Preparado para:

REFORM/SC2022/126
DELIVERABLE 4
MÓDULO 4
REGRESSÃO LINEAR

DESIGNING A NEW VALUATION MODEL
FOR RURAL PROPERTIES IN PORTUGAL

Parte II

Formador: Luís Teles Morais | Nova SBE

Lisboa, 29 junho 2023

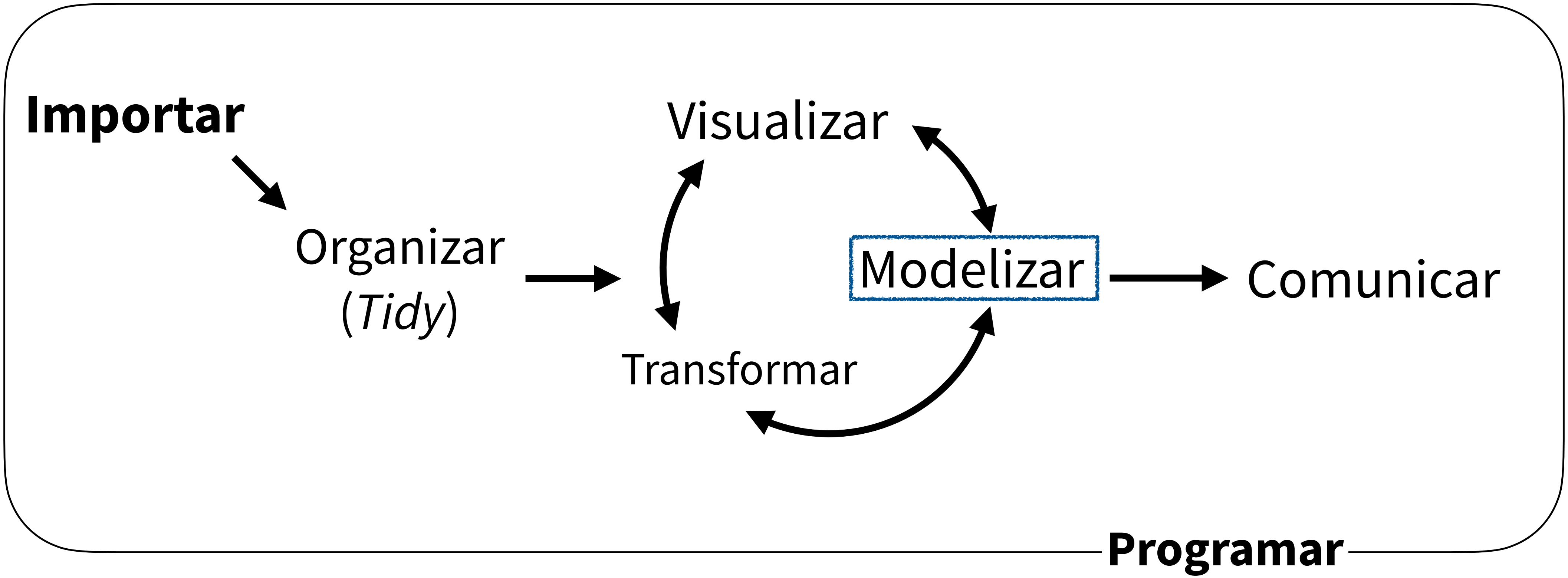
This project is carried out with funding by the European Union via the Structural Reform Support Programme and in cooperation with the Directorate General for Structural Reform Support of the European Commission

Programa

Módulos	Duração
Módulo 1 – Introdução ao R: - O que é o R? - Como instalar e configurar o R. - Sintaxe básica e comandos. - Tipos de dados, objetos e classes.	**4 Horas**
Módulo 2 – Gestão e tratamento de dados em R: - Carregar dados no R. - Perceber as estruturas de dados e *subsetting*. - Limpeza de dados: *missing values, outliers* e transformações - Juntar bases de dados	**8 Horas**
Módulo 3 – Estatística básica em R: - Estatísticas descritivas: medidas de dispersão central e variação. - Distribuições probabilísticas: variáveis discretas e contínuas. - Testes de hipóteses.	**8 Horas**

Módulos	Duração
Módulo 4 – Regressão Linear: - O modelo classico linear. - Estimação de parametros segundo o MMQ. - Testes de hipóteses: significância estatística e ajuste do modelo. - Modelo de regressão múltipla. - Testar as premissas: multicolinearidade, heteroscedasticidade e normalidade dos resíduos. - Critérios de seleção dos modelos.	**12 Horas**
Módulo 5 – O modelo: - Estrutura do modelo e premissas – Perceber o modelo (4 Hours). - Uso e tratamento dos dados (4 Hours). - Descrição do modelo (4 Hours). - Aplicação do modelo a cada piloto (12 Hours). - Aplicação autónoma do modelo a uma região (8 Hours).	**32 Horas**

Ciência de dados



Modelos em R

Definição de um modelo linear

$$Y = \alpha + \beta X + \varepsilon$$

- Ex.: Y altura, X largura
- α - constante (ordenada na origem
- β - coeficiente de regressão / declive
- ε - erro do modelo

porque não resíduo?

