Lição 11: Correlação e Regressão

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

```
idade <- c(1, 2, 3, 4, 5, 5, 5, 6, 7, 8, 8, 9, 11, 12, 12)

altura <- c(60, 65, 97, 98, 100, 105, 107, 105, 119, 122,
            125, 132, 142, 147, 153)

criancas <- as.data.frame(cbind(idade, altura))
```

Na Lição 9, vimos testes estatísticos que se aplicam a variáveis nominais; na Lição 10, vimos testes que se aplicam a variáveis numéricas, quando queremos comparar dois grupos ou comparar uma média com outra média conhecida. Nesta lição, veremos outros testes estatísticos que se aplicam a variáveis numéricas, em relação a outra variável também numérica.

Você já sabe o que fazer no início da sessão: carregar pacotes necessários! Carregue o pacote tidyverse.

```
library(tidyverse)
```

E, nesta sessão, também vamos usar outro pacote! Carregue o pacote chamado GGally.

```
library(GGally)
```

Comecemos com um exemplo não linguístico. Existe correlação entre a idade e a altura de crianças?

- sim
- não
- não sei

Sabemos, por simples observação cotidiana, que sim! Vamos analisar este caso com alguns dados. Deixei disponíveis para você dois vetores numéricos no dataframe `criancas`. Inspecione-o.

```
criancas
```

```
##    idade altura
## 1      1     60
```

```
## 2       2     65
## 3       3     97
## 4       4     98
## 5       5    100
## 6       5    105
## 7       5    107
## 8       6    105
## 9       7    119
## 10      8    122
## 11      8    125
## 12      9    132
## 13     11    142
## 14     12    147
## 15     12    153
```

Esse dataframe contém as idades (em anos) e as alturas (em centímetros) de 15 crianças. Vamos agora fazer um gráfico de dispersão que relaciona idade e altura, por meio da função `geom_point()`, que já havíamos visto na Lição 7. Para as coordenadas, use `idade` para o eixo x e `altura` para o eixo y (Figura 11.1).

```
ggplot(criancas, aes(x = idade, y = altura)) +
  geom_point()
```

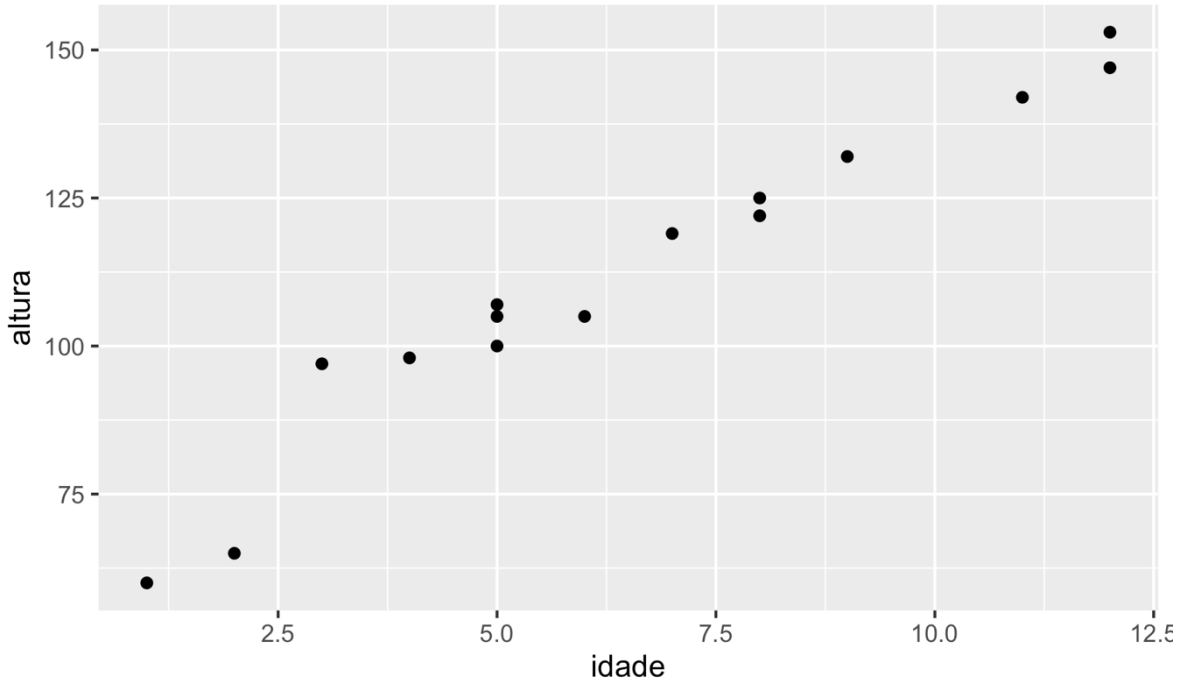

Figura 11.1: Gráfico de dispersão da altura por idade das crianças. Fonte: própria.

A figura plotada mostra claramente a correlação entre idade e altura das crianças: quanto mais velha a criança, maior sua altura. A disposição dos pontos lembra uma reta

inclinada, e é justamente a inclinação da reta que nos faz concluir que existe uma correlação entre essas duas variáveis.

A inspeção gráfica é sempre um excelente ponto de partida, mas os testes estatísticos nos permitem chegar a estimativas mais precisas sobre se há correlação ou não entre duas variáveis. Quando temos duas variáveis numéricas, o teste indicado é o teste de correlação, que no R é feito por meio da função `cor.test()`. Esta função toma como argumentos dois vetores numéricos de igual extensão. Aplique-a então aos vetores idade e altura do dataframe `criancas`.

```
cor.test(criancas$idade, criancas$altura)
```

```
##
##  Pearson's product-moment correlation
##
## data:  criancas$idade and criancas$altura
## t = 14.693, df = 13, p-value = 1.782e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9132826 0.9906151
## sample estimates:
##       cor
## 0.9711854
```

Faz diferença se trocarmos a ordem dos vetores? Vamos testar. Aplique a função `cor.test()` com os vetores na ordem inversa daquela que você usou na linha anterior.

```
cor.test(criancas$altura, criancas$idade)
```

```
##
##  Pearson's product-moment correlation
##
## data:  criancas$altura and criancas$idade
## t = 14.693, df = 13, p-value = 1.782e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9132826 0.9906151
## sample estimates:
##       cor
## 0.9711854
```

Vemos que o resultado é idêntico. Essa é uma primeira propriedade do teste de correlação de Pearson: ele não pressupõe uma direção da correlação; apenas avalia se duas variáveis x e y se correlacionam. Vejamos então o resultado de um teste de Pearson, cuja estrutura, a esta altura, já deve ser familiar a você.

As duas primeiras linhas indicam o teste realizado e o conjunto de dados a que se aplicou o teste. Em seguida, o R informa o valor-*t*, os graus de liberdade e o valor de significância, de modo muito semelhante ao resultado do teste-t. Numa correlação de Pearson, os graus de liberdade são o número de pares (x, y) – aqui, 15 –, menos 2: n - 2 = 13. As linhas seguintes enunciam a hipótese alternativa, o intervalo de confiança e um valor de correlação.

O valor de correlação gerado pelo teste de correlação de Pearson é chamado de *r*. Ele é calculado a partir da fórmula que aparece no *script* do Anexo A, e leva em conta o número de observações n, a soma simples e dos quadrados de x, a soma simples e dos quadrados de y, e o produto x * y. Após esta lição, separe um momento para estudá-la.

Para nossos propósitos práticos, o resultado da fórmula de *r* é sempre um número que vai de -1 a +1, que indica não só se há correlação entre x e y, mas também a força da correlação. Um valor de -1, ou próximo dele, indica uma forte correlação negativa: quanto mais x, menos y. Um valor de +1, ou próximo dele, indica uma forte correlação positiva: quanto mais x, mais y. Um valor próximo a zero indica ausência de correlação.

