunos do que a turma B. Vamos agora calcular a média de cada turma, para verificar qual teve uma nota média mais alta. Você se lembra, do Ensino Médio, que a média é o resultado da soma de todos os elementos dividida pelo número de elementos do conjunto. No R, uma função para realizar operações de adição é `sum()` e para mostrar o número de elementos de um vetor é `length()`. O operador de divisão é /. De posse dessas informações, calcule a média das notas dos alunos da turma A.

```
sum(notas_turmaA) / length(notas_turmaA)
```

```
## [1] 7.06129
```

Agora calcule a média para as notas da turma B.

```
sum(notas_turmaB) / length(notas_turmaB)
```

```
## [1] 7.04
```

Você pode ter imaginado que, para uma medida tão comum como a média, o R tem um jeito mais fácil de chegar a esse resultado – e você tem razão! A função `mean()` faz justamente o que foi feito acima. Aplique-a às notas da turma A para verificar o mesmo resultado.

```
mean(notas_turmaA)
```

```
## [1] 7.06129
```

E aplique a função `mean()` às notas da turma B.

```
mean(notas_turmaB)
```

```
## [1] 7.04
```

Vemos que as médias das duas turmas são bem parecidas. Vejamos agora outra medida de tendência central: a mediana. Dessa talvez você não se lembre, já que não é de uso tão corrente no cotidiano. Se um conjunto de medições for colocado na ordem crescente, a mediana é a observação bem no ponto médio desse conjunto ordenado. Quando um conjunto tem um número ímpar de observações, como é o caso de

`notas_turmaA`, a mediana é o valor de n/2 (ou seja, 31/2 = 15,5), arredondado para cima (=16). A mediana será, então, o 16º valor do vetor colocado em ordem crescente. Colocado de outro modo, a mediana é o valor que separa a metade inferior da metade superior da amostra (15 observações para cada lado).

Coloque os elementos do vetor `notas_turmaA` com uso da função `sort()`. Verifique qual é o valor do 16º elemento para achar a mediana.

```
sort(notas_turmaA)
```

```
##  [1]  0.0  0.5  1.5  1.5  2.1  5.0  5.5  6.0  7.0  7.0  7.0  7.0  7
.2
## [14]  7.5  8.0  8.1  8.1  8.2  8.2  8.8  8.9  9.0  9.3  9.3  9.3  9
.5
## [27]  9.5  9.9 10.0 10.0 10.0
```

Você deve ter encontrado o valor 8,1, certo? Vamos fazer o mesmo agora para `notas_turmaB`. Do mesmo modo que acima, primeiro se colocam os termos em ordem crescente. Neste caso, em que o vetor tem um número par de observações, tomamos dois números para calcular a mediana: n/2 arredondado para baixo e n/2 arredondado para cima. Como `notas_turmaB` tem 10 elementos, a mediana é a média da 5ª e da 6ª observação dos elementos organizados em ordem crescente. Aplique então a função `sort()` para descobrir quais são a 5ª e a 6ª medição de `notas_turmaB`.

```
sort(notas_turmaB)
```

```
##  [1] 0.0 0.0 6.5 7.5 8.5 9.3 9.4 9.5 9.8 9.9
```

Você deve ter encontrado os valores 8,5 e 9,3. A média entre esses dois valores é 8,9, que é a mediana de `notas_turmaB`. Mas você deve estar pensando: "Por que essa complicação toda? Eu vou ter que ficar colocando os números na ordem crescente e procurando no meio da distribuição qual é a observação n/2 arredondada pra cima ou pra baixo? Eu sei que o R tem um jeito mais fácil de calcular a mediana!" E é claro que tem! É a função `median()`. Aplique-a agora às notas da turma A.

```
median(notas_turmaA)
```

```
## [1] 8.1
```

Aplique a função `median()` às notas da turma B.

```
median(notas_turmaB)
```

```
## [1] 8.9
```

Mesmos valores que calculamos previamente, certo? O motivo de eu não ter ido direto ao ponto é porque é fácil aplicar ferramentas computacionais sem saber direito o que se está fazendo. Média e mediana são duas medidas de cálculo fácil, que muitas vezes podem ser feitas à mão ou com uma calculadora, desde que o número de observações não seja muito extenso. Mas por mais banais que pareçam ser essas medidas, *todos* os testes estatísticos que existem derivam em menor ou maior grau delas. É importante ter clareza sobre como essas medidas são calculadas.

Há ainda uma terceira medida de tendência central, chamada moda. A moda é o valor que ocorre mais frequentemente em um conjunto de dados. Interessantemente, o R não tem uma função própria para calcular a moda de um conjunto de dados, mas eu deixei uma tal função disponível para você nesta lição. Ela se chama Mode (com 'M' maiúsculo). Aplique-a às notas dos alunos da turma A.

```
Mode(notas_turmaA)
## [1] 7
```

E agora aplique-a às notas dos alunos da turma B.

```
Mode(notas_turmaB)
## [1] 0
```

Ok! Temos então três medidas para cada conjunto de dados, resumidas na Tabela 6.1.

Tabela 6.1: Medidas de tendência central das turmas A e B.

	Turma A	Turma B
Média	7,06	7,04
Mediana	8,1	8,9
Moda	7,0	0,0

Fonte: própria.

