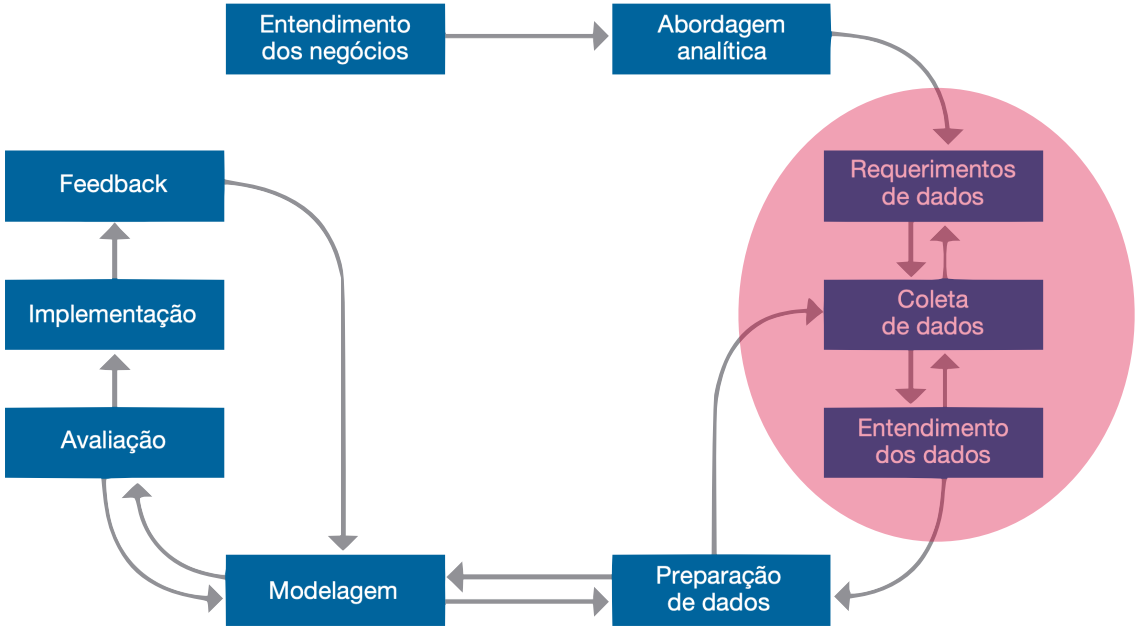
PROF. EDUARDO PÉCORA

AULA 02 – DATA SCIENCE PPGOLD

NOSSA CAMINHADA

- Ler um banco de dados usando o Python
- Gravar essas informações em um DataFrame
- Analisar a estrutura deste dataFrame
- Identificar possíveis problemas

AS 10 QUESTÕES-CHAVE



Trabalhando com os dados:

3) Qual dado será necessário?

4) De onde virão e como eu terei acesso?

5) Os dados que coletados são representativos?

6) Será necessário algum trabalho de manipulação adicional?

Source: Rollings John B., Foundational methodology for data science. IBM white paper
https://www.ibm.com/downloads/cas/B1WQ0GM2

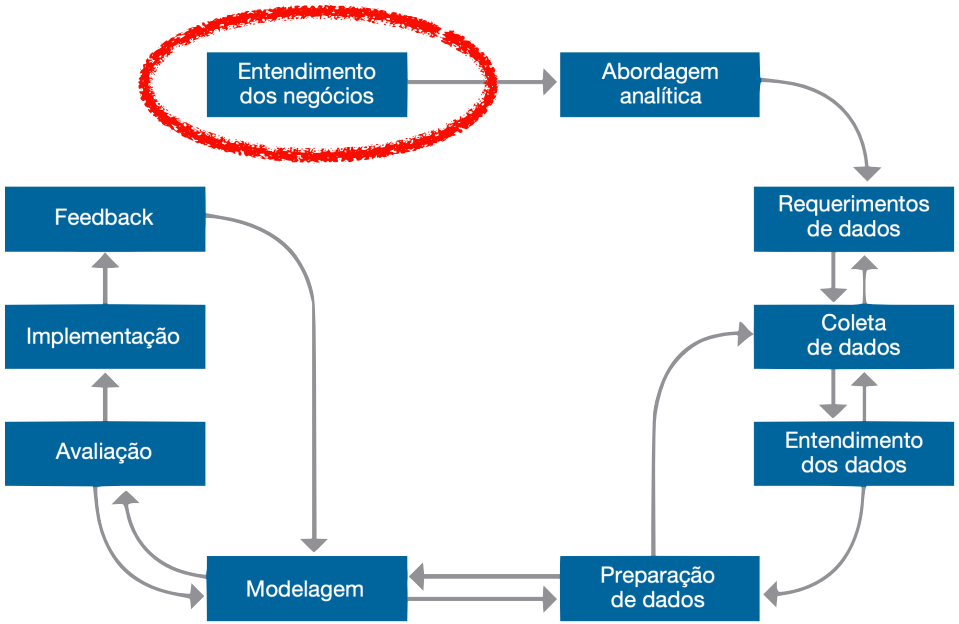
EDUARDO PÉCORA

DATA ANALYSIS WITH PYTHON

PORQUE DATA ANALYSIS?

- Os dados estão por toda a parte
- Ciência de dados nos ajuda a responder importantes questões com os dados
- Ciência de dados tem um papel fundamental em:
  - Descobrir informações úteis
  - Responder questões com base científica
  - Prever situações futuras e desconhecidas

ENTENDIMENTO DOS NEGÓCIOS



Qual é o problema que estou tentando resolver?