O nosso teste de correlação entre idade e altura das crianças dá um valor de r de Pearson igual a 0,97, o que indica forte correlação positiva entre as duas variáveis. O intervalo de confiança de 95% estima que o r de Pearson poderia ter sido entre 0,91 e 0,99, um intervalo que não inclui zero, daí o valor de significância ser abaixo de 0,05.

Assim como o teste-t, o teste de correlação tem uma versão paramétrica e uma não paramétrica, a depender de se as distribuições seguem ou não a distribuição normal. Aplique o teste de Shapiro ao vetor `idade` para verificar sua normalidade.

```
shapiro.test(criancas$idade)

##
##  Shapiro-Wilk normality test
##
## data:  criancas$idade
## W = 0.95595, p-value = 0.6225
```

Como $p > 0{,}05$, podemos assumir que a distribuição dos valores de `idade` segue a distribuição normal. Aplique agora o teste de Shapiro ao vetor `altura`.

```
shapiro.test(criancas$altura)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  criancas$altura
## W = 0.95079, p-value = 0.537
```

Ambas as variáveis seguem a distribuição normal, de modo que o teste de correlação de Pearson foi apropriado. Se este não tivesse sido o caso, a solução seria aplicar uma variante do teste de correlação, adicionando novo argumento a `cor.test()`: method = "spearman", para aplicar o teste de correlação de Spearman. Apenas como prática, aplique o teste de correlação com o método de Spearman às variáveis `idade` e `altura` para ver o resultado.

```
cor.test(criancas$idade, criancas$altura, method = "spearman")

##
##  Spearman's rank correlation rho
##
## data:  criancas$idade and criancas$altura
## S = 8.5479, p-value = 2.96e-11
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9847359
```

Começamos esta lição com a pergunta: "Existe correlação entre a idade e a altura das crianças?", e vimos, pelo teste de correlação, que existe uma forte correlação entre elas. Mas o que significa "haver correlação" entre duas variáveis? Um dos interesses em encontrar correlações reside no fato de que, se duas variáveis se correlacionam, podemos estimar o valor de uma se temos o valor da outra. Por exemplo, quando vemos uma criança, já temos uma ideia de qual é a sua idade. De modo semelhante, também podemos estimar qual é a altura de uma criança se sabemos quantos anos ela tem.

Quando sabemos que há correlação entre duas variáveis, podemos formalizar tal conhecimento dentro de um *modelo estatístico*, que nos ajuda a chegar a estimativas mais precisas. Façamos isso com nossos dados de idade e altura das crianças. Quando duas variáveis têm uma relação linear, como é o caso dessas duas variáveis, podemos criar um *modelo linear* por meio da função `lm()` – linear model. Essa função toma como primeiro argumento uma fórmula no formato y ~ x, que já usamos na função `t.test()`. Nesta

fórmula, y é uma variável *dependente*, x é uma variável *independente* e ~ pode ser glosado como "explicado por"; em outras palavras, estamos testando se y pode ser explicado por ou depende de ou tem correlação com x. Entre altura e idade, qual é a variável *dependente*?

- altura, pois a altura depende da idade
- idade, pois a idade depende da altura

Sim, é a altura que depende da idade! Na função `cor.test()`, não precisamos definir a direção da correlação, mas na função `lm()` isso é necessário. Aplique então a função `lm()` à fórmula altura ~ idade. Como segundo argumento, informe o conjunto de dados com data = criancas. Guarde o resultado em um objeto chamado `modelo`.

```
modelo <- lm(altura ~ idade, data = criancas)
```

Vejamos o resultado da função. Digite `modelo` no Console.

```
modelo
```

```
##
## Call:
## lm(formula = altura ~ idade, data = criancas)
##
## Coefficients:
## (Intercept)        idade
##      62.504        7.545
```

O resultado de um modelo linear gera dois coeficientes: um valor de interceptação (também chamado de coeficiente linear) e um coeficiente angular. Em nosso modelo, o coeficiente linear é 62,5 e o coeficiente angular é 7,5. O que são esses números?

Para bem entendê-los, vamos dar um passeio em suas memórias das aulas de Matemática... vai aqui uma questão fácil: y + 3 = 5. Qual é o valor de y? (Desculpe-me se estou ofendendo sua inteligência... prometo que isso vai chegar a um lugar!)

- 2
- 3
- 4
- depende!

E em y - 10 = 15, qual é o valor de y?

- 25
- 30
- 35
- depende!

E em x + y = 30, qual é o valor de y?

- 2
- 20
- 40
- depende!

Quando temos duas variáveis, x e y, o valor de y depende do valor de x. A relação entre duas variáveis é chamada de função, que pode ser denotada genericamente pela expressão y = f(x).

Uma reta no plano cartesiano segue uma função de primeiro grau no formato y = a + bx. Há aí dois novos termos, a e b. O primeiro, "a", é o valor de y quando x é igual a zero (experimente colocar x = 0 na expressão y = a + bx). "a", portanto, é o ponto em que a reta intersecciona o eixo y.

O termo "b" denota em quanto aumenta ou diminui o valor de y a cada unidade de x. Se b = 0, o valor de y é uma constante (ou seja, não é uma variável); se b > 0, a reta é ascendente; se b < 0, a reta é descendente.

Os dois valores gerados em nosso modelo linear denotam justamente os valores de "a" e "b", respectivamente o coeficiente linear e o coeficiente angular. Nosso modelo então segue a função y = 62,5 + 7,5x. Vamos visualizar isso no gráfico. A linha de comando neste ponto do *script* é a mesma que rodamos anteriormente para plotar o gráfico, com a adição da função `geom_smooth()`, que usa justamente um modelo linear `lm` para plotar a linha. Rode-a agora para visualizar a Figura 11.2.

```
ggplot(criancas, aes(x = idade, y = altura)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey")
```

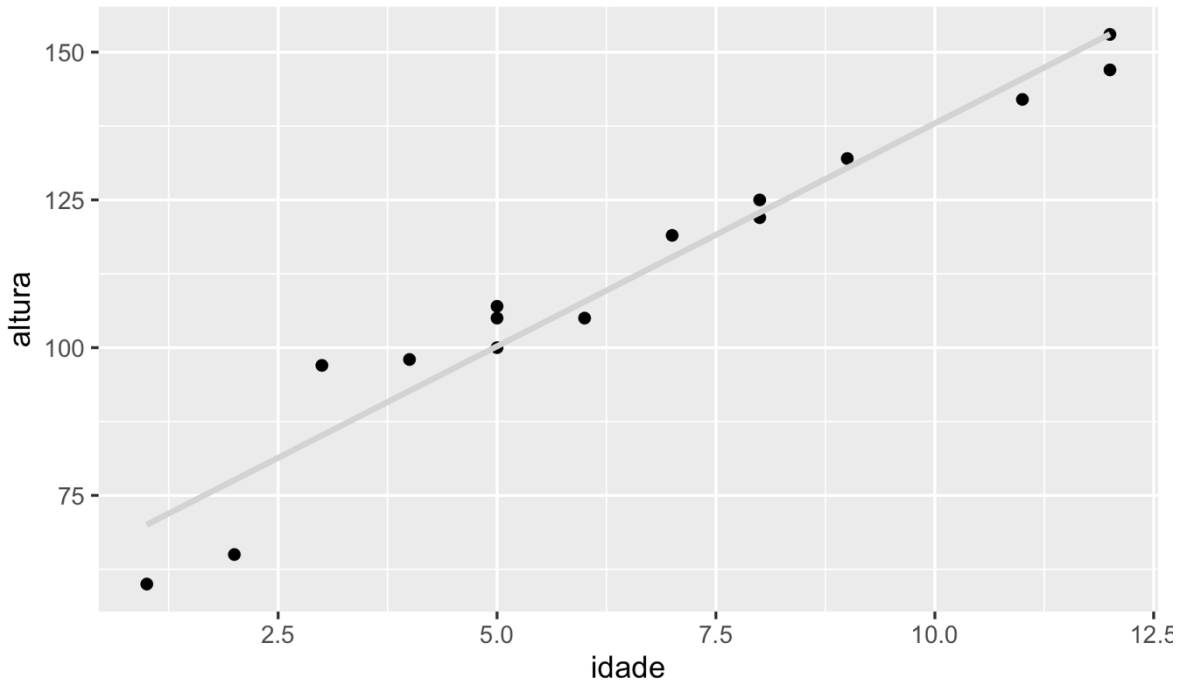

Figura 11.2: Gráfico de dispersão da altura por idade das crianças, com linha de regressão. Fonte: própria.