Turma A: 7,06, 8,1 e 7,0 e Turma B: 7,04, 8,9 e 0,0 para média, mediana e moda respectivamente. Embora a média de ambas as turmas tenha sido praticamente a mesma,

há diferenças nas medidas de mediana e moda. Você pode estar se perguntando agora: "Qual é a melhor e qual eu devo usar?"

Pois bem, retire essas perguntas imediatamente da cabeça! A questão aqui não é a melhor ou a pior, mas sim o que cada uma informa. Vale a pena fazer as três medições em todo conjunto de dados numéricos que você tiver. Quando essas três medidas são parecidas entre si, isso significa que você está diante de um conjunto equilibrado de dados, em que todos os valores se distribuem de modo mais ou menos simétrico em torno do ponto médio.

Graficamente, tal conjunto de dados se distribuiria mais ou menos como a imagem do meio da Figura 6.1.

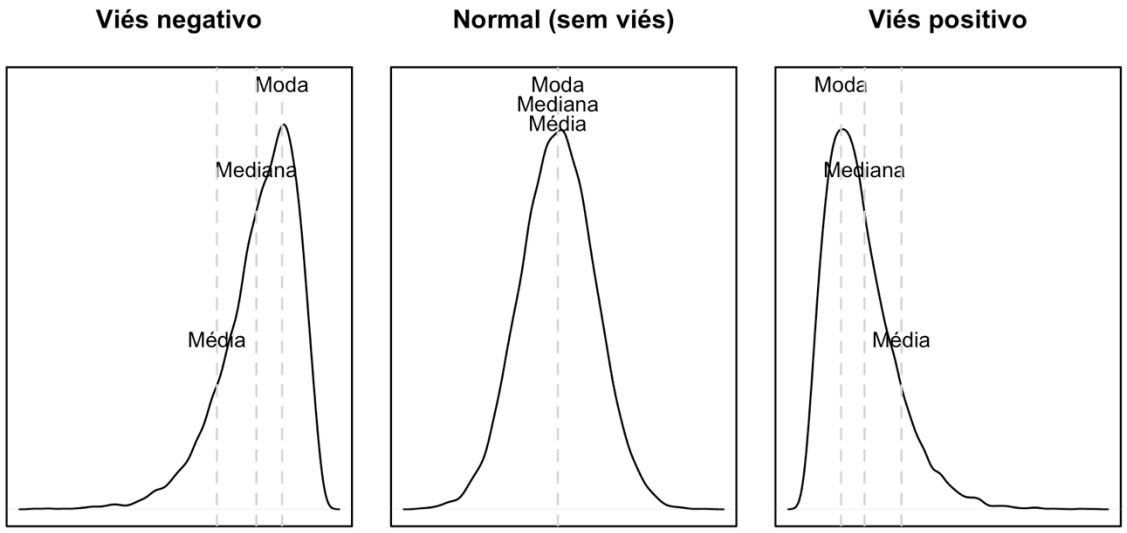

Figura 6.1: Distribuições. Fonte: própria.

Quando as três medidas diferem mais entre si, isso é indicativo de que a distribuição de seus dados é assimétrica, ou seja, que ela provavelmente tem um viés negativo ou positivo. A mediana tanto da turma A quanto da turma B está acima de 8,0, o que significa que pelo menos metade dos alunos de cada turma tirou uma nota acima de 8,0, mas a média está "mascarando" isso – o viés, aqui, é negativo. Na situação contrária – a mediana abaixo da média –, o viés seria positivo. Esses vieses estão ilustrados na Figura 6.1.

Algo semelhante tende a ocorrer com a moda. Quando uma distribuição é simétrica, o valor mais frequente (o ponto mais alto da distribuição) tende a estar próximo da média. Isso parece ser mais verdadeiro para a turma A do que para a turma B, em que a moda difere radicalmente dos demais valores de medidas centrais.

Também há que se levar em conta o número de elementos de um conjunto para avaliar o quanto essas três medidas são representativas dele. Na turma B, há apenas 10 alunos, 3 vezes menos do que na turma A. Isso significa que cada aluno da turma B contribui mais para a média do que cada aluno da turma A.

Façamos um teste: acrescente a cada um dos vetores notas_turmaA e notas_turmaB uma nova nota 0. Digite `notas_turmaA2 <- c(notas_turmaA, 0)`. (Certifique-se que você entende o que está sendo feito nesta linha de comando; se não, reveja a Lição 1.)

```
notas_turmaA2 <- c(notas_turmaA, 0)
```

Vamos fazer o mesmo agora para a turma B. Digite `notas_turmaB2 <- c(notas_turmaB, 0)`.

```
notas_turmaB2 <- c(notas_turmaB, 0)
```

Aplique agora a função `mean()` a notas_turmaA2.

```
mean(notas_turmaA2)
## [1] 6.840625
```

E aplique a função `mean()` ao vetor notas_turmaB2.

```
mean(notas_turmaB2)
## [1] 6.4
```

Veja que a média da turma A baixou de 7,06 para 6,84 (=0,22 de diferença), enquanto a média da turma B baixou de 7,04 para 6,4 (=0,64). A média do segundo grupo baixou quase três vezes mais do que a do primeiro! Isso é porque uma única observação "conta mais" num grupo menor de dados do que num grupo maior e, portanto, tem mais potencial para mudar os resultados. A lição a se tirar daqui é que quanto menor o número de dados de que se dispõe, mais cautela se deve ter na leitura de resultados dos testes estatísticos!