Definição de um modelo linear

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

não observados

Estimar $\hat{\alpha}$ $\hat{\beta}$

- Ex.: Y altura, X largura
- α - constante (ordenada na origem
- β - coeficiente de regressão / declive
- ε - <u>erro</u> do modelo

- Hip.: linearidade
- Parâmetros e estimativas a partir dos dados

i = 1, 2, …, N

Método dos mínimos quadrados (OLS)

- *Ordinary Least Squares:* minimizar os resíduos

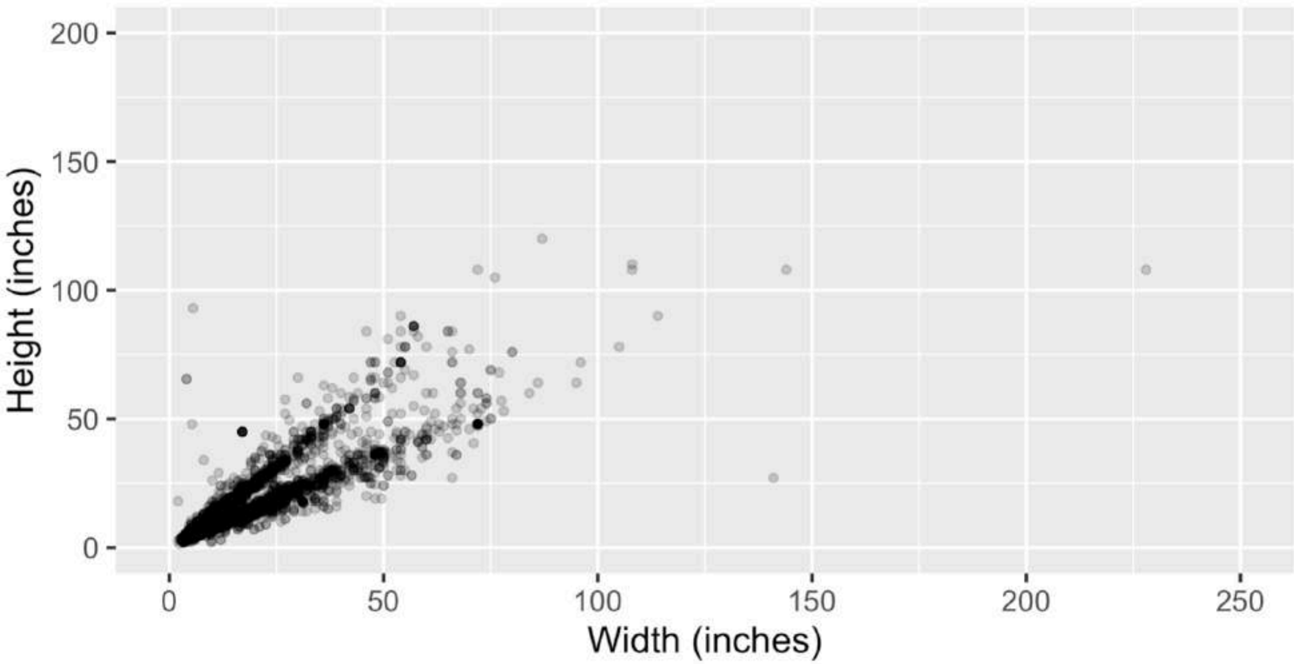
$$\min_{\hat{\alpha},\hat{\beta}} \sum_{i=1}^{n} \varepsilon_i^2 = \min_{\hat{\alpha},\hat{\beta}} \sum_{i=1}^{n} \left[Y_i - \left(\hat{\alpha} + \hat{\beta} X_i \right) \right]^2$$

- Mostra-se que:

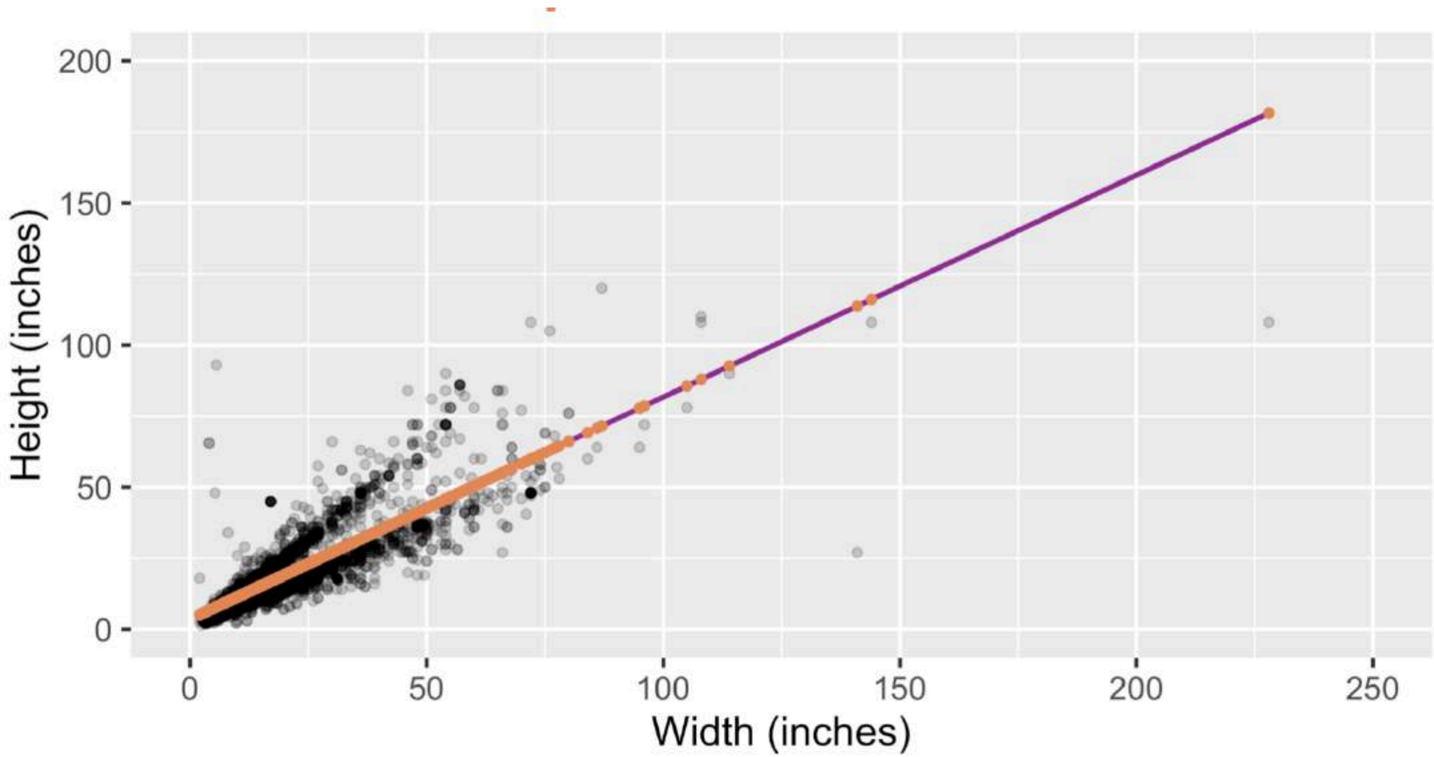
$$\hat{\alpha} = \bar{Y} - \hat{\beta}\bar{X} \qquad \hat{\beta} = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (X_i - \bar{X})^2}$$

