



# PREPARAÇÃO DOS DADOS

André Gustavo Adami  
Daniel Luis Notari

# INTRODUÇÃO

Muito do trabalho realizado em um projeto de aprendizado de máquina está focado na coleta, análise e preparação dos dados

A análise nos permite entender melhor os dados e selecionar os métodos/técnicas/algoritmos que deverão ser aplicados para produzir modelos que permitam discriminar classes ou extrair conhecimento dos dados

A preparação visa eliminarmos qualquer redundância ou dado que não contribua para o objetivo proposto



# INTRODUÇÃO

Nos projetos de aprendizado de máquina, empresas gastam em torno 80% do tempo em rotulamento dos dados, limpeza e preparação

Existem mais passos nestas tarefas do que construção dos modelos, ciência dos dados e implantação do sistema

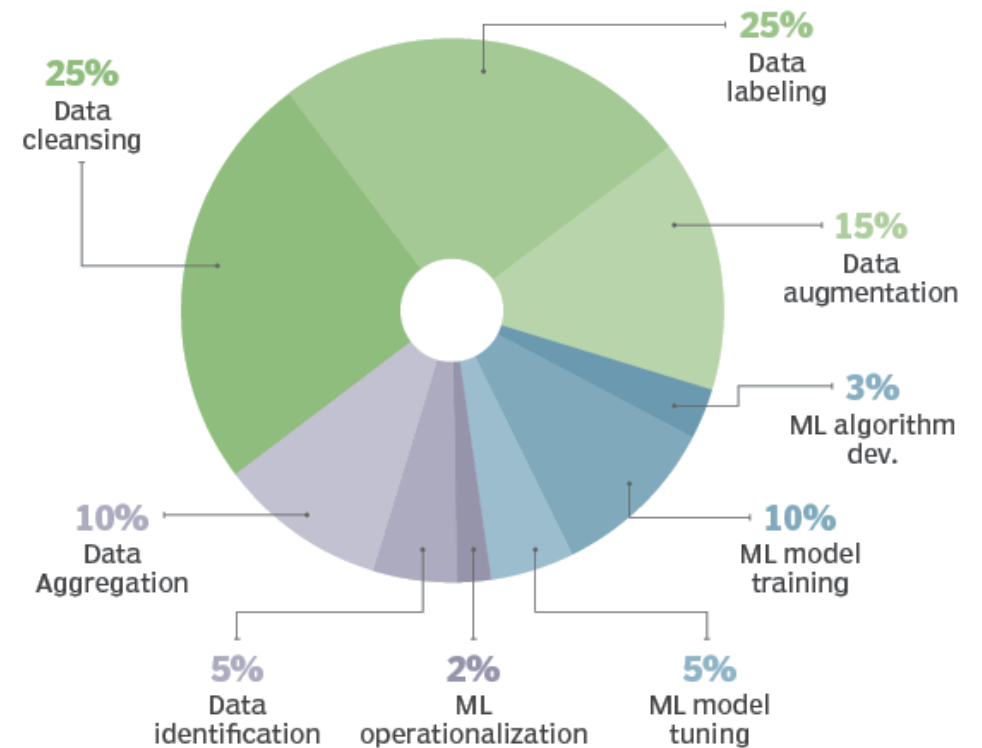
O envolvimento de humanos (capacitados) em todas as etapas ainda é necessário

<https://searchenterpriseai.techtarget.com/feature/Data-preparation-for-machine-learning-still-requires-humans>

<https://towardsdatascience.com/why-ai-models-absolutely-need-humans-to-stay-awesome-8fce149a8bf>

<https://www.found.co.uk/blog/human-oversight-machine-learning>

<https://www.kdnuggets.com/2022/07/data-preparation-raw-data-machine-learning.html>



# INTRODUÇÃO

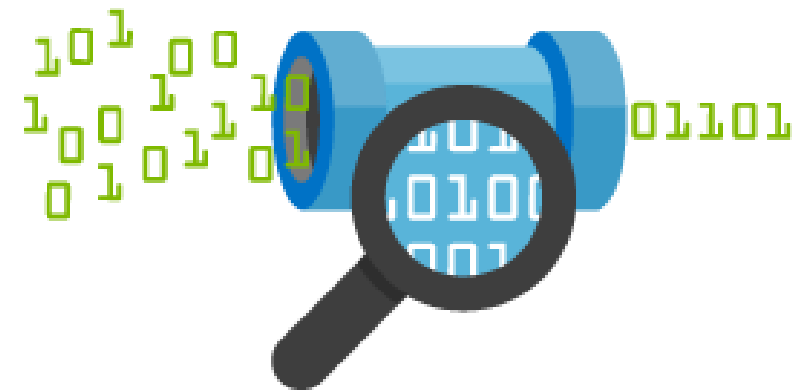
Alguns dos passos na preparação dos dados



Integração dos Dados



Limpeza dos Dados



Normalização/Transformação  
dos Dados

<https://www.kdnuggets.com/2022/07/data-preparation-raw-data-machine-learning.html>

# LIMPEZA DOS DADOS



Harvard  
Business  
Review

DATA

## Bad Data Costs the U.S. \$3 Trillion Per Year

by Thomas C. Redman

September 22, 2016

Dados com problemas podem  
levar a resultados com problemas



O processo de limpeza dos dados tem por objetivo corrigir ou remover dados incorretos, corrompidos, formatados incorretamente, irrelevantes, duplicados, incompletos de um conjunto de dados

Quanto melhor os dados,  
mais simples serão os modelos!