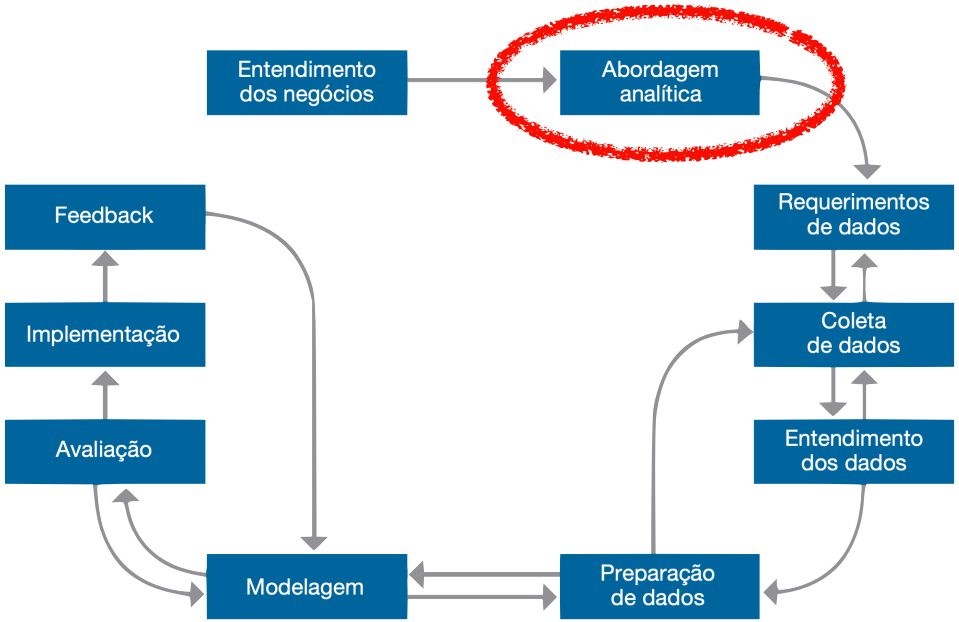
Source: Rollings John B., Foundational methodology for data science. IBM white paper
https://www.ibm.com/downloads/cas/B1WQ0GM2

CHICO QUER VENDER O SEU CARRO

Quanto ele deve pedir pelo carro?

O preço não pode ser nem muito alto nem muito baixo.

