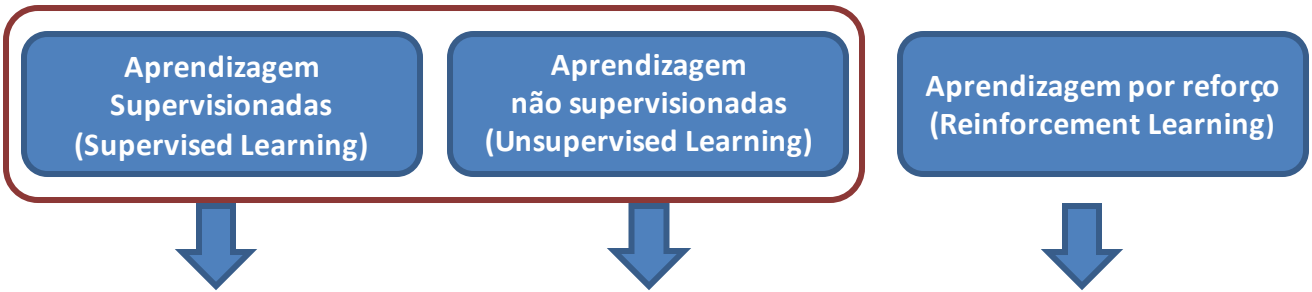
INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Aplicação de Técnicas de Aprendizagem de Máquina utilizando R

Mário de Noronha Neto e Richard Demo Souza

Alguns tipos de aprendizado de máquina

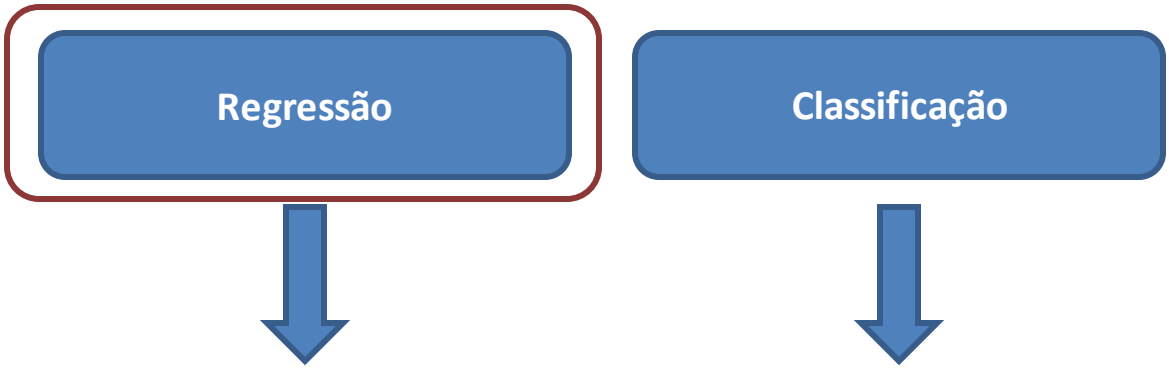


O processo de treinamento do modelo é realizado com um conjunto de dados em que as entradas e saídas são conhecidas

O processo de treinamento do modelo é realizado com um conjunto de dados em que apenas as entradas são fornecidas

O processo de treinamento é realizado a partir de recompensas nas interações com o ambiente

Técnicas de Aprendizagem Supervisionada abordadas neste curso:



Regressão

As técnicas de regressão normalmente são utilizadas para modelar relações complexas entre dados, estimando o impacto das variáveis no resultado de saída e extrapolando esta relação para resultados futuros.

Esta técnicas pode ser aplicada em diversas tarefas, entre elas podemos citar:

- Quantificação da relação causal entre um evento e a resposta, como por exemplo em ensaios clínicos de medicamentos, testes de segurança de engenharia ou pesquisa de marketing

- Identificação de padrões que podem ser usados para prever comportamentos futuros, como por exemplo previsão de sinistros, danos causados por desastres naturais, resultados eleitorais e taxas de criminalidade.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Regressão

- Quando o modelo de regressão é representado por uma reta, chamamos o modelo de Regressão Linear. Neste caso, podemos ter os modelos de Regressão Linear Simples (uma única variável independente) ou de Regressão Linear Múltipla (duas ou mais variáveis independentes).