# LIMPEZA DOS DADOS

O processo de limpeza pode variar de um conjunto de dados para outro

Algumas estratégias para limpeza dos dados

1. Remover valores irrelevantes, sem variabilidade (constantes) ou duplicados (redução da dimensionalidade, modelos mais simples)
2. Tratar dados faltantes
3. Verificar a validade dos dados (intervalo de valores correto / outliers, variáveis com um único tipo de dado, validação entre campos, mesma unidade de medida)

# LIMPEZA DOS DADOS: DADOS FALTANTES

A maioria dos algoritmos de aprendizado não lida com dados faltantes

Em um primeiro momento, devemos verificar se existem **dados faltantes**, isto é, alguns dados não estão disponíveis por alguma falha na coleta, problema no armazenamento ou característica dos dados



É importante tentar entender o porquê da falta de dados

- Acontece aleatoriamente em toda a base de dados (a falta não está relacionada com o valor da variável ou qualquer outra variável dos dados) – falhas
- Acontece aleatoriamente em função de um ou mais variáveis do conjunto de dados
- Acontece de forma não aleatória (proposital)

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

Crie um diretório de trabalho e grave o arquivo prep-dados.txt que está no AVA da disciplina

Crie um novo script e salve no diretório criado

- Não esqueça de definir como diretório de trabalho: `setwd()`
- Limpar todas as variáveis de memória: limpar: `rm(list=ls())`

Faça a leitura do arquivo prep-dados.txt

```
dados = read.csv("prep-dados.txt", header=T)
```

ou

```
dados = read.table(file.choose(), header=T, sep=",")
```



# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

Vamos analisar os dados...

```
> str(dados)
```

```
'data.frame':      8143 obs. of  7 variables:
```

```
$ date      : chr  "2015-02-04 17:51:00" "2015-02-04 17:51:59" "2015-02-04 17:53:00" "2015-02-04 17:54:00" ...
```

```
$ Temperature : chr  "?" "23.15" "23.15" "23.15" ...
```

```
$ Humidity    : chr  "27.272" "27.2675" "27.245" "27.2" ...
```

```
$ Light       : num  426 430 426 426 426 ...
```

```
$ CO2         : num  NaN 714 714 708 704 ...
```

```
$ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
```

```
$ Occupancy   : int  1 1 1 1 1 1 1 1 1 1 ...
```

**Algum problema?**

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

Qual é a temperatura média?

```
> mean(dados$Temperature)
```

```
[1] NA
```

Warning message:

In mean.default(dados\$Temperature) :

argumento não é numérico nem lógico: retornando NA

Qual é o intervalo e o desvio padrão de CO2?

```
> range(dados$CO2)
```

```
[1] NaN NaN
```

```
> sd(dados$CO2)
```

```
[1] NA
```

Cuidado com valores “Not Applicable”,  
“NA”, “None”, “Null”, “NaN” ou “INF”!  
Eles podem representar a mesma coisa:  
**o valor está faltando**

# LIMPEZA DOS DADOS: DADOS FALTANTES

Entenda que não existe uma receita pronta como lidar, por isso, é difícil de estabelecer um senso comum nesta tarefa

Pode-se categorizar as técnicas em dois grupos

- **Substituir os valores faltantes:** a questão é que valor deverá ser utilizado na substituição? Esta substituição vai adicionar nova informação ou somente será um reflexo dos padrões já encontrados nos dados? É possível resulta em um viés nos dados (**bias**)
- **Remover as observações ou variáveis:** o problema é que remover observações reduz a quantidade de amostras para estimar o modelo. A informação faltante pode ser informativo em algumas situações. E em algumas aplicações, o problema em si tem esta característica de nem sempre ter todas as informações

# LIMPEZA DOS DADOS: DADOS FALTANTES

No caso de substituição dos valores faltantes, é possível

1. Substituir pela média/mediana/moda com base em todos os dados ou dados da classe
2. Utilizar alguma forma de regressão para estimar os valores faltantes
3. Copiar/estimar os valores a partir de observações similares (próximas)

No caso de remoção de dados, é possível

1. Se os valores faltantes são aleatórios, pode-se remover as observações (ou linhas)
2. No caso de valores faltantes em uma variável, pode-se remover somente a variável
  - Existem diversas regras de quantidade mínima:  $> 50\%$ ,  $> 65\%$ , ... dos dados faltando

Estas opções são **sub-ótimas** e, por isso, a sua utilização deve ser avaliada cuidadosamente

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

No nosso arquivo de dados de exemplo, verificamos que alguns valores estão faltando

- Valores ausentes podem ser identificados pelas “?” ou “NA” (Not Available)
- Problemas de condicionamento numérico podem resultar em valores “NaN” (*Not a Number*)

Façamos a leitura do arquivo `prep-dados.txt` levando em conta esta informação

```
dados = read.csv("prep-dados.txt", header=T, na.strings="?")
```


- O NaN é um valor reconhecido por qualquer linguagem de programação, por isso não precisa ser tratado na leitura). É tratado como NA

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

## Verificar a estrutura do data.frame

`str(dados)`

```
'data.frame':      8143 obs. of  7 variables:
 $ date          : chr  "2015-02-04 17:51:00" "2015-02-04 17:51:59" "2015-02-04 17:53:00" "2015-02-04
17:54:00" ...
 $ Temperature   : num  NA 23.1 23.1 23.1 23.1 ...
 $ Humidity      : num  27.3 27.3 27.2 27.2 27.2 ...
 $ Light         : num  426 430 426 426 426 ...
 $ CO2           : num  NaN 714 714 708 704 ...
 $ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
 $ Occupancy     : int  1 1 1 1 1 1 1 1 1 1
```