parece familiar?

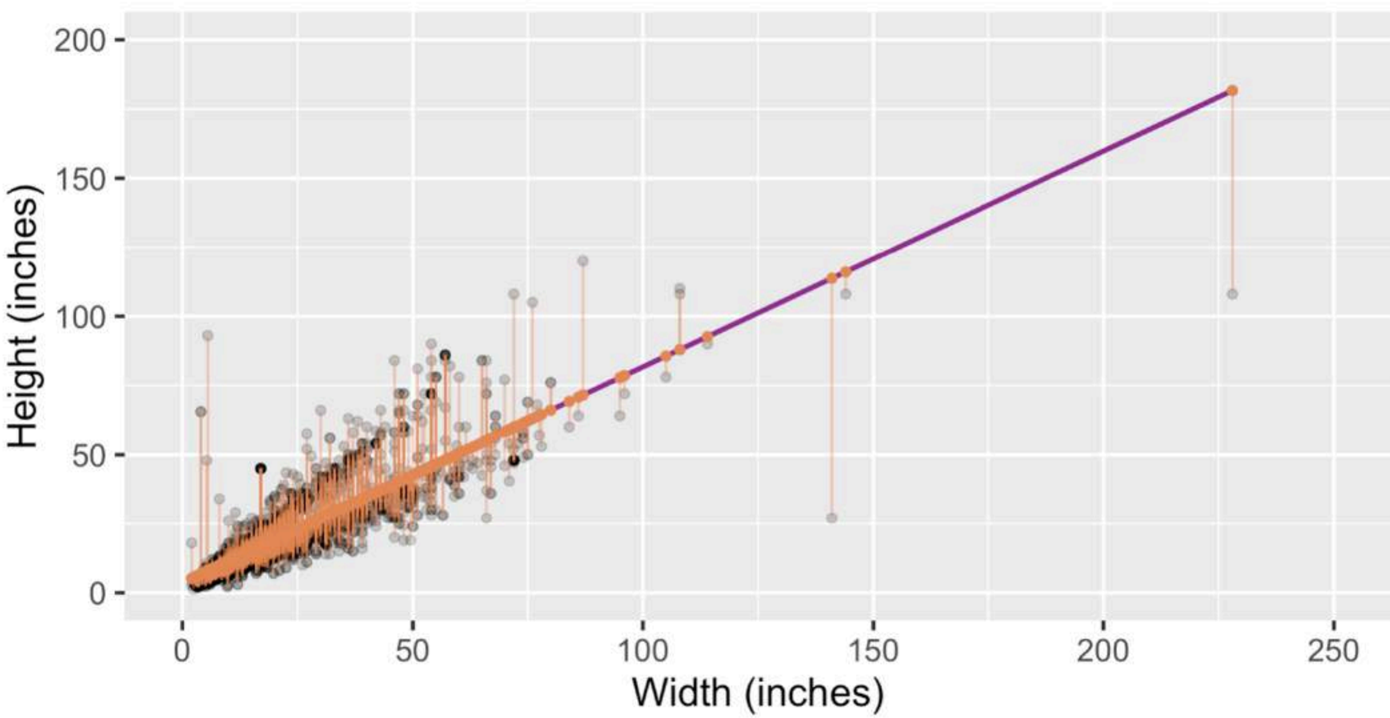
Minimizar os resíduos



Minimizar os resíduos



Minimizar os resíduos



“Bondade do ajustamento”

- R2: medida de ajustamento do modelo aos dados

Fonte da variação	Soma dos quadrados
Variação explicada	$ESS = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2$
Variação residual	$SSR = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$
Variação total	$TSS = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$

TSS=ESS+SSR

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{SSR}{TSS},$$
$$0 \leq R^2 \leq 1$$

- R2 = 1: toda a variação dos dados pode ser explicada pelo modelo
- R2 = 0: vice-versa

Funções em R para estimar diferentes modelos

function	package	fits
lm()	**stats**	**linear models**
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines

modelr

Funções tidy para trabalhar com modelos no *tidyverse*

```
library(tidyverse)
library(modelr)
```

lm()

linear **m**odel

fórmulas no R

A equação de um modelo define-se no R em fórmulas, onde apenas é necessário indicar as variáveis dependente e independentes

$$y = \alpha + \beta x + \epsilon$$

$$y \sim x$$

Var. dependente

til

Var. independente

lm()

Função base de modelos lineares:

```
modelo <- lm(y ~ x, data = babynames)
```

Fórmula
(equação da regressão a estimar)

Tabela de dados
(tibble ou data.frame) onde as variáveis do modelo se localizam

%>% .

