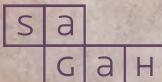


# PREPARAÇÃO E ANÁLISE EXPLORATÓRIA DE DADOS

Leandro Botelho Alves de Miranda



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS

# Correlações usando Pandas, Numpy e Seaborn

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Calcular correlações lineares baseadas no coeficiente de Pearson e usando as bibliotecas Pandas e Numpy.
- Identificar as principais correlações presentes em bases de dados.
- Reconhecer a multicolinearidade em conjuntos de dados.

## Introdução

No cenário atual, no qual a tecnologia está cada vez mais robusta e prática, ferramentas estatísticas são importantes na observação de dados, permitindo a criação de evidências de forma mais rápida. A correlação é um dos conceitos estatísticos bastante aplicado em análises de dados. Em geral, ela mede a relação linear entre duas variáveis de interesse, de acordo com diferentes tipos de dados (por exemplo, compras de um usuário e seu grau de satisfação).

Algumas das métricas mais comumente usadas nesse domínio são as correlações lineares baseadas no coeficiente de Pearson, que são implementadas em bibliotecas como Pandas e Numpy. Além disso, ferramentas de visualização de correlação também são relevantes, principalmente para facilitar uma análise analítica dos dados; para isso, Seaborn aparece como uma grande ferramenta gráfica. Também é importante lidar com cenários com variáveis categóricas e contínuas, que estão presentes em base de dados; as análises de correlações em mais de duas variáveis também são bastante presentes nesses estudos.

Neste capítulo você vai estudar as técnicas para o cálculo de correlações lineares baseadas no coeficiente de Pearson a partir das bibliotecas Pandas e Numpy. Também vai ver como identificar as principais correlações presentes em bases de dados utilizando tabelas cruzadas (*cross-tabs*).

Por fim, vai ver como reconhecer a forte correlação entre duas ou mais variáveis, conhecida como multicolinearidade.

## 1 Correlações lineares

Em ciência de dados, ao estudarmos uma variável de interesse, em geral usamos medidas de tendência central, dispersão, variância, etc. Com duas ou mais variáveis além dessas medidas individuais, é interessante analisarmos se há algum relacionamento entre elas, isto é, se determinado valor de uma variável implica no(s) valor(es) de outra(s) variável(eis). Por exemplo, você pode verificar se existe associação entre o índice de desenvolvimento humano (IDH) e a renda *per capita* de uma cidade ou distrito; entre o consumo de produtos *on-line* entre distintos usuários, etc.

A **correlação** é uma técnica para investigar a relação entre duas variáveis quantitativas e contínuas, sendo uma variável independente e uma variável dependente. Uma **variável independente** é uma medida que não é dependente de nenhum valor de outra variável. Uma **variável dependente** é uma medida que dependerá do valor de outra variável.

O coeficiente de correlação de Pearson ( $r$ ) e o coeficiente de determinação ( $R^2$ ) são medidas da força da associação entre as duas variáveis. A seguir, veremos em mais detalhes esses coeficientes e as ferramentas de manipulação e visualização de dados correlacionados.

### Correlação de Pearson e $R^2$

Como vimos, é interessante realizarmos um estudo sobre o par de valores de uma variável. Assim, dadas duas amostras, uma da variável  $X$ , composta por um conjunto de amostras ( $X_1, \dots, X_n$ ), e outra da variável  $Y$ , composta por um conjunto de amostras ( $Y_1, \dots, Y_n$ ), o coeficiente de correlação de Pearson é estimado com a seguinte expressão (MORETTIN; BUSSAB, 2017):

$$r = \frac{1}{n} \sum_{i=1}^n \left( \frac{X_i - \bar{X}}{STD(X)} \right) \left( \frac{Y_i - \bar{Y}}{STD(Y)} \right) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

$$r = \frac{n \cdot (\sum_{i=1}^n XY) - (\sum_{i=1}^n X)(\sum_{i=1}^n Y)}{\sqrt{n(\sum_{i=1}^n X^2) - (\sum_{i=1}^n X)^2} \sqrt{n(\sum_{i=1}^n Y^2) - (\sum_{i=1}^n Y)^2}} = \frac{n((\sum XY) - (\sum X)(\sum Y))}{\sqrt{n(\sum X^2) - (\sum X)^2} \sqrt{n(\sum Y^2) - (\sum Y)^2}}$$

Sendo a função STD a função desvio-padrão. Essa correlação é representada pela média dos produtos dos valores padronizados das variáveis.

Existe outra forma de representar a correlação de Pearson. Seja  $COV(X, Y)$  a covariância entre as duas variáveis  $X$  e  $Y$ , a fórmula da covariância é dada por:

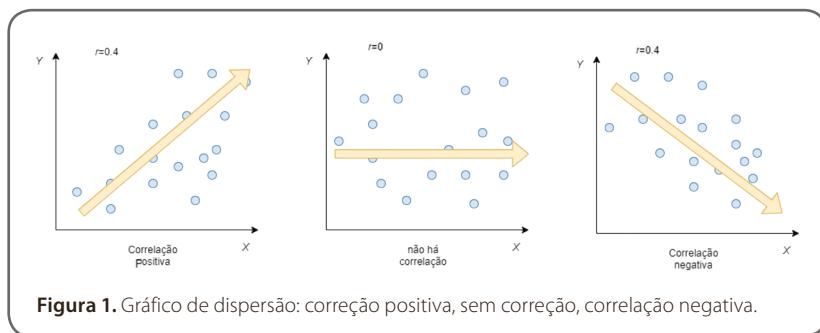
$$COV(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Tal equação representa a média dos produtos dos valores centrados das variáveis. Simplificando essa equação, você pode ter o seguinte cálculo:

$$r = \frac{1}{n} \sum_{i=1}^n \left( \frac{COV(X, Y)}{STD(X) STD(Y)} \right)$$

O coeficiente de correlação de Pearson é representado