A linha plotada no gráfico é uma linha de regressão. Ela representa os pontos pelos quais deve passar a reta de modo que a soma das distâncias entre as observações (os pontos) e a linha seja a menor possível. Ela cruza o eixo y, altura, em 62,5cm e aumenta 7,5cm a cada unidade de x. Isso significa que nosso modelo prevê que as crianças nascem com cerca de 62,5 cm e que crescem cerca de 7,5 cm por ano.

Você pode ter achado a altura de 62,5 cm muito grande para um recém-nascido. Mas veja que isso é um modelo, não a realidade! Em parte, nosso modelo é menos preciso do que poderia ser porque estamos levando em conta apenas uma variável – a idade – para prever a altura. Certamente há outras que influenciam a altura das crianças: a altura dos pais, a alimentação etc. Adiante, no curso, veremos como incluir mais variáveis num modelo estatístico, o que faz com que os parâmetros sejam ajustados. Mas mesmo com a inclusão de outras tantas variáveis, o modelo nunca vai se equiparar com a realidade.

Um pequeno conto de Jorge Luis Borges, chamado Del Rigor en la Ciencia, ilustra bem este ponto: “En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el mapa de una sola Provincia ocupaba toda una Ciudad, y el mapa del Imperio, toda una

Provincia. Con el tiempo, estos Mapas Desmesurados no satisficieron y los Colegios de Cartógrafos levantaron un Mapa del Imperio, que tenía el tamaño del Imperio y coincidía puntualmente con él. Menos Adictas al Estudio de la Cartografía, las Generaciones Siguientes entendieron que ese dilatado Mapa era Inútil y no sin Impiedad lo entregaron a las Inclemencias del Sol y los Inviernos. En los desiertos del Oeste perduran despedazadas Ruinas del Mapa, habitadas por Animales y por Mendigos; en todo el País no hay otra reliquia de las Disciplinas Geográficas. Suárez Miranda, Viajes de Varones Prudentes, Libro Cuarto, Cap. XLV, Lérida, 1658."

Um modelo é necessariamente uma representação simplificada da realidade – caso contrário, corre-se o risco de ter um modelo inútil, tal como o mapa que ocupa todo o Império. Ainda que o modelo se distancie da realidade, ele é útil para apreender o todo: para descrevê-lo, explicá-lo e fazer previsões.

Vejamos então o que mais o modelo nos fornece. No R, o resultado da função `lm()` é, minimamente, os dois coeficientes necessários para plotar a linha de regressão, mas o R também gera outros valores dentro do modelo. Aplique a função `str()` a `modelo` para ver essas outras informações.

```
str(modelo)
```

```
## List of 12
##  $ coefficients : Named num [1:2] 62.5 7.55
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "idade"
##  $ residuals    : Named num [1:15] -10.049 -12.595 11.86 5.315 -0.2
31 ...
##   ..- attr(*, "names")= chr [1:15] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:15] -433 97.72 15.47 8.28 2.09 ...
##   ..- attr(*, "names")= chr [1:15] "(Intercept)" "idade" "" "" ...
##  $ rank         : int 2
##  $ fitted.values: Named num [1:15] 70 77.6 85.1 92.7 100.2 ...
##   ..- attr(*, "names")= chr [1:15] "1" "2" "3" "4" ...
##  $ assign       : int [1:2] 0 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:15, 1:2] -3.873 0.258 0.258 0.258 0.258 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:15] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:2] "(Intercept)" "idade"
##   .. ..- attr(*, "assign")= int [1:2] 0 1
##   ..$ qraux: num [1:2] 1.26 1.26
##   ..$ pivot: int [1:2] 1 2
##   ..$ tol  : num 1e-07
##   ..$ rank : int 2
```

```
##   ..- attr(*, “class”)= chr “qr”
##  $ df.residual  : int 13
##  $ xlevels      : Named list()
##  $ call         : language lm(formula = altura ~ idade, data = cria
ncas)
##  $ terms        :Classes 'terms', 'formula'  language altura ~ idad
e
##   .. ..- attr(*, “variables”)= language list(altura, idade)
##   .. ..- attr(*, “factors”)= int [1:2, 1] 0 1
##   .. .. ..- attr(*, “dimnames”)=List of 2
##   .. .. .. ..$ : chr [1:2] “altura” “idade”
##   .. .. .. ..$ : chr “idade”
##   .. ..- attr(*, “term.labels”)= chr “idade”
##   .. ..- attr(*, “order”)= int 1
##   .. ..- attr(*, “intercept”)= int 1
##   .. ..- attr(*, “response”)= int 1
##   .. ..- attr(*, “.Environment”)=<environment: R_GlobalEnv>
##   .. ..- attr(*, “predvars”)= language list(altura, idade)
##   .. ..- attr(*, “dataClasses”)= Named chr [1:2] “numeric” “numeric
”
##   .. .. ..- attr(*, “names”)= chr [1:2] “altura” “idade”
##  $ model        :'data.frame':   15 obs. of  2 variables:
##   ..$ altura: num [1:15] 60 65 97 98 100 105 107 105 119 122 ...
##   ..$ idade : num [1:15] 1 2 3 4 5 5 5 6 7 8 ...
##   ..- attr(*, “terms”)=Classes 'terms', 'formula'  language altura
~ idade
##   .. .. ..- attr(*, “variables”)= language list(altura, idade)
##   .. .. ..- attr(*, “factors”)= int [1:2, 1] 0 1
##   .. .. .. ..- attr(*, “dimnames”)=List of 2
##   .. .. .. .. ..$ : chr [1:2] “altura” “idade”
##   .. .. .. .. ..$ : chr “idade”
##   .. .. ..- attr(*, “term.labels”)= chr “idade”
##   .. .. ..- attr(*, “order”)= int 1
##   .. .. ..- attr(*, “intercept”)= int 1
##   .. .. ..- attr(*, “response”)= int 1
##   .. .. ..- attr(*, “.Environment”)=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, “predvars”)= language list(altura, idade)
##   .. .. ..- attr(*, “dataClasses”)= Named chr [1:2] “numeric” “nume
ric”
##   .. .. .. ..- attr(*, “names”)= chr [1:2] “altura” “idade”
##  - attr(*, “class”)= chr “lm”
```

Assim como na função de qui-quadrado (Lição 9), o resultado da função `lm()` é uma lista com diversas medidas estatísticas, que podem ser acessadas pelo operador `$`. Digite `modelo$fitted.values` e guarde o resultado em um objeto chamado `previsao`.

```
previsao <- modelo$fitted.values
```

Visualize o vetor `previsao`.

```
previsao
```

```
##         1         2         3         4         5         6
7
##  70.04928  77.59459  85.13990  92.68521 100.23052 100.23052 100.230
52
##         8         9        10        11        12        13
14
## 107.77583 115.32114 122.86645 122.86645 130.41176 145.50238 153.047
69
##        15
## 153.04769
```

No modelo linear, os valores previstos são aqueles que coincidiriam com a linha de regressão. Na figura, vemos que a maior parte das observações não coincide exatamente com a linha. A diferença entre os valores observados e os valores previstos é o resíduo. Acesse os valores de resíduos com `modelo$residuals` e guarde-os em um objeto chamado `residuos`.

```
residuos <- modelo$residuals
```

Visualize o vetor `residuos`.

```
residuos
```

```
##            1            2            3            4            5
## -10.04928458 -12.59459459  11.86009539   5.31478537  -0.23052464
##            6            7            8            9           10
##   4.76947536   6.76947536  -2.77583466   3.67885533  -0.86645469
##           11           12           13           14           15
##   2.13354531   1.58823529  -3.50238474  -6.04769475  -0.04769475
```