Utiliza-se o ponto final quando queremos passar uma tabela a uma função, noutro local que não o 1.º argumento

```
mod_e <- wages %>%
  lm(log(income) ~ education, data = .)
```

wages (tabela) **passa para aqui**

$

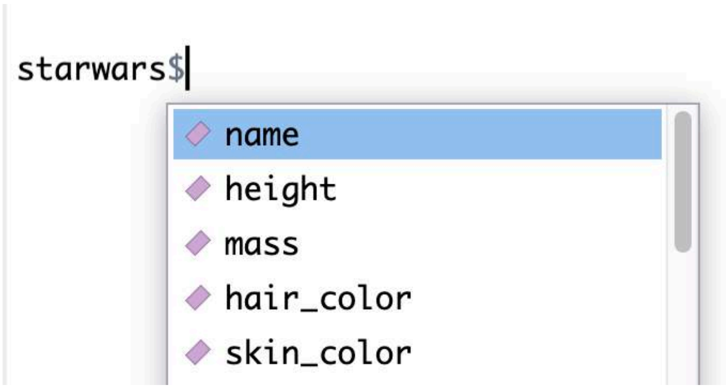
Para obter uma componente de um objeto (e.g. **variáveis** de um **tibble**)

```
> starwars$height
 [1] 172 167  96 202 150 178 165  97 183 182 188 180 228
[14] 180 173 175 170 180  66 170 183 200 190 177 175 18
...
```

nome da variável sem aspas

N.º da observação

- Útil para visualizar rapidamente as diferentes variáveis:



Experimente

Corra 2 modelos lineares:

`mass` = α + β x `height`, com os dados starwars

`price` = α + β x `Width_in`, com os dados pp

e examine os outputs, atribuindo-os aos objetos *modelo_star* e *modelo_quadros*, respetivamente

```
modelo_star <- lm(mass ~ height, data = starwars)
modelo_star

Call:
lm(formula = mass ~ height, data = starwars)

Coefficients:
(Intercept)       height
   -13.8103       0.6386
```

```
modelo_quad <- lm(price ~ Width_in, data = pp)

Error in lm.fit(x, y, offset = offset, singular.ok =
singular.ok, ...) :
  NA/NaN/Inf in 'y'
In addition: Warning message:
In storage.mode(v) <- "double" : NAs introduced by
coercion
```

```
modelo_quad <- lm(price ~ Width_in, data = pp)
     🤬‼️
Error in lm.fit(x, y, offset = offset, singular.ok =
singular.ok, ...) :
  NA/NaN/Inf in 'y'  🤓
In addition: Warning message:
In storage.mode(v) <- "double" : NAs introduced by
coercion
```

```
pp$price
> ...
 [988] "1,240.0"                    "502"                    "231.5"
 [991] "231.5"                      "500"                    "1,401.0"
 [994] "1,401.0"                    "3,000.0"                "3,000.0"
 [997] "9,200.0"                    "1,800.0"                "1,201.0"
[1000] "1,201.0"
 [ reached getOption("max.print") -- omitted 2393
entries ]
```

```
pp$price
> ...
 [988] "1,240.0"                    "502"                      "231.5"
 [991] "231.5"                      "500"                      "1,401.0"
 [994] "1,401.0"                    "3,000.0"                  "3,000.0"
 [997] "9,200.0"                    "1,800.0"                  "1,201.0"
[1000] "1,201.0"
 [ reached getOption("max.print") -- omitted 2393
entries ]
```

str_replace_all()

Substituir partes de “strings” (valores alfanuméricos)

```
str_replace_all(“TerrenoRustico”, “Terreno”, “Predio”)
```

Valor
ou **vector** de **valores**

Expressão a substituir

Expressão a introduzir

```
> “PredioRustico”
```

```
pp <- pp %>% mutate(price = str_replace_all(price, ",", ""))
pp$price
> ...
 [988] "1240.0"             "502"
 [991] "231.5"              "500"               "1401.0"
 [994] "1401.0"             "3000.0"            "3000.0"
 [997] "9200.0"             "1800.0"            "1201.0"
[1000] "1201.0"
 [ reached getOption("max.print") -- omitted 2393 entries ]
```

string vazia (“”) é diferente de **NA**

unique()

Devolve o conjunto de valores diferentes existentes num vetor (= remove as repetições).
Permite ver rapidamente conteúdo de variável qualitativa

```
> starwars$skin_color %>% unique()
 [1] "blond"          NA               "none"           "brown"
 [5] "brown, grey"    "black"          "auburn, white" "auburn, grey"
...
```

funciona com pipe %>%

útil para descobrir **NAs**

útil antes de utilizar **group_by**

modelr

Para aquecer

Os dados de Paris contêm quadros com vários formatos: não só quadrangulares mas redondos, ovais etc.

Crie uma nova tabela **pp_rect** que contenha apenas as observações que verifiquem as seguintes condições:

1. apenas quadros quadrangulares (variável **Shape**)
2. apenas quadros que contenham valores (não NA) para **Width_in** e **Height_in**

Finalmente, na nova tabela **pp_rect** assegure que a variável **price** é do tipo *numeric*, utilizando a função **as.numeric.**

```r
pp_rect <- pp %>% filter(Shape == "squ_rect",
                         !is.na(Width_in), !is.na(Height_in)) %>%
                  mutate( price = str_replace_all(price, ",", ""),
                          price = as.numeric(price))
pp_rect$price / 2
```

```
...
 [958]   116.00   600.00   250.50   212.50   760.00   360.00   430.50    75.00   250.50   400.50
 [969]    65.00   406.00    24.00   349.50    24.50    24.50   205.00   525.00   237.75   237.75
 [980]   203.00    36.00   125.25   125.25   184.00    75.50   130.00   130.00  4525.00   900.00
 [991]   500.50   401.00    50.00    80.00   125.00   125.00    36.00   151.00    12.00  1000.00
 [ reached getOption("max.print") -- omitted 2083 entries ]
```

```r
modelo_quad <- lm(price ~ Width_in, data = pp_rect)
modelo_quad

Call:
lm(formula = price ~ Width_in, data = pp_rect)

Coefficients:
(Intercept)     Width_in
     376.77        19.47
```