ABORDAGEM ANALÍTICA

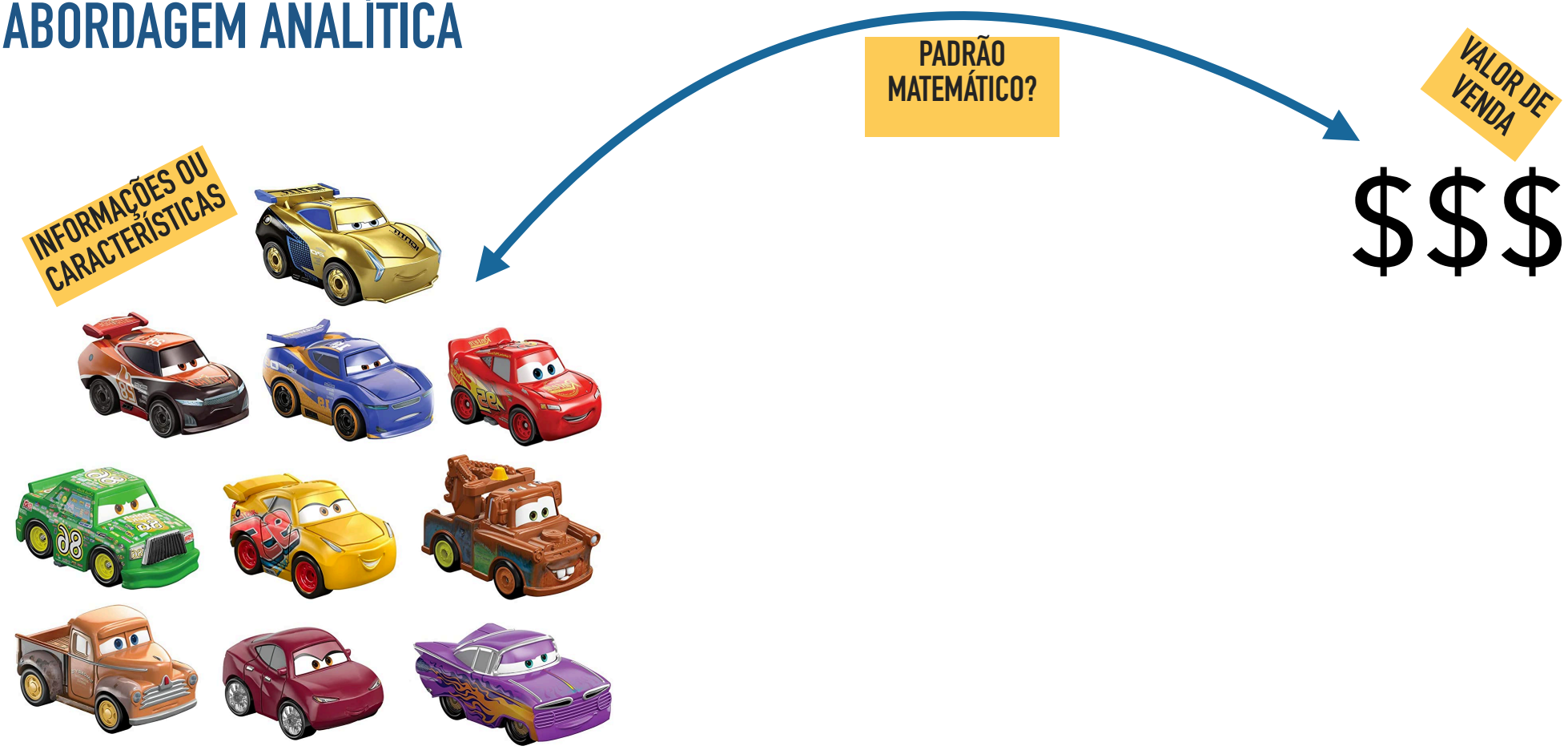


Como eu posso usar dados para resolver esse problema?

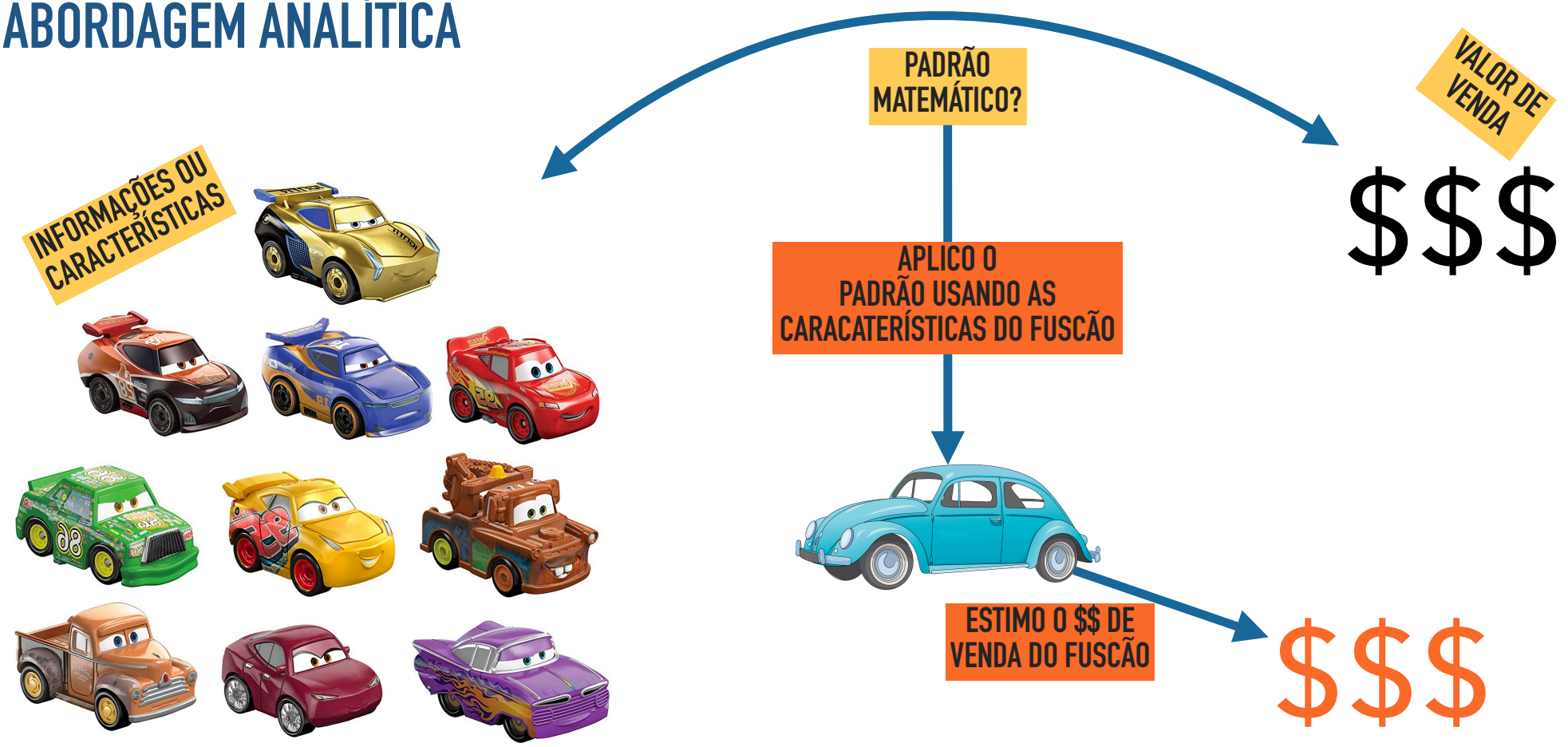
A resposta para esta pergunta depende do entendimento do negócio.