Veja que os resíduos são exatamente a diferença entre o valor observado – a altura – e os valores previstos no modelo. Digite `criancas$altura - modelo$fitted.values` para obter o mesmo resultado acima.

```
criancas$altura - modelo$fitted.values
```

```
##            1            2            3            4            5
## -10.04928458 -12.59459459  11.86009539   5.31478537  -0.23052464
##            6            7            8            9           10
##   4.76947536   6.76947536  -2.77583466   3.67885533  -0.86645469
##           11           12           13           14           15
##   2.13354531   1.58823529  -3.50238474  -6.04769475  -0.04769475
```

Os primeiros dois valores são negativos porque os valores observados estão abaixo dos valores previstos. O terceiro e o quarto valores, por sua vez, são positivos, e

assim por diante. Rode a próxima linha de comando, já pronta, que plota segmentos que indicam as diferenças entre valores observados e previstos (Figura 11.3).

```
ggplot(criancas, aes(x = idade, y = altura)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey") +
  geom_segment(aes(xend = idade, y = altura, yend = previsao), alpha =
.2)
```

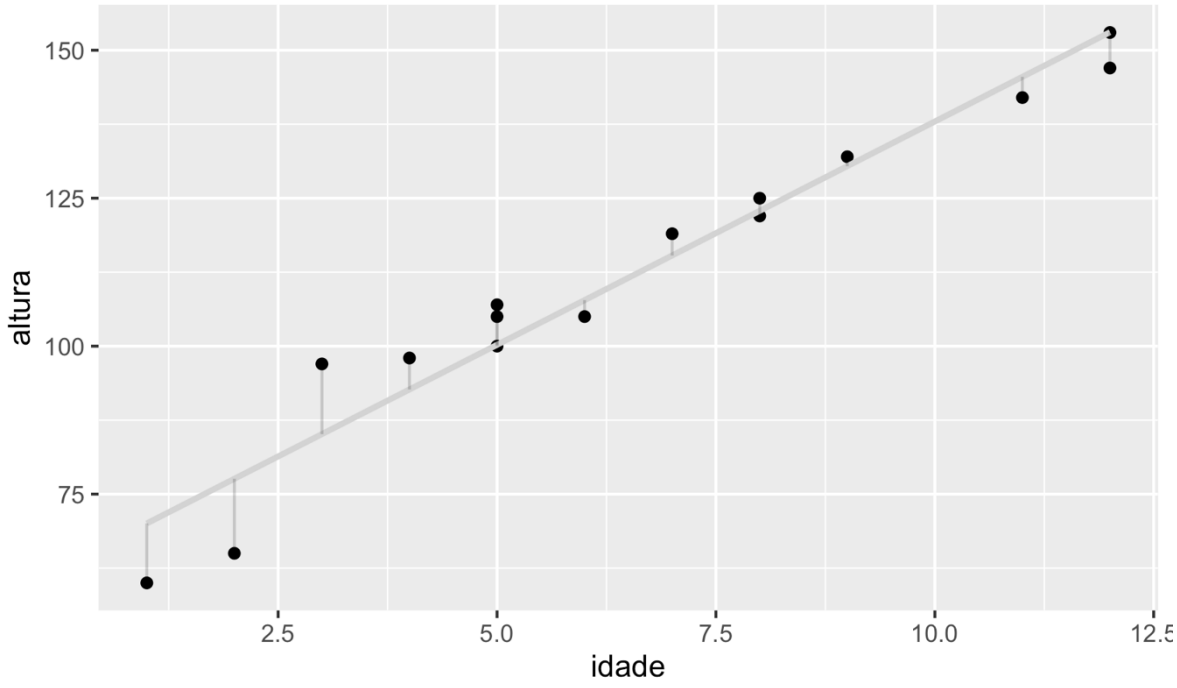

Figura 11.3: Gráfico de dispersão da altura por idade das crianças, com linha de regressão e diferenças entre valores observados e previstos. Fonte: própria.

Vamos ver os demais resultados da função linear por meio de `summary()`, para obter o resumo. Aplique essa função a `modelo`.

```
summary(modelo)
```

```
##
## Call:
## lm(formula = altura ~ idade, data = criancas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5946  -3.1391  -0.0477   4.2242  11.8601
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.5040     3.7691   16.58 3.98e-10 ***
## idade         7.5453     0.5135   14.69 1.78e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.651 on 13 degrees of freedom
## Multiple R-squared:  0.9432, Adjusted R-squared:  0.9388
## F-statistic: 215.9 on 1 and 13 DF,  p-value: 1.782e-09
```

O resumo do modelo traz, em primeiro lugar, a fórmula que usamos na função `lm()` – altura ~ idade. Em seguida, traz uma visão geral dos resíduos: o valor mínimo, o primeiro quartil (1Q), a mediana, o terceiro quartil (3Q) e o valor máximo. Vimos esses termos da Lição 7, quando falamos de boxplots. Ao inspecionar os resíduos, verifique se o valor da mediana está próximo de zero e se os valores min-max e 1Q-3Q são razoavelmente simétricos. Em nosso `modelo`, estamos dentro dessa expectativa, pois a mediana (-0,05) está bem próxima de zero, os valores mínimo (-12,6) e máximo (11,8) são semelhantes em valores absolutos, assim como os valores 1Q (-3,14) e 3Q (4,22). Colocado em outros termos, nossos resíduos seguem uma distribuição normal.

Por que esperamos simetria em torno de zero para os resíduos? Os resíduos são a diferença entre os valores observados e os valores previstos pelo modelo, certo? Visto de outro modo, os resíduos são os valores que nosso modelo não foi capaz de prever perfeitamente – o que já é esperado, pois, como visto, modelos não são a realidade. Contudo, se nosso modelo "erra", é bom que ele "erre" tanto para mais quanto para menos, pois daí podemos ter certa segurança de que nossas estimativas não estão muito longe do que se poderia observar. Veja que, em nossos dados, não temos nenhuma observação para uma criança com 10 anos; no entanto, esperamos que a estimativa do modelo – i.e., y = 62,5 + (7,5 * 10) = 137,5 cm – não esteja tão longe da realidade.

Em seguida, o resumo mostra os coeficientes em uma tabela. As estimativas, na primeira coluna, já são suas conhecidas. O valor 'Intercept' é o coeficiente linear, ou seja, o valor de y quando x é zero. O valor na segunda linha é a estimativa para a idade, o coeficiente angular, que é quanto muda o valor do eixo y a cada unidade de x.

A segunda coluna se refere ao erro padrão da estimativa (ver Lições 6 e 10). Esta é uma medida da precisão das previsões: quanto menor esse valor, maior é o grau de precisão do modelo.

O valor-*t*, na terceira coluna, é computado pela divisão Estimate / Std.Error. Faça essa conta: divida o valor da estimativa do coeficiente linear, 62.5040, por seu erro padrão, 3.7691.

```
62.504 / 3.7691
```

```
## [1] 16.58327
```

Exatamente o valor-*t* calculado, certo? E faça a divisão do valor da estimativa do coeficiente angular, 7.5453, por seu erro padrão, 0.5135.

```
7.5453 / 0.5135
```

```
## [1] 14.69387
```

Como vimos na Lição 10, sobre teste-t, o valor-*t* pode ser consultado numa tabela de distribuição t para determinar a significância. Em nosso modelo, ambas as estimativas têm valor de significância $p < 0{,}05$. Mas, para poder interpretar esse valor de significância, precisamos saber qual era a hipótese alternativa e a hipótese nula sob teste!

O modelo linear testa a hipótese nula de que a estimativa tem valor zero. Em nosso modelo, 62,5 cm é bastante diferente de zero e, levando em conta o erro padrão, o valor-*p* reflete isso. Mas peraí! 62,5 cm é a estimativa do tamanho das crianças quando elas nascem. Você esperaria que um bebê nascesse com 0 centímetros? Claro que não! Você nunca levantaria essa hipótese! Isso significa que esse valor de significância, para nós, não quer dizer *na-da*!