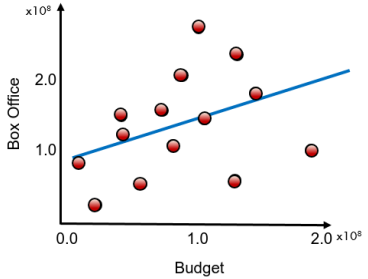
- A regressão também pode ser aplicada em outras formas de relação entre as variáveis independentes e dependente (ex.: Regressão Polinomial) e também em algumas tarefas de classificação (ex.: Regressão Logística).

Fonte: Brett Lantz – Machine Learning with R – Second Edition

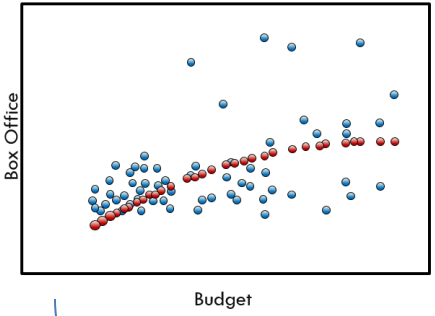
Regressão - Exemplos

$$y_\beta(x) = \beta_0 + \beta_1 x$$

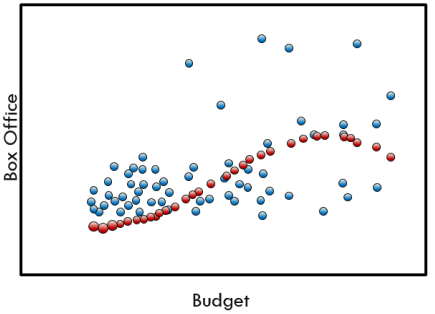
Regressão Linear Simples



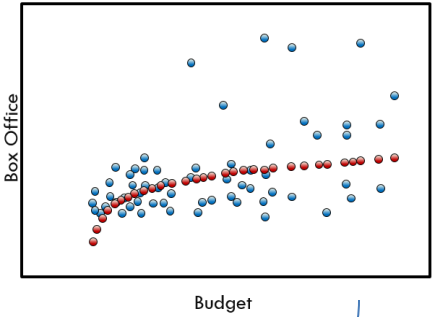
$$y_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$



$$y_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



$$y_\beta(x) = \beta_0 + \beta_1 \log(x)$$

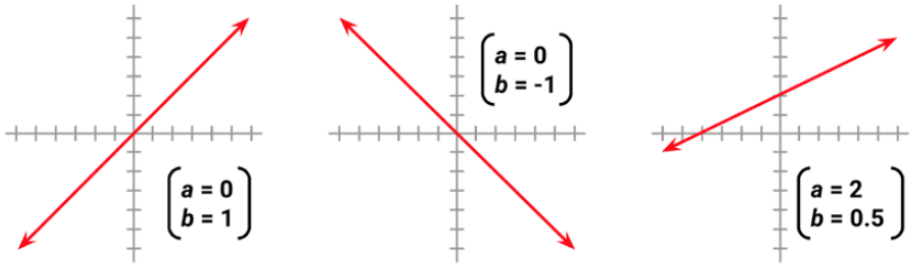


Regressão polinomial

Fonte: Material do Programa INTEL NERVANA AI ACADEMY, disponibilizado ao IFSC

Regressão Linear Simples

O método de Regressão Linear Simples consiste em especificar uma relação entre uma variável **numérica dependente** (valor a ser previsto) com uma variável **numérica independente** (preditora) através de uma reta.



