Apresentação de Resultados

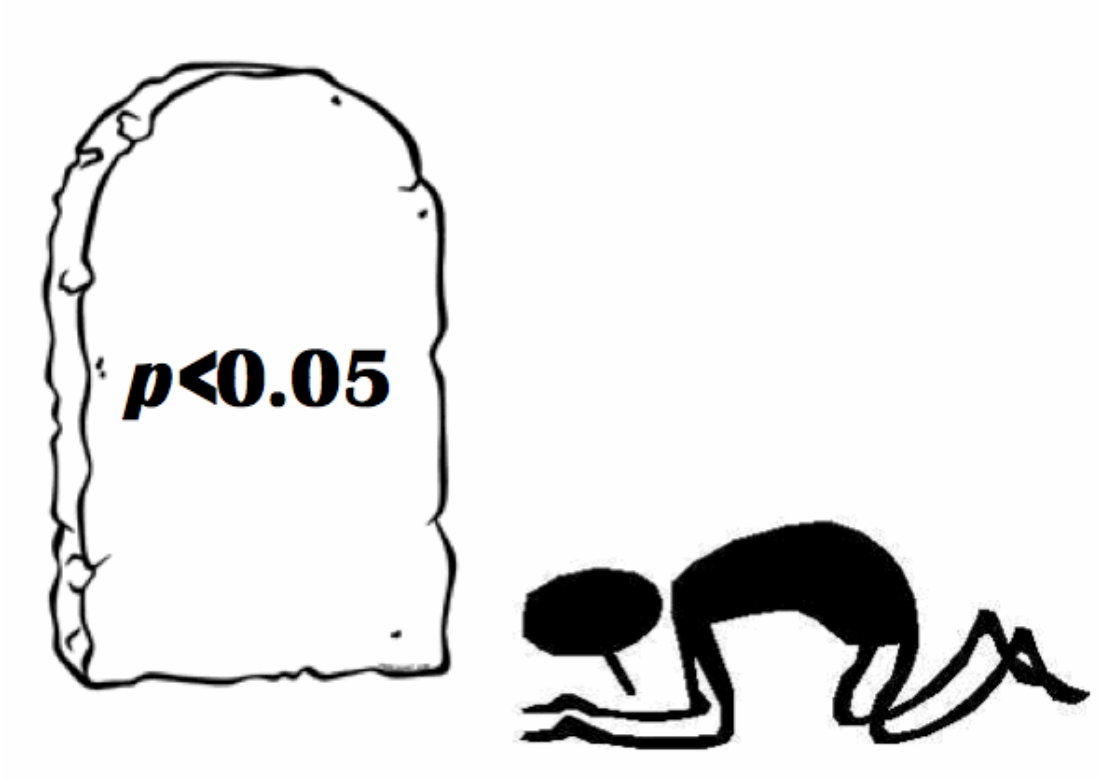
Manejo, Visualização e Compartilhamento de Dados

Nicholas A. C. Marino
github.com/nacmarino/compartilhaR

Estrutura da Aula

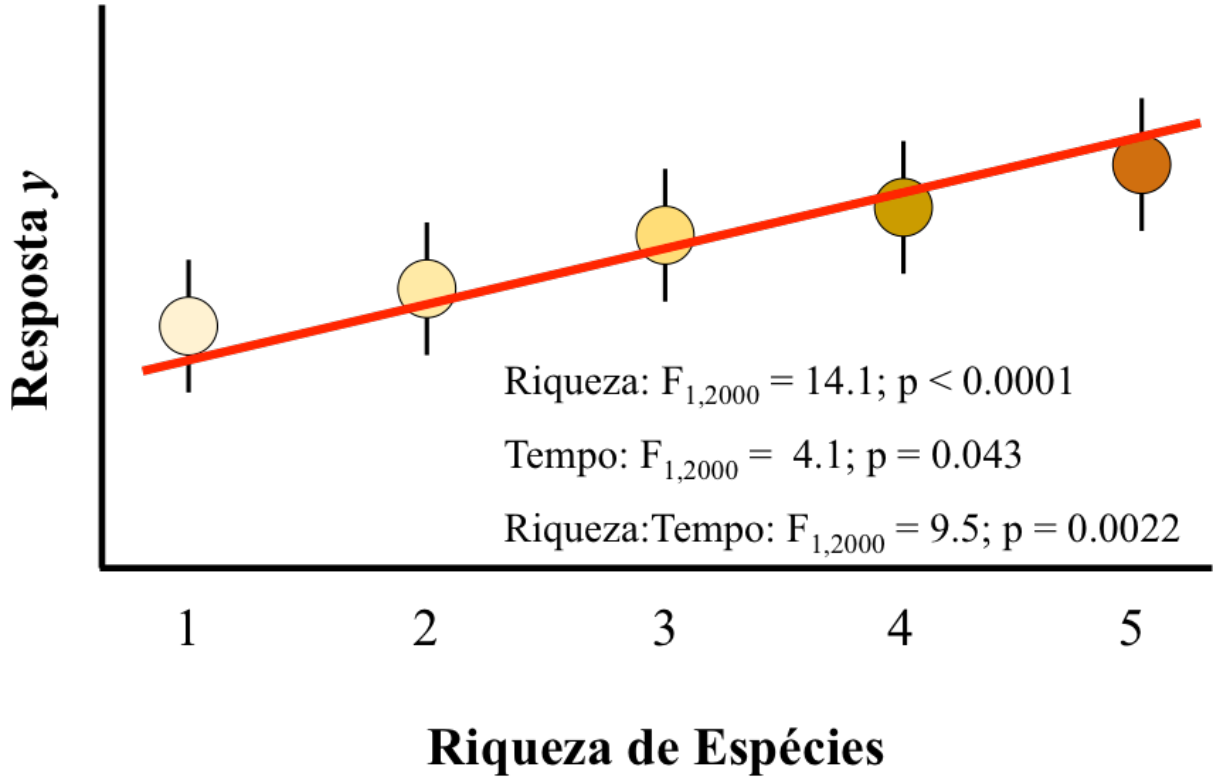
1. O significado de um modelo estatístico
2. Extraindo os parâmetros e predições de um modelo estatístico
3. Visualização dos resultados de um modelo estatístico