Média, mediana e moda resumem, cada qual, um conjunto de dados em um único número. Mas interessa também saber como os dados se dispersam. Afinal, um conjunto A (5, 5, 5) e outro B (0, 5, 10) têm ambos a mesma média e a mesma mediana (=5), mas são bem diferentes. Para capturar a diferença entre um tal conjunto A e um tal conjunto B, calculamos *medidas de dispersão*.

A *variância* é uma das medidas de dispersão. Seu cálculo é também bastante simples e está ilustrado na Tabela 6.2 (10 primeiros valores de `notas.turmaA`):

Tabela 6.2: Cálculo da variância.

1: Obs	**2**: Obs - μ	**3**: (Obs - μ)^2
7,0	-0,06	0,0036
10,0	2,94	8,6436
10,0	2,94	8,6436
0,5	-6,56	43,0346
10,0	2,94	8,6436
8,2	1,14	1,2996
9,5	2,44	5,9536
8,1	1,04	1,0816
5,0	-2,06	4,2436
8,9	1,84	3,3856
...	...	...
		4: Σ = 264,1136
		5: Σ / 30 = 8,8038

Fonte: própria.

Toma-se cada uma das observações (1), e dela se subtrai o valor da média (2). No vetor notas_turmaA, isso seria: 7 – 7,06; 10 – 7,06; 10 – 7,06 etc. Em seguida, eleva-se cada resultado das subtrações ao quadrado (3). Os quadrados servem o propósito de eliminar o sinal negativo – como você se lembra do Ensino Médio, qualquer número elevado ao quadrado é positivo. Somam-se então os valores quadrados (4). O resultado da soma é dividido pelo número de observações N (se o cálculo for da *população*) ou por n - 1 (se o cálculo for da *amostra*) (5). Não vou entrar em detalhes do porquê disso aqui.

Os curiosos podem consultar este site: http://duramecho.com/Misc/WhyMinusOneInSd.html.

No R, a função para calcular a variância é `var()`. Aplique-a às notas da turma A.

```
var(notas_turmaA)
## [1] 8.803785
```

Aplique agora a função `var()` às notas da turma B.

```
var(notas_turmaB)
## [1] 14.92044
```

Vemos que a variância é maior na turma B. Isso significa que a perfórmance dos estudantes da turma B foi menos homogênea, mais dispersa, do que os da turma A. Mas esses números são um pouco difíceis de interpretar. O que significam 8,8 ou 14,9 de variância? Uma medida mais "intuitiva" é o desvio padrão.

O desvio padrão é a raiz quadrada da variância. Lembra que elevamos todos os valores de diferença entre uma observação e a média ao quadrado? A raiz quadrada reverte essa operação. No R, a função para calcular o desvio padrão é `sd()` (=standard deviation). Aplique-a agora ao vetor `notas_turmaA`.

```
sd(notas_turmaA)
## [1] 2.967117
```

E aplique `sd()` ao vetor `notas_turmaB`.

```
sd(notas_turmaB)
## [1] 3.862699
```

O desvio padrão da turma A é 2,96 e da turma B é 3,86. Esses valores já são mais facilmente interpretáveis: em média, os alunos da turma A desviaram 2,96 pontos da média, e os alunos da turma B desviaram mais. Novamente, a interpretação é que a perfórmance dos alunos da turma A foi relativamente mais homogênea do que a dos alunos da turma B.

Outra medida estatística útil, e que aparecerá nos modelos de regressão linear (Lições 12 e 13), é o erro padrão. Não há uma função específica no R para computá-lo, mas sua definição matemática é bastante simples: o erro padrão é igual ao desvio padrão,

dividido pela raiz quadrada do número de observações. Aplique então essa fórmula a notas_turmaA. (A função para calcular a raiz quadrada é `sqrt()` e para computar o número de observações é `length()`).

```
sd(notas_turmaA) / sqrt(length(notas_turmaA))
## [1] 0.53291
```

E faça o cálculo do erro padrão para notas_turmaB.

```
sd(notas_turmaB) / sqrt(length(notas_turmaB))
## [1] 1.221493
```

Mas para que servem todas essas medidas? Um dos objetivos da análise estatística é criar modelos ou fazer previsões (vamos falar mais sobre isso adiante no curso). As medidas de tendência central resumem em poucos números uma distribuição que pode ter dezenas, centenas, milhares, milhões de dados. As medidas de dispersão dão um indicativo do quanto as medidas de tendência central conseguem prever uma determinada medição. Quanto maior a variância, desvio padrão e erro padrão, menos informativas são as medidas de tendências centrais. A previsão de uma medida dificilmente se refere a um valor exato. Na maior parte das vezes, tais medidas vão ser base para estimar *probabilidades*.

Que tal aplicar esse conhecimento em algo *muito mais importante*, como a análise da altura das vogais médias pretônicas na fala de migrantes nordestinos? Primeiro, defina como diretório de trabalho aquele que, em seu computador, contém o arquivo Pretonicas.csv.

```
setwd("~/Dropbox/_R/swirl/Introducao_a_Estatistica_para_Linguistas/data")
```

N.B.: O diretório em seu computador provavelmente vai ser diferente!

