Lição 7: Variáveis Numéricas: Gráficos

Na lição anterior, fizemos algumas tabelas com medidas de médias, medianas e desvio padrão a partir da planilha de dados Pretonicas.csv. Rode as linhas de comando a seguir para carregá-la novamente, se necessário.

```
# Definir diretório de trabalho

#setwd()

# Importar planilha de dados

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"))
                                        )
                       )
```

N.B.: Defina como diretório de trabalho aquele que contém o arquivo Pretonicas.csv.

Você pode ter achado difícil fazer sentido de tantos números – médias, medianas, desvios padrão... E de fato é! A compreensão de muitos dados estatísticos é sempre mais fácil por meio de gráficos – tanto para você, para entender o que está acontecendo em seus dados (gráficos exploratórios), quanto para seu leitor, ao qual futuramente você vai querer comunicar resultados (gráficos explanatórios). Nesta lição, vamos aprender a fazer alguns tipos de gráficos adequados para variáveis numéricas: *gráficos de linhas*, *gráficos de dispersão*, *boxplots* e *histogramas*.

Primeiro, carregue o pacote tidyverse. (Minha ideia é condicionar você a sempre carregar os pacotes necessários no início da sessão! Tá funcionando?)

```
library(tidyverse)
```

Agora, refamiliarize-se com o conjunto de dados de vogais pretônicas. Inspecione a estrutura do dataframe `pretonicas` com `str()`. Os níveis das variáveis VOGAL e AMOSTRA já foram definidos na importação dos dados.

```
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>
```

Cheque a ordem dos níveis das vogais por meio da função `levels()` novamente. Digite `levels(pretonicas$VOGAL)`. Essa é a ordem em que as vogais aparecerão nos gráficos adiante.

```
levels(pretonicas$VOGAL)
```

```
## [1] "i" "e" "a" "o" "u"
```

Esse conjunto de dados contém as medições de F1 e F2 em Hertz, de valores brutos e normalizados (pelo método de Lobanov), para cada vogal pretônica. Nesta lição, trabalharemos com os valores normalizados (F1.NORM e F2.NORM).

Na Lição 5, plotamos um gráfico de linhas para as proporções de realização de /r/ em três lojas de departamento em Nova Iorque. O gráfico de linhas foi ali empregado porque a variável store, naquele conjunto de dados, pode ser considerada uma variável ordinal quanto ao grau de prestígio das lojas. Por outro lado, variáveis numéricas, como são as medidas de F1 e F2, sempre são intrinsicamente ordinais também: tem-se valores que vão de menor para maior.

Reveja o quadro de vogais do IPA (Figura 6.2). Se o imaginamos como um plano cartesiano, cada vogal é identificada por uma coordenada no eixo x e no eixo y. No entanto, há variação no espaço vocálico a depender do indivíduo, do item lexical, da comunidade etc.

Vamos plotar o espaço vocálico das pretônicas na fala de paraibanos residentes em São Paulo (PBSP), em comparação com a dos paulistanos nativos. Para isso, vamos

usar as médias de valores de F1 e F2 normalizados para cada comunidade, de modo a obter as coordenadas para o eixo x (medidas de F2) e eixo y (medidas de F1).

No *script* desta lição, a maior parte dos comandos contém a estrutura dos códigos que vamos usar aqui – cabe a você preenchê-lo com os dados relevantes! Como visto anteriormente, na maior parte do tempo, trabalhamos com *scripts*, sem a necessidade de digitar linhas de comando extensas. Se você conhece a sintaxe e os argumentos das funções, saberá como adaptá-los para suas necessidades.

Voltemos então para nosso gráfico de linhas. Para obter as médias de F1.NORM e F2.NORM para cada VOGAL e para cada AMOSTRA, vamos usar a função `summarize()`, como vimos na última lição. Examine o esqueleto da linha de comando no *script*.

```
# Não rodar! Estrutura do código:

novo.df <- df %>%
  group_by(VAR, VAR) %>%
  summarize(novaVAR1 = mean(VAR),
            novaVAR2 = mean(VAR)) %>%
  print()
```

Primeiro, preencha o nome do novo dataframe em que vamos guardar as medidas das médias: `medias`. Em seguida, explicite o dataframe do qual vamos extrair os dados: `pretonicas`. Vamos agrupar os dados por VOGAL e AMOSTRA. Por fim, vamos computar as medidas de F1.NORM e F2.NORM (nessa ordem) e guardá-las, respectivamente, em colunas chamadas `media_F1` e `media_F2`. Aqui, como estamos criando um novo dataframe com `<-`, usamos a função `print()` para que o R já mostre o resultado. Ao terminar, revise a digitação e rode com CTRL + ENTER.

```
medias <- pretonicas %>%
  group_by(VOGAL, AMOSTRA) %>%
  summarize(media_F1 = mean(F1.NORM),
            media_F2 = mean(F2.NORM)) %>%
  print()

## # A tibble: 10 × 4
## # Groups:   VOGAL [5]
##    VOGAL AMOSTRA media_F1 media_F2
##    <fct> <fct>      <dbl>    <dbl>
##  1 i     PBSP        373.    1688.
##  2 i     SP2010      379.    1717.
##  3 e     PBSP        432.    1612.
##  4 e     SP2010      423.    1606.
```

```
##  5 a     PBSP        475.    1377.
##  6 a     SP2010      488.    1369.
##  7 o     PBSP        445.    1217.
##  8 o     SP2010      435.    1237.
##  9 u     PBSP        386.    1184.
## 10 u     SP2010      395.    1204.
```

Temos agora as médias de F1 e F2 por vogal e por amostra, e podemos prosseguir para a plotagem do gráfico! Vamos plotar, primeiramente, um gráfico de linhas que representa o espaço vocálico de paraibanos em São Paulo e de paulistanos nativos, para que possamos compará-los. Cada ponto será uma das vogais i, e, a, o e u, e elas serão ligadas por uma linha. Já vimos, na Lição 5, que o ggplot2 precisa, pelo menos: (i) do dataframe do qual extrair as informações; (ii) dos parâmetros estéticos; e (iii) da geometria do gráfico a ser plotado – as duas primeiras linhas do comando. Mantenha as demais linhas com o #, por enquanto.

```
# Não rodar! Estrutura do código:

ggplot(df, aes(x = VAR, y = VAR, color = VAR)) +
  geom_line() +
#  geom_label() +
#  scale_x_reverse() +
#  scale_y_reverse() +
#  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
#  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

De qual dataframe vamos extrair os dados das médias de F1 e F2?

- ds
- medias
- pretonicas

Queremos plotar um gráfico do espaço vocálico, em que os eixos x e y vão representar as medidas de F1 e F2. Qual variável do dataframe em questão representa o eixo x?

- media_F1
- media_F2
- F1.NORM
- F2.NORM

Qual variável do dataframe em questão deve ocupar o eixo y?

- media_F1
- media_F2
- F1.NORM
- F2.NORM

No argumento color, queremos definir uma cor diferente para a linha dos paraibanos e outra para a dos paulistanos. Qual variável categoriza o local de origem dos falantes? (Desculpa por essa pergunta tão óbvia...)

- AMOSTRA
- VOGAL
- PARTICIPANTE

Com essas informações, já conseguimos plotar uma primeira versão do gráfico. Preencha o comando com essas informações, substituindo df e as palavras VAR, e rode com CTRL + ENTER. O resultado se encontra na Figura 7.1.