Visualização dos  
dados  
`View(dados)`

Os valores ausentes foram substituídos por NA! Mas continuam a existir no conjunto de dados

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

## Estimar algumas medidas estatísticas dos dados

`summary(dados)`

	date		Temperature		Humidity		Light		CO2		HumidityRatio		Occupancy
2015-02-04 17:51:00:	1	Min.	:19.00	Min.	:16.75	Min.	: 0.0	Min.	: 412.8	Min.	:0.002674	Min.	:0.0000
2015-02-04 17:51:59:	1	1st Qu.:	19.70	1st Qu.:	20.29	1st Qu.:	0.0	1st Qu.:	439.0	1st Qu.:	0.003078	1st Qu.:	0.0000
2015-02-04 17:53:00:	1	Median	:20.50	Median	:26.25	Median	: 0.0	Median	: 453.5	Median	:0.003801	Median	:0.0000
2015-02-04 17:54:00:	1	Mean	:20.64	Mean	:25.75	Mean	: 119.5	Mean	: 606.5	Mean	:0.003863	Mean	:0.2123
2015-02-04 17:55:00:	1	3rd Qu.:	21.39	3rd Qu.:	30.53	3rd Qu.:	256.4	3rd Qu.:	638.4	3rd Qu.:	0.004352	3rd Qu.:	0.0000
2015-02-04 17:55:59:	1	Max.	:23.15	Max.	:39.12	Max.	:1546.3	Max.	:2028.5	Max.	:0.006476	Max.	:1.0000
(Other)	:8137	NA's	:209	NA's	:64			NA's	:1				

A leitura somente padronizou que os valores ausentes fossem interpretados ausentes

`colSums(is.na(dados))`

date	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
0	209	64	0	1	0	0

Como não são muitos valores, vamos remover as linhas que contém estes dados ausentes

# LIMPEZA DOS DADOS: DADOS FALTANTES - PRÁTICA

A remoção pode ser realizada com a função `na.omit()`

```
dados = na.omit(dados)
```

```
summary(dados)
```

	date		Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
2015-02-04 17:51:59:	1	Min.	:19.00	Min. :16.75	Min. : 0.0	Min. : 412.8	Min. :0.002678	Min. :0.0000
2015-02-04 17:53:00:	1	1st Qu.:	19.70	1st Qu.:20.50	1st Qu.: 0.0	1st Qu.: 439.0	1st Qu.:0.003125	1st Qu.:0.0000
2015-02-04 17:54:00:	1	Median :	20.50	Median :26.29	Median : 0.0	Median : 454.5	Median :0.003809	Median :0.0000
2015-02-04 17:55:00:	1	Mean :	20.64	Mean :25.86	Mean : 122.3	Mean : 611.2	Mean :0.003886	Mean :0.2168
2015-02-04 17:55:59:	1	3rd Qu.:	21.39	3rd Qu.:30.60	3rd Qu.: 272.8	3rd Qu.: 655.4	3rd Qu.:0.004359	3rd Qu.:0.0000
2015-02-04 17:57:00:	1	Max.	:23.15	Max. :39.12	Max. :1546.3	Max. :2028.5	Max. :0.006476	Max. :1.0000

Como eram os valores antes da operação?

	date		Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
2015-02-04 17:51:00:	1	Min.	:19.00	Min. :16.75	Min. : 0.0	Min. : 412.8	Min. :0.002674	Min. :0.0000
2015-02-04 17:51:59:	1	1st Qu.:	19.70	1st Qu.:20.29	1st Qu.: 0.0	1st Qu.: 439.0	1st Qu.:0.003078	1st Qu.:0.0000
2015-02-04 17:53:00:	1	Median :	20.50	Median :26.25	Median : 0.0	Median : 453.5	Median :0.003801	Median :0.0000
2015-02-04 17:54:00:	1	Mean :	20.64	Mean :25.75	Mean : 119.5	Mean : 606.5	Mean :0.003863	Mean :0.2123
2015-02-04 17:55:00:	1	3rd Qu.:	21.39	3rd Qu.:30.53	3rd Qu.: 256.4	3rd Qu.: 638.4	3rd Qu.:0.004352	3rd Qu.:0.0000
2015-02-04 17:55:59:	1	Max.	:23.15	Max. :39.12	Max. :1546.3	Max. :2028.5	Max. :0.006476	Max. :1.0000
		NA's	:209	NA's :64		NA's :1		