Vamos carregar a planilha em um dataframe chamado `pretonicas`. Para tanto, carregue o pacote tidyverse.

```
library(tidyverse)
```

Use a função `read_csv()` para carregar a planilha. Nela, defina as variáveis AMOSTRA e VOGAL como `col_factor()`; a primeira tem os níveis "PBSP" e "SP2010", e a segunda tem os níveis "i", "e", "a", "o" e "u".

```
pretonicas <- read_csv("Pretonicas.csv",
                       col_types = cols(AMOSTRA = col_factor(levels =
c("PBSP", "SP2010")),
                                        VOGAL = col_factor(levels = c(
"i", "e", "a", "o", "u"))
                                        )
                       )
```

E você também já sabe que a primeira coisa a se fazer após carregar os dados é inspecioná-los. Aplique `str()` ao dataframe `pretonicas` para se certificar de que os dados foram carregados corretamente e ter uma visão global deles.

```
str(pretonicas)
```

```
## spec_tbl_df [2,415 × 27] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PALAVRA      : chr [1:2415] "fazer" "quatorze" "casou" "casado"
...
##  $ Transc.Fon   : chr [1:2415] "f<a>-'zer" "k<a>-'tor-ze" "k<a>-'zo
w" "k<a>-'za-do" ...
##  $ VOGAL        : Factor w/ 5 levels "i","e","a","o",..: 3 3 3 3 3
3 4 3 3 3 ...
##  $ F1           : num [1:2415] 487 686 731 621 845 ...
##  $ F2           : num [1:2415] 1666 1414 1168 1275 1574 ...
##  $ F1.NORM      : num [1:2415] 397 476 494 450 540 ...
##  $ F2.NORM      : num [1:2415] 1517 1386 1258 1314 1469 ...
##  $ CONT.PREC    : chr [1:2415] "f" "k" "k" "k" ...
##  $ CONT.SEG     : chr [1:2415] "z" "t" "z" "z" ...
##  $ VOGAL.SIL.SEG: chr [1:2415] "e" "o" "ow" "a" ...
##  $ F1.SIL.SEG   : num [1:2415] 498 462 529 842 509 ...
##  $ F2.SIL.SEG   : num [1:2415] 2001 1126 1009 1239 2351 ...
##  $ F1.SEG.NORM  : num [1:2415] 328 317 338 433 331 ...
##  $ F2.SEG.NORM  : num [1:2415] 1518 1095 1038 1149 1687 ...
##  $ VOGAL.TONICA : chr [1:2415] "e" "o" "ow" "a" ...
##  $ DIST.TONICA  : num [1:2415] 1 1 1 1 1 1 1 1 1 1 ...
##  $ ESTR.SIL.PRET: chr [1:2415] "CV" "CV" "CV" "CV" ...
##  $ Begin.Time.s : num [1:2415] 20.4 20.6 33.6 36.5 40.3 ...
##  $ End.Time.s   : num [1:2415] 20.4 20.6 33.6 36.5 40.4 ...
##  $ Duration.ms  : num [1:2415] 19.1 20.2 40.7 25.2 34.7 ...
##  $ AMOSTRA      : Factor w/ 2 levels "PBSP","SP2010": 1 1 1 1 1 1 1
1 1 1 ...
##  $ PARTICIPANTE : chr [1:2415] "MartaS" "MartaS" "MartaS" "MartaS"
...
##  $ SEXO         : chr [1:2415] "feminino" "feminino" "feminino" "fe
minino" ...
##  $ IDADE        : num [1:2415] 32 32 32 32 32 32 32 32 32 32 ...
##  $ IDADE.CHEGADA: num [1:2415] 18 18 18 18 18 18 18 18 18 18 ...
##  $ ANOS.SP      : num [1:2415] 14 14 14 14 14 14 14 14 14 14 ...
```

```
##  $ CONTEXTO     : chr [1:2415] “ai aqui j\u0087 tem treze ano vai f
azer quatorze” “ai aqui j\u0087 tem treze ano vai fazer quatorze” “a\u
0092 depois ele voltou a gente casou e viemos” “que l\u0087 voc\u0090
s\u0097 podia sair se fosse casado n\u008e se fosse pra” ...
##  - attr(*, “spec”)=
##   .. cols(
##   ..   PALAVRA = col_character(),
##   ..   Transc.Fon = col_character(),
##   ..   VOGAL = col_factor(levels = c(“i”, “e”, “a”, “o”, “u”), orde
red = FALSE, include_na = FALSE),
##   ..   F1 = col_double(),
##   ..   F2 = col_double(),
##   ..   F1.NORM = col_double(),
##   ..   F2.NORM = col_double(),
##   ..   CONT.PREC = col_character(),
##   ..   CONT.SEG = col_character(),
##   ..   VOGAL.SIL.SEG = col_character(),
##   ..   F1.SIL.SEG = col_double(),
##   ..   F2.SIL.SEG = col_double(),
##   ..   F1.SEG.NORM = col_double(),
##   ..   F2.SEG.NORM = col_double(),
##   ..   VOGAL.TONICA = col_character(),
##   ..   DIST.TONICA = col_double(),
##   ..   ESTR.SIL.PRET = col_character(),
##   ..   Begin.Time.s = col_double(),
##   ..   End.Time.s = col_double(),
##   ..   Duration.ms = col_double(),
##   ..   AMOSTRA = col_factor(levels = c(“PBSP”, “SP2010”), ordered =
FALSE, include_na = FALSE),
##   ..   PARTICIPANTE = col_character(),
##   ..   SEXO = col_character(),
##   ..   IDADE = col_double(),
##   ..   IDADE.CHEGADA = col_double(),
##   ..   ANOS.SP = col_double(),
##   ..   CONTEXTO = col_character()
##   .. )
##  - attr(*, “problems”)=<externalptr>
```