É importante reforçar este ponto: quem faz os testes e os interpreta é o pesquisador. O R simplesmente gera os valores que está programado para gerar. No modelo linear, ele vai gerar valores de significância para avaliar o quanto cada estimativa difere de zero. Mas cabe a você saber o que é verdadeiramente relevante ou não!

Vamos para a próxima estimativa e sua significância. Aqui, o R avaliou a hipótese nula de que o coeficiente angular – 7,5 cm – difere de zero. Pare um pouco para pensar o que significaria um coeficiente zero neste modelo: isso implicaria que as crianças não crescem, pois a cada ano teriam a mesma altura; isso também implicaria que altura e idade não se correlacionam, pois a altura seria a mesma não importa a idade da criança, e não seria possível estimar o valor de uma variável a partir de outra. O valor de

significância, aqui, mostra que há uma correlação significativa entre idade e altura. Esta sim é uma medida relevante para nós.

Logo abaixo da tabela de coeficientes, o R mostra o significado dos asteriscos, segundo os níveis α mais comuns: 0,001, 0,01 e 0,05. Três asteriscos indicam $p < 0{,}001$, dois asteriscos indicam p entre 0,001 e 0,01, e um asterisco indica p entre 0,01 e 0,05. O ponto final indica um valor pouco acima de 0,05. Como já vimos na Lição 9, os valores de significância são sensíveis ao tamanho da amostra. Um valor pouco acima de 0,05 pode indicar, por exemplo, que o resultado do teste pode mudar se o pesquisador obtiver mais dados ou, ainda, se o pesquisador "limpar" os dados de valores atípicos, que podem estar causando ruído na distribuição.

Ao pé do resultado, o R mostra o valor do erro padrão dos resíduos de acordo com os graus de liberdade (cf. `sqrt(sum(modelo$residuals^2)/13)`); um valor de zero aqui significaria que o modelo prevê as observações perfeitamente (o que quase nunca é o caso).

O R também fornece dois valores chamados R^2 "R ao quadrado", o segundo dos quais "ajustado". Sua interpretação é o quanto de variabilidade na variável dependente é explicada pelas variáveis incluídas no modelo; aqui, temos que a idade explica cerca de 94% da variação em altura. Este valor nada mais é do que o valor de r de Pearson elevado ao quadrado. Você se lembra do r de Pearson calculado acima, para a mesma correlação entre idade e altura? Ele também pode ser acessado por meio de `cor.test(criancas$idade, criancas$altura)$estimate`. Faça isso agora.

```
cor.test(criancas$idade, criancas$altura)$estimate
```

```
##       cor
## 0.9711854
```

Agora calcule o quadrado da expressão acima. Copie o último comando e adicione ^ 2 à expressão acima para calcular o R^2.

```
cor.test(criancas$idade, criancas$altura)$estimate ^ 2
```

```
##      cor
## 0.943201
```

Este é o valor de R^2 do modelo. O valor de R^2 ajustado é um ajuste de R^2 a depender do número de variáveis independentes incluídas no modelo (aqui, apenas uma) e o tamanho da amostra (aqui, 15). A fórmula de R^2 ajustado está no *script* do Anexo A.

A estatística-F, por fim, é uma medida útil quando se tem interesse em comparar diferentes modelos estatísticos, a fim de determinar qual deles mais bem explica a variação na variável dependente. Ela é usada para calcular um valor de significância do modelo como um todo, para além da significância de variáveis independentes específicas.

Que tal agora aplicar esse conhecimento a fenômenos linguísticos? Rode a linha de comando a seguir para carregar o dataframe `cov`, com os dados da planilha Covariaveis.csv. Para tanto, defina como diretório de trabalho aquele que contém essa planilha em seu computador.

```
# Definir diretório de trabalho

#setwd()

# Carregar dados

cov <- read_csv("Covariaveis.csv")
```

Veja a estrutura do dataframe `cov`.

```
str(cov)

## spec_tbl_df [118 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PARTICIPANTE : chr [1:118] "AdolfoF" "AdrianaP" "AmandaA" "Amara
lM" ...
##  $ SEXO.GENERO  : chr [1:118] "masculino" "feminino" "feminino" "ma
sculino" ...
##  $ IDADE        : num [1:118] 21 24 20 60 73 32 45 61 70 63 ...
##  $ FAIXA.ETARIA : chr [1:118] "1a" "1a" "1a" "3a" ...
##  $ ESCOLARIDADE : chr [1:118] "EnsSuperior" "EnsSuperior" "EnsSuper
ior" "EnsMedio" ...
##  $ REGIAO       : chr [1:118] "central" "central" "periferica" "per
iferica" ...
##  $ ZONA         : chr [1:118] "centro" "centro" "leste" "norte" ...
##  $ CLASSE.SOCIAL: chr [1:118] "b2" "c1" "c1" "c2-d" ...
##  $ ORIGEM.PAIS  : chr [1:118] "interior" "mista" "paulistanos" "int
erior" ...
##  $ EN           : num [1:118] 16 16 58 6 48 58 68 62 6 14 ...
##  $ RT           : num [1:118] 44 4 34 80 18 54 28 20 42 12 ...
##  $ R0           : num [1:118] 47 38 53 61 55 52 39 49 46 59 ...
##  $ CN           : num [1:118] 45 6 7 12 1 15 4 0 17 7 ...
##  $ CV3PP        : num [1:118] 8 8 19 14 7 18 8 5 24 23 ...
##  $ CV1PP        : num [1:118] 0 0 0 0 0 0 0 0 40 0 ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   PARTICIPANTE = col_character(),
##   ..   SEXO.GENERO = col_character(),
##   ..   IDADE = col_double(),
##   ..   FAIXA.ETARIA = col_character(),
##   ..   ESCOLARIDADE = col_character(),
##   ..   REGIAO = col_character(),
##   ..   ZONA = col_character(),
##   ..   CLASSE.SOCIAL = col_character(),
##   ..   ORIGEM.PAIS = col_character(),
##   ..   EN = col_double(),
##   ..   RT = col_double(),
##   ..   R0 = col_double(),
##   ..   CN = col_double(),
##   ..   CV3PP = col_double(),
##   ..   CV1PP = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Veja também a planilha de dados por meio da função `View()`.

```
View(cov)
```

N.B.: Resultado aqui omitido.

`cov` contém dados de 118 falantes paulistanos, com suas respectivas características sociais e proporções de emprego de variantes de seis variáveis sociolinguísticas: (i) ditongação de /e/ nasal [~ej], como em "fazenda" (por oposição à variante monotongada [~e]; (ii) realização de /r/ em coda, como em "mulher", como retroflexo (por oposição ao tepe); (iii) apagamento de /r/ em coda (por oposição à sua realização como tepe ou retroflexo); (iv) concordância nominal não padrão, como em "os menino" (por oposição à variante padrão); (v) concordância de 3PP não padrão, como "eles foi" (por oposição à variante padrão); e (vi) concordância de 1PP não padrão, como "nós vai" (por oposição à variante padrão). Embora sejam todas variáveis nominais, elas estão aqui representadas pelas proporções de uso de uma das variantes, de modo que se tornaram numéricas.

Esses dados foram assim organizados para investigar se certas variáveis sociolinguísticas "andam juntas", ou seja, se falantes que tendem a empregar a variante x de uma variável A também tendem a empregar a variante y de uma variável B. Por exemplo: falantes que tendem a não fazer a concordância nominal padrão também

tendem a fazer a concordância não padrão de 3PP? (ver Oushiro, 2016) Visualize a distribuição desses dados no gráfico de dispersão (Figura 11.4).