```
ggplot(medias, aes(x = media_F2, y = media_F1, color = AMOSTRA)) +
  geom_line() +
#  geom_label() +
#  scale_x_reverse() +
#  scale_y_reverse() +
#  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
#  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

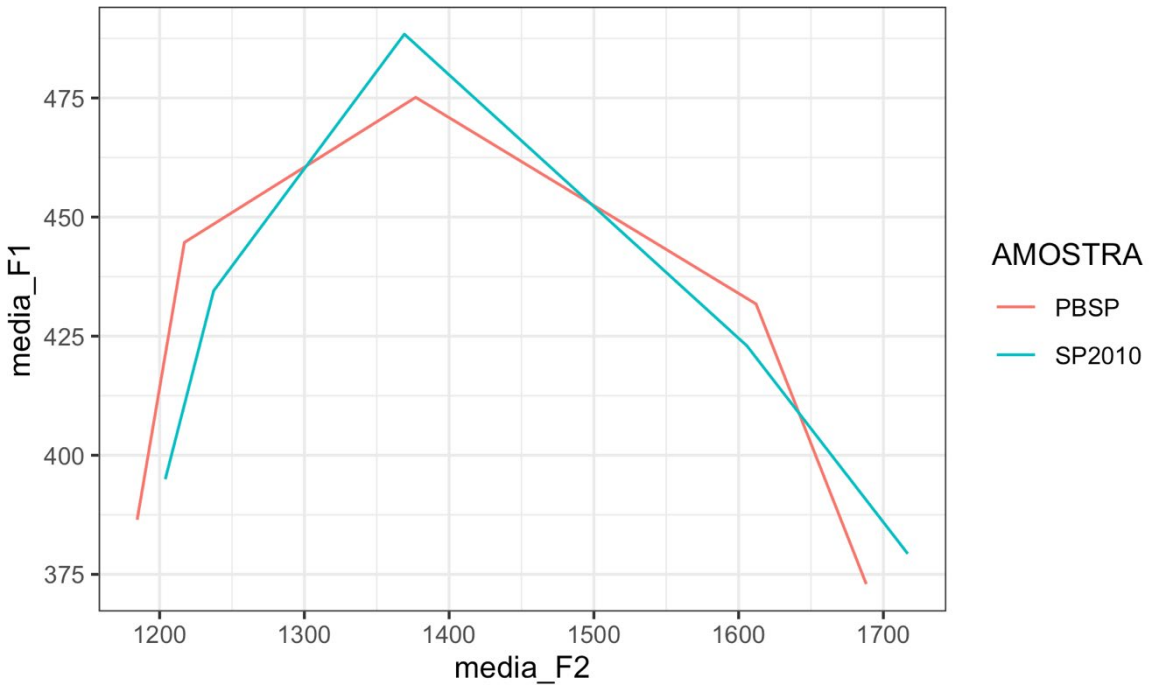

Figura 7.1: Gráfico de linhas com espaço vocálico de paraibanos e paulistanos. Fonte: própria.

Eita! Se você está familiarizado com a representação das vogais no quadro do IPA, deve ter percebido que tem um problema nesse gráfico! Os eixos x e y, ambos de variáveis numéricas/contínuas, estão ordenados de modo crescente: os valores aumentam de baixo para cima e da esquerda para a direita. Mas lembre-se de suas aulas de Fonética: os valores de F1 e de F2 são convencionalmente representados de modo invertido, para deixar as vogais anteriores à esquerda e as vogais fechadas na parte de cima.

Conseguimos consertar isso rapidinho com as funções `scale_x_reverse()` e `scale_y_reverse()`. Retire o comentário # dessas duas linhas e rode o comando novamente (Figura 7.2).

```
ggplot(medias, aes(x = media_F2, y = media_F1, color = AMOSTRA)) +
  geom_line() +
#  geom_label() +
  scale_x_reverse() +
  scale_y_reverse() +
#  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
#  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

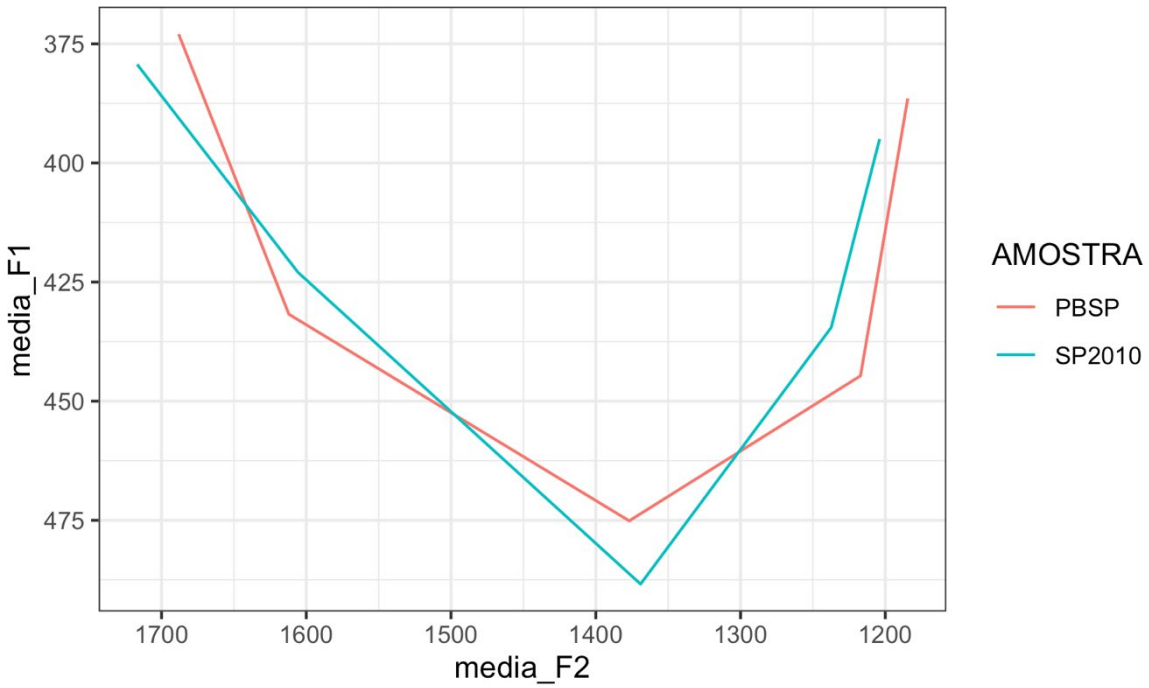

Figura 7.2: Gráfico de linhas com espaço vocálico de paraibanos e paulistanos com eixos x e y revertidos. Fonte: própria.

Bem melhor, não? Um linguista bem treinado já consegue ver esse gráfico e tirar várias informações interessantes. Mas vamos deixá-lo mais informativo, adicionando as vogais para facilitar sua interpretação. Para isso, podemos usar a geometria `geom_label()`, que insere rótulos em gráficos e que neste código está logo abaixo de `geom_line()`. Mas, para que as vogais sejam devidamente mapeadas aos rótulos, precisamos especificar qual variável contém os rótulos, com o argumento label = VOGAL dentro dos parâmetros estéticos `aes()`. Execute então esses dois passos: retire o # da linha com geom_label, e insira o argumento label em `aes()` (Figura 7.3).