Na Lição 2, também vimos a função `summary()`, que fornece uma visão global dos dados com medidas estatísticas relevantes. Aplique-a agora ao objeto `pretonicas`.

```
summary(pretonicas)
```

```
##    PALAVRA           Transc.Fon        VOGAL         F1
##  Length:2415        Length:2415        i:409   Min.   : 122.5
##  Class :character   Class :character   e:686   1st Qu.: 426.4
##  Mode  :character   Mode  :character   a:415   Median : 510.4
##                                        o:639   Mean   : 524.6
##                                        u:266   3rd Qu.: 618.7
##                                                Max.   :1095.4
##
##        F2            F1.NORM         F2.NORM        CONT.PREC
##  Min.   : 672.9   Min.   :302.1   Min.   : 946.9   Length:2415
##  1st Qu.:1144.6   1st Qu.:395.3   1st Qu.:1247.4   Class :character
```

```
##  Median :1432.9   Median :425.7   Median :1412.4   Mode  :character
##  Mean   :1477.2   Mean   :427.1   Mean   :1437.8
##  3rd Qu.:1786.3   3rd Qu.:455.0   3rd Qu.:1613.8
##  Max.   :2593.5   Max.   :578.1   Max.   :1994.9
##
##    CONT.SEG         VOGAL.SIL.SEG        F1.SIL.SEG
##  Length:2415        Length:2415        Min.   : 169.6
##  Class :character   Class :character   1st Qu.: 488.9
##  Mode  :character   Mode  :character   Median : 602.9
##                                        Mean   : 611.3
##                                        3rd Qu.: 725.1
##                                        Max.   :2163.0
##
##    F2.SIL.SEG      F1.SEG.NORM     F2.SEG.NORM     VOGAL.TONICA
##  Min.   : 620.8   Min.   :252.1   Min.   : 883.7   Length:2415
##  1st Qu.:1277.5   1st Qu.:341.2   1st Qu.:1213.6   Class :character
##  Median :1478.6   Median :375.6   Median :1293.7   Mode  :character
##  Mean   :1547.0   Mean   :372.3   Mean   :1333.4
##  3rd Qu.:1747.0   3rd Qu.:403.6   3rd Qu.:1442.0
##  Max.   :3182.9   Max.   :648.5   Max.   :1974.5
##
##   DIST.TONICA    ESTR.SIL.PRET       Begin.Time.s
##  Min.   :1.000   Length:2415        Min.   :   7.95
##  1st Qu.:1.000   Class :character   1st Qu.: 378.90
##  Median :1.000   Mode  :character   Median : 938.50
##  Mean   :1.192                      Mean   :1133.67
##  3rd Qu.:1.000                      3rd Qu.:1709.12
##  Max.   :5.000                      Max.   :4735.69
##
##    End.Time.s        Duration.ms       AMOSTRA     PARTICIPANTE
##  Min.   :   7.966   Min.   : 4.211   PBSP  :1171   Length:2415
##  1st Qu.: 378.928   1st Qu.:16.271   SP2010:1244   Class :character
##  Median : 938.521   Median :20.775                 Mode  :character
##  Mean   :1133.695   Mean   :21.896
##  3rd Qu.:1709.143   3rd Qu.:26.661
##  Max.   :4735.718   Max.   :97.504
##
##      SEXO               IDADE       IDADE.CHEGADA      ANOS.SP
##  Length:2415        Min.   :30.00   Min.   :13.00   Min.   :14.00
##  Class :character   1st Qu.:31.00   1st Qu.:14.00   1st Qu.:15.00
##  Mode  :character   Median :35.00   Median :17.00   Median :17.00
##                     Mean   :34.63   Mean   :16.51   Mean   :18.82
##                     3rd Qu.:37.00   3rd Qu.:18.00   3rd Qu.:23.00
##                     Max.   :42.00   Max.   :21.00   Max.   :25.00
##                                     NA's   :1244    NA's   :1244
##    CONTEXTO
##  Length:2415
##  Class :character
##  Mode  :character
##
##
##
##
```

Veja que, para variáveis numéricas, como F1 e F2, a função `summary()` fornece os valores de média e de mediana, além de outras medidas como mínimo, máximo e quartis. Veremos essas outras medidas com mais detalhes na próxima lição. Por ora, voltemos às medidas de tendências centrais.

O arquivo Pretonicas.csv contém medições de F1 e F2 de vogais pretônicas de 7 falantes paraibanos que migraram para a cidade de São Paulo (amostra PBSP). Além disso, o arquivo também contém as mesmas medições para 7 paulistanos nativos (amostra SP2010), que servem como um parâmetro de comparação para os padrões de fala dos migrantes.