```
ggplot(cov, aes(x = CV3PP, y = CN)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey")
```

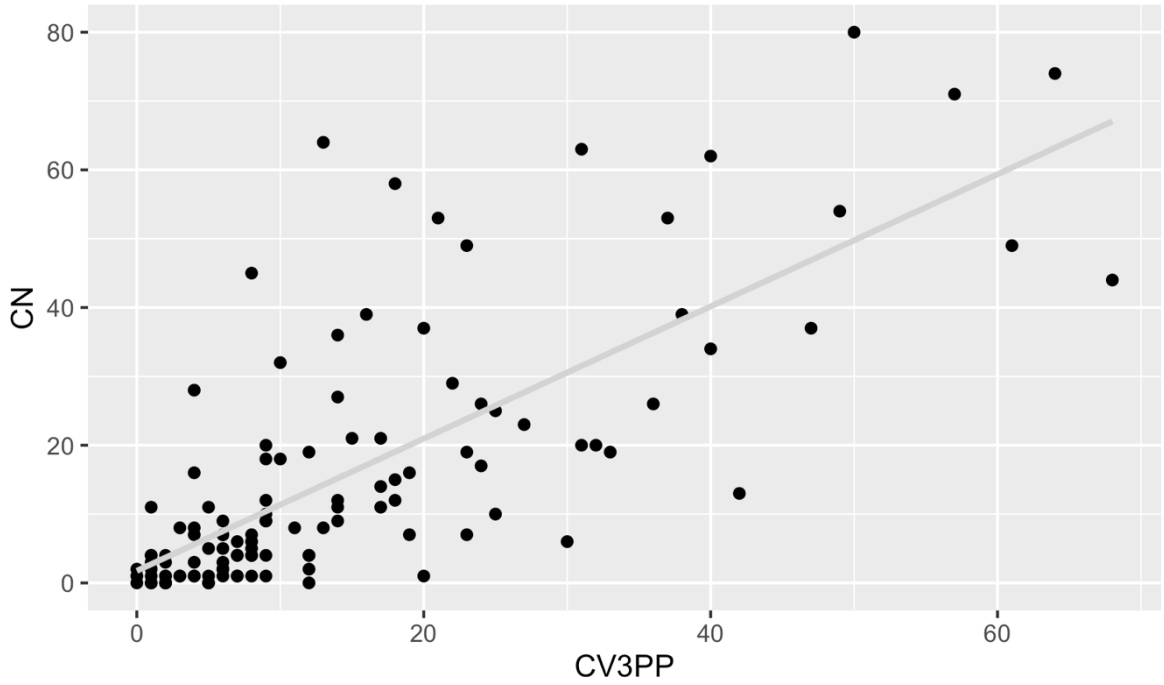

Figura 11.4: Gráfico de dispersão das proporções de uso de concordância verbal não padrão e de concordância nominal não padrão na amostra de Oushiro (2016). Fonte: própria.

Faça um teste de Shapiro para testar se a distribuição de dados de `cov$CV3PP` segue a distribuição normal.

```
shapiro.test(cov$CV3PP)

##
##  Shapiro-Wilk normality test
##
## data:  cov$CV3PP
## W = 0.81503, p-value = 7.137e-11
```

Faça um teste de Shapiro para testar se a distribuição de dados de `cov$CN` segue a distribuição normal.

```
shapiro.test(cov$CN)

##
##  Shapiro-Wilk normality test
```

```
##
## data:  cov$CN
## W = 0.78386, p-value = 6.832e-12
```

Como a distribuição de ambos não é normal, faça um teste de correlação com a adição do argumento method = "spearman".

```
cor.test(cov$CV3PP, cov$CN, method = "spearman")

##
##  Spearman's rank correlation rho
##
## data:  cov$CV3PP and cov$CN
## S = 66236, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7581044
```

A que conclusão o pesquisador pode chegar a partir do teste acima?

- existe correlação entre o uso de CN e CV3PP
- não existe correlação entre o uso de CN e CV3PP

O teste indica que existe correlação entre o uso de CN e CV3PP, com $\rho = 0{,}75$ e $p < 0{,}001$.

Crie um modelo linear, chamado `modelo1`, com a fórmula CN ~ CV3PP como primeiro argumento, e data = cov como segundo argumento.

```
modelo1 <- lm(CN ~ CV3PP, data = cov)
```

Acima, colocamos CN ~ CV3PP, o que assume que a proporção de emprego de CN não padrão depende da proporção de emprego de CV3PP não padrão. No entanto, isso não necessariamente é verdade. Fizemos isso por exigência do formato de fórmula para o primeiro argumento de `lm()`.

Visualize o resultado de `modelo1` por meio de `summary()`.

```
summary(modelo1)

##
## Call:
## lm(formula = CN ~ CV3PP, data = cov)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.081  -6.188  -2.716   2.203  49.763
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.75595    1.64045    1.07    0.287
## CV3PP        0.96011    0.07913   12.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.56 on 116 degrees of freedom
## Multiple R-squared:  0.5593, Adjusted R-squared:  0.5555
## F-statistic: 147.2 on 1 and 116 DF,  p-value: < 2.2e-16
```

A que conclusão o pesquisador pode chegar a partir do teste acima?

- existe correlação entre o uso de CN e CV3PP
- não existe correlação entre o uso de CN e CV3PP

O `modelo1` prevê que, quando a proporção de emprego de CV3PP não padrão é zero, a proporção de CN não padrão é 1,8%, e que a cada 1% de aumento da proporção de emprego de CV3PP não padrão, a proporção de CN não padrão também aumenta em cerca de 1% (os valores das estimativas). Também vemos que os resíduos não se distribuem de modo muito simétrico; compare, por exemplo, o valor mínimo (-29) ao valor máximo (49). Isso pode ser visto no gráfico: diferentemente do caso da correlação entre idade e altura das crianças, aqui os pontos se dispersam bem mais e alguns deles se distanciam mais da linha de regressão – sobretudo para cima. O valor máximo de 49 é a maior distância entre o valor observado e o valor previsto pelo modelo.

Podemos repetir o teste acima para cada par de variáveis, o que daria um total de 15 testes (EN-RT, EN-R0, EN-CN etc.). Uma função muito útil para calcular uma série de correlações entre pares de variáveis é `ggpairs()`, do pacote GGally, que carregamos no início da lição.

A função `ggpairs()` toma como argumento o dataframe. É possível também definir um subconjunto de colunas, o que, em nosso caso, vai ser necessário, pois nem todas as variáveis do dataframe `cov` são numéricas. As proporções de uso das variantes /~e/ ditongado, /r/ retroflexo, apagamento de /r/, CN não padrão, CV3PP não padrão e CV1PP não padrão se encontram nas colunas 10 a 15 (Figura 11.5).