$$y = a + bx$$

Variável dependente

Variável independente

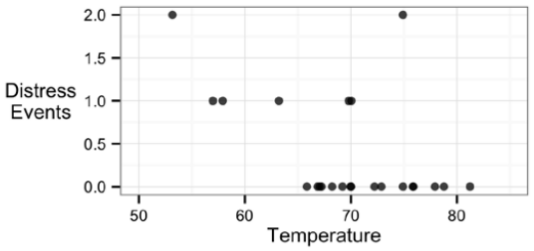
interceptação do eixo vertical

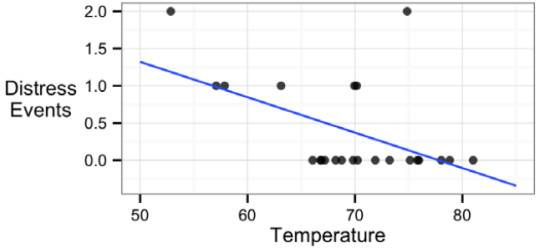
Inclinação da reta **(slope)**

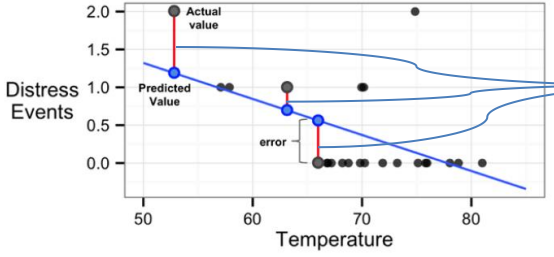
Neste exemplo, o objetivo (trabalho que a máquina realizará) é encontrar valores de '*a*' e *'b'* que **representem da melhor forma** a relação entre *'x'* e *'y'*.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Regressão Linear Simples







Os erros são conhecidos como **resíduos**

O objetivo é minimizar a soma dos quadrados dos resíduos (erros)

$$\sum (y_i - \hat{y}_i)^2 = \sum e_i^2$$

Definição do erro (*e*) em função da diferença do **valor atual** para o **valor previsto**

valor atual

valor previsto

Fonte: Brett Lantz – Machine Learning with R – Second Edition

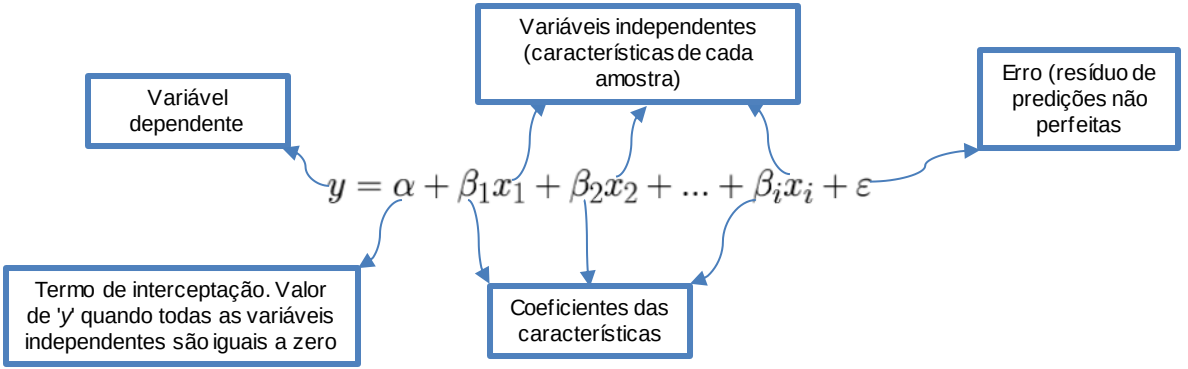
Regressão Linear Múltipla

A maioria das análises de casos reais que utilizam a técnica de regressão linear, possui mais de uma variável independente. Portanto, na prática, a Regressão Linear Múltipla é mais usada do que a Regressão Linear Simples. Algumas características desta técnica são:

Strengths	Weaknesses
• By far the most common approach for modeling numeric data • Can be adapted to model almost any modeling task • Provides estimates of both the strength and size of the relationships among features and the outcome	• Makes strong assumptions about the data • The model's form must be specified by the user in advance • Does not handle missing data • Only works with numeric features, so categorical data requires extra processing • Requires some knowledge of statistics to understand the model

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Regressão Linear Múltipla



Considerando que o termo interceptador é uma constante como qualquer outro coeficiente, podemos denotá-lo como '*βo'*. Como o termo interceptador é descorrelacionado de qualquer variável independente, podemos considerar que '*βo'* seja multiplicado por um termo *'xo',* o qual é uma constante de valor 1. Desta forma, a expressão da Regressão Linear Múltipla pode ser escrita como:

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i + \varepsilon$$

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Correlação e Multicolinearidade entre variáveis

A correlação é um número que mede a relação entre duas variáveis. No caso de uma correlação linear, esses valores podem variar entre –1 e 1. Os valores extremos indicam perfeita relação, enquanto valores próximos do zero indicam que não há relação linear entre as variáveis.