A questão mais geral por trás desses dados é descobrir se alguns migrantes paraibanos se acomodaram aos padrões da nova comunidade, e quais padrões são esses. Para isso, foram extraídas medições de F1 e F2 de cerca de 130 a 180 vogais pretônicas da fala de cada um desses participantes, com especial interesse nas vogais médias /e/ e /o/ (como em 'relógio' e 'romã'), em contextos linguísticos que favorecem o abaixamento dessas vogais (as realizações média-baixas [ɛ] e [ɔ]). Em princípio se espera que os paulistanos tenham vogais médias relativamente mais altas do que os paraibanos, mas que alguns dos migrantes tenham se aproximado do novo padrão.

Linguística e acusticamente, a altura das vogais é medida pelo F1, em Hertz. Quanto mais alto é o valor de F1, mais *baixa* é a vogal. Coloquei o quadro de vogais do IPA na Figura 6.2 para facilitar essa visualização.

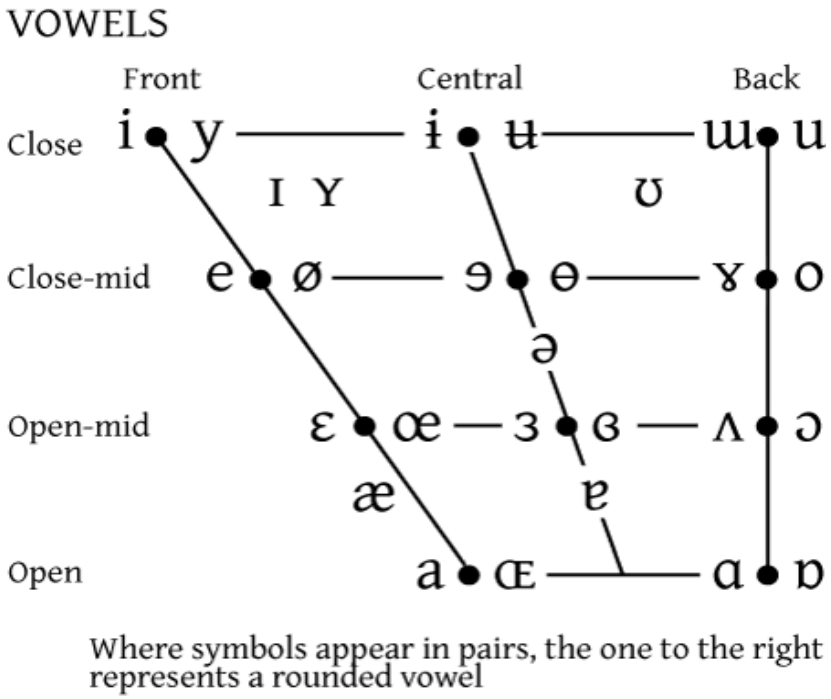

Figura 6.2: Vogais no IPA. Fonte: International Phonetic Association.[9]

No português brasileiro, as vogais [i] e [u] costumam ter valores mais baixos de F1, as vogais [e] e [o] um pouco mais altos, as vogais [ɛ] e [ɔ] mais altos ainda, e a vogal [a] os valores mais altos de F1. (Consulte o manual de Barbosa & Madureira, 2015 para mais informações!). Na página https://en.wikipedia.org/wiki/IPA_vowel_chart_with_audio, você pode visualizar o diagrama de vogais do International Phonetic Alphabet e escutar os sons das vogais.

Da lista de variáveis desse dataframe, qual é a variável dependente?

- CONT.PREC
- F1
- F2
- Palavra
- Transc.Fon

[9] Disponível em https://www.internationalphoneticassociation.org/sites/default/files/IPA_Kiel_2015.pdf.

- Vogal

Uma das questões que queremos responder é: qual é a altura média de cada vogal (principalmente /e/ e /o/) para cada comunidade (PB vs SP)? Considerando-se que há 5 vogais pretônicas e 2 grupos, precisaríamos calcular a média 10 vezes. Além disso, temos interesse, minimamente, na mediana e no desvio padrão, para ter uma ideia de se os dados se distribuem simetricamente ou não, e o grau de sua dispersão. Você ficou cansado só de pensar em fazer tudo isso?

A boa notícia é que, como sempre no R, alguém também já pensou que isso dá muito trabalho e criou uma função para fazer isso mais rapidamente. No tidyverse, isso pode ser feito com as funções `group_by()` – que você já conhece – e `summarize()`, ambas do dplyr.