Amamos valores de *p* e resultados significativos…



…mas olhamos muito pouco para o que esses resultados significam e como eles devem ser apresentados

- Como você interpreta essa figura? Ela está boa ou existe algo de estranho e/ou errado nela?

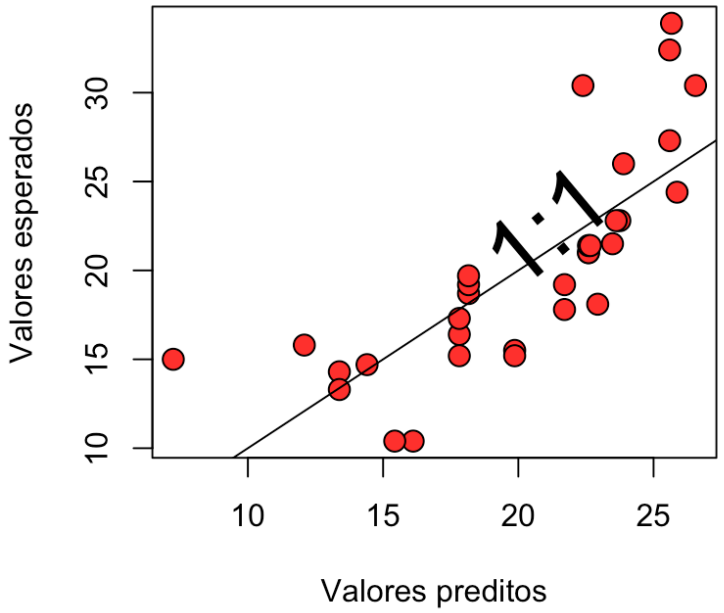


Alguns problemas comuns ao apresentar resultados em figuras e tabelas

1. Valores de alguns parâmetros dos testes estatísticos não batem com a forma como o dado é apresentado (*e.g.*, medida categórica, graus de liberdade contínuos);
2. Apresentado gráfica do resutlado não bate com a análise de dados (*e.g.*, variável preditora é contínua, mas figura o apresentada como uma categoria);
3. Resultados gráficos não consideram efeitos de interações e outras variáveis que descrevem a estrutura do modelo (*e.g.*, auto-correlação e não-independência);
4. Métricas descritivas de tabelas e figuras não batem com aquelas vindas dos modelos (*e.g.*, cálculo da média simples quando existem efeitos mistos).

O que queremos mostrar?

- Existem muitos objetivos implícitos ao utilizarmos uma análise estatísticas, mas dois deles que se destacam no contexto desta aula são:
  - Estimar os parâmetros que melhor caracterizem a população e/ou sub-populações estudadas;
  - Desenvolver um modelo que seja capaz de predizer novos valores da variável resposta `y` uma vez que se forneçam novos valores de `x` (x1, x2, x3,…).
- Este último objetivo é particularmente marcante e, inclusive, é um dos critérios de validação de um modelo - ao plotarmos os valores observados *vs* os valores preditos pelo modelo.



O que queremos mostrar?

- Sempre que apresentamos graficamente ou em uma tabela os resultados de um modelo, devemos apresentar:
  1. os valores dos parâmetros estimados pelo modelo; e/ou,
  2. de que forma o modelo prediz que `y` e `x` estão relacionados considerando a estrutura geral do modelo (_i.e._outras covariáveis, não-independência, auto-correlação, heterogeneidade nas variâncias, efeitos mistos,…).