A multicolinearidade acontece quando duas variáveis independentes são fortemente correlacionadas.

Uma correlação elevada entre as variáveis dependente e independente é algo positivo, dado que o objetivo é prever o valor da variável dependente através de valores de variáveis independentes.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Exemplo: Predição de despesas médicas

Passo 1: Coleta de dados

Dataset utilizado:

O *dataset* utilizado neste exemplo (insurance.*csv*) contém despesas médicas hipotéticas para pacientes nos EUA. Este conjunto de dados foi elaborado pelo autor do livro "Machine Learning with R" com base em dados demográficos do *US Census Bureal*, portanto reflete condições reais. O *dataset* possui 1338 exemplos de beneficiários de planos de saúde com características do paciente, bem como o total de despesas médicas gastas por ano com o plano.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Exemplo: Predição de despesas médicas

Características dos dados:

- `age`: An integer indicating the age of the primary beneficiary (excluding those above 64 years, since they are generally covered by the government).
- `sex`: The policy holder's gender, either male or female.
- `bmi`: The body mass index (BMI), which provides a sense of how over- or under-weight a person is relative to their height. BMI is equal to weight (in kilograms) divided by height (in meters) squared. An ideal BMI is within the range of 18.5 to 24.9.
- `children`: An integer indicating the number of children/dependents covered by the insurance plan.
- `smoker`: A yes or no categorical variable that indicates whether the insured regularly smokes tobacco.
- `region`: The beneficiary's place of residence in the US, divided into four geographic regions: northeast, southeast, southwest, or northwest.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
> insurance <- read.csv("insurance.csv", stringsAsFactors = TRUE)

> str(insurance)
'data.frame':     1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 4 levels "northeast","northwest",..: ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
```

Variáveis independente

Variável dependente

Variáveis categóricas!

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

Como modelos de regressão necessitam que todas as variáveis sejam numéricas, as variáveis categóricas devem ser convertidas/codificadas para variáveis numéricas. Uma forma de fazer esta conversão é utilizando técnica **Dummy coding**. Algumas funções do R fazem isto automaticamente.

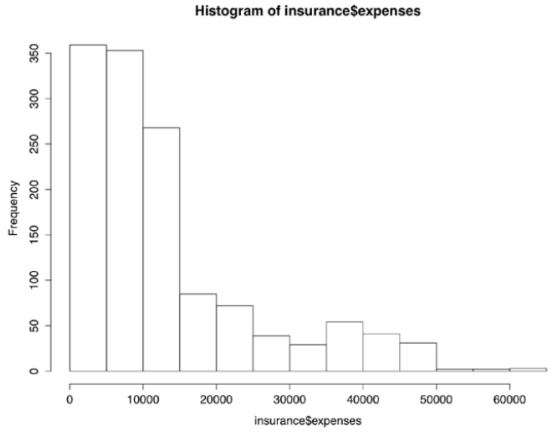
Region
northwest
northeast
southeast

Regionnorthwest	Regionsoutheast	Regionsouthwest	Regionnortheast
1	0	0	0
0	0	0	1
0	1	0	0

Passo 2: Explorando e preparando os dados

```
> summary(insurance$expenses)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1122    4740    9382   13270   16640   63770
```

```
> hist(insurance$expenses)
```




```
> table(insurance$region)

northeast northwest southeast southwest
      324       325       364       325
> table(insurance$sex)

female   male
   662    676
> table(insurance$smoker)

  no  yes
1064  274
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

Matriz de Correlação

```
> cor(insurance[c("age", "bmi", "children", "expenses")])
                age        bmi   children   expenses
age      1.0000000 0.10934101 0.04246900 0.29900819
bmi      0.1093410 1.00000000 0.01264471 0.19857626
children 0.0424690 0.01264471 1.00000000 0.06799823
expenses 0.2990082 0.19857626 0.06799823 1.00000000
```