Veja a linha de comando neste ponto do *script*, que precisará ser completada por você.

```
# Não rodar! Estrutura do código:

df %>%
  group_by(VAR, VAR) %>%
  summarize(nomeVar = mean(VAR))
```

Nela, pedimos para que o R pegue um dataframe e, primeiro, agrupe o dados de acordo com duas variáveis; em seguida, pedimos para que o R compute as médias de outra variável (com a função `mean()`), e guarde o resultado em uma variável chamada nomeVar. Substitua então os termos adequados nos pontos correspondentes: nosso dataframe é `pretonicas`, e queremos agrupar os dados por AMOSTRA e VOGAL; a variável sobre a qual vamos computar as médias é F1; e o nome que vamos atribuir a essa nova variável com as médias é `media_F1`.

```
pretonicas %>%
  group_by(AMOSTRA, VOGAL) %>%
  summarize(media_F1 = mean(F1))

## # A tibble: 10 × 3
## # Groups:   AMOSTRA [2]
##    AMOSTRA VOGAL media_F1
##    <fct>   <fct>    <dbl>
##  1 PBSP    i         410.
##  2 PBSP    e         579.
```

```
##  3 PBSP    a         695.
##  4 PBSP    o         617.
##  5 PBSP    u         446.
##  6 SP2010  i         350.
##  7 SP2010  e         477.
##  8 SP2010  a         660.
##  9 SP2010  o         508.
## 10 SP2010  u         396.
```

O resultado é um tibble, que tem as colunas AMOSTRA, VOGAL e media_F1, e que informa, para cada vogal de cada amostra, qual é a média.

Note que a ordem em que as variáveis são colocadas em `group_by()` determina a ordem de apresentação dos resultados. Para comparar, crie uma nova linha de comando, semelhante à de cima, mas que inverte a ordem das variáveis de agrupamento.

```
pretonicas %>%
  group_by(VOGAL, AMOSTRA) %>%
  summarize(media_F1 = mean(F1))
```

```
## # A tibble: 10 × 3
## # Groups:   VOGAL [5]
##    VOGAL AMOSTRA media_F1
##    <fct> <fct>      <dbl>
##  1 i     PBSP        410.
##  2 i     SP2010      350.
##  3 e     PBSP        579.
##  4 e     SP2010      477.
##  5 a     PBSP        695.
##  6 a     SP2010      660.
##  7 o     PBSP        617.
##  8 o     SP2010      508.
##  9 u     PBSP        446.
## 10 u     SP2010      396.
```

A ordem VOGAL, AMOSTRA torna mais diretamente comparáveis os valores de F1. Vemos que, para todas as vogais, os valores de F1 são maiores (= vogais mais baixas) para os paraibanos do que para os paulistanos.

Dentro da função `summarize()`, podemos já incluir outras medidas estatísticas como novos argumentos. A partir da última linha de comando, mantenha a ordem VOGAL, AMOSTRA para o agrupamento em `summarize()`, e inclua, além do cálculo da média, o cálculo da mediana e do desvio padrão de F1 (nessa ordem), atribuindo a essas novas variáveis os nomes `mediana_F1` e `sd_F1` respectivamente.

```
pretonicas %>%
  group_by(VOGAL, AMOSTRA) %>%
```

```
  summarize(media_F1 = mean(F1),
            mediana_F1 = median(F1),
            sd_F1 = sd(F1)
            )
```

```
## # A tibble: 10 × 5
## # Groups:   VOGAL [5]
##    VOGAL AMOSTRA media_F1 mediana_F1 sd_F1
##    <fct> <fct>      <dbl>      <dbl> <dbl>
##  1 i     PBSP        410.       408.  77.3
##  2 i     SP2010      350.       353.  88.9
##  3 e     PBSP        579.       564. 113.
##  4 e     SP2010      477.       476. 109.
##  5 a     PBSP        695.       663. 127.
##  6 a     SP2010      660.       661. 118.
##  7 o     PBSP        617.       598. 116.
##  8 o     SP2010      508.       509. 114.
##  9 u     PBSP        446.       439.  68.4
## 10 u     SP2010      396.       402. 105.
```

Nesta lição, vimos as medidas de tendência central – média, mediana, moda – e de dispersão – variância, desvio padrão e erro padrão. Em seguida, em um conjunto real de dados, aplicamos a função `summarize()` para rapidamente visualizar algumas dessas medidas por subconjuntos de dados.

O corpus do Projeto SP2010 (http://projetosp2010.fflch.usp.br/) – gravações, transcrições e fichas dos informantes – está todo disponível gratuitamente *on-line*. Convido você a visitar essa página!

Para saber mais

Recomendo a leitura do capítulo 4 de Dalgaard (2008) sobre Estatística Descritiva.

Exercícios

Para estes exercícios, usaremos novamente o conjunto de dados de vogais pretônicas.

1. Carregue o pacote tidyverse.
2. Defina como diretório de trabalho aquele que, em seu computador, contém a planilha Pretonicas.csv.
3. Importe a planilha Pretonicas.csv em um dataframe chamado `pretonicas`. Defina a variável AMOSTRA como factor, com os níveis “PBSP” e “SP2010”; a

variável VOGAL como factor, com os níveis “i”, “e”, “a”, “o” e “u”; e a variável PARTICIPANTE também como factor (sem necessidade de definir a ordem dos níveis).