```
ggplot(medias, aes(x = media_F2, y = media_F1,
                   color = AMOSTRA, label = VOGAL)) +
  geom_line() +
  geom_label() +
  scale_x_reverse() +
  scale_y_reverse() +
#  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
#  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

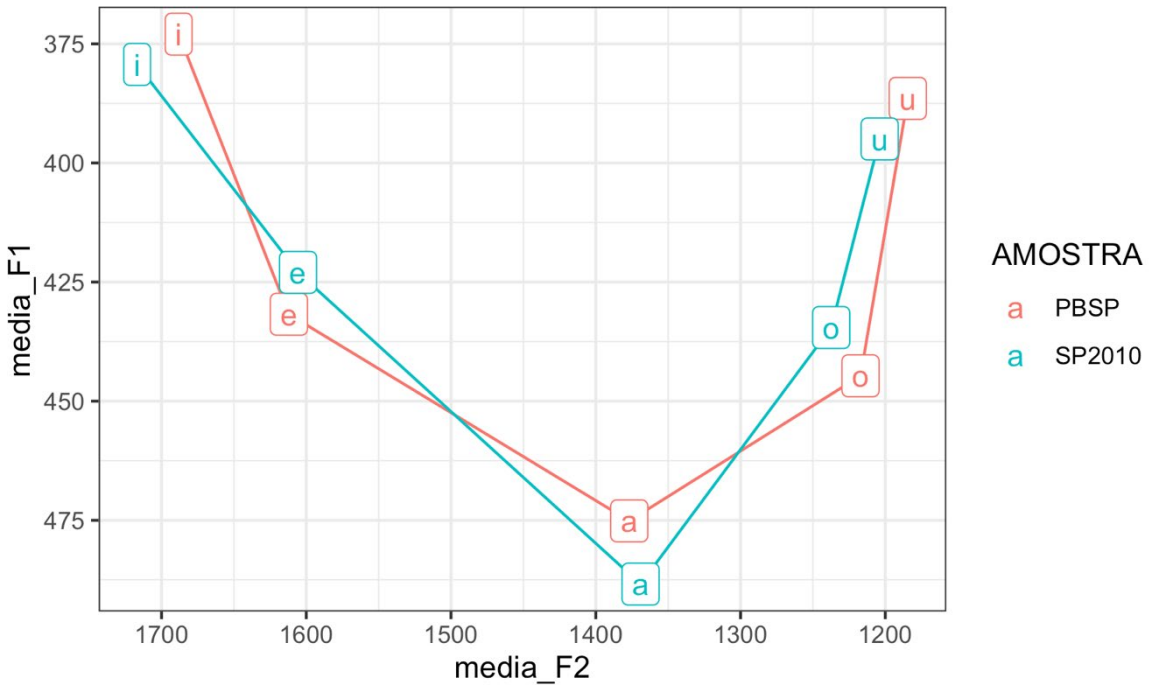

Figura 7.3: Gráfico de linhas com espaço vocálico de paraibanos e paulistanos com eixos x e y revertidos e `geom_label()`. Fonte: própria.

As duas últimas linhas desse comando já são conhecidas por você: a função `ggtitle()` insere um título na figura e a função `labs()` permite definir o nome dos eixos. Já deixei esses textos prontos. Basta descomentar as linhas (apagar o #) e rodá-las (Figura 7.4).

```
ggplot(medias, aes(x = media_F2, y = media_F1, color = AMOSTRA, label
= VOGAL)) +
  geom_line() +
  geom_label() +
  scale_x_reverse() +
  scale_y_reverse() +
  ggtitle("Valores médios de F1 e F2 normalizados nas amostras PBSP e
SP2010") +
  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

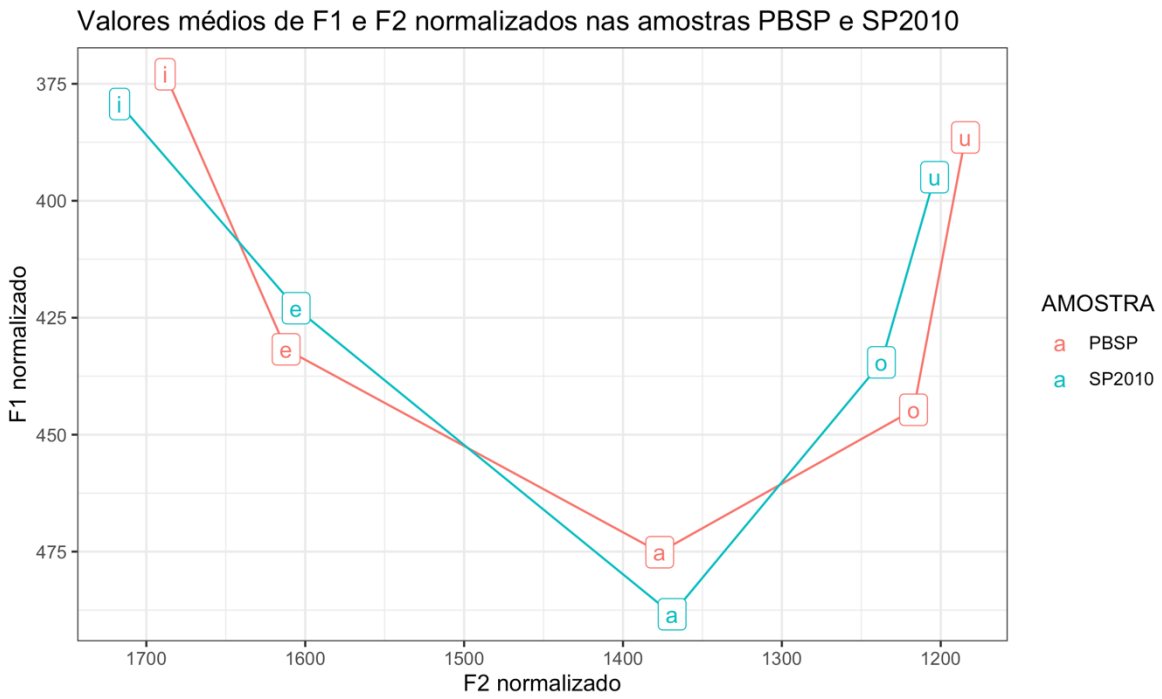

Figura 7.4: Gráfico de linhas com espaço vocálico de paraibanos e paulistanos com eixos x e y revertidos, `geom_label()`, título e rótulos dos eixos. Fonte: própria.

Na última lição, já havíamos visto que as médias de altura (F1) das vogais /e/ e /o/ dos parabainos têm valores mais altos – i.e., tendem a ser realizações mais baixas – do que as vogais dos paulistanos. Essa figura, no entanto, é capaz de informar também a posição relativa quanto ao traço [+- anterior]. Para mais bem visualizá-las, clique sobre Zoom na aba Plots. Caso queira exportar o gráfico, lembre-se das funções `png()` e `dev.off()`, que vimos na Lição 5.

Nesse gráfico que plotamos, as milhares de vogais dos participantes paraibanos e paulistanos foram resumidas em poucos pontos, as médias de F1 e F2. Vimos na última lição que, para além das medidas de tendência central, também é importante conhecer a dispersão dos dados. Um gráfico de dispersão nos permite visualizar não só pontos isolados, mas também a distribuição de todos os dados da amostra.

Neste próximo comando, não há linhas comentadas, pois você já conhece a maior parte das funções: `ggplot()` – que define o dataframe e os parâmetros gráficos, `geom_point()` – que plota pontos, `scale_x_reverse()` e `scale_y_reverse()` – que

invertem os eixos x e y, `ggtitle()` – que inclui um título, `labs()` – que inclui rótulos para os eixos, e `theme_bw()` – que define o tema visual.

```
# Não rodar! Estrutura do código:

ggplot(df, aes(x = VAR, y = VAR, color = VAR)) +
  geom_point() +
  scale_x_reverse() +
  scale_y_reverse() +
  facet_grid(. ~ VAR) +
  ggtitle("Dispersão das medidas de F1 e F2 normalizados nas amostras
PBSP e SP2010") +
  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

A única função nova é `facet_grid()`. Essa função permite criar vários subgráficos semelhantes, de acordo com algum critério, que serão dispostos num grid. Aqui, vamos criar dois gráficos de dispersão, um para a amostra PBSP, e outro para SP2010.