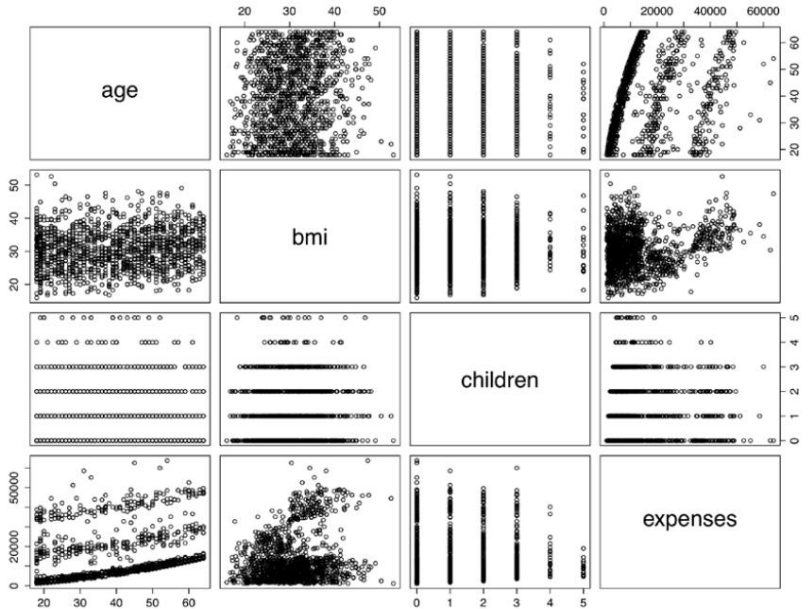
A matriz de correlação é simétrica, ou seja, cor(x, y) = cor(y,x). Além disto, os elementos da diagonal serão sempre iguais a 1, pois existe uma correlação perfeita entre a variável e ela mesmo.

Nenhum dos elementos da matriz possui forte correlação, entretanto podemos observar que existe uma correlação positiva entre *age* e *expenses*, *bmi* e *expenses*, e *children* e *expenses*. Isto significa que um aumento em *age*, *bmi* e *children*, implica em um aumento em *expenses.*

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
> pairs(insurance[c("age", "bmi", "children", "expenses")])
```

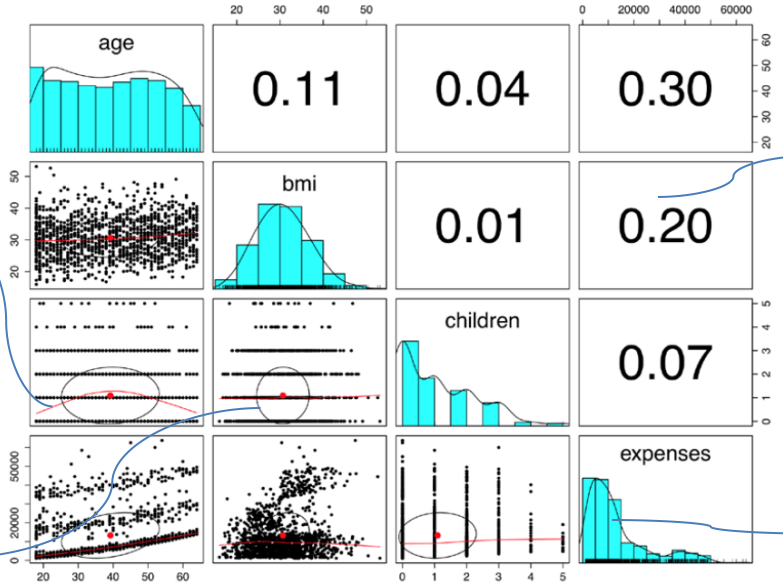


Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
> pairs.panels(insurance[c("age", "bmi", "children", "expenses")])
```

Loess Curve: Indica a relação entre as variáveis. Neste caso, observamos que passoas mais novas e mais velhas tem menos filhos. Relação não linear!



Matriz de correlação

Elipse de correlação: Fornece uma vizualização da da intensidade de correlação. O ponto central indica o ponto do valor médio para as variáveis *x* e *y*. Quanto mais oval for a elipse, menos correlação entre as variáveis.