$\hat{\alpha}$ $\hat{\beta}$

Como interpretar?

```
modelo_quad <- lm(price ~ Width_in, data = pp_rect)
modelo_quad

Call:
lm(formula = price ~ Width_in, data = pp_rect)

Coefficients:
(Intercept)     Width_in
     376.77        19.47
```

$\hat{\alpha}$ $\hat{\beta}$

Incerteza?

broom

Transforma output de modelos em tabelas (tidy)

```r
library(tidyverse)
library(broom)
```

broom

Três funções úteis:

1. **tidy()** - devolve coeficientes e estatísticas principais do modelo
2. **glance()** - dá testes de diagnóstico do modelo
3. **augment()** - obtém, para cada observação, valores previstos, resíduos, e outras medidas relevantes

tidy()

Obtém resultados essenciais do modelo

```
modelo_quad %>% tidy()
```

```
# A tibble: 2 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    368.      59.3       6.21 6.15e-10
2 Width_in        19.6      2.22      8.81 1.95e-18
```

glance()

Diagnóstico da qualidade do ajustamento

```r
modelo_quad %>% glance()
```

```
# A tibble: 1 × 12
  r.squared adj.r.squared sigma statistic  p.value    df  logLik    AIC    BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>   <dbl>  <dbl>  <dbl>
1    0.0242        0.0239 1941.      77.7 1.95e-18     1 -28200. 56405. 56424.
```

augment()

Dá tabela com os dados utilizados no modelo e variáveis relacionadas com a estimação, como valores previstos e resíduos

```
modelo_quad %>% augment()
```

```
> modelo_quad %>% augment %>% arrange(.resid)
# A tibble: 3,137 × 9
   .rownames price Width_in .fitted .resid    .hat .sigma .cooksd .std.resid
   <chr>     <dbl>    <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
 1 966          24      228   4832. -4808. 0.0561   1939. 0.193       -2.55
 2 965         100      144   3187. -3087. 0.0199   1940. 0.0262      -1.61
 3 964          24      108   2482. -2458. 0.0101   1940. 0.00825     -1.27
 4 1169         90      108   2482. -2392. 0.0101   1940. 0.00782     -1.24
 5 517         252.     105   2424. -2172. 0.00941  1941. 0.00601     -1.12
 6 824         120       96   2247. -2127. 0.00755  1941. 0.00461     -1.10
 7 967          48       87   2071. -2023. 0.00591  1941. 0.00325     -1.05
 8 2787       1200      141   3128. -1928. 0.0190   1941. 0.00973     -1.00
 9 945         100       84   2013. -1913. 0.00541  1941. 0.00265     -0.988
10 901          60       74   1817. -1757. 0.00390  1941. 0.00161     -0.907
# ℹ 3,127 more rows
```

Experimente

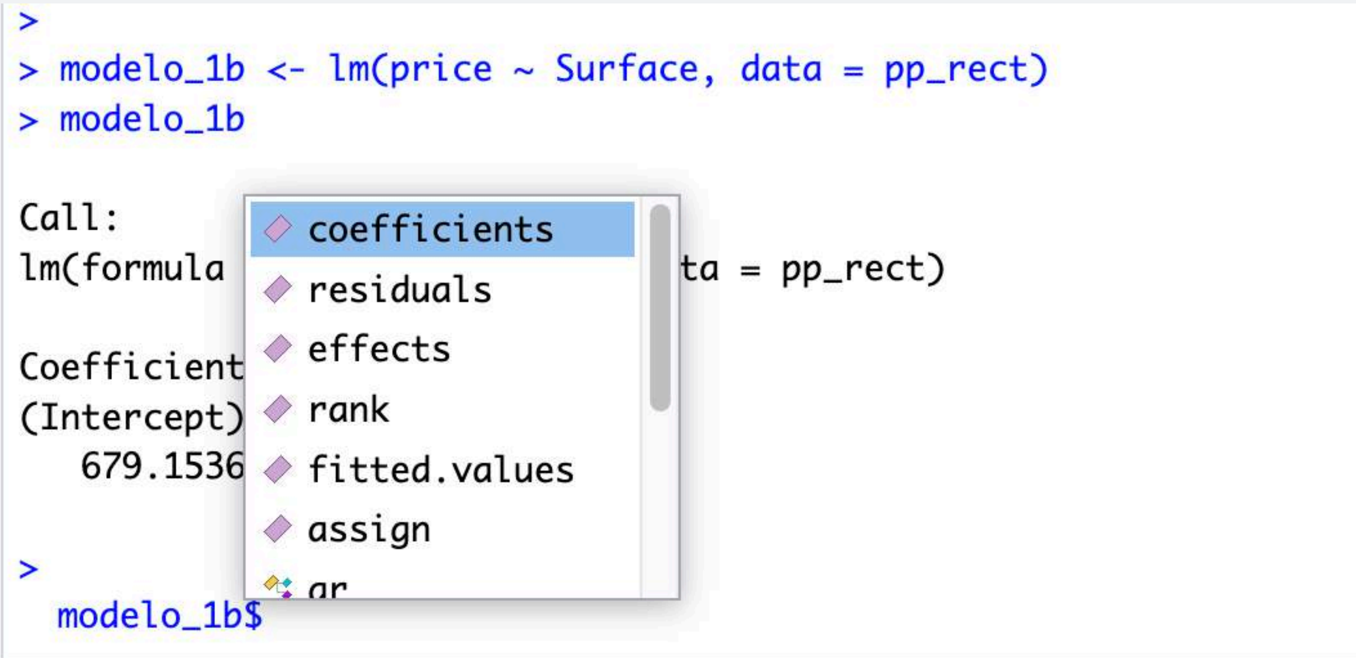
Utilizando os dados filtrados **pp_rect**:

1. Estime 2 modelos separadamente, que relacionam o **preço** (price) como variável independente, com:
    a. a altura do quadro (**Height_in**)
    b. a área de superficie (**Surface**)

E guarde-os em objetos respetivos.

```
> modelo_1a <- lm(price ~ Height_in, data = pp_rect)
> modelo_1a

Call:
lm(formula = price ~ Height_in, data = pp_rect)

Coefficients:
(Intercept)    Height_in
      493.6         14.9
```

```
> modelo_1b <- lm(price ~ Surface, data = pp_rect)
> modelo_1b

Call:
lm(formula = price ~ Surface, data = pp_rect)

Coefficients:
(Intercept)      Surface
   679.1536       0.1903
```


>
> modelo_1b <- lm(price ~ Surface, data = pp_rect)
> modelo_1b
Call:
lm(formula
ta = pp_rect)
Coefficient
(Intercept)
679.1536
>
modelo_1b$
coefficients
residuals
effects
rank
fitted.values
assign
qr

Hipóteses de um modelo OLS

Recorde: pelo CLT, para 1 variável a média da amostra é um bom estimados da média da população — ou seja, não é <u>enviesado.</u>

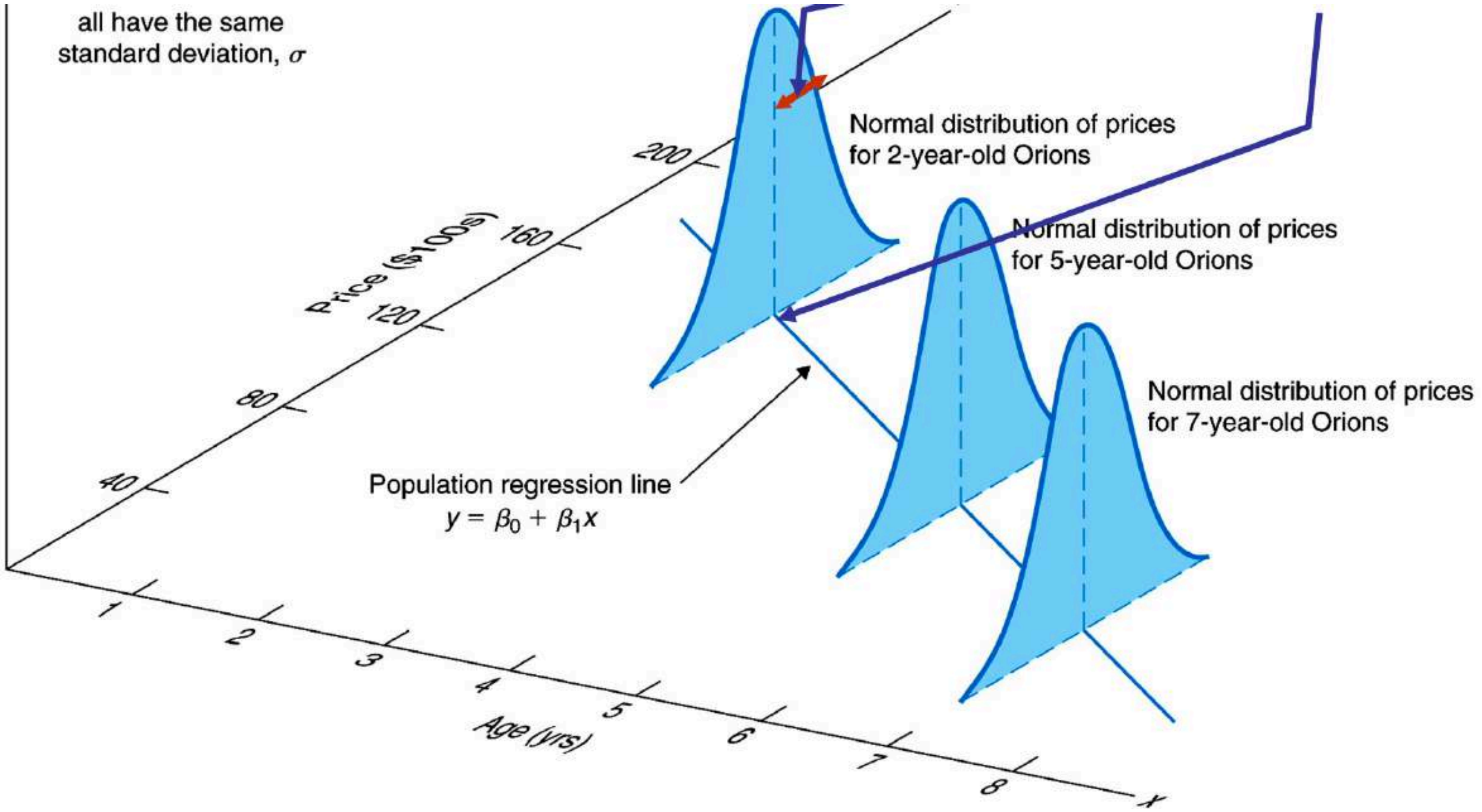
$$Y = \alpha + \beta X + \varepsilon \longrightarrow Y_i = \hat{\alpha} + \hat{\beta} X_i + e_i$$

Da mesma forma, os **estimadores OLS** são bons estimadores dos parâmetros, sob <u>certas hipóteses</u>.