- Como interpretar os resultados do modelo abaixo?

```
library(tidyverse)
library(readxl)
library(emmeans)
ilhas <- read_excel(path = "../99 - dados para exemplos/ilhas.xlsx", na = "NA")
modelo1 <- lm(log(riqueza) ~ ilha, data = ilhas)
emmeans(object = modelo1, specs = "ilha")
```

O que queremos mostrar?

```
emmeans(object = modelo1, specs = "ilha")
```

```
##  ilha       emmean         SE df lower.CL upper.CL
##  costeira 3.322795 0.09812881 78 3.127436 3.518155
##  oceanica 2.372455 0.09812881 78 2.177096 2.567814
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

O que queremos mostrar?

- E o que este resultado do modelo sugere?

```
coef(modelo1)
```

```
##  (Intercept) ilhaoceanica
##    3.3227951   -0.9503401
```

O que queremos mostrar?

- Existem duas formas de ver a mesma coisa no *output* de um modelo:
  - A estimativa dos parâmetros que descrevem cada nível de cada variável: ilhas costeiras possuem, em média, 3.32 espécies, enquanto ilhas oceanicas possuem 2.37 espécies em média (ambas na escala log).
  - O modelo que prediz o valor da riqueza de espécies em uma ilha, dependendo da sua distância com relação ao continente é: $y = \beta_0 + \beta_1 x_1 = \beta_{costeira} + \beta_{oceanica} x_{oceanica} = 3.32 - 0.95 x_{oceanica}$.

Como interpretar o exemplo abaixo?

```
modelo2 <- lm(log(riqueza) ~ log(area) * ilha, data = ilhas)
coef(modelo2)
```

```
##            (Intercept)             log(area)          ilhaoceanica
##             3.07289602             0.09202304             -1.34772797
## log(area):ilhaoceanica
##             0.09847926
```

Como interpretar o exemplo abaixo?

- log(riqueza) = $\beta_{costeira}$ + β_{area} x_{area} + $\beta_{oceanica}$ $x_{oceanica}$ + $\beta_{area:oceanica}$ $x_{area}x_{oceanica}$
- log(riqueza) = 3.07 + 0.09 x_{area} - 1.35 $x_{oceanica}$ + 0.1 $x_{area}x_{oceanica}$

```
modelo2 <- lm(log(riqueza) ~ log(area) * ilha, data = ilhas)
coef(modelo2)
```

```
##            (Intercept)              log(area)          ilhaoceanica
##             3.07289602             0.09202304             -1.34772797
## log(area):ilhaoceanica
##             0.09847926
```

Exercício 1

- Com base no modelo abaixo, qual seria o valor predito para a riqueza de espécies de uma ilha oceanica com área de 2680.297 (valor não está na escala log)?
- E qual seria o valor predito para uma ilha costeira de mesma área?
- O que estes dois valores representam?

$$\log(\text{riqueza}) = 3.07 + 0.09\, x_{\text{area}} - 1.35\, x_{\text{oceanica}} + 0.1\, x_{\text{area}} x_{\text{oceanica}}$$

Então, como faríamos para mostrar…

- O valor esperado para a riqueza de espécies para ilhas costeiras e oceanicas, considerando o efeito da área nesta estimativa (conforme sugerido pelo modelo)?

```
emmeans(object = modelo2, specs = "ilha", by = "area")
```

```
## area = 2680.297:
##  ilha       emmean         SE df lower.CL upper.CL
##  costeira 3.799297 0.12269435 76 3.554930 4.043664
##  oceanica 3.228933 0.09919047 76 3.031378 3.426488
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

- Note, portanto, que os parametros estimados pelo modelo e o modelo utilizado para determinar a relação entre `y` e `x` (x1, x2, x3,…) caminham lado a lado, e ambos devem ser empregados para demonstrar numericamente e graficamente as relações estudadas.

Um outro exemplo, através do Exercício 2

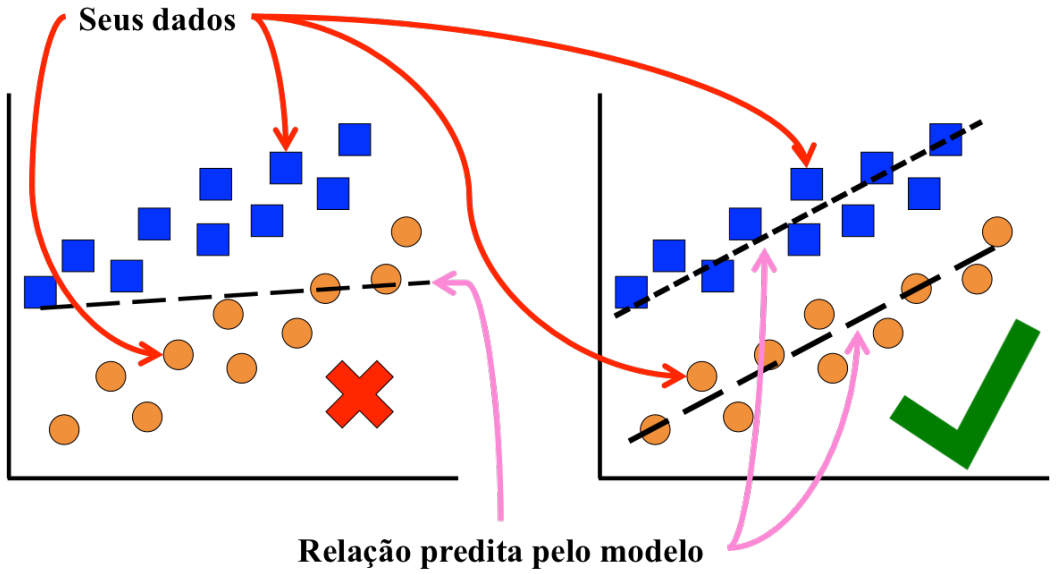
- Qual o valor de riqueza de espécies esperado para ilhas costeiras que tenham os seguintes tamanhos abaixo:
- Crie uma figura demonstrando esta relação.

```
data.frame(area = c(66810, 2420, 55850, 40840, 81110, 71020, 81420, 18520, 46440, 54000))
```

```
##     area
## 1  66810
## 2   2420
## 3  55850
## 4  40840
## 5  81110
## 6  71020
## 7  81420
## 8  18520
## 9  46440
## 10 54000
```