```
ggpairs(cov, columns = 10:15)
```

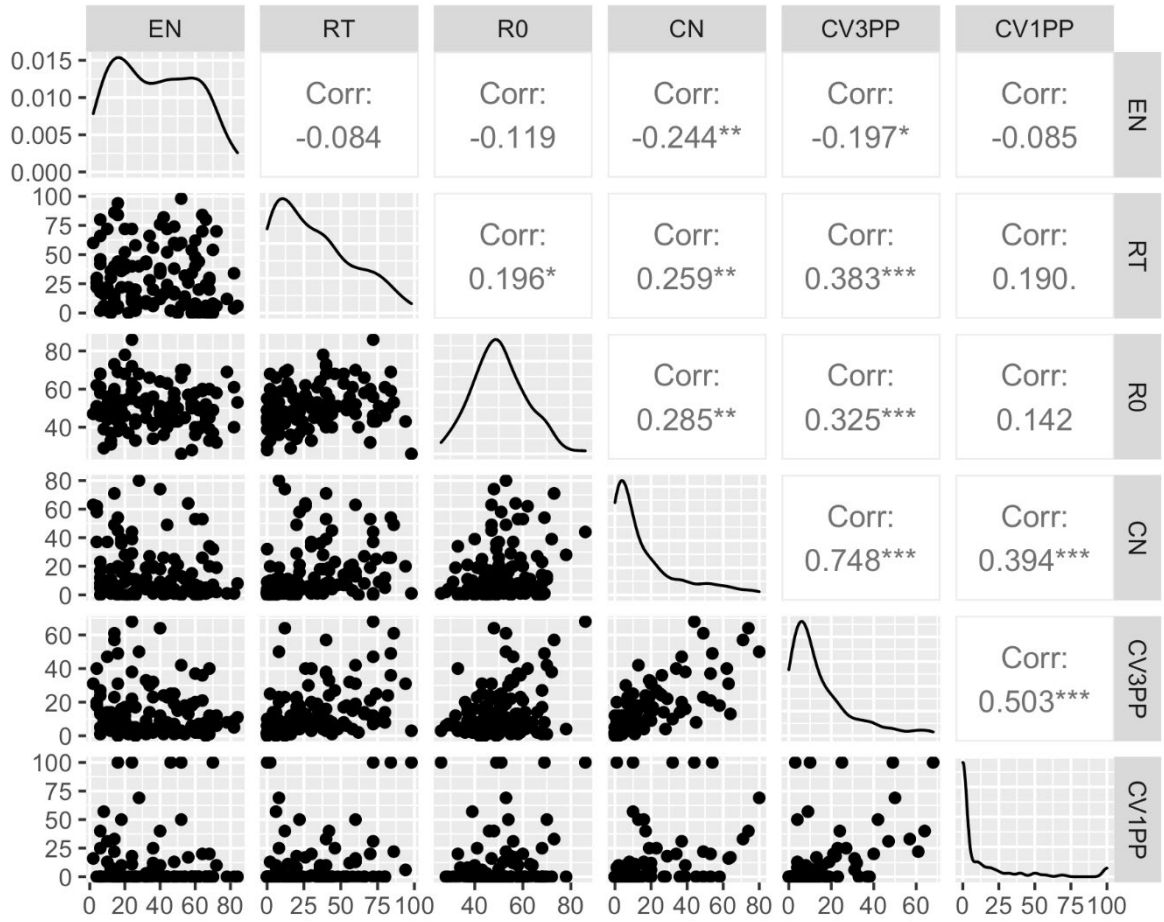

Figura 11.5: Distribuição dos dados de ditongação de /e/ nasal, /r/ retroflexo, apagamento de /r/, CN não padrão, CV3PP não padrão e CV1PP não padrão nos dados de Oushiro (2016). Fonte: própria.

A figura mostra, na linha diagonal, o gráfico de densidade das proporções das variantes indicadas nas colunas/linhas. Vemos aí claramente que a maioria das variáveis não tem uma distribuição normal. Na parte superior, a função calculou o r de Pearson, com os respectivos valores de significância, para cada par de variáveis – valores que se encontram no cruzamento das respectivas linhas. Na parte inferior, visualizam-se os gráficos de dispersão.

Da figura, qual par de variáveis apresenta a correlação mais forte?

- CN e CV3PP
- CV3PP e CV1PP
- EN e CN

- RT e R0

Da figura, qual dos pares a seguir não apresenta uma correlação significativa?

- CV3PP e CV1PP
- EN e R0
- R0 e CN
- R0 e CV3PP

Da figura, a distribuição de qual das variáveis mais se aproxima da distribuição normal?

- EN
- CN
- CV1PP
- R0

Com a função `ggpairs()`, também é possível adicionar uma variável fatorial para visualizar a distribuição para diferentes grupos. A partir da última linha de comando, acrescente o argumento `ggplot2::aes(color = REGIAO))` para definir as cores do gráfico.