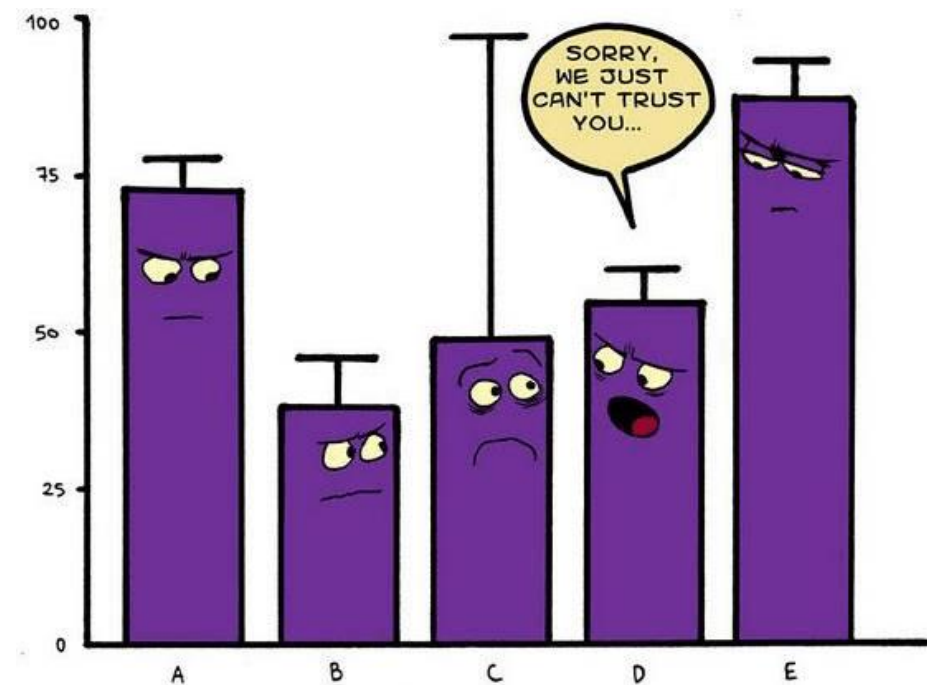
# DETECÇÃO DE VALORES ATÍPICOS

Um dos problemas que os dados podem apresentar é a ocorrência de valores atípicos (*outliers*)

- Problemas na medição, entrada dos dados, condução do experimento, processamento, amostragem, integração de diferentes sistemas, entre outros

Estes precisam ser identificados, avaliados e removidos, se necessário

**Lembre-se: só porque um outlier existe, não quer dizer que ele é um valor incorreto**



# DETECÇÃO DE VALORES ATÍPICOS

O processo de detecção pode ser realizado em função de uma ou múltiplas variáveis e a distribuição destas variáveis pode ser determinada (paramétrico) não (não-paramétrico)

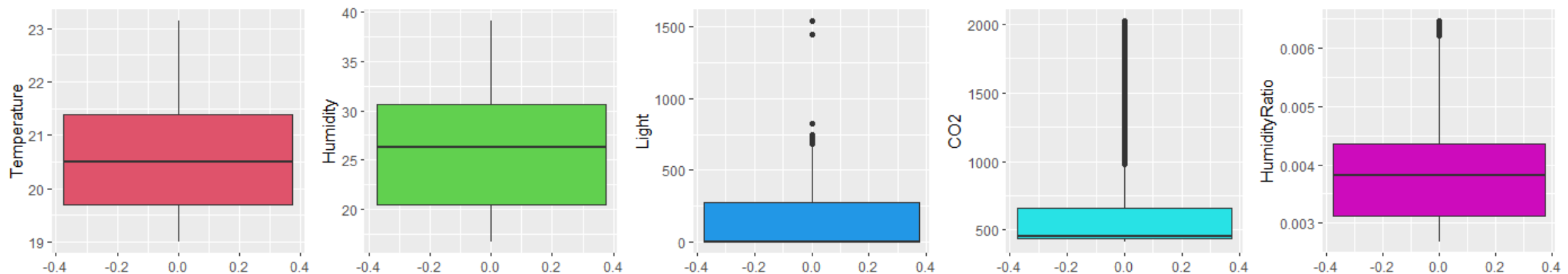
## Métodos

- Z-Score ou Análise do valor extremo (paramétrico)
- Modelos estatísticos ou probabilísticos (paramétrico)
- Métodos de projeção (não-paramétrico)
- ...

# DETECÇÃO E REMOÇÃO DE OUTLIERS - PRÁTICA

Avaliando o gráfico boxplot dos dados, pode-se verificar que os atributos Light, CO2 e HumidityRatio possuem valores atípicos

```
library(gridExtra)
library(ggplot2)
p = list()
for (i in 2:6) {
  p[[i-1]] = ggplot(dados, aes_string(y=names(dados)[i])) + geom_boxplot(fill = i) +
    theme(legend.position="none")
}
do.call(grid.arrange,c(p,ncol=5))
```

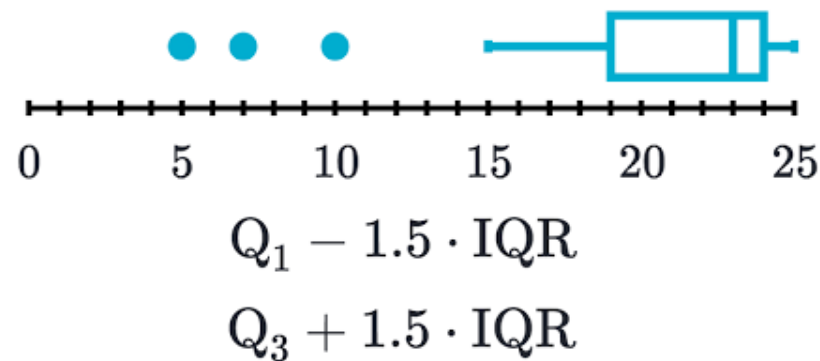


# DETECÇÃO E REMOÇÃO DE OUTLIERS - PRÁTICA

O método definido por Tukey (1977) utiliza o gráfico boxplots para detectar valores atípicos

Para uma variável contínua, valores atípicos são as observações que ficam fora de  $1,5 \cdot \text{IQR}$  (*Inter Quartile Range*)

- IQR é o intervalo entre o quartil de 25% ( $Q_1$ ) e o quartil 75% ( $Q_3$ )



Tukey, J.W., 1977. Exploratory Data Analysis. Addison-Wesley, Reading, MA.