<u>Hip. OLS</u>

- Linearidade
- Resíduos não correlacionados com a variável independente (**<u>exogeneidade</u>**)
- Amostra i.i.d. - independente e identicamente distribuída

Hipóteses de um modelo OLS



Significância estatística

Sob estas hipóteses, podemos ter uma ideia da <u>significância</u> do coeficiente de declive: traduz ele realmente uma relação entre as variáveis?

$$H_0 : \beta = 0 \qquad \text{vs} \qquad H_1 : \beta \neq 0$$

Sob hipóteses OLS, numa amostra grande:

- $\hat{\beta}$ acerta em média (é centrada)
- A variância, ou o erro padrão, de $\hat{\beta}$ tem uma distribuição Normal pelo T. do Limite Central

→ **Podemos avaliar a <u>estatística-t</u>:**

$$t = \frac{\hat{\beta}}{SE\left(\hat{\beta}\right)}$$

Significância estatística

$$H_0 : \beta = 0 \qquad \text{vs} \qquad H_1 : \beta \neq 0$$

Sob hipóteses OLS, numa amostra grande, mostra-se que

$$t = \frac{\hat{\beta}}{SE\left(\hat{\beta}\right)} \sim N(0,1)$$

Interpretação de modelos: **queremos erros-padrão pequenos em relação a ^β**

Quando | **t** | **> 1.96** (= valor crítico):

- rejeita-se H0 (= **β é diferente de zero)**
- com um nível de significância de **5%**

Significância estatística

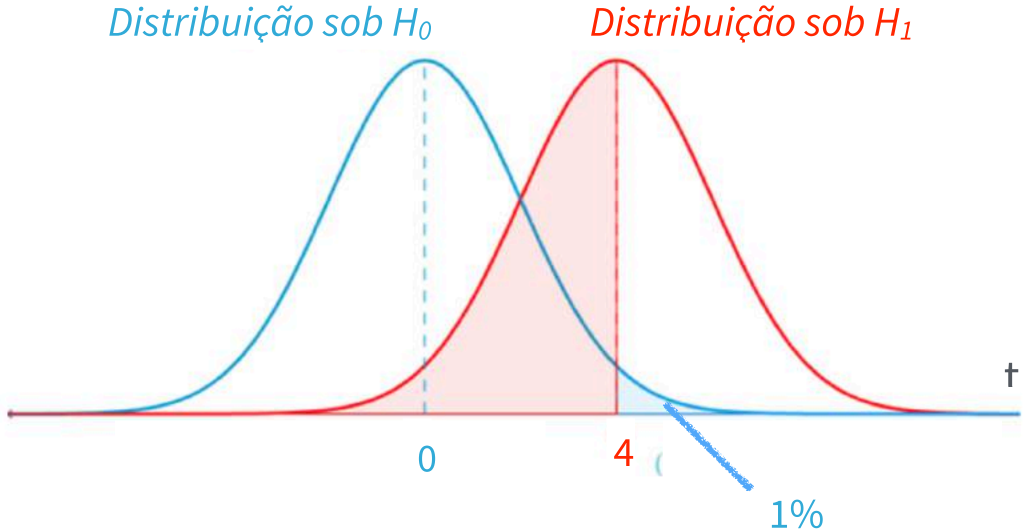
Interpretação de modelos: **queremos p-value pequenos**

$$H_0 : \beta = 0 \qquad \text{vs} \qquad H_1 : \beta \neq 0$$

O mesmo teste pode ser visto rapidamente com **p-values:**

$$t = \frac{\hat{\beta}}{SE\left(\hat{\beta}\right)} \sim N(0,1)$$

valor da **estatística *t* grande**
<=> **valor-p pequeno**



- **p-value = 2% < nível de significância = 5%**
  - rejeita-se H0 => **<u>estimativa de β é significativa</u>**

Experimente

Utilizando os dados filtrados **pp_rect**:

1. Estime 2 modelos separadamente, que relacionam o **preço** (price) como variável independente, com a altura do quadro (**Height_in**) e a área de superficie (**Surface**). E guarde-os em objetos respetivos.
2. Utilizando funções **broom** obtenha, para cada modelo, uma única tabela que contenha, pelo menos, os coeficientes de declive, o p-value e o R2.
   - As estimativas são significativas ao nível de 5%?
   - Pode concluir-se que as dimensões de diferentes quadros explicam bem as diferenças nos seus preços de venda?

```
> modelo_2 %>% tidy() %>%
+   cross_join(modelo_2 %>% glance %>% select(r.squared))
# A tibble: 2 × 6
  term        estimate std.error statistic  p.value r.squared
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)    494.       60.8      8.12 6.84e-16    0.0124
2 Height_in       14.9       2.39     6.23 5.45e-10    0.0124
```

```
> modelo_2b %>% tidy() %>%
+   cross_join(modelo_2b %>% glance %>% select(r.squared))
# A tibble: 2 × 6
  term        estimate std.error statistic  p.value r.squared
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)  679.       41.1        16.5 8.14e-59    0.0111
2 Surface        0.190     0.0324      5.87 4.76e- 9    0.0111
```

Modelos de regressão linear **múltipla**

fórmulas no R

Para acrescentar múltiplas variáveis dependentes, basta somá-las à fórmula do modelo

```
price ~ Surface + year
```

```
modelo_3 <- pp_rect %>%
        lm(price ~ Surface + Height_in, data = .)

modelo_3 %>% tidy()
# A tibble: 3 × 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 538.        73.7          7.29 3.93e-13
2 Surface       0.0661     0.0627       1.05 2.92e- 1
3 Height_in    10.7        4.64         2.31 2.09e- 2
```