Nesse comando, só falta definir os argumentos de `ggplot()`. Vamos plotar a localização (medidas de F1 e F2) de todas as vogais da planilha original, de modo que o dataframe será `pretonicas`. O eixo x será ocupado pelas medidas de F2.NORM, e o eixo y, pelas medidas de F1.NORM. Para mais bem visualizar os níveis de VOGAL, vamos plotá-las cada uma com uma cor. Em `facet_grid()`, defina a variável AMOSTRA. Substitua então os termos df e VAR na linha de comando pelos dados relevantes, revise o código e rode com CTRL + ENTER. O resultado se encontra na Figura 7.5.

```
ggplot(pretonicas, aes(x = F2.NORM, y = F1.NORM, color = VOGAL)) +
  geom_point() +
  scale_x_reverse() +
  scale_y_reverse() +
  facet_grid(. ~ AMOSTRA) +
  ggtitle("Dispersão das medidas de F1 e F2 normalizados nas amostras
PBSP e SP2010") +
  labs(x = "F2 normalizado", y = "F1 normalizado") +
  theme_bw()
```

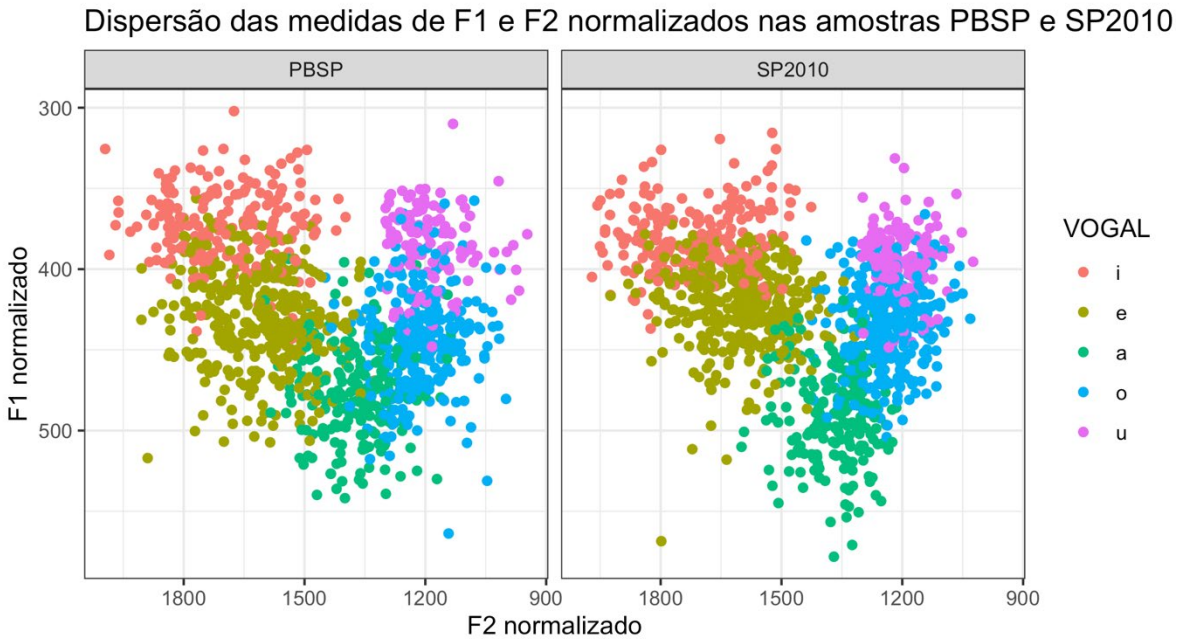

Figura 7.5: Gráfico de dispersão com ggplot. Fonte: própria.

Não ficou bacana? São várias observações que podem ser feitas sobre esse gráfico como, por exemplo, a maior dispersão das vogais /e/ e /o/ dos migrantes paraibanos do que aquelas dos paulistanos (talvez, justamente, pela situação de migração e contato com outro dialeto?...)

Vamos ver agora outro tipo de gráfico para variáveis numéricas, o boxplot. Antes de plotarmos um, vamos ver quais são suas características. Na Figura 7.6 está um esquema do que representa cada elemento de um boxplot.

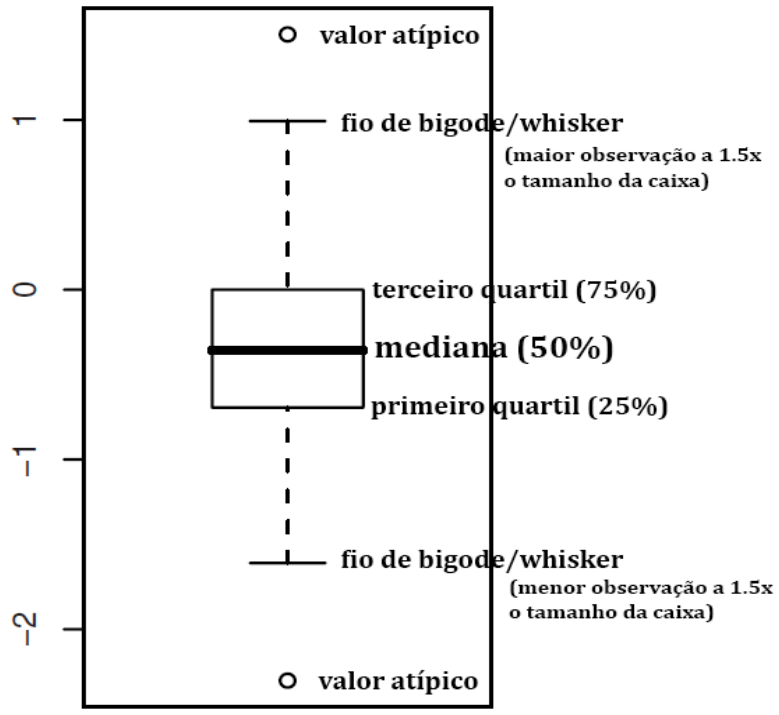

Figura 7.6: Estrutura do boxplot. Fonte: própria.

Comecemos pelo meio. A mediana é normalmente representada por uma linha mais escura e, como você já sabe, ela é a observação central quando se colocam os dados numa ordem crescente. Semelhantes à mediana são as medidas de primeiro quartil e terceiro quartil; mas, diferentemente da mediana (que representa o ponto em 50% da distribuição), eles indicam, respectivamente, os valores em 25% e 75% da distribuição.

Em seguida temos os whiskers, ou fios de bigode. Há um para cima e outro para baixo da "caixa" que forma o meio da distribuição. Eles são calculados tendo o primeiro e o terceiro quartil como referências. Subtrai-se o valor do primeiro quartil do terceiro, o que dá a extensão da caixa. Esse valor é multiplicado por 1,5, o que dará a extensão aproximada do fio do bigode para cima e para baixo.

Por fim, qualquer valor para além dos fios de bigode, para cima e para baixo, é considerado *outlier*, ou valor atípico. Desse modo, o boxplot também permite visualizar a distribuição e a dispersão dos dados.

No boxplot que vamos plotar, vamos comparar a distribuição de dados das 5 vogais para as 2 amostras. Aqui, assim como no gráfico de dispersão, não há linhas

comentadas, porque você já é capaz de interpretá-las. Falta substituir devidamente os valores df e VAR.

```
# Não rodar! Estrutura do código:

ggplot(df, aes(x = VAR, y = VAR, color = VAR)) +
  geom_boxplot(notch = FALSE) +
  scale_y_reverse() +
  labs(x = "Amostra", y = "F1 normalizado") +
  facet_grid(. ~ VAR) +
  theme_bw()
```

Desta vez, vamos fazer um pouco diferente. Veja a Figura 7.7: este deve ser o resultado de seu código! A partir dela, determine o que deve entrar em df, quais variáveis definem os parâmetros estéticos de x, y e color, e qual variável define as facetas.