# PRÁTICA: DETECÇÃO E REMOÇÃO DE OUTLIERS

Para estimar os valores atípicos

```
quartis = quantile(dados$Light, probs=c(.25, .75));  
limiar  = 1.5 * IQR(dados$Light)  
outliers = c(which(dados$Light < (quartis[1]-limiar)),  
              which(dados$Light > (quartis[2]+limiar)))
```

**OU**

```
outliers = which(dados$Light %in% boxplot.stats(dados$Light)$out)  
dadosSemOutliers = dados[-outliers,]
```

# PRÁTICA: DETECÇÃO E REMOÇÃO DE OUTLIERS

## Antes

summary(dados)

	date		Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
2015-02-04	17:51:59:	1	Min. :19.00	Min. :16.75	Min. : 0.0	Min. : 412.8	Min. :0.002678	Min. :0.0000
2015-02-04	17:53:00:	1	1st Qu.:19.70	1st Qu.:20.50	1st Qu.: 0.0	1st Qu.: 439.0	1st Qu.:0.003125	1st Qu.:0.0000
2015-02-04	17:54:00:	1	Median :20.50	Median :26.29	Median : 0.0	Median : 454.5	Median :0.003809	Median :0.0000
2015-02-04	17:55:00:	1	Mean :20.64	Mean :25.86	Mean : 122.3	Mean : 611.2	Mean :0.003886	Mean :0.2168
2015-02-04	17:55:59:	1	3rd Qu.:21.39	3rd Qu.:30.60	3rd Qu.: 272.8	3rd Qu.: 655.4	3rd Qu.:0.004359	3rd Qu.:0.0000
2015-02-04	17:57:00:	1	Max. :23.15	Max. :39.12	Max. :1546.3	Max. :2028.5	Max. :0.006476	Max. :1.0000

## Depois

summary(dadosSemOutliers)

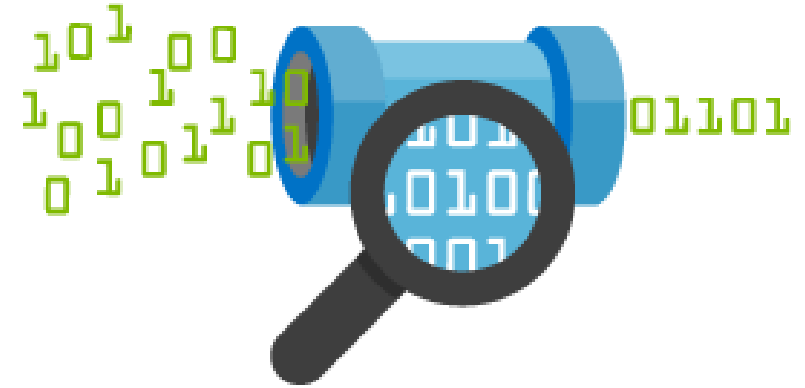
	date		Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
2015-02-04	17:51:59:	1	Min. :19.00	Min. :16.75	Min. : 0.00	Min. :412.8	Min. :0.002678	Min. :0.0000
2015-02-04	17:53:00:	1	1st Qu.:19.60	1st Qu.:19.89	1st Qu.: 0.00	1st Qu.:438.0	1st Qu.:0.003048	1st Qu.:0.0000
2015-02-04	17:54:00:	1	Median :20.29	Median :25.39	Median : 0.00	Median :449.5	Median :0.003738	Median :0.0000
2015-02-04	17:55:00:	1	Mean :20.47	Mean :24.96	Mean : 82.07	Mean :508.8	Mean :0.003698	Mean :0.1286
2015-02-04	17:55:59:	1	3rd Qu.:21.10	3rd Qu.:28.20	3rd Qu.: 29.33	3rd Qu.:481.5	3rd Qu.:0.004235	3rd Qu.:0.0000
2015-02-04	17:57:00:	1	Max. :23.15	Max. :36.26	Max. :611.50	Max. :979.0	Max. :0.005621	Max. :1.0000

**Recomenda-se primeiro obter resultados com os dados sem o tratamento de valores atípicos para comparação**

# TRANSFORMAÇÃO

A transformação de dados tem por objetivo alterar os dados de forma que mantenha a mesma informação, mas de uma forma que facilite a sua representação e manipulação pelos algoritmos de aprendizado de máquina

- Codificação de dados categóricos
  - Dados categóricos podem ter variações (caracteres especiais ou capitalização)
  - A padronização destes valores não só garante a qualidade dos dados, mas também economiza armazenamento
- Normalização/padronização
- Mitigação de viés (bias)
- Assimetria de distribuição (logaritmo, raiz cúbica ou quadrada)



# TRANSFORMAÇÃO

A falta ou baixo nível de padronização dos dados também podem criar dados inconsistentes

A padronização tem por objetivo resolver diferenças de unidades de medida, escalas e representações

- As unidades de medida devem ser as mesmas para o mesmo atributo (centímetros x metros)
- Formatos de dados devem ser os mesmos (representação de datas DDMMAAAA x AAAAMMDD)
- Correção de inconsistências de representação de dados categóricos (capitalização de letras, erros de grafia, espaços em branco, caracteres especiais)



# TRANSFORMAÇÃO

Datas e horas podem ser boas fontes de informação para determinados problemas

Uma técnica para utilizar datas ou horas em algoritmos de aprendizagem é convertê-los para outra representação

- Mês
- Ano
- Dia
- Dia da semana
- Semana do ano
- Hora
- Minuto corrido (1.440 minutos no dia)
- Segundo corrido (86.400 segundos no dia)



# TRANSFORMAÇÃO: PRÁTICA

Para manipular data e horas, pode-se utilizar o pacote lubridate

```
library(lubridate)
```

Converter a string para uma data (representação interna)

```
dados$date = ymd_hms(dados$date)
```

Minutos corrido no dia

```
dados$Minutos = int_length(interval(date(dados$date),  
                                     ymd_hms(dados$date))) / 60.0
```

Dia do mês

```
dados$Diames = day(date(dados$date))
```

# DADOS CATEGÓRICOS

Uma variável categórica pode assumir um conjunto limitado de valores (rótulos)

- gênero, estado civil, nacionalidade, tamanho de vestuário, estados, espécies, componentes, formação escolar, ...