Qual a utilidade de um modelo estatístico, então?

- Um modelo estatístico pode (e deve) ser utilizado para predição, e não só para descrever a relação ou a significância da relação entre duas ou mais variáveis.
  - Predição restrita ao universo amostral que deu origem ao modelo;
  - Dentro deste universo, predição pode ser feita para qualquer novo valor das variáveis preditoras;
- Portanto, o que precisamos demonstrar na forma da tabela são os valores dos parâmetros estimados pelo modelo, e, graficamente, de que forma o modelo que você estabeleceu explica a relação entre as variáveis.



Extraindo os parâmetros e predições de um modelo estatístico

- Você tem duas opções principais para extrair as predições de um modelo e, em todas as duas, o ideal é que você consiga colocar estes valores dentro de um `data.frame` ou `tibble`, porque:
  - São classes de objetos que apresentam grande ferramental para a sua manipulação.
  - Facilita a exportação desses resultados para uso posterior (*e.g.*, criação de uma figura em um outro script, dados brutos para a criação de uma tabela de resultados).
- Então, o objetivo geral vai ser sempre colocar todo e qualquer resultado que nos interessa em uma daquelas duas classes de objeto e, caso necessário seja armazenar um outro conjunto de resultados ou um conjunto de resultados com estrutura diferente (e.g., número de dimensões), podemos armazenar estes múltiplos `data.frame` ou `tibble` em uma `list` e exportá-la no formato `.rds`.

Extraindo estimativas de parâmetros de categorias

- Como vimos, podemos utilizar o pacote `emmeans` para realizar diversos tipos de pós-teste para uma análise de dados.
- O objeto resultante das funções deste pacote não são nem um `data.frame` e nem um `tibble`, mas podemos usar a coerção para fazer isso.

```
class(emmeans(object = modelo2, specs = "ilha", by = "area"))
```

```
## [1] "emmGrid"
## attr(,"package")
## [1] "emmeans"
```

Extraindo estimativas de parâmetros de categorias

```
data.frame(emmeans(object = modelo2, specs = "ilha", by = "area"))
```

```
##       ilha     area   emmean         SE df lower.CL upper.CL
## 1 costeira 2680.297 3.799297 0.12269435 76 3.554930 4.043664
## 2 oceanica 2680.297 3.228933 0.09919047 76 3.031378 3.426488
```

```
class(data.frame(emmeans(object = modelo2, specs = "ilha", by = "area")))
```

```
## [1] "data.frame"
```

```
# vamos armazenar o resultado deste processamento em um objeto para uso posterior
diferencas_ilha <- data.frame(emmeans(object = modelo2, specs = "ilha", by = "area"))
```

Exerício 3

1. Crie um modelo onde você também incluirá o efeito do tamanho do arquipélago como uma variável preditora, em interação com todas as demais variáveis preditoras do `modelo2`.
2. Extraia o valor da estimativa dos parâmetros que descrevem a riqueza de espécies em cada um dos tipos de ilha (tamanho x distância), considerando o efeito do tamanho da ilha.

Extraindo predições vindas de variáveis contínuas

- Como vimos anteriormente, podemos estimar o valor esperado de uma variável resposta para cada valor de uma variável preditora contínua que tivermos - uma vez que o modelo estatístico tenha sido estabelecido.
- Muitas das funções que ajustam modelos no R possuem uma função companheira que serve para extrairmos as predições do modelo, a função `predict`.
- Essa função tem sempre alguns argumentos importantes, mas que podem variar de um tipo de modelo ao outro:
  - `object`: nome do objeto que comporta o modelo a partir do qual você quer extrair as predições;
  - `se.fit`: um argumento lógico que indica se você quer que os valores do erro associado a cada estimativa sejam apresentados (indisponível quando trabalhamos com modelos mistos);
  - `newdata`: um `data.frame` contendo os valores que você quer usar para gerar novas predições. Este `data.frame` deve conter colunas com as mesmas variáveis preditoras que você usou ao ajustar o modelo -