Source: Rollings John B., Foundational methodology for data science. IBM white paper
https://www.ibm.com/downloads/cas/B1WQ0GM2

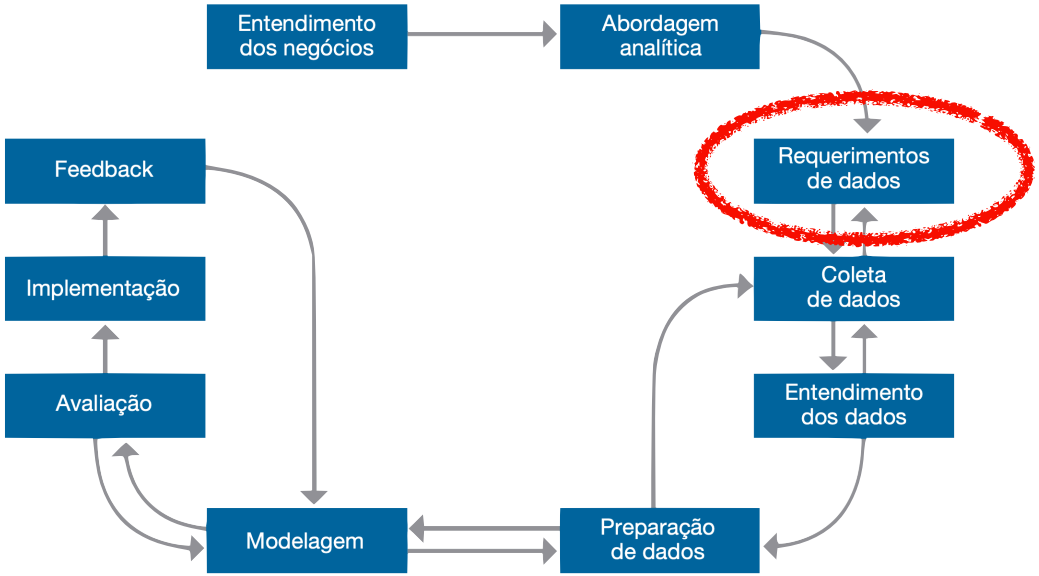
ABORDAGEM ANALÍTICA



ABORDAGEM ANALÍTICA



ABORDAGEM ANALÍTICA



Quais dados serão necessários?

De onde virão e como eu terei acesso?

Os dados que forem coletados serão representativos?

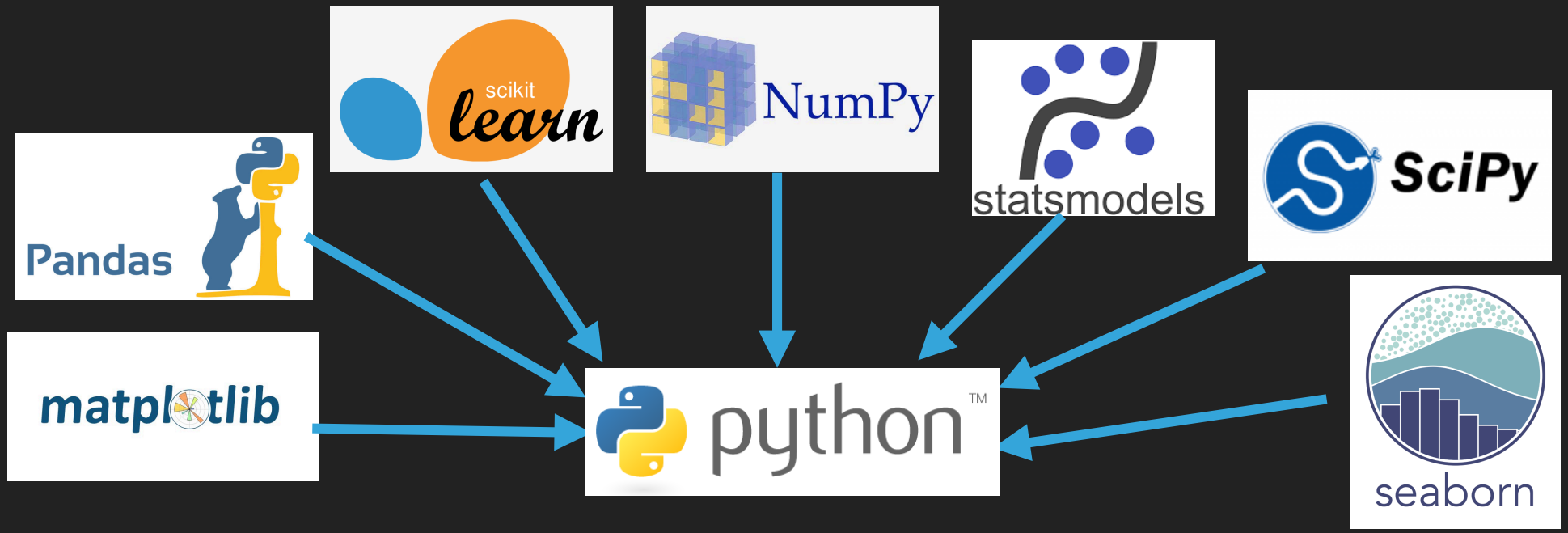
Source: Rollings John B., Foundational methodology for data science. IBM white paper
https://www.ibm.com/downloads/cas/B1WQ0GM2

PREÇOS ESTIMADOS DE CARROS USADOS

- Como o Chico pode determinar o melhor preço para o seu carro?
  - Existem dados de preços de carros usados?
  - Quanto as características afetam os preços?
    - Cor, marca, potência do motor, algo mais?
  - Precisamos fazer as perguntas corretas quando se trata de dados.