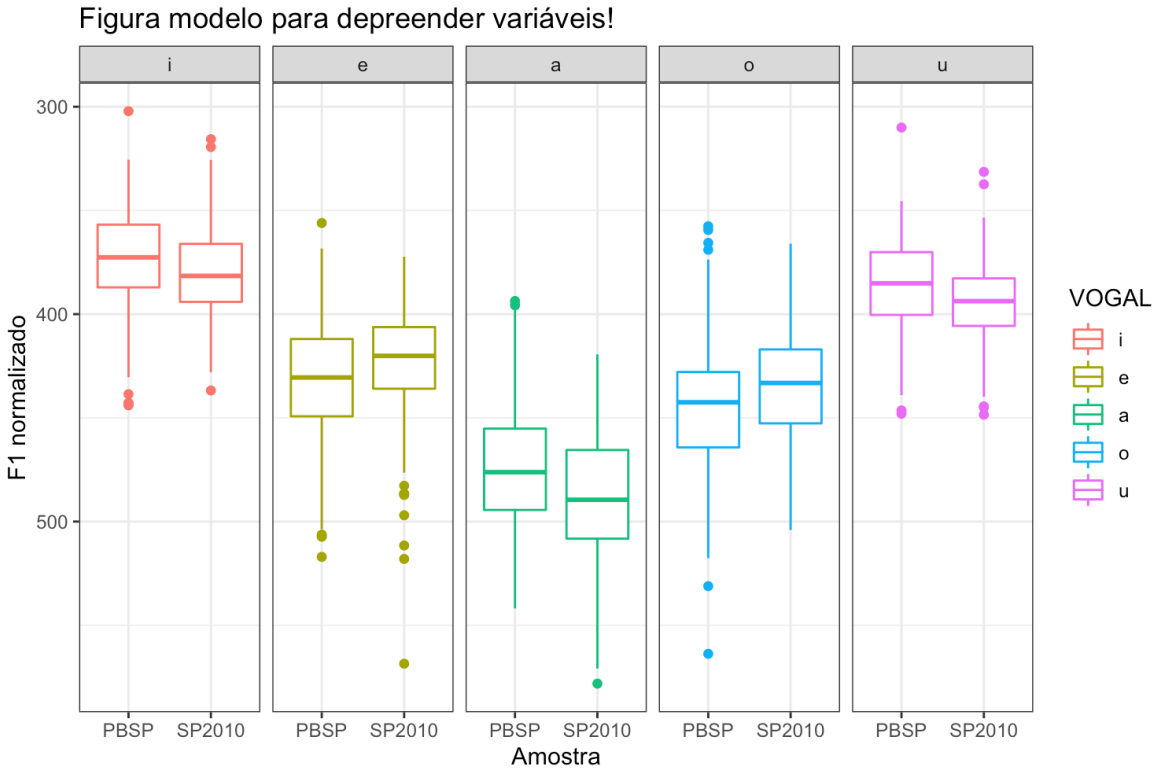

Figura 7.7: Boxplot a ser plotado. Fonte: própria.

Qual dataframe compreende os dados de interesse?

- `df`
- `medias`
- `pretonicas`

Qual variável ocupa o eixo x?

- AMOSTRA
- PBSP
- SP2010

Qual variável ocupa o eixo y?

- F1
- F1 normalizado
- F1.NORM

Qual variável define as cores dos boxplots?

- AMOSTRA
- F1.NORM
- VOGAL

Qual variável define o grid de facetas?

- AMOSTRA
- F1.NORM
- VOGAL

De posse dessas informações, substitua os termos `df` e `VAR` devidamente no *script*, revise o código e rode com CTRL + ENTER. O resultado deve ser o mesmo da Figura 7.7.

```
ggplot(pretonicas, aes(x = AMOSTRA, y = F1.NORM, color = VOGAL)) +
  geom_boxplot(notch = FALSE) +
  scale_y_reverse() +
  labs(x = "Amostra", y = "F1 normalizado") +
  facet_grid(. ~ VOGAL) +
  theme_bw()
```

Esse exercício de visualizar o gráfico que se quer plotar e, a partir dele, determinar o código, é algo que deve acontecer com frequência na prática. Muitas vezes, temos uma figura como modelo, e queremos reproduzi-la em nossos dados; em outros momentos, você pode simplesmente imaginar o gráfico que quer plotar, e o desafio é traduzi-lo em R!

Você pode estar se perguntando o que faz o argumento notch = FALSE em `geom_boxplot()`. Para descobrir, mude-o para TRUE e rode o comando (Figura 7.8).

```
ggplot(pretonicas, aes(x = AMOSTRA, y = F1.NORM, color = VOGAL)) +
  geom_boxplot(notch = TRUE) +
  scale_y_reverse() +
  labs(x = "Amostra", y = "F1 normalizado") +
  facet_grid(. ~ VOGAL) +
  theme_bw()
```

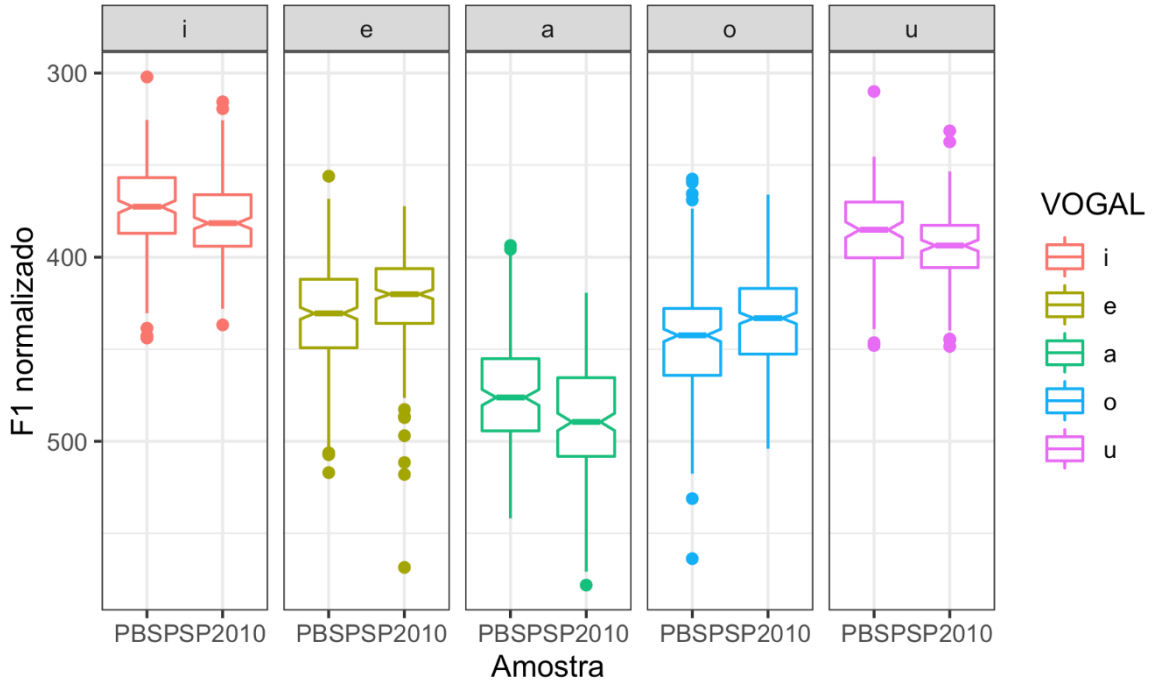

Figura 7.8: Boxplot com notch = TRUE. Fonte: própria.

O argumento notch estabelece se queremos ou não plotar um entalhe na altura da mediana, por meio de um valor lógico (T ou TRUE para verdadeiro ou F ou FALSE para falso). Mas ele não é apenas uma questão estética.

Na figura, vemos claramente que a vogal /a/ é a mais baixa e que as vogais /i/ e /u/ são mais altas. Vemos também que os paulistanos tendem a realizar as vogais /i/, /a/ e /u/ pretônicas como mais baixas que os paraibanos, mas que as vogais médias /e/ e /o/ são mais altas. No entanto, há também uma boa sobreposição entre os intervalos de cada vogal quando comparamos as duas amostras. Não é o caso que os paulistanos sempre realizam /e/ e /o/ pretônicas mais altos e os paraibanos tenham realizações sempre mais baixas. Pelos fios de bigode, vemos também que às vezes um /e/ pode ser tão alto quanto um /i/, ou tão baixo quanto um /a/.