Extraindo predições vindas de variáveis contínuas

- No uso mais simples da função `predict` extraimos as predições para cada observação utilizando como *input* os mesmos valores utilizados para ajustar o modelo - isto é, o próprio `data.frame` que usamos no argumento `data`.

```
predict(object = modelo2)
```

```
##         1         2         3         4         5         6         7
## 3.1107403 3.3591310 3.4147562 3.1855771 3.7090043 2.4242250 3.4098415
##         8         9        10        11        12        13        14
## 2.0701253 2.8221200 3.4745708 2.5290203 3.1686538 3.4468751 2.7719651
##        15        16        17        18        19        20        21
## 2.4700845 3.0660693 2.1925117 3.7138670 3.5774598 1.6125216 2.4198936
##        22        23        24        25        26        27        28
## 1.6717559 3.5475339 0.9980755 3.3877297 3.0275752 1.5576257 2.1076351
##        29        30        31        32        33        34        35
## 2.2419372 1.9301209 3.4423748 3.3080863 3.3185639 2.0810901 3.2695506
##        36        37        38        39        40        41        42
## 4.0279326 3.2689297 3.4238288 2.2746249 2.4258953 2.8628274 2.5384782
##        43        44        45        46        47        48        49
## 3.6372807 3.6496577 3.5824192 3.0548469 2.3009938 2.9037372 2.9113264
##        50        51        52        53        54        55        56
## 2.6646507 3.0584803 3.4269642 1.9326762 3.1377065 3.3181979 2.8851255
##        57        58        59        60        61        62        63
## 1.9028449 3.3880951 3.1635830 1.2704287 2.9347014 3.3709655 3.2516050
##        64        65        66        67        68        69        70
## 3.2552826 3.4374856 1.2252579 3.1114028 3.1415387 3.6203270 3.9817362
```

Extraindo predições vindas de variáveis contínuas

- No entanto, também podemos fornecer um novo conjunto de dados a partir dos quais novas predições serão geradas.

```
predizer <- data.frame(area = seq(from = 1, to = 140000, by = 1), ilha = "costeira")
predizer
```

```
## # A tibble: 140,000 x 2
##     area ilha
##    <dbl> <fct>
##  1     1 costeira
##  2     2 costeira
##  3     3 costeira
##  4     4 costeira
##  5     5 costeira
##  6     6 costeira
##  7     7 costeira
##  8     8 costeira
##  9     9 costeira
## 10    10 costeira
## # ... with 139,990 more rows
```

Extraindo predições vindas de variáveis contínuas

- Uma vez que tenhamos criado esta nova tabela, podemos então usar a função `predict` alimentado o argumento `newdata` com ela.

```
predict(object = modelo2, newdata = predizer)
```

```
##        1        2        3        4        5        6        7        8
## 3.072896 3.136682 3.173994 3.200467 3.221001 3.237779 3.251965 3.264253
##        9       10       11       12       13       14       15       16
## 3.275091 3.284787 3.293558 3.301565 3.308930 3.315750 3.322099 3.328038
##       17       18       19       20       21       22       23       24
## 3.333617 3.338877 3.343852 3.348572 3.353062 3.357343 3.361434 3.365350
##       25       26       27       28       29       30       31       32
## 3.369107 3.372716 3.376189 3.379536 3.382765 3.385885 3.388902 3.391824
##       33       34       35       36       37       38       39       40
## 3.394655 3.397402 3.400070 3.402662 3.405184 3.407638 3.410028 3.412358
##       41       42       43       44       45       46       47       48
## 3.414630 3.416848 3.419013 3.421129 3.423197 3.425219 3.427198 3.429136
##       49       50       51       52       53       54       55       56
## 3.431033 3.432892 3.434715 3.436501 3.438254 3.439974 3.441663 3.443321
##       57       58       59       60
## 3.444950 3.446550 3.448123 3.449670
```

Extraindo predições vindas de variáveis contínuas

- Em todo o caso, o que você precisa fazer para poder ter o `data.frame` com o qual plotar os resultados é unir o resultado do `predict` à tabela que foi usada para dar origem a esses valores.

```
library(tidyverse)
cbind.data.frame(predizer, preditos = predict(object = modelo2, newdata = predizer)) %>%
  tbl_df
```

```
## # A tibble: 140,000 x 3
##     area ilha     preditos
##  * <dbl> <fct>       <dbl>
##  1     1 costeira     3.07
##  2     2 costeira     3.14
##  3     3 costeira     3.17
##  4     4 costeira     3.20
##  5     5 costeira     3.22
##  6     6 costeira     3.24
##  7     7 costeira     3.25
##  8     8 costeira     3.26
##  9     9 costeira     3.28
## 10    10 costeira     3.28
## # ... with 139,990 more rows
```

O pacote `broom`

- O pacote `broom` oferece algumas funções para tornar a extração dos resultados dos modelos bem atrativa, através de três funções:
  - `augment`: retorna uma tabela com os valores preditos pelo modelo, bem como diversas métricas relativas à influência de cada observação para as estimativas feitas pelo modelo.
  - `glance`: retorna uma tabela com as métricas de ajuste do modelo.
  - `tidy`: retorna a tabela do `summary` do modelo já formata de forma mais elegante.