BANCO DE DADOS

UCI Donald Bren School of Information & Computer Sciences

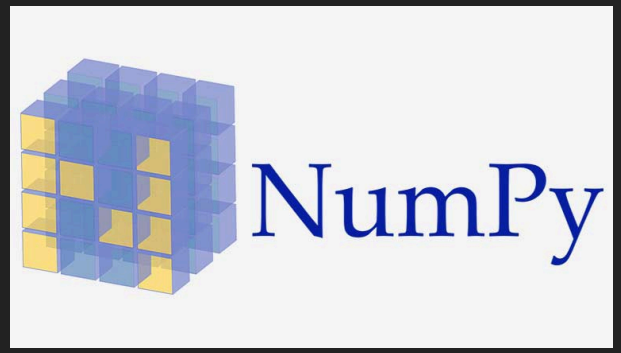


PYTHON E SUA BIBLIOTECAS

PYTHON PACKAGES FOR DATA SCIENCE

SCIENTIFIC COMPUTING LIBRARIES

Data Structure & Tools



Arrays & Matrices

Integrals,
differential equations,
optimization

PYTHON PACKAGES FOR DATA SCIENCE

VISUALIZATION LIBRARIES

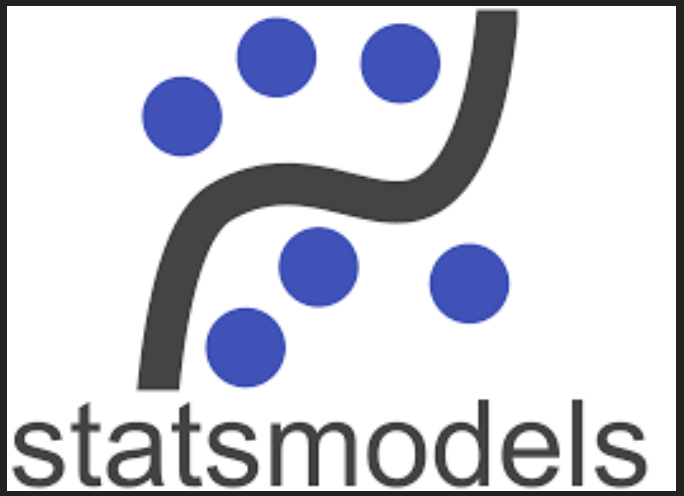
Plots & Graphs

Heatmaps, violin plots, time series

PYTHON PACKAGES FOR DATA SCIENCE

ALGORITHMIC LIBRARIES

Machine Learning, regression, classification



Explore data, estimate statistical models, statistical tests

DS_PPGOLD_AULA02.IPYNB

- Importar as bibliotecas
- Ler os dados a partir de um arquivo CSV
- Gravar em um data Frame

IMPORTAR AS BIBLIOTECAS

```
# import libraries
import pandas as pd
```

IMPORTAR AS BIB AS

SINAL DE COMENTÁRIO NO CÓDIGO, ESTA LINHA O PYTHON DESCONSIDERA

```
# import libraries
import pandas as pd
```

IMPORTAR AS BIBLIOTECAS

COMANDO DE IMPORTAÇÃO

IMPORTAR AS BIBLIOTECAS

```
# import libraries
import pandas as pd
```

NOME DA BIBLIOTECA A SER IMPORTADA

IMPORTAR AS BIBLIOTECAS

```
# import libraries
import pandas as pd
```

"ALIAS" APELIDO DA BIBLIOTECA NO SEU CÓDIGO

IMPORTAR AS BIBLIOTECAS

```
[ ]  # import libraries
     import pandas as pd  # Biblioteca para os DataFrames
     import numpy as np   # Biblioteca Numérica
```

LER O ARQUIVO CSV

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

LER O ARQUIVO CSV

VARIÁVEL ONDE GUARDAREI O NOME DO ARQUIVO

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

LER O ARQUIVO CSV

O NOME DO ARQUIVO

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

LER O ARQUIVO CSV

REPAREM QUE USAREI ESSA VARIÁVEL NA SEGUNDA LINHA DE PROGRAMAÇÃO

```
[3]  # Ler o ar                    o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

LER O ARQUIVO CSV

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

O NOME DO MEU DATAFRAME

LER O ARQUIVO CSV

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

COMANDO DE LEITURA DE UM ARQUIVO CSV

LEMBREM QUE É UM MÉTODO QUE ESTÁ NA BIBLIOTECA PANDAS, POR ISSO COMEÇA COM PD

LER O ARQUIVO CSV

```
[3]  # Ler o arquivo, e atribua-o à variável "df".
     my_file = "auto.csv"

     # Atribuir o arquivo a uma váriável DataFrame
     df = pd.read_csv( my_file , header=None )
```

ARGUMENTOS DO MÉTODO READ_CSV

NOME DO ARQUIVO

ARQUIVO SEM CABEÇALHO

EXISTEM OUTROS (...)

ARQUIVO AUTO.CSV EM UM EDITOR DE TEXTO

SEM CABEÇALHO

CAMPOS SEPARADOS POR VÍRGULAS

- IO tools (text, CSV, HDF5, ...)
  - CSV & text files
  - JSON
  - HTML
  - LaTeX
  - XML
  - Excel files
  - OpenDocument Spreadsheets
  - Binary Excel (.xlsb) files
  - Clipboard
  - Pickling
  - msgpack
  - HDF5 (PyTables)
  - Feather
  - Parquet
  - ORC
  - SQL queries
  - Google BigQuery
  - Stata format
  - SAS formats
  - SPSS formats
  - Other file formats
  - Performance considerations

A PANDAS LÊ DIVERSOS OUTROS FORMATOS TAMBÉM

https://pandas.pydata.org/

```
# import libraries
import pandas as pd
```

"PD" É PANDAS

```
# Atribuir o arquivo a uma váriável DataFrame
df = pd.read_csv( my_file , header=None )
```

MAS "DF" É "PD"

ENTÃO "DF" É PANDAS

DANDO UM "LOOK" NOS DADOS

- Verificar estrutura do dataFrame
- Existem valores faltantes ?
- As variáveis são categóricas ou numéricas?
- O intervalos dos dados estão corretos?

HEAD AND TAIL