Histograma

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 3: Treinando o modelo

Para a regressão linear, utilizaremos a função lm*()*. Esta função está inclusa no pacote *stats*, já incluso na instalação padrão do R.

Multiple regression modeling syntax

using the `lm()` function in the `stats` package

Building the model:

```
m <- lm(dv ~ iv, data = mydata)
```

- `dv` is the dependent variable in the `mydata` data frame to be modeled
- `iv` is an R formula specifying the independent variables in the `mydata` data frame to use in the model
- `data` specifies the data frame in which the `dv` and `iv` variables can be found

The function will return a regression model object that can be used to make predictions. Interactions between independent variables can be specified using the * operator.

Making predictions:

```
p <- predict(m, test)
```

- `m` is a model trained by the `lm()` function
- `test` is a data frame containing test data with the same features as the training data used to build the model.

The function will return a vector of predicted values.

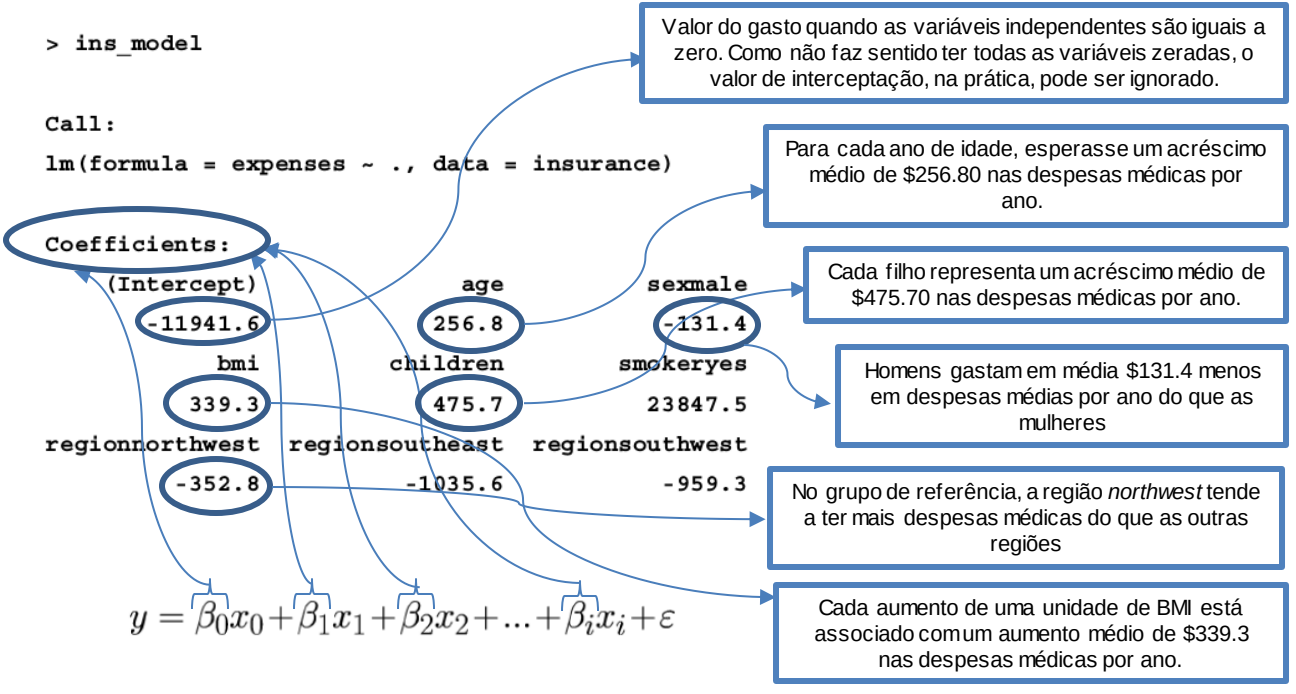
Example:

```
ins_model <- lm(charges ~ age + sex + smoker,
                data = insurance)
ins_pred <- predict(ins_model, insurance_test)
```

```
> ins_model <- lm(expenses ~ age + children + bmi + sex +
    smoker + region, data = insurance)
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Analisando o modelo



Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 4: Avaliando o desempenho do modelo