Podem ser

- Nominais (categórica): conjunto finito de valores discretos, sem relacionamento entre eles
- Ordinais: conjunto finito de valores discretos, com uma ordem entre eles

Mas poucos algoritmos trabalham diretamente com variáveis categóricas ou quando a variabilidade é baixa (conjunto de valores pequenos)

- Máquinas de Vetor de Suporte, Árvores de Decisão, Modelos de Markov Discretos e K-vizinhos mais próximos

Uma maneira de lidar com dados categóricos é apagar 🤖, exceto quando eles podem trazer alguma informação

# DADOS CATEGÓRICOS

Uma técnica é convertê-los em valores numéricos

- **Codificação one-hot:** para um conjunto de  $c$  categorias, este método codifica ou transforma a variável em  $c$  variáveis binárias, onde a categoria é definida com o valor 1 na respectiva coluna
- **Codificação dummy:** semelhante ao *one-hot*, mas a codificação é em  $c - 1$  variáveis, onde uma das categorias é representada pelo valor 0 em todas as  $c - 1$  variáveis
- **Codificação ordinal:** um valor inteiro é atribuído a cada categoria/rótulo

Categoria	pequeno	médio	grande
Pequeno	1	0	0
Médio	0	1	0
Grande	0	0	1

Categoria	médio	grande
Pequeno	0	0
Médio	1	0
Grande	0	1

Categoria	Ordinal
Pequeno	0
Médio	1
Grande	2

# DADOS CATEGÓRICOS

Codificações *one-hot* e *dummy* são mais apropriadas para modelos lineares, mas aumenta o número de variáveis *Maldição da Dimensionalidade*

Codificação ordinal não aumenta o número de variáveis, mas não é apropriada para modelos lineares (o relacionamento de ordem pode não ser verdadeiro)

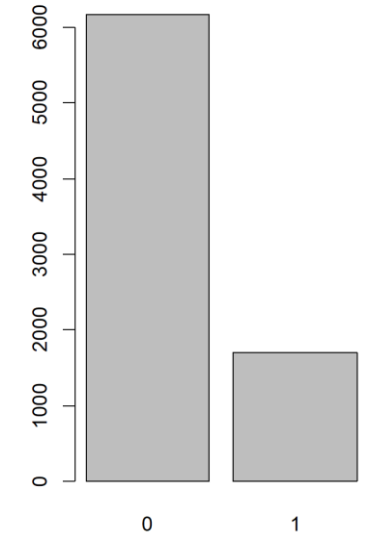


<https://heartbeat.fritz.ai/hands-on-with-feature-engineering-techniques-encoding-categorical-variables-be4bc0715394>

# TRANSFORMAÇÃO: PRÁTICA

Os rótulos das classes podem ser padronizados utilizando *factor*

```
unique(dados$Occupancy)
[1] 1 0
barplot(table(dados$Occupancy))
```

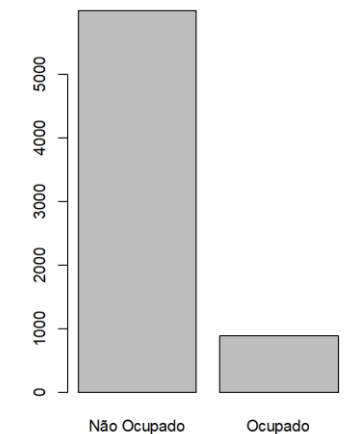


Além disso, garante a uniformidade dos valores

```
dados$Occupancy = factor(dados$Occupancy,
                          labels=c("Não Ocupado", "Ocupado"))
```

ou

```
dados$Occupancy = as.factor(dados$Occupancy);
levels(dados$Occupancy)[1] = "Não Ocupado"
levels(dados$Occupancy)[2] = "Ocupado"
```



# NORMALIZAÇÃO

Como os dados podem ter características que variam em magnitude, unidade e intervalo, esta variação pode afetar o algoritmo de aprendizado

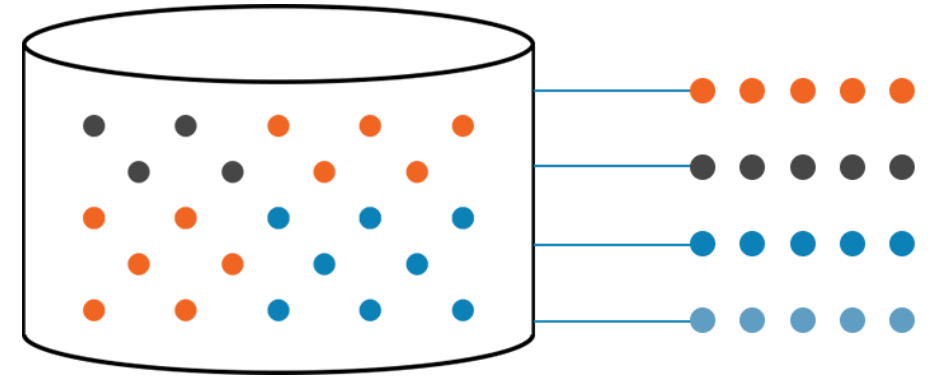
A normalização tem por objetivo transformar os dados a fim de torná-los mais apropriados para os algoritmos de aprendizagem de máquina, isto é, cada característica contribuirá igualmente