```
# mostrar as 10 primeiras linhas usando o método dataframe.head()
print("As primeiras 10 linhas do dataframe são:")
df.head(10)
```

MÉTODO HEAD

ARGUMENTO
DE LINHAS = 10

DADOS SOBRE AS VENDAS DE CARROS USADOS

As primeiras 10 linhas do dataframe são:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four	130	mpfi	3.47	2.68	9.0	111	5000	21	27	13495
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	ohcv	six	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	ohc	four	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	ohc	five	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450
5	2	?	audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	15250
6	1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	17710
7	1	?	audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	18920
8	1	158	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	3086	ohc	five	131	mpfi	3.13	3.40	8.3	140	5500	17	20	23875
9	0	?	audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52.0	3053	ohc	five	131	mpfi	3.13	3.40	7.0	160	5500	16	22	?

FALTAM OS NOMES DAS COLUNAS

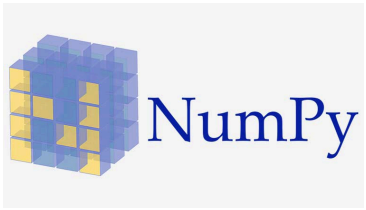
As primeiras 10 linhas do dataframe são:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four	130	mpfi	3.47	2.68	9.0	111	5000	21	27	13495
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	dohc	four	130	mpfi	3.47	2.68	9.0	111	5000	21	27	16500
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	ohcv	six	152	mpfi	2.68	3.47	9.0	154	5000	19	26	16500
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	ohc	four	109	mpfi	3.19	3.40	10.0	102	5500	24	30	13950
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	ohc	five	136	mpfi	3.19	3.40	8.0	115	5500	18	22	17450
5	2	?	audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	15250
6	1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	17710
7	1	?	audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	ohc	five	136	mpfi	3.19	3.40	8.5	110	5500	19	25	18920
8	1	158	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	3086	ohc	five	131	mpfi	3.13	3.40	8.3	140	5500	17	20	23875
9	0	?	audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52.0	3053	ohc	five	131	mpfi	3.13	3.40	7.0	160	5500	16	22	?

EXISTEM TAMBÉM ALGUNS "BURACOS" NESTE DATAFRAME

DS_PPGOLD_AULA02.IPYNB

- Cabeçalho
- Interrogações “?” (dados faltantes)



NOMES DAS COLUNAS (HEADERS)

O cabeçalho está na documentação dos dados

https://archive.ics.uci.edu/ml/datasets/Automobile

Defino uma variável chamada **headers** tipo LISTA com os nomes de cada coluna

```
# Cria a headers list
headers = ["symboling","normalized-losses","make","fuel-type","aspiration", "num-of-doors","body-style",
           "drive-wheels","engine-location","wheel-base", "length","width","height","curb-weight","engine-type",
           "num-of-cylinders", "engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
           "peak-rpm","city-mpg","highway-mpg","price"]
print("headers\n", headers)
```

NOMES DAS COLUNAS (HEADERS)

Atribuo a variável **headers** aos nomes das colunas do DATAFRAME

```
# Print DataFrame com os Headers
df.columns = headers
df.head(10)
```

O QUE FAZER COM AS INTERROGAÇÕES "?"

- Neste dataFrame o simbolo de "?" significa valores faltantes.
- Nós cuidaremos deles mais tarde, mas por enquanto só vamos substituí-los por NaN (Not a Number)