Uma questão que se coloca, ao observar os boxplots, é se essas diferenças na altura das vogais são reais, ou se se devem simplesmente à distribuição dos dados. Mais especificamente, podemos nos perguntar: as vogais /e/ e /o/ dos paraibanos que vivem em São Paulo são realmente mais baixas do que as dos paulistanos? Para responder a essa questão, vamos precisar de ferramentas além da estatística descritiva (que temos implementado por meio de tabelas e gráficos). Vamos precisar de *testes estatísticos* e de ferramentas da *estatística inferencial*.

Os entalhes dos boxplots, no entanto, já são uma primeira pista para saber se podemos afirmar que as vogais /e/ e /o/ de paraibanos são em média mais baixas do que as dos paulistanos. Elas representam um intervalo de confiança – algo que veremos com mais detalhes na próxima lição. Por ora, basta indicar que, quando os entalhes não se sobrepõem, é provável que a diferença entre as amostras seja estatisticamente significativa.

Vamos, por fim, ver outro tipo de gráfico que se aplica a variáveis numéricas: o histograma. Quando tratamos de variáveis nominais, vimos que frequências são a medida de quantas vezes algo aconteceu numa amostra. O histograma "quebra" uma variável numérica em intervalos (p.ex., de 200 a 300, de 301 a 400 etc.) e representa graficamente a frequência dentro de cada intervalo. Isso permite visualizar em qual ou quais intervalos se concentram os dados.

Veja o código para plotar um histograma no *script*, no qual aproveito para revisar o pipe, do tidyverse: `%>%` permite encadear diversas operações – pegar o resultado da operação à esquerda e usá-lo no comando seguinte.

```
# Não rodar! Estrutura do código:

pretonicas %>%
  filter(VOGAL == "e" & AMOSTRA == "PBSP") %>%
  ggplot(., aes(x = VAR)) +
  geom_histogram(binwidth = n, fill = "white", color = "black") +
  labs(x = "F1 normalizado", y = "Frequência") +
  theme_bw()
```

Vamos plotar um histograma das medições de F1.NORM da vogal /e/ na amostra dos paraibanos. No código, primeiro pegamos o dataframe `pretonicas` e, com

`filter()`, criamos um subconjunto de dados da vogal /e/ na amostra PBSP. Este novo subconjunto de dados (sem nome!) é o dataframe utilizado para fazer o gráfico.

Para plotar um histograma, precisamos apenas de uma variável numérica. Vamos usar F1.NORM novamente. Na geometria `geom_histogram()`, podemos definir o tamanho dos intervalos (binwidth) – se de 5 em 5, 10 em 10 unidades etc.; com qual cor preencher (fill) as barras e com qual cor definir o contorno das barras (color). No código, inclua a variável F1.NORM e o número 20 para o tamanho dos intervalos de cada barra. O resultado se encontra na Figura 7.9.

```
pretonicas %>%
  filter(VOGAL == "e" & AMOSTRA == "PBSP") %>%
  ggplot(., aes(x = F1.NORM)) +
  geom_histogram(binwidth = 20, fill = "white", color = "black") +
  labs(x = "F1 normalizado", y = "Frequência") +
  theme_bw()
```

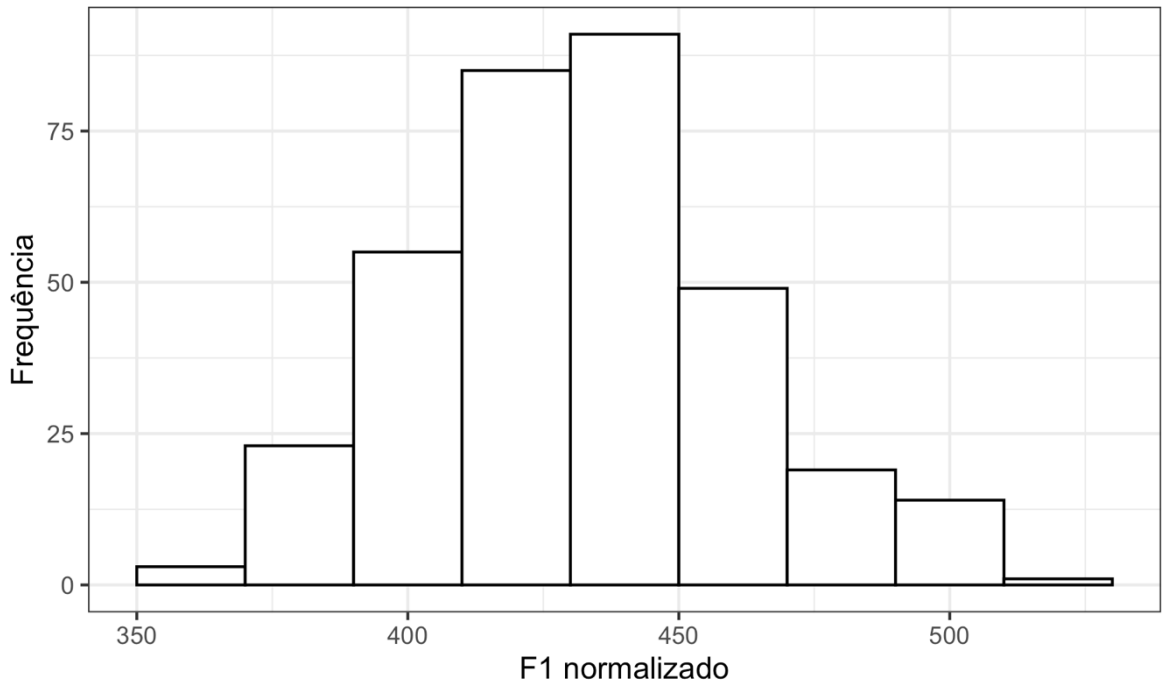

Figura 7.9: Histograma com ggplot. Fonte: própria.

Um histograma se parece com um gráfico de barras, mas não devemos confundi-los. Vimos que gráficos de barras se aplicam a variáveis nominais (como as lojas de departamento em Nova Iorque). Um histograma, diferentemente, se aplica a variáveis

contínuas, de modo que as barras são representadas no contínuo do eixo x. São informações diferentes.

O que o histograma nos informa? Ele mostra como se *distribuem* e se *dispersam* os dados da variável numérica. Nesse sentido, o histograma é uma representação visual das medidas de desvio padrão / variância, bem como das medidas de tendência central (média, mediana, moda). Quanto menos dispersos os dados, maior será sua concentração em torno de um ponto médio.

A figura mostra que a maior parte das medições de F1.NORM da vogal /e/ de paraibanos se concentra entre 410 e 430 Hz e entre 430 e 450 Hz – as duas barras com maior número de ocorrências (frequência). Os intervalos são de 20 Hz porque assim definimos com binwidth acima. Também houve ocorrências de /e/ com medidas acima e abaixo desses valores, mas eles foram menos frequentes.

Vamos checar como os dados se dispersam em torno da média e da mediana. O comando para computar a média e a mediana já está pronto neste ponto do *script*. Basta rodá-lo – mas certifique-se de que entende o que está sendo feito aí!

```
med_centrais_PBSPe <- pretonicas %>%
  filter(., VOGAL == "e" & AMOSTRA == "PBSP") %>%
  summarize(media = mean(F1.NORM),
            mediana = median(F1.NORM)) %>%
  print()
```

```
## # A tibble: 1 × 2
##   media mediana
##   <dbl>   <dbl>
## 1  432.    431.
```

Poderíamos inspecionar os valores de `media` e `mediana` no Console. No entanto, melhor do que isso, é visualizá-los! Vamos plotar, sobre o histograma, duas linhas que indicam esses pontos.

A linha de comando aqui reproduz o mesmo histograma que acabamos de criar, mas acrescenta mais duas linhas com a função `geom_vline()` que, como o nome já indica, plota uma linha vertical. Dentro dessa função, especificamos em qual ponto a linha vertical deve cortar o eixo x (xintercept): respectivamente, nos valores da média e

da mediana do dataframe que acabamos de computar. A média será representada por uma linha azul e a mediana por uma linha vermelha (Figura 7.10).

```
pretonicas %>%
  filter(., VOGAL == "e" & AMOSTRA == "PBSP") %>%
  ggplot(., aes(x = F1.NORM)) +
  geom_histogram(binwidth = 20, fill = "white", color = "black") +
  labs(x = "F1 normalizado", y = "Frequência") +
  theme_bw() +
  geom_vline(xintercept = med_centrais_PBSPe$media, color = "blue") +
  geom_vline(xintercept = med_centrais_PBSPe$mediana, color = "red")
```

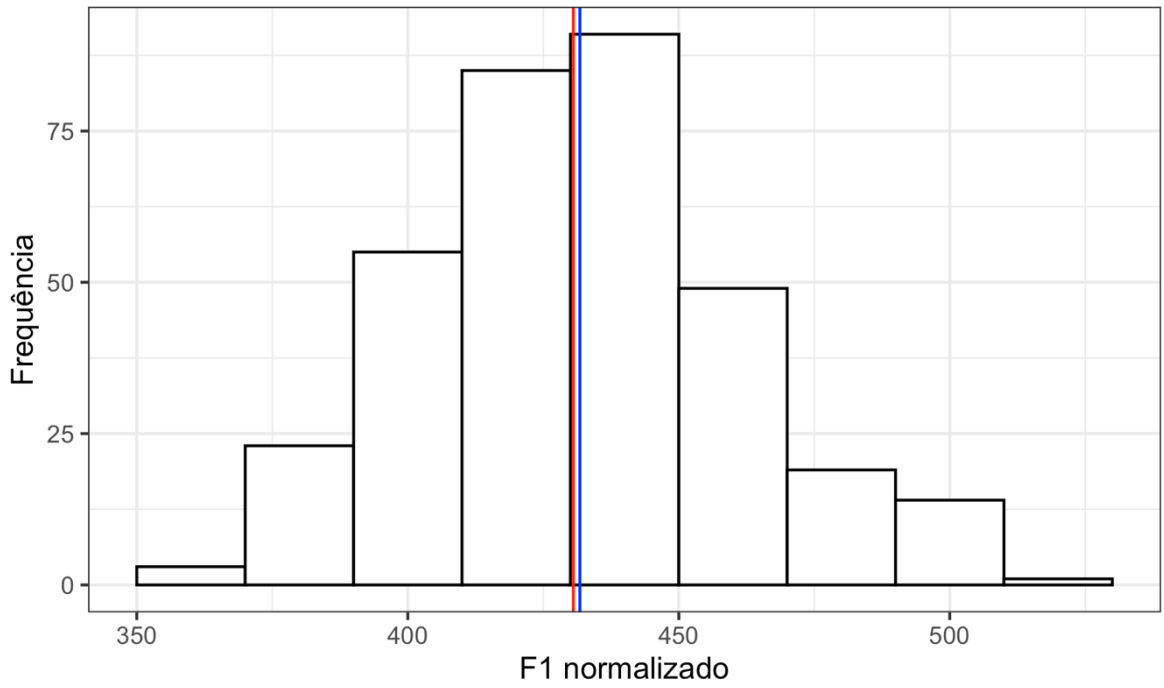

Figura 7.10: Histograma com indicação dos valores de média e de mediana. Fonte: própria.

Vemos que a média e a mediana são praticamente coincidentes, próximas ao topo da distribuição. Você se lembra da Figura 6.1, vista na lição passada, que representa modelos idealizados da distribuição e dispersão dos dados? Nosso histograma representa uma distribuição real.

Com qual das figuras você acha que nosso histograma mais se parece?

- com a figura do meio, uma distribuição normal
- com a figura à esquerda, com viés negativo
- com a figura à direita, com viés positivo

Nosso histograma se parece com uma distribuição normal e, na verdade, isso não é surpreendente. Temos usado os dados de F1 e de F2 *normalizados* de acordo com o método de Lobanov. O que a normalização faz é justamente isso: transformar os dados em uma distribuição que se aproxima da distribuição normal.

Você já deve ter percebido que o termo "distribuição normal", aqui, é o um termo técnico. Ela se refere a dados que se distribuem na forma de uma curva de sino. Veremos as propriedades dessa distribuição com mais detalhes na próxima lição.

Para saber mais

Na Lição 5 e nesta, vimos apenas alguns poucos tipos de gráficos: de barras, de linhas, histogramas, dispersão e boxplots. Mas existem vários outros que você deve explorar. No site The R Graph Gallery (http://www.r-graph-gallery.com/), há uma enorme coleção de gráficos produzidos no R, com seus respectivos códigos. Outros tipos de gráficos podem mais bem expressar as relações que você quer mostrar.

Além disso, recomendo a consulta constante ao livro "R Graphics Cookbook", de Winston Chang. Trata-se realmente de um livro de receitas, que apresenta "problemas" – o que você quer fazer – e o passo a passo para solucioná-los. Uma versão dele se encontra *on-line* em https://r-graphics.org/.

A mensagem aqui é: não deixe as ferramentas conduzirem sua análise! Você, como pesquisador ou pesquisadora, deve estabelecer o que é mais adequado para os seus dados.

Exercícios

Nesta lista de exercícios, você vai precisar do dataframe `pretonicas`, que foi usado na lição. Siga os comandos da lição caso precise recarregá-lo no R.

1. Nesta lista, você vai precisar dos pacotes tidyverse e RColorBrewer. Carregue-os.
2. Nesta lição, fizemos um histograma da distribuição das medições de F1 normalizado da vogal /e/ dos dados da amostra PBSP. Crie um novo histograma

semelhante, mas agora com os dados de F1 (i.e., sem normalização.). Para tanto, com auxílio do pipe `%>%` e a partir do dataframe `pretonicas`, (i) crie um subconjunto apenas dos dados da vogal "e" na amostra PBSP; (ii) determine o parâmetro estético para o eixo x; (iii) use a geometria de histograma com bindwidth = 30, com barras "lightblue" e bordas das barras "gray"; (iv) nomeie o eixo x como "F1" e o eixo y como "Frequência"; (v) intitule o gráfico "Medidas de F1 da vogal /e/ da amostra PBSP"; e (vi) use o tema `theme_minimal()`.

3. Reproduza o gráfico acima para a vogal /o/ dos paraibanos. Modifique apenas o estritamente necessário!
4. Compare os dois gráficos que acabou de plotar. Você pode voltar a gráficos anteriores ou posteriores clicando sobre a flecha para a esquerda e para a direita no topo da aba Plots. Pelo histograma, qual das vogais deve ter maiores valores de média e mediana? Justifique sua resposta.
5. Determine os valores de média e mediana das vogais /e/ e /o/ dos paraibanos. Para tanto, com auxílio do pipe e a partir do dataframe `pretonicas`, (i) crie um subconjunto dos dados das vogais "e" e "o" na amostra "PBSP"; (ii) agrupe os dados por VOGAL; e (iii) calcule as medidas de média e de mediana, nomeando as colunas media_F1 e mediana_F1 respectivamente.
6. Crie dois histogramas da distribuição dos dados da vogal /e/, um para cada amostra (PBSP e SP2010), cada qual com uma cor diferente. Para tanto, com auxílio do pipe e a partir do dataframe `pretonicas`, (i) crie um subconjunto dos dados da vogal /e/; (ii) defina os parâmetros gráficos para x e para fill dentro da função `ggplot()`; (iii) use a geometria de histograma com binwidth = 20 e alpha = 0.4; (iv) nomeie o eixo x como "F1" e o eixo y como "Frequência"; (v) use a função `facet_grid()` para que os histogramas de cada amostra sejam dispostos um sobre o outro; (vi) use a paleta "Pastel2" do RColorBrewer; e (vii) aplique o tema `theme_minimal()`.
7. Faça boxplots dos dados de F1 da vogal /e/ por PARTICIPANTE, diferenciando os falantes da amostra PBSP e SP2010 com cores distintas. Para tanto, com auxílio