# NORMALIZAÇÃO

A normalização é importante porque

- No caso de classificadores que utilizam distância Euclidiana, a característica com o maior intervalo terá o maior efeito na distância
- No caso de classificadores que utilizam o gradiente descendente (redes neurais artificiais, por exemplo), a convergência é muito mais rápida quando as características são normalizadas
- Métodos de redução de dimensionalidade buscam direções que maximizem a variância também podem sofrer com diferentes intervalos de valores





# NORMALIZAÇÃO: MÉTODOS

**Max-min** $_{x,n}$ : transformação linear mapeia dados em um novo domínio  $[novomin_x, novomax_x]$

$$x' = \left( \frac{x - min_x}{max_x - min_x} \right) (novomax_x - novomin_x) + novomin_x$$

Muito utilizado para normalização para o intervalo  $[0;1]$ , isto é,  $novomin_x=0$  e  $novomax_x=1$

**Score-z (standard scaler)**: transformação para os dados tenham as propriedades de uma distribuição normal padrão com  $\mu=0$  and  $\sigma=1$

$$x' = \left( \frac{x - \mu_x}{\sigma_x} \right)$$

# NORMALIZAÇÃO: MÉTODOS

**Robust scaler:** remover a mediana e escala os dados de acordo com o intervalo inter-quartis (IQR). Deve ser robusto a outliers

$$x' = \left( \frac{x - \text{Mediana}_x}{IQR_x} \right)$$

# NORMALIZAÇÃO: PRÁTICA

## Max-min

```
maxmin = as.data.frame(lapply(dados[,2:6],  
                             function(y) (y - min(y))/(max(y) - min(y))))
```

```
summary(maxmin)
```

Temperature	Humidity	Light	CO2	HumidityRatio
Min. :0.0000	Min. :0.0000	Min. :0.00000	Min. :0.00000	Min. :0.0000
1st Qu.:0.1446	1st Qu.:0.1612	1st Qu.:0.00000	1st Qu.:0.04459	1st Qu.:0.1258
Median :0.3108	Median :0.4430	Median :0.00000	Median :0.06490	Median :0.3603
Mean :0.3541	Mean :0.4211	Mean :0.13420	Mean :0.16968	Mean :0.3467
3rd Qu.:0.5060	3rd Qu.:0.5871	3rd Qu.:0.04797	3rd Qu.:0.12141	3rd Qu.:0.5291
Max. :1.0000	Max. :1.0000	Max. :1.00000	Max. :1.00000	Max. :1.0000

# NORMALIZAÇÃO: PRÁTICA

## Score Z

```
escorez = as.data.frame(lapply(dados[,2:6], function(y) (y - mean(y))/sd(y))))
```

ou

```
escorez = as.data.frame(scale(dados[,2:6]))
```

```
summary(escorez)
```

Temperature	Humidity	Light	CO2	HumidityRatio
Min. : -1.5262	Min. : -1.63007	Min. : -0.5014	Min. : -0.7188	Min. : -1.48233
1st Qu.: -0.9030	1st Qu.: -1.00585	1st Qu.: -0.5014	1st Qu.: -0.5299	1st Qu.: -0.94430
Median : -0.1864	Median : 0.08487	Median : -0.5014	Median : -0.4439	Median : 0.05797
<b>Mean : 0.0000</b>	<b>Mean : 0.00000</b>	<b>Mean : 0.0000</b>	<b>Mean : 0.0000</b>	<b>Mean : 0.00000</b>
3rd Qu.: 0.6548	3rd Qu.: 0.64279	3rd Qu.: -0.3222	3rd Qu.: -0.2045	3rd Qu.: 0.77962
Max. : 2.7839	Max. : 2.24118	Max. : 3.2344	Max. : 3.5175	Max. : 2.79306

# NORMALIZAÇÃO: PRÁTICA

## Robust Scaler

```
robustScaler = as.data.frame(lapply(dados[,2:6],  
                                   function(y) (y - median(y))/IQR(y)))
```

```
summary(robustScaler)
```

Temperature	Humidity	Light	CO2	HumidityRatio
Min. : -0.8600	Min. : -1.04021	Min. : 0.000	Min. : -0.8448	Min. : -0.89349
1st Qu.: -0.4600	1st Qu.: -0.66159	1st Qu.: 0.000	1st Qu.: -0.2644	1st Qu.: -0.58139
<b>Median : 0.0000</b>	<b>Median : 0.00000</b>	<b>Median : 0.000</b>	<b>Median : 0.0000</b>	<b>Median : 0.00000</b>
Mean : 0.1197	Mean : -0.05148	Mean : 2.798	Mean : 1.3640	Mean : -0.03363
3rd Qu.: 0.5400	3rd Qu.: 0.33841	3rd Qu.: 1.000	3rd Qu.: 0.7356	3rd Qu.: 0.41861
Max. : 1.9067	Max. : 1.30793	Max. : 20.847	Max. : 12.1724	Max. : 1.58654

# PRÁTICA: NORMALIZAÇÃO DOS DADOS

Como será que fica a distribuição dos dados?