```
df=df.replace('?',np.NaN)
df.head(10)
```

O QUE FAZER COM AS INTERROGAÇÕES "?"

MÉTODO REPLACE DA BIBLIOTECA PANDAS

CONSTANTE NAN DA BIBLIOTECA NUMPY

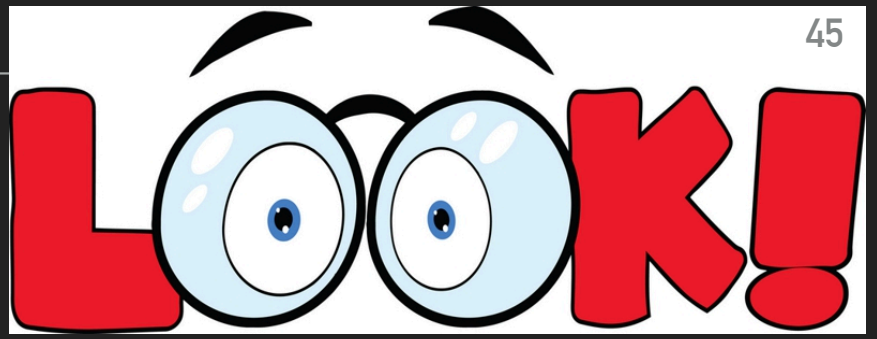
```
df=df.replace('?',np.NaN)
df.head(10)
```

https://numpy.org/

EDUARDO PÉCORA

ANÁLISES INICIAIS DE DADOS EM PYTHON

UMA OLHADA DE RELANCE (BASIC INSIGHTS ABOUT THE DATA)



- Entender os dados antes de começar as análises
- Devemos checar
  - Tipos dos dados
  - Distribuição dos dados
- Identificar potenciais problemas com os dados

RESUMÃO DOS NOSSOS DADOS

No.	Attribute name	attribute range	No.	Attribute name	attribute range
1	symboling	-3, -2, -1, 0, 1, 2, 3.	14	curb-weight	continuous from 1488 to 4066.
2	normalized-losses	continuous from 65 to 256.	15	engine-type	dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
3	make	audi, bmw, etc.	16	num-of-cylinders	eight, five, four, six, three, twelve, two.
4	fuel-type	diesel, gas.	17	engine-size	continuous from 61 to 326.
5	aspiration	std, turbo.	18	fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
6	num-of-doors	four, two.	19	bore	continuous from 2.54 to 3.94.
7	body-style	hardtop, wagon, etc.	20	stroke	continuous from 2.07 to 4.17.
8	drive-wheels	4wd, fwd, rwd.	21	compression-ratio	continuous from 7 to 23.
9	engine-location	front, rear.	22	horsepower	continuous from 48 to 288.
10	wheel-base	continuous from 86.6 120.9.	23	peak-rpm	continuous from 4150 to 6600.
11	length	continuous from 141.1 to 208.1.	24	city-mpg	continuous from 13 to 49.
12	width	continuous from 60.3 to 72.3.	25	highway-mpg	continuous from 16 to 54.
13	height	continuous from 47.8 to 59.8.	26	price	continuous from 5118 to 45400.

https://archive.ics.uci.edu/ml/datasets/Automobile

DATASET

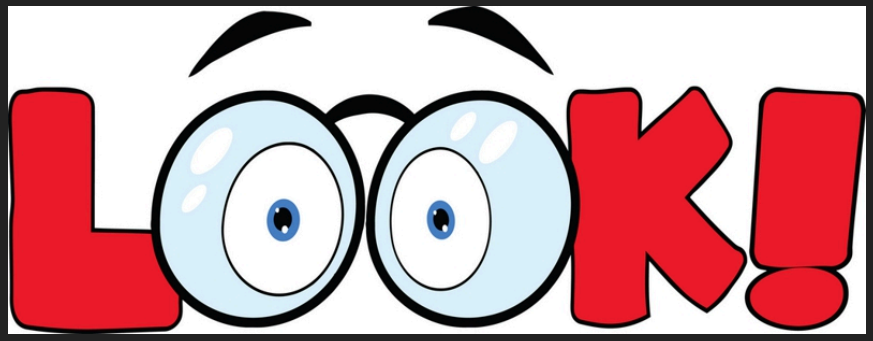
No.	Attribute name	attribute range	No.	Attribute name	attribute range
1	symboling	-3, -2, -1, 0, 1, 2, 3.	14	curb-weight	continuous from 1488 to 4066.
2	normalized-losses	continuous from 65 to 256.	15	engine-type	dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
3	make	audi, bmw, etc.	16	num-of-cylinders	eight, five, four, six, three, twelve, two.
4	fuel-type	diesel, gas.	17	engine-size	continuous from 61 to 326.
5	aspiration	std, turbo.	18	fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
6	num-of-doors	four, two.	19	bore	continuous from 2.54 to 3.94.
7	body-style	hardtop, wagon, etc.	20	stroke	continuous from 2.07 to 4.17.
8	drive-wheels	4wd, fwd, rwd.	21	compression-ratio	continuous from 7 to 23.
9	engine-location	front, rear.	22	horsepower	continuous from 48 to 288.
10	wheel-base	continuous from 86.6 120.9.	23	peak-rpm	continuous from 4150 to 6600.
11	length	continuous from 141.1 to 208.1.	24	city-mpg	continuous from 13 to 49.
12	width	continuous from 60.3 to 72.3.	25	highway-mpg	continuous from 16 to 54.
13	height	continuous from 47.8 to 59.8.	26	price	continuous from 5118 to 45400.

Target (Label)

DATA TYPES

Pandas Type	Native Python Type	Description
object	string	numbers and strings
int64	int	Numeric characters
float64	float	Numeric characters with decimals
datetime64, timedelta[ns]	N/A (but see the datetime module in Python's standard library)	time data.

PORQUE CHECAR OS DADOS ?



- Potenciais erros de tipos
- Compatibilidade com os métodos em Python

DS_PPGOLD_AULA02.IPYNB

- Verificar os tipos de dados
- Corrigir os tipos de dados

TIPOS DOS DADOS

```
# Imprimir os tipos de dados de cada coluna do DataFrame
df.dtypes
```

propriedade **dtypes** da PANDAS

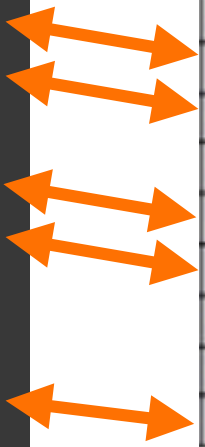
COMPARAR O DATAFRAME COM AS ESPECIFICAÇÕES

```
symboling           int64
normalized-losses   object
make                object
fuel-type           object
aspiration          object
num-of-doors        object
body-style          object
drive-wheels        object
engine-location     object
wheel-base         float64
length             float64
width              float64
height             float64
```

No.	Attribute name	attribute range
1	symboling	-3, -2, -1, 0, 1, 2, 3.
2	normalized-losses	continuous from 65 to 256.
3	make	audi, bmw, etc.
4	fuel-type	diesel, gas.
5	aspiration	std, turbo.
6	num-of-doors	four, two.
7	body-style	hardtop, wagon, etc.
8	drive-wheels	4wd, fwd, rwd.
9	engine-location	front, rear.
10	wheel-base	continuous from 86.6 120.9.
11	length	continuous from 141.1 to 208.1.
12	width	continuous from 60.3 to 72.3.
13	height	continuous from 47.8 to 59.8.

COMPARAR O DATAFRAME COM AS ESPECIFICAÇÕES