```
> summary(ins_model)

Call:
lm(formula = expenses ~ ., data = insurance)

Residuals:
     Min       1Q   Median       3Q      Max
-11302.7  -2850.9   -979.6   1383.9  29981.7

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -11941.6      987.8 -12.089  < 2e-16 ***
age                256.8       11.9  21.586  < 2e-16 ***
sexmale           -131.3      332.9  -0.395 0.693255
bmi                339.3       28.6  11.864  < 2e-16 ***
children           475.7      137.8   3.452 0.000574 ***
smokeryes        23847.5      413.1  57.723  < 2e-16 ***
regionnorthwest   -352.8      476.3  -0.741 0.458976
regionsoutheast  -1035.6      478.7  -2.163 0.030685 *
regionsouthwest   -959.3      477.9  -2.007 0.044921 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,  Adjusted R-squared:  0.7494
F-statistic: 500.9 on 8 and 1329 DF,  p-value: < 2.2e-16
```

1

2

3

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 4: Avaliando o desempenho do modelo

1. Resíduos: fornece um sumário estatístico para os erros (resíduos) na predição. Por exemplo, um erro máximo de 29981.7 indica que para este caso o modelo errou de aproximadamente $30,000.00 abaixo do valor real. Por outro lado, 50% (entre o primeiro e terceiro quartil) foram de $2,850.90 acima do valor real e de $1,383,90 abaixo do valor real.

2. p-value: Indica o nível de significância da característica. Valore muito pequenos sugerem que é extremamente improvável que a característica não tenha relação com a variável dependente. (***) indica o grau máximo de significância.

3. R-squared value: Fornece uma medida de quão bem o modelo como um todo explica os valores da variável independente. É similar ao coeficiente de correlação, em que quanto mais próximo de 1, melhor o modelo representa/explica os dados. Neste exemplo, podemos dizer que nosso modelo explica em torno de 75% os dados analisados. O **adjusted R-squared value** corrige o R-squared penalizando modelos com muitas características.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Atividades

Como observado anteriormente, a relação entre idade e despesas médicas não é constante para todos os valores de idade. Desta forma pode-se fazer uso de outras relações (não lineares) para tentar melhorar o desempenho do sistema. Tente adicionar a seguinte característica:

```
> insurance$age2 <- insurance$age^2
```

Inclua esta nova característica no modelo e compare com o modelo inicial. Observe o **R-squared value** e o **adjusted R-squared value**

Atividades

Algumas características não são cumulativas e tem efeito apenas após um determinado valor. No exemplo apresentado neste encontro, observamos que o BMI tem impacto muito baixo em indivíduos dentro da escala normal, mas impacto elevado em indivíduos com valores acima de 30. É possível criar um indicador binário de obesidade e incluí-lo como característica. Para isso, execute o seguinte comando:

```
> insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)
```

Inclua esta nova característica no modelo e compare com o modelo inicial. Observe o **R-squared value** e o **adjusted R-squared value**

Atividades

Em algumas situações algumas características podem ter impactos combinados na variável dependente. Por exemplo, fumantes e obesidade podem ter efeitos separados, mas quando combinados podem ter efeitos piores do que a soma dos efeitos considerados de forma isolada. No R é possível combinar o efeito de duas características através do operador (*). Para isso, crie o modelo da seguinte forma:

```
ins_model31 <- lm(expenses ~ age + children + bmi + sex +
                    bmi*smoker + smoker + region, data = insurance)
```

Inclua esta nova característica no modelo e compare com o modelo inicial. Observe o **R-squared value** e o **adjusted R-squared value**

Atividades

Agora inclua todas as modificações feitas anteriormente e avalie o resultado.

Exemplo: Predizendo a qualidade de um vinho

Este exemplo foi retirado de um curso ofertado pelo MIT: https://www.edx.org/course/analytics-edge-mitx-15-071x-3

O objetivo e realizar uma análise sobre a qualidade do vinho com base em características como condições de chuva, temperatura e idade. Explore o arquivo wine_ex.r e analise os resultados.