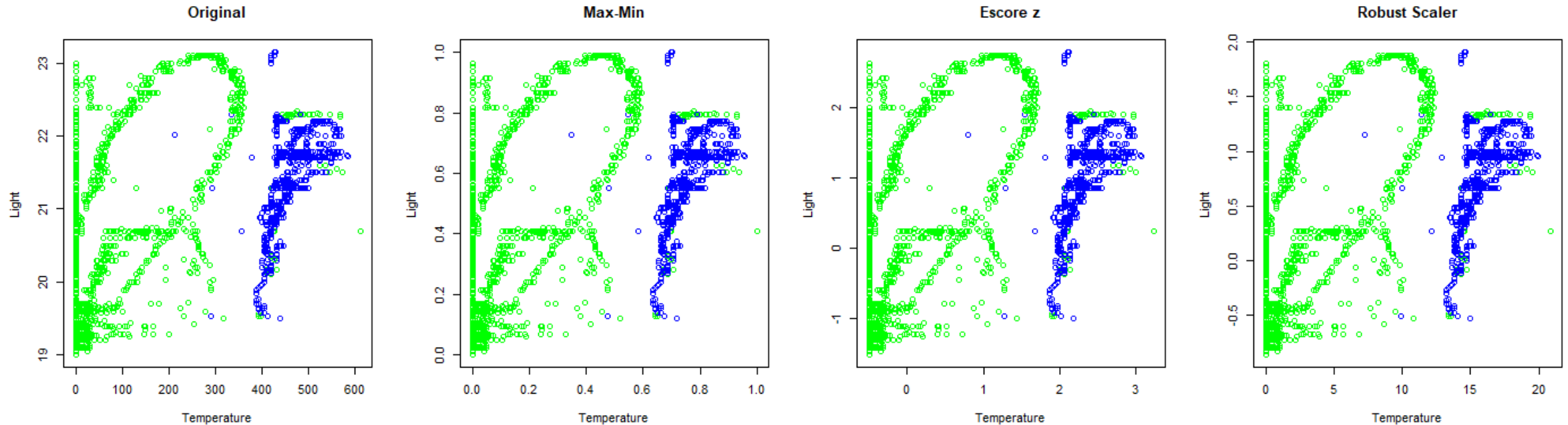
```
par(mfrow=c(1,3))  
cores = c("green","blue")  
plot(dados$Light, dados$Temperature,  
      col=cores[dados$Occupancy],  
      xlab='Temperature',ylab='Light',main="Original")
```

```
plot(maxmin$Light, maxmin$Temperature, col=cores[dados$Occupancy],  
      xlab='Temperature',ylab='Light', main="Max-Min")
```

```
plot(escorez$Light,escorez$Temperature, col=cores[dados$Occupancy],  
      xlab='Temperature',ylab='Light', main="Escore z")
```

```
plot(robustScaler$Light,robustScaler$Temperature,col=cores[dados$Occupancy],  
      xlab='Temperature',ylab='Light', main="Robust Scaler")
```

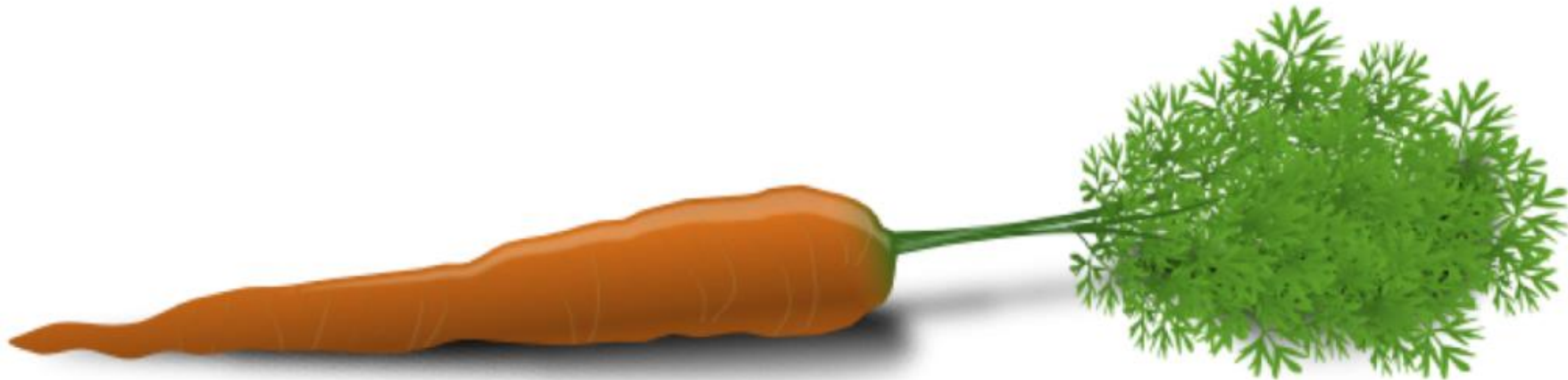
# PRÁTICA: NORMALIZAÇÃO DOS DADOS



**Alguma diferença?**

# PACOTE CARET

O pacote caret (**C**lassification **A**nd **R**Egression **T**raining) é um conjunto de funções para criar modelos preditivos (tanto para classificação como regressão)



```
install.packages("caret", dependencies = TRUE)
```

<http://topepo.github.io/caret/index.html>



# PACOTE CARET

## Limpeza dos Dados

- Detectar variáveis com um valor único (ou variância zero)

```
library(caret)  
indVariaveis = nearZeroVar(dados[,2:6])
```

## Transformação

```
→ dummyConv = dummyVars(~ ., data=dados,fullRank = F)  
→ novoDataFrame = data.frame(predict(dummyConv,newdata=dados))
```

Variável da classe deve ser adicionada, pois na conversão ela é removida

# PACOTE CARET

## Normalização

- Max-Min

```
maxminParams = preProcess(dados[,2:6], method=c("range"))  
maxmin = predict(maxminParams, dados[,2:6])
```

- Escore-Z

```
escorezParams = preProcess(dados[,2:6], method=c("center", "scale"))  
escorez = predict(escorezParams, dados[,2:6])
```