```
curb-weight           int64
engine-type          object
num-of-cylinders     object
engine-size           int64
fuel-system          object
bore                 object
stroke               object
compression-ratio   float64
horsepower           object
peak-rpm             object
city-mpg              int64
highway-mpg           int64
price                object
dtype: object
```



No.	Attribute name	attribute range
14	curb-weight	continuous from 1488 to 4066.
15	engine-type	dohc, dohcv, l, ohc, ohcf, ohcv, rot
16	num-of-cylinders	eight, five, four, six, three, twelve,
17	engine-size	continuous from 61 to 326.
18	fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spd
19	bore	continuous from 2.54 to 3.94.
20	stroke	continuous from 2.07 to 4.17.
21	compression-ratio	continuous from 7 to 23.
22	horsepower	continuous from 48 to 288.
23	peak-rpm	continuous from 4150 to 6600.
24	city-mpg	continuous from 13 to 49.
25	highway-mpg	continuous from 16 to 54.
26	price	continuous from 5118 to 45400.

EXISTEM ERROS, TEREMOS QUE CORRIGIR. MAS ANTES VAMOS EXPLORAR MAIS UM POUCO

Eduardo Pécora

DESCRIBE

```
# Este método fornecerá várias estatísticas resumidas, excluindo valores NaN

df.describe()
```

método **describe** da PANDAS

	symboling	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	city-mpg	highway-mpg
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	10.142537	25.219512	30.751220
std	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	3.972040	6.542142	6.886443
min	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	7.000000	13.000000	16.000000
25%	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	8.600000	19.000000	25.000000
50%	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	9.000000	24.000000	30.000000
75%	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	9.400000	30.000000	34.000000
max	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	23.000000	49.000000	54.000000

DESCRIBE

```
# describe all the columns in "df"
df.describe(include = "all")
```

método **describe** da PANDAS com o argumento **“all”**

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	eng loca
count	205.000000	164	205	205	205	203	205	205	
unique	NaN	51	22	2	2	2	5	3	
top	NaN	161	toyota	gas	std	four	sedan	fwd	
freq	NaN	11	32	185	168	114	96	120	
mean	0.834146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
std	1.245307	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
min	-2.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
25%	0.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
50%	1.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
75%	2.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
max	3.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

INFO

```
# look at the info of "df"
df.info()
```

método **info** da PANDAS

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  164 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   aspiration         205 non-null    object
 5   num-of-doors       203 non-null    object
 6   body-style         205 non-null    object
 7   drive-wheels       205 non-null    object
 8   engine-location    205 non-null    object
 9   wheel-base         205 non-null    float64
 10  length             205 non-null    float64
 11  width              205 non-null    float64
 12  height             205 non-null    float64
 13  curb-weight        205 non-null    int64
 14  engine-type        205 non-null    object
 15  num-of-cylinders   205 non-null    object
 16  engine-size        205 non-null    int64
 17  fuel-system        205 non-null    object
 18  bore               201 non-null    object
 19  stroke             201 non-null    object
 20  compression-ratio  205 non-null    float64
 21  horsepower         203 non-null    object
 22  peak-rpm           203 non-null    object
 23  city-mpg           205 non-null    int64
 24  highway-mpg        205 non-null    int64
 25  price              201 non-null    object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB
```

COMO NÃO HÁ STRINGS NAS COLUNAS QUE DEVERIAM SER NUMÉRICAS, PODEMOS SIMPLESMENTE TROCAR O TIPO, SENÃO TERÍAMOS QUE ARRUMAR ANTES

Johnny Appleseed

ASTYPE

método **astype()** da PANDAS

```
#Trocando o tipo da coluna 'normalized-losses'
df['normalized-losses'] = df['normalized-losses'].astype(float)
```

ESTOU SUBSTITUINDO A COLUNA "NORMALIZED-LOSSES" DO DATAFRAME

PELA MESMA COLUNA COM O TIPO MODIFICADO

REPLICAR PARA AS OUTRAS COLUNAS

```
#Trocando o tipo das colunas com erro

df['normalized-losses'] = df['normalized-losses'].astype(float)
df['price'] = df['price'].astype(float)
df['horsepower'] = df['horsepower'].astype(float)
df['peak-rpm'] = df['peak-rpm'].astype(float)
df['bore'] = df['bore'].astype(float)
df['stroke'] = df['stroke'].astype(float)
df['normalized-losses'] = df['normalized-losses'].astype(float)

df.info()

# Dá pra fazer tudo de uma única vez também !

#df[["bore", "peak-rpm","stroke", "price"]] = df[["bore", "peak-rpm", "stroke","price"]].astype("float")
#df[["normalized-losses"]] = df[["normalized-losses"]].astype("int")
```

EXPORTE O SEU ARQUIVO

```
[ ]  path = "Auto_limpo.csv"

     #path = "C:/windows/.../Auto.csv"
     #path = "/Users/pecorajr/Desktop/DELETE/Auto.csv"

     df.to_csv(path, index = False)
```

PROF. EDUARDO PÉCORA

LAB02 – REPLIQUEM TUDO ISSO PARA OS VINHOS