4. Inspecione o dataframe `pretonicas` com a função `str()`.
5. As variáveis F1.NORM e F2.NORM contêm valores dos formantes normalizados pelo método de Lobanov (1971). A normalização é um procedimento padrão em análises acústicas para minimizar a variação em medições de formantes por conta de diferenças anatômicas entre falantes (p.ex., mulheres tendem a medidas mais altas de formantes do que os homens). Visualize os 6 primeiros elementos da coluna F1.NORM.
6. Calcule a mediana de F1.NORM por VOGAL e por AMOSTRA (nessa ordem). Use o pipe e as funções `group_by()` e `summarize()`, nomeando a coluna para as medianas como mediana_F1.NORM.
7. Calcule a média de F1.NORM por VOGAL e por AMOSTRA (nessa ordem). Nomeie a coluna para as médias como media_F1.NORM.
8. No cálculo da média de F1 com vogais não normalizadas, nesta lição, havíamos visto que as medidas de F1 para paraibanos eram maiores para todas as vogais. Esse resultado se mantém com as vogais normalizadas? Explique sua resposta.
9. No cálculo da média de F1 com vogais normalizadas (F1.NORM), quais vogais têm medidas de F1 maiores (= vogais mais baixas) para paraibanos do que para paulistanos?
    a. /a/ e /o/
    b. /e/ e /o/
    c. /i/ e /e/
    d. /i/ e /o/
    e. /i/ e /u/
10. Se quisermos olhar apenas para as vogais /e/ e /o/, podemos criar um subconjunto de dados com a função `filter()`. Aplique esta função para criar o

subconjunto de dados de vogais médias. Guarde esse subconjunto num dataframe chamado `medias_pretonicas`.

11. Normalmente se verifica bastante variação entre diferentes indivíduos. Um dos interesses em comparar o padrão de paulistanos e migrantes paraibanos é tentar descobrir quais dos migrantes mais se acomodaram ao padrão paulistano (ao menos quanto às vogais médias pretônicas). Para observar o padrão individual, calcule a média de F1.NORM por AMOSTRA e por PARTICIPANTE, inicialmente apenas para a vogal "e". Nomeie a coluna com as médias como media_F1.NORM_e. Use como conjunto de dados o dataframe recém-criado `medias_pretonicas`.
12. Inspecionando o resultado da linha de comando anterior, responda: qual dos paraibanos mais se distancia da média de F1.NORM da vogal /e/ dos paulistanos (= 423 Hz, como calculado mais acima)?
    a. HenriqueA
    b. JoaoS
    c. JosaneV
    d. JosueO
    e. MarinalvaS
    f. MartaS
    g. PedroC
13. Inspecionando o mesmo dataframe, responda: qual dos paraibanos tem uma média de F1.NORM mais próxima da média paulistana para a vogal /e/ (= 423 Hz)?
    a. HenriqueA
    b. JoaoS
    c. JosaneV
    d. JosueO
    e. MarinalvaS
    f. MartaS
    g. PedroC

14. Calcule agora a média de F1.NORM por AMOSTRA e por PARTICIPANTE apenas para a vogal "o". Nomeie a coluna com as médias como media_F1.NORM_o. Use como conjunto de dados o dataframe `medias_pretonicas`.
15. Inspecionando o dataframe gerado pela última linha de comando, responda: qual dos paraibanos tem uma média de F1.NORM mais próxima da média paulistana para a vogal /o/ (= 435 Hz)?
    a. HenriqueA
    b. JoaoS
    c. JosaneV
    d. JosueO
    e. MarinalvaS
    f. PedroC
16. Qual dos paraibanos tem uma média de F1.NORM mais distante da média paulistana para a vogal /o/ (= 435 Hz)?
    a. HenriqueA
    b. JoaoS
    c. JosaneV
    d. JosueO
    e. MarinalvaS
    f. MartaS
    g. PedroC
17. A partir do dataframe `medias_pretonicas`, crie um dataframe chamado `medidas_medias_pretonicas`, com o cálculo da média, da mediana, do desvio padrão e do erro padrão de F1.NORM por VOGAL, AMOSTRA e PARTICIPANTE (nessa ordem). Nomeie as colunas das medições, respectivamente, como media_F1.NORM, mediana_F1.NORM, sd_F1.NORM e ep_F1.NORM.
18. Aplique a função `View()` ao dataframe `medidas_medias_pretonicas`.
19. Acima, aplicamos a função `View()`, pois não é possível visualizar todas as linhas do dataframe por meio do formato tibble. No entanto, o dplyr tem outras

funções que permitem rearranjar os dados de modo a facilitar certas visualizações. A função `arrange()` reorganiza os dados a partir de certas colunas. Para visualizar as medidas de F1.NORM dos paraibanos em ordem crescente, siga os seguintes passos, usando pipe: a partir de `medidas_medias_pretonicas`, filtre os dados da vogal "e" e aplique a função `arrange()` a AMOSTRA e media_F1.NORM.

20. Quem é o paulistano ou a paulistana com vogais /e/ pretônicas relativamente mais baixas?
    a. AliceC
    b. AnaS
    c. LucianoT
    d. MauricioB
    e. NelsonF
    f. RenataC
    g. RobsonF
21. Visualize agora o dataframe na ordem crescente das medidas de desvio padrão da vogal "o", com os dados dos paraibanos antes dos dados dos paulistanos.
22. Quem é o paraibano ou a paraibana com maior dispersão de medidas da altura de vogais /o/ pretônicas?
    a. HenriqueA
    b. JoaoS
    c. JosaneV
    d. JosueO
    e. MartaS
    f. MarinalvaS
    g. PedroC
23. Visualize o dataframe na ordem crescente das medidas de desvio padrão da vogal "e", sem ordenar as amostras.

24. Comparando as medidas de desvio padrão de F1.NORM entre paulistanos e paraibanos, em qual dos grupos há mais dispersão?
    a. PBSP
    b. SP2010