O pacote **broom**

```
library(broom)
augment(x = modelo2)
```

```
## # A tibble: 80 x 10
##    log.riqueza. log.area. ilha     .fitted .se.fit   .resid   .hat .sigma
##  *        <dbl>     <dbl> <chr>      <dbl>   <dbl>    <dbl>  <dbl>  <dbl>
##  1         3.04     0.411 costeira    3.11  0.0763 -0.0662  0.0411  0.379
##  2         3.30     3.11  costeira    3.36  0.0601 -0.0633  0.0255  0.379
##  3         3.14     3.71  costeira    3.41  0.0630 -0.279   0.0280  0.377
##  4         2.30     7.67  oceanica    3.19  0.0960 -0.883   0.0651  0.364
##  5         3.53     6.91  costeira    3.71  0.105  -0.183   0.0784  0.378
##  6         2.56     3.67  oceanica    2.42  0.0597  0.141   0.0252  0.378
##  7         3.76     3.66  costeira    3.41  0.0626  0.351   0.0277  0.377
##  8         2.48     1.81  oceanica    2.07  0.0658  0.415   0.0305  0.376
##  9         2.56     5.76  oceanica    2.82  0.0726 -0.257   0.0373  0.378
## 10         3.47     4.36  costeira    3.47  0.0686 -0.00883 0.0333  0.379
## # ... with 70 more rows, and 2 more variables: .cooksd <dbl>,
## #   .std.resid <dbl>
```

O pacote **broom**

```
glance(x = modelo2)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC
## *     <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl>
## 1     0.776         0.767 0.376      87.9 1.20e-24     4  -33.3  76.5
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

O pacote **broom**

```
tidy(x = modelo2)
```

```
## # A tibble: 4 x 5
##   term                  estimate std.error statistic  p.value
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)              3.07      0.0819     37.5  8.48e-51
## 2 log(area)                0.0920    0.0207      4.44 3.00e- 5
## 3 ilhaoceanica            -1.35      0.118     -11.5  3.19e-18
## 4 log(area):ilhaoceanica   0.0985    0.0272      3.62 5.33e- 4
```

Exercício 4

- Extraia as métricas relacionadas ao ajuste dos três modelos que utilizamos até aqui (`glance`; 2 modelos que utilizamos como exemplo e mais o modelo que você trabalhou no exercício anterior) e armazene eles em um único objeto.
- Adicione algum tipo de identificador às informações de cada modelo.
- Exporte este objeto para algum diretório do seu computador, utilizando a extensão `.csv`.

Visualização dos resultados de um modelo estatístico {#anchor3}

- Como vimos até aqui, podemos gerar os valores preditos por um modelo e adicioná-los a um `data.frame`, com o intuito de utilizar esta informação, por exemplo, para confeccionar uma figura.