Experimente

1. Volte a estimar com os dados **pp_rect**, mas agora com três variáveis explicativas: a altura, a área de superfície e o ano **year**.
    - O que aconteceu às estimativas do efeito da altura e da área, comparado com os modelos anteriores?
    - O que aconteceu ao R2?
    - Porque é que a estimativa da constante α ficou tão grande (em modulo)?

```
modelo_4 <- p_rect %>%
        lm(price ~ Surface +Height_in + year, data = .)

modelo_4 %>% tidy() %>%
    cross_join((modelo_4 %>% glance %>% select(r.squared)))
```

```
# A tibble: 4 × 6
  term        estimate std.error statistic  p.value r.squared
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept) -110909.    12576.     -8.82 1.90e-18    0.0373
2 Surface         0.0392     0.0620     0.633 5.27e- 1    0.0373
3 Height_in      15.8        4.61      3.43 6.04e- 4    0.0373
4 year           62.8        7.09      8.86 1.30e-18    0.0373
```

Interpretação com variáveis múltiplas

```
# A tibble: 4 × 6
  term         estimate std.error statistic  p.value r.squared
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept) -110909.   12576.       -8.82 1.90e-18    0.0373
2 Surface          0.0392    0.0620    0.633 5.27e- 1    0.0373
3 Height_in       15.8       4.61      3.43 6.04e- 4    0.0373
4 year            62.8       7.09      8.86 1.30e-18    0.0373
```

- **Declive - superfície: Tudo o resto constante (*ceteris paribus*)**, por cada polegada adicional de superfície, espera-se que o preço do quadro seja, em média, 0.04 Francos mais elevado. *Quadros maiores tendem a ter preços mais elevados.*

Interpretação com variáveis múltiplas

```
# A tibble: 4 x 6
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) -110909.    12576.      -8.82 1.90e-18
2 Surface          0.0392     0.0620    0.633 5.27e- 1
3 Height_in       15.8        4.61      3.43 6.04e- 4
4 year            62.8        7.09      8.86 1.30e-18
```

Altura mais importante do que superfície?!

- **Declive - Height_in: Tudo o resto constante (*ceteris paribus*)**, por cada polegada adicional em altura, espera-se que o preço do quadro seja, em média, 15.8 Francos mais elevado. *Quadros maiores tendem a ter preços mais elevados.*

Interpretação com variáveis múltiplas

```
# A tibble: 4 × 6
  term         estimate std.error statistic  p.value r.squared
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept) -110909.    12576.      -8.82 1.90e-18    0.0373
2 Surface          0.0392     0.0620   0.633 5.27e- 1    0.0373
3 Height_in       15.8        4.61     3.43  6.04e- 4    0.0373
4 year            62.8        7.09     8.86  1.30e-18    0.0373
```

- **Declive - ano: Tudo o resto constante (*ceteris paribus*)**, espera-se que por cada ano a mais da data do leilão, o preço do quadro seja em média, 62.8 Francos mais elevado. *Quanto mais tarde ocorreu o leilão, mais elevados tendem a ser os preços.*

Interpretação com variáveis múltiplas

```
# A tibble: 4 x 6
  term          estimate std.error statistic  p.value r.squared
  <chr>            <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept) -110909.    12576.      -8.82  1.90e-18    0.0373
2 Surface          0.0392     0.0620   0.633 5.27e- 1    0.0373
3 Height_in       15.8        4.61     3.43  6.04e- 4    0.0373
4 year            62.8        7.09     8.86  1.30e-18    0.0373
```

- **Intercept:** Espera-se que um quadro com **zero** de área de superfície, **zero** de altura e vendido num leilão no ano **zero** tenha em média um preço de -111,000 francos. (Faz sentido?)

Multicolinearidade

Quando as variáveis independentes estão fortemente correlacionadas:

- Estimativas instáveis
- Impossível distinguir efeito de diferentes variáveis

Forma fácil de inspecionar:

- Matriz de correlações
- -> Todos os coeficientes de correlação cruzados

```
pp_rect %>% select(Surface, Height_in, year) %>%
cor()
```

```
             Surface  Height_in       year
Surface    1.0000000  0.8563879 -0.1130731
Height_in  0.8563879  1.0000000 -0.1611672
year      -0.1130731 -0.1611672  1.0000000
```

Endogeneidade dos resíduos

Quando as variáveis independentes estão fortemente correlacionadas:

- Estimativas instáveis
- Impossível distinguir efeito de diferentes variáveis

Forma fácil de inspecionar:

- Matriz de correlações
- -> Todos os coeficientes de correlação cruzados

```
pp_rect %>% select(Surface, Height_in, year) %>%
cor()
```

```
            Surface  Height_in       year
Surface   1.0000000  0.8563879 -0.1130731
Height_in 0.8563879  1.0000000 -0.1611672
year     -0.1130731 -0.1611672  1.0000000
```

Experimente

1. Volte a estimar com os dados **pp_rect**, mas agora com três variáveis explicativas: a altura, a área de superfície e o ano **year**.
    - O que aconteceu às estimativas do efeito da altura e da área, comparado com os modelos anteriores?
    - O que aconteceu ao R2?
    - Porque é que a estimativa da constante α ficou tão grande (em modulo)?
2. Faça a seguinte alteração aos dados e volte a estimar o modelo pedido acima:
    1. Em vez de **year**, use uma nova variável **no_year**, que seja o número de anos volvidos desde o primeiro ano disponível nos dados
    - Que diferenças nota nos resultados?

```r
modelo_4b <- pp_rect %>% mutate(no_year = year - min(year)) %>%
                  lm(price ~ Surface +Height_in + no_year, data = .)

modelo_4b %>% tidy() %>%
    cross_join((modelo_4b %>% glance %>% select(r.squared)))
```

```
# A tibble: 4 × 6
  term        estimate std.error statistic  p.value r.squared
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept) -67.0       99.8      -0.672 5.02e- 1    0.0373
2 Surface       0.0392     0.0620    0.633 5.27e- 1    0.0373
3 Height_in    15.8        4.61      3.43  6.04e- 4    0.0373
4 no_year      62.8        7.09      8.86  1.30e-18    0.0373
```

Obrigado e até à próxima!

luis.morais@novasbe.pt