do pipe e a partir do dataframe pretonicas, (i) crie um subconjunto de dados da vogal “e”; (ii) defina os parâmetros gráficos x, y e color dentro da função `ggplot()`; (iii) use a geometria para boxplots sem o notch; (iv) inverta a escala dos valores do eixo y (para que os valores de F1 sejam representados do modo fonético convencional); (v) nomeie o eixo x como “Falante”, o eixo y como “F1”, e a variável da legenda como “Amostra”; (vi) intitule o gráfico como “Medidas de F1 da vogal /e/ por falantes das amostras PBSP e SP2010”; e (vii) use o tema `theme_light()`. Nota: se você achar que os nomes dos falantes não estão bem dispostos na figura, veja-a com Zoom, pois os nomes não estão de fato encavalados!

8. Para comparar, reproduza o mesmo gráfico acima para os dados de F1 da vogal /e/ com normalização (F1.NORM). Faça modificações (i) no parâmetro estético relevante; (ii) no rótulo do eixo y, de “F1” para “F1 normalizado”; e (iii) no título do gráfico, trocando igualmente “F1” por “F1 normalizado”.
9. Comparando os gráficos, em qual deles há maior dispersão das medições da altura da vogal /e/: com F1 ou com F1.NORM? Explique sua resposta.
10. Determine a diferença da dispersão das medidas da altura da vogal /e/ numericamente. Para tanto, com auxílio do pipe e a partir do dataframe `pretonicas`, (i) crie um subconjunto de dados da vogal “e”; e (ii) compute as médias de desvio padrão de F1 e de F1.NORM, nomeando as colunas como sd_F1 e sd_F1.NORM respectivamente.
11. Nesta lição, plotamos um gráfico de dispersão que representa todas as vogais “i”, “e”, “a”, “o” e “u” de paraibanos migrantes e paulistanos, e em que cada vogal é identificada por pontos de cores distintas. No ggplot2, também é possível representar cada ponto por um caractere específico – p.ex., as próprias vogais – por meio da geometria `geom_text()`. Façamos então um tal gráfico. Para tanto, com auxílio do pipe e a partir do dataframe `pretonicas`, (i) defina os parâmetros gráficos x, y e color dentro da função `ggplot()`, de modo que o eixo x represente o traço [+- anterior] de vogais normalizadas, o eixo y represente o traço [+- alto]

de vogais normalizadas e cada vogal tenha uma cor diferente; (ii) use a geometria `geom_text()` com o argumento aes(label = VOGAL); (iii/iv) inverta a escala dos eixos x e y, de modo que a representação do espaço vocálico siga a convenção fonética; (v) use a função `facet_grid()` de modo que sejam plotados os espaços vocálicos separados por AMOSTRA, dispostas lado a lado; (vi) intitule o gráfico "Dispersão das medidas de F1 e F2 normalizados nas amostras PBSP e SP2010"; (vii) nomeie o eixo x como "F2 normalizado" e o eixo y como "F1 normalizado"; e (viii) adicione theme(legend.position = "none") ao final, para que não seja plotada a legenda, que agora não é mais necessária.

12. Nesta lição, fizemos um gráfico de linhas que representa os espaços vocálicos de paraibanos migrantes e paulistanos, com as médias de F1 e F2 normalizados. Você vai fazer agora um gráfico semelhante, mas com as medidas das vogais de F1 e F2 não normalizados. Para tanto, faça primeiro, com auxílio do pipe e a partir do dataframe `pretonicas`, um dataframe que computa as médias de F1 e de F2 para cada vogal e para cada amostra: (i) guarde o resultado em dataframe chamado `medias`; (ii) agrupe os dados por VOGAL e AMOSTRA; (iii) calcule as médias de F1 e de F2, nomeando as colunas como media_F1 e media_F2 respectivamente; (iv) visualize o resultado em tela com `print()`.
13. Com auxílio do pipe e a partir do dataframe `medias`, (i) defina os parâmetros x, y, color e label dentro da função `ggplot()`; (ii/iii) empregue as geometrias de linha e de rótulo (label); (iv/v) inverta as escalas dos eixos x e y, de modo que os espaços vocálicos sejam representados da maneira convencional em Fonética; (vi) intitule o gráfico como "Valores médios de F1 e F2 nas amostras PBSP e SP2010"; (vii) nomeie o eixo x como "F2 (Hz)", o eixo y como "F1 (Hz)" e a variável da legenda como "Amostra"; (viii) use a paleta "Paired" do pacote RColorBrewer na função `scale_color_brewer()`; e (ix) use o tema `theme_classic()`.
14. Reproduza o gráfico da Figura 7.11, que foi feito a partir dos dados de pretonicas. Ele consiste nos espaços vocálicos de cada um dos sete falantes paraibanos, separados por seu gênero. Atente-se a todos os detalhes (nomes de eixos, cores,

título, tema etc.). Como mencionado na lição, um bom modo de aprimorar a representação gráfica de dados é tentar reproduzir gráficos vistos em outras fontes, ou mesmo visualizar um gráfico em sua mente e tentar fazê-lo no ggplot2.

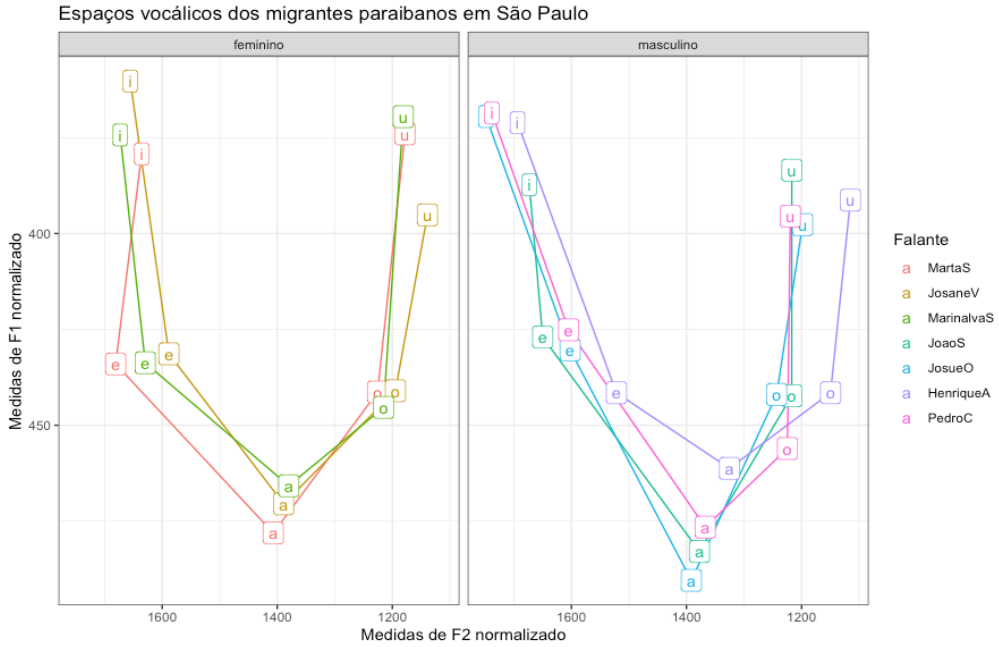

Figura 7.11: Gráfico para reprodução no ggplot2. Fonte: própria.