```
ilhas <- mutate(ilhas, log_riqueza_predita = predict(modelo2))
ilhas
```

```
## # A tibble: 80 x 13
##    ID     ilha  arquipelago riqueza   area produtividade populacao habitat
##    <chr>  <chr> <chr>         <dbl>  <dbl>         <dbl>     <dbl>   <dbl>
##  1 ilha_1 cost… medio            21 1.51e0          1.57      268.       2
##  2 ilha_… cost… medio            27 2.24e1          4.56      259.       2
##  3 ilha_… cost… pequeno          23 4.11e1          2.19      281.       3
##  4 ilha_… ocea… pequeno          10 2.13e3       2013.        145.       5
##  5 ilha_… cost… medio            34 1.00e3        244.        516.       4
##  6 ilha_… ocea… medio            13 3.92e1          3.17      579.       3
##  7 ilha_… cost… grande           43 3.89e1         18.4       389.       2
##  8 ilha_… ocea… grande           12 6.12e0          4.40      232.       3
##  9 ilha_… ocea… pequeno          13 3.17e2         14.3       156.       4
## 10 ilha_… cost… medio            32 7.86e1         35.8       403.       3
## # ... with 70 more rows, and 5 more variables: montanha <chr>,
## #   mamiferos <dbl>, temperatura <dbl>, precipitacao <dbl>,
## #   log_riqueza_predita <dbl>
```

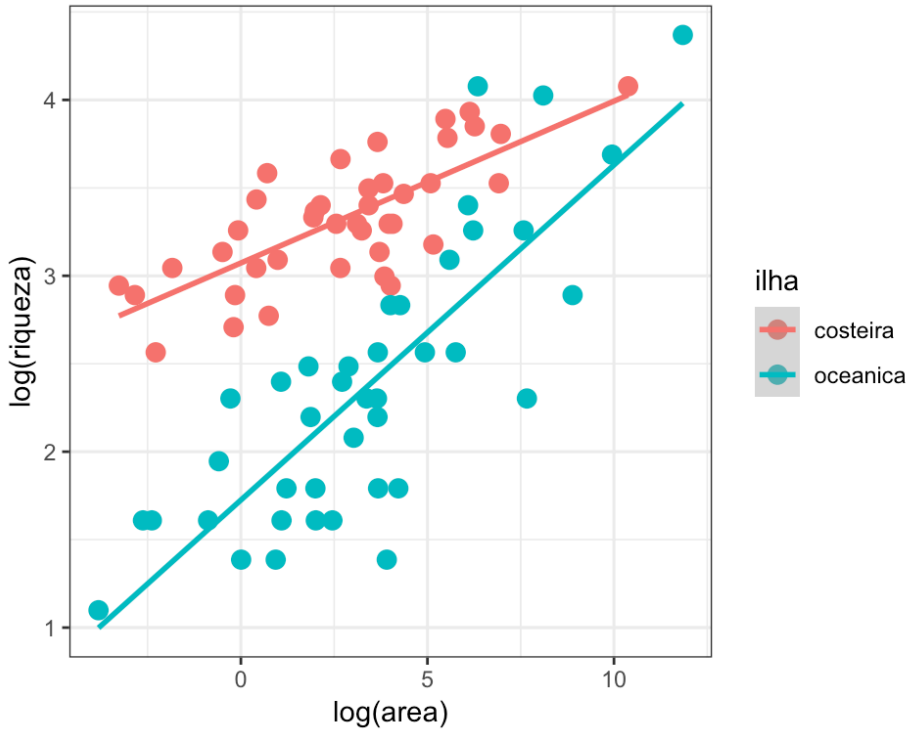
Visualização dos resultados de um modelo estatístico

- Com esta informação em mãos, podemos agora proceder para a confecção da figura que descreva de que forma nosso modelo prediz o formato da relação entre as variáveis analisadas.

```
ggplot(data = ilhas) +
  geom_point(mapping = aes(x = log(area), y = log(riqueza), colour = ilha), size = 3) +
  geom_smooth(mapping = aes(x = log(area), y = log_riqueza_predita, colour = ilha),
              method = "lm") +
  theme_bw()
```

Visualização dos resultados de um modelo estatístico

```
ggplot(data = ilhas) +
  geom_point(mapping = aes(x = log(area), y = log(riqueza), colour = ilha), size = 3) +
  geom_smooth(mapping = aes(x = log(area), y = log_riqueza_predita, colour = ilha),
              method = "lm") +
  theme_bw()
```



Exercício 5

- Crie uma figura que descreva o valor esperado para a riqueza de espécies em ilhas oceânicas e costeiras conforme extraímos e armazenamos no objeto **diferencas_ilhas**.

```
diferencas_ilha
```

```
##       ilha     area   emmean         SE df lower.CL upper.CL
## 1 costeira 2680.297 3.799297 0.12269435 76 3.554930 4.043664
## 2 oceanica 2680.297 3.228933 0.09919047 76 3.031378 3.426488
```

O pacote `visreg`

- O pacote `visreg` é extremamente útil para gerararmos visualizações rápidas do resultado dos modelos estatísticos que estabelecemos.
- A função-base deste pacote é capaz de interagir com praticamente qualquer tipo de modelo que normalmente utilizamos e, portanto, pode nos ajudar bastante a entender o resultado de nossas análises.

```
modelo2
```

```
##
## Call:
## lm(formula = log(riqueza) ~ log(area) * ilha, data = ilhas)
##
## Coefficients:
##            (Intercept)              log(area)           ilhaoceanica
##                3.07290                 0.09202              -1.34773
## log(area):ilhaoceanica
##                0.09848
```

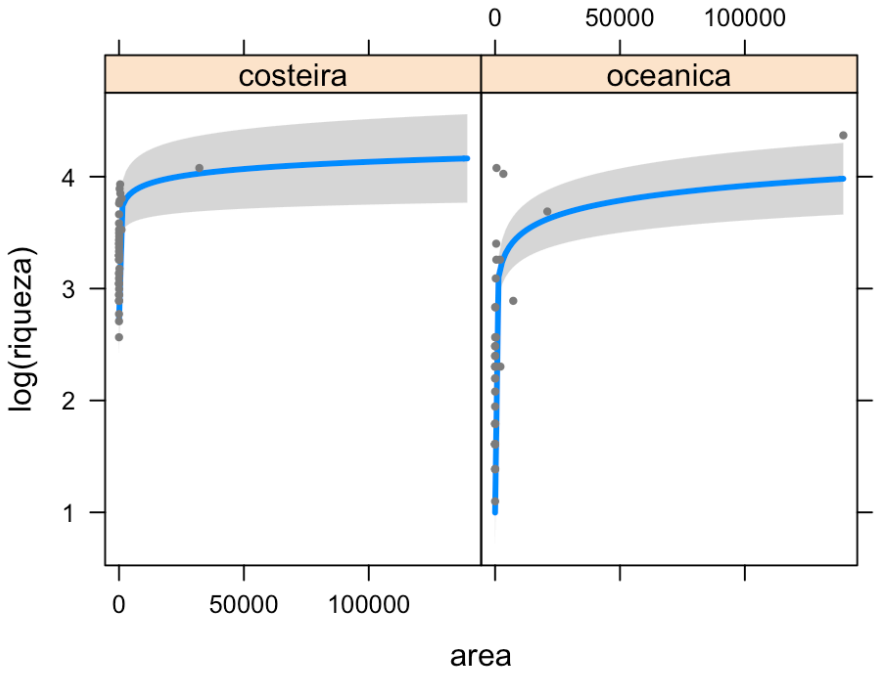
O pacote **visreg**

```
library(visreg)
visreg(fit = modelo2, xvar = "ilha")
```

```
## Conditions used in construction of plot
## log.area.: 3.305367
## riqueza: 20.5
```

O pacote **visreg**

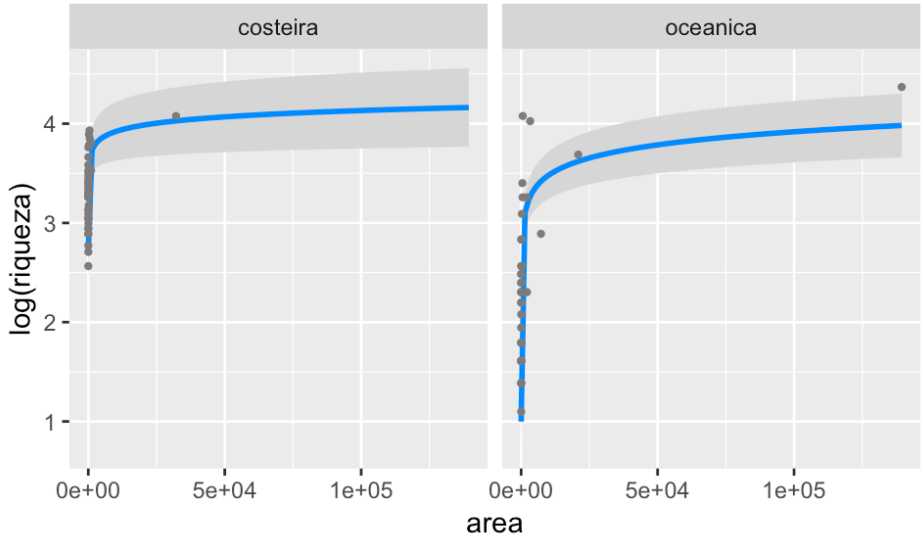
```
visreg(fit = modelo2, xvar = "area", by = "ilha")
```



O pacote `visreg`

- O `visreg` interage bastante com o `lattice`, mas tem um argumento que permite que a visualização seja feita nos moldes do `ggplot2`.
- Além disso, ele tem uma série de outros argumentos que permitem outros tipos de manipulação aquilo que será plotado.

```
visreg(fit = modelo2, xvar = "area", by = "ilha", gg = TRUE)
```



O pacote `visreg`

- Podemos também utilizar o resultado da função `visreg` para produzir um `data.frame` de onde vamos utilizar os dados para customizar a figura da maneira como bem entendermos.
- O jeito de fazer isso é passando para o `visreg` um argumento que indique que ele não precisa plotar a figura, e atribuir o resultado dessa função a um objeto.
- Se você conferir, este objeto será uma lista, que contém dois `data.frame` que podemos utilizar: `res` e `fit`.
    - `res`: valores ajustados das observações;
    - `fit`: valores utilizados para a construção da reta de regressão.

```
exemplo <- visreg(fit = modelo2, xvar = "area", by = "ilha", plot = FALSE)
exemplo$fit
exemplo$res
```

Algumas palavras de cuidado

- Muitas destas técnicas podem ser usadas para averiguarmos se os resultados de uma análise estatística fazem sentido ou não, e devem ser utilizadas para tal.
- Todas as técnicas apresentadas aqui no intuito de demonstrar como apresentar seus resultados assumem que o modelo que estamos ajustando está correto e, também, que tenhamos desvendado os potenciais viéses e artefatos existentes no conjunto de dados que estamos trabalhando.
- Finalmente, busque sempre apresentar graficamente seus resultados de forma que eles combinem com o modo que você os analisou e com os resultados encontrados, por exemplo:
  - Se existe uma interação entre duas ou mais variáveis, apresente uma figura onde você contemple o resultado da interação;
  - Se você considerou que uma variável preditora é contínua, apresenta os valores de `x` no formato de uma regressão, e não com uma medida de médio e erro associado (o que seria o padrão esperado para uma análise utilizando variáveis categóricas).