```
ggpairs(cov, columns = 10:15, ggplot2::aes(color = REGIAO))
```

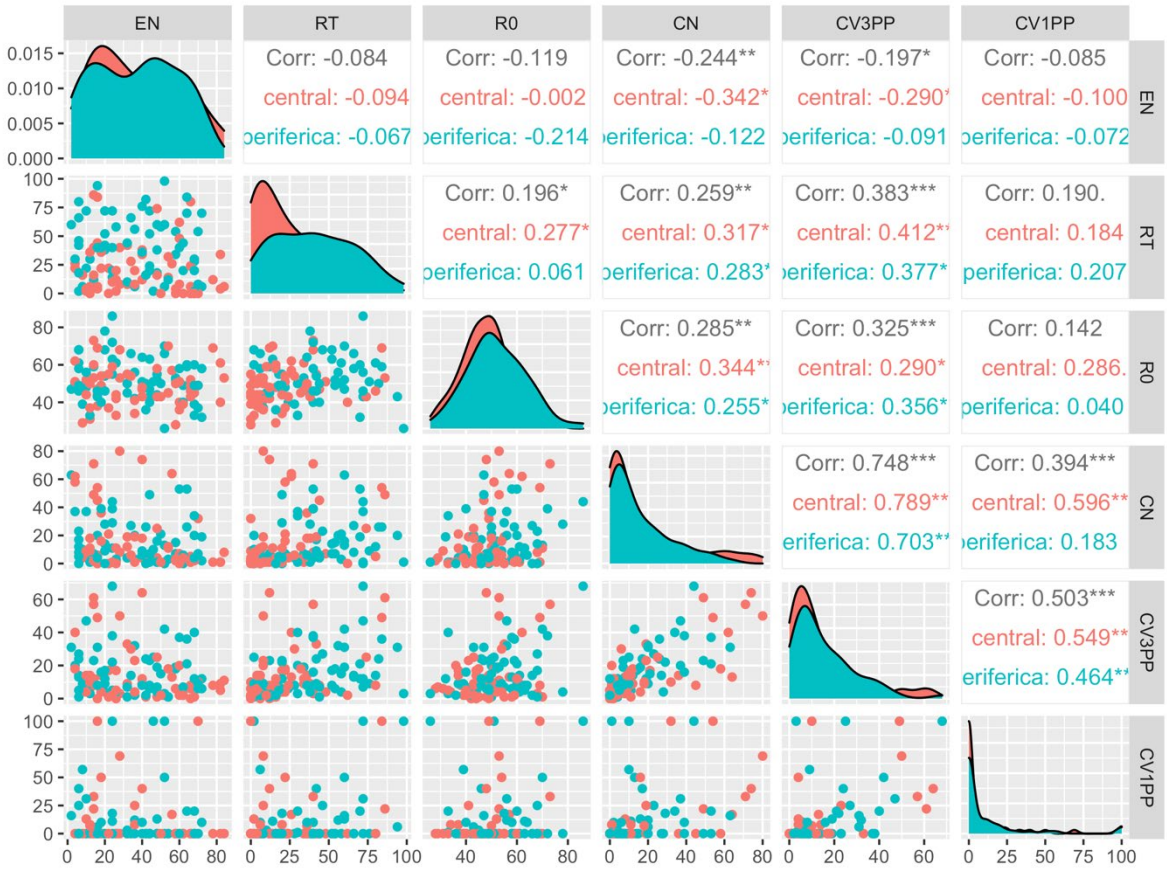

Figura 11.6: Distribuição dos dados de ditongação de /e/ nasal, /r/ retroflexo, apagamento de /r/, CN não padrão, CV3PP não padrão e CV1PP não padrão nos dados de Oushiro (2016), com falantes separados por região de residência. Fonte: própria.

A função plotou agora os gráficos de dispersão, de densidade e os valores de r de Pearson para central e periferica, o que permite comparar o comportamento desses dois grupos de falantes. Dica: clique em Zoom para ter uma melhor visualização dos dados.

Ao reportar testes de correlação, informe o valor de r de Pearson ou de Spearman, os graus de liberdade e o valor-*p*. Em modelos lineares, as medidas estatísticas do `summary()` são relevantes. Na próxima lição, veremos com mais detalhes como reportar esses dados.

Após tantos testes de correlação e modelos lineares, cabe ressaltar um mantra da Estatística: *correlação não é sinônimo de motivação*! Ainda que você tenha encontrado uma correlação significativa, seja por meio do teste de correlação, seja por meio de um modelo linear, isso não significa que y é motivado por x. A relação (ou não) entre duas

variáveis só pode ser explicada pelo pesquisador, que deve ser maximamente crítico quanto aos resultados. Um site que bem ilustra a afirmação acima é o Spurious Correlations (http://tylervigen.com/old-version.html), que traz uma série de correlações absurdas.

Os resultados de testes estatísticos não provam nada. Eles são apenas uma ferramenta auxiliar do pesquisador para explicar os fenômenos sob análise, mas todo o processo – desde o levantamento de hipóteses até seu teste e interpretação – deve ser guiado pelo bom senso.

Para saber mais

Recomendo a leitura do capítulo 6 de Dalgaard (2008), do capítulo 6 de Levshina (2015) e das páginas 138–146 de Gries (2019) para fixar o conteúdo visto aqui. No *script*, deixei um código para fazer gráficos de pares por meio do pacote languageR (Baayen, 2008).

Exercícios

Observe os gráficos da Figura 11.7 para responder as questões.

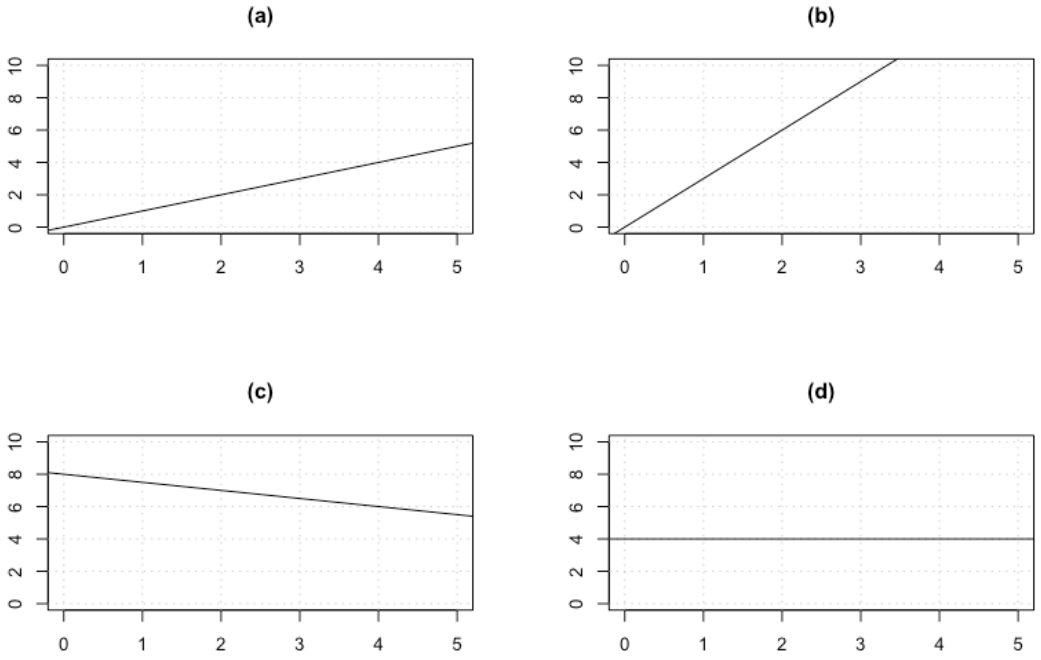

Figura 11.7: Gráficos para leitura de coeficientes linear e angular. Fonte: própria.

1. Quais são os valores dos coeficientes linear e angular na figura (a)?
    a. 0, 0
    b. 0, 1
    c. 0, 2
    d. 4, 2
2. Qual é o coeficiente angular da figura (b)?
3. Qual é o coeficiente angular da figura (c)?
4. Qual é o coeficiente linear da figura (d)?
5. Carregue o pacote tidyverse.
6. As próximas questões se baseiam nos dados de vogais médias pretônicas. Defina como diretório de trabalho a pasta que, em seu computador, tem o arquivo Pretonicas.csv.
7. Carregue os dados num dataframe chamado `pretonicas`. Defina a variável VOGAL como factor.

8. Cheque a estrutura do dataframe `pretonicas`.
9. Crie um subconjunto de dados da VOGAL "o" da AMOSTRA "PBSP", e guarde-o num dataframe chamado `PBSP_o`.
10. O abaixamento de vogais médias pretônicas, em palavras como "relógio" e "romã", é descrito como um fenômeno de harmonia vocálica: a vogal pretônica abaixa quando a vogal da sílaba seguinte é baixa [ɛ, a, ɔ]. Você tem interesse em testar a hipótese de que a altura da vogal /o/, de acordo com a medida de F1 normalizada (a variável F1.NORM), correlaciona-se com a altura da vogal da sílaba seguinte (a variável F1.SEG.NORM). O primeiro passo é plotar o gráfico de dispersão. Qual das duas variáveis deve ser colocada no eixo x (i.e., qual é a variável independente)? Justifique sua resposta.
    a. F1.SEG.NORM
    b. F1.NORM
    c. tanto faz!
11. Plote um gráfico simples de dispersão das medidas de F1.NORM e F1.SEG.NORM da vogal /o/ na fala de paraibanos com os eixos x e y devidamente especificados. Adicione a função `geom_smooth()`, com argumento method = "lm", para já plotar uma linha de regressão.
12. Você quer fazer um teste de correlação. Para decidir qual teste aplicar, o que deve fazer antes?
    a. aplicar o teste de Shapiro para verificar se as variáveis seguem a distribuição normal
    b. aplicar o teste-t para ver se as variáveis podem ser correlacionadas
    c. aplicar o teste de correlação de Pearson para determinar o r
13. Aplique o teste de Shapiro no vetor com as medidas de F1.NORM da vogal /o/.
14. Aplique o teste de Shapiro no vetor com as medidas de F1.SEG.NORM da vogal /o/.
15. Pelo resultado dos testes de Shapiro acima, qual teste é mais adequado para se aplicar aos dados?

a. teste-t
b. teste de qui-quadrado
c. teste de correlação de Spearman
d. teste de correlação de Pearson

16. Aplique o teste identificado na questão 15 para testar a hipótese de que as medidas de F1 da vogal pretônica e da vogal da sílaba seguinte se correlacionam.
17. A que conclusão o pesquisador pode chegar a partir do teste acima?
    a. não há correlação significativa entre F1.NORM e F1.SEG.NORM, com r = 0,27.
    b. há correlação significativa entre F1.NORM e F1.SEG.NORM, com r = 0,27.
    c. não há correlação significativa entre F1.NORM e F1.SEG.NORM, com r = 4,07.
    d. há correlação significativa entre F1.NORM e F1.SEG.NORM, com r = 4,69.
18. Crie um modelo linear, chamado `modelo`, para testar se há correlação entre F1.NORM e F1.SEG.NORM nos dados da vogal /o/ na fala de paraibanos.
19. Visualize o resumo dos resultados do modelo gerado.
20. De acordo com os resultados do modelo, qual afirmação seguinte não é verdadeira?
    a. A distribuição dos resíduos, neste modelo, não segue uma distribuição normal.
    b. A variável F1.SEG.NORM explica 6,7% da variação nos valores de F1.NORM da vogal /o/.
    c. Quando F1.SEG.NORM é igual a zero, a estimativa do valor de F1.NORM é 355,7 Hz.
    d. A cada unidade de F1.SEG.NORM, o valor de F1.NORM cresce 0,23 Hz.
    e. A significância da estimativa de F1.SEG.NORM está entre 0,01 e 0,05.
    f. A variável F1.SEG.NORM se correlaciona significativamente com a variável F1.NORM.
21. Qual é o valor do coeficiente linear nesse modelo?

22. Qual é o valor do coeficiente angular nesse modelo?
23. O que significa o valor-p = 4.08e-06 para F1.SEG.NORM?
    a. é baixa a probabilidade de que a verdadeira estimativa seja zero
    b. é baixa a probabilidade de que a diferença seja significativa
    c. é baixa a probabilidade que F1.SEG.NORM correlacione-se com F1.NORM
24. Aplique o teste de Shapiro aos resíduos do modelo para verificar a normalidade ou não da distribuição.
25. O que informa o teste de Shapiro sobre os resíduos do modelo?
    a. a distribuição não é normal
    b. a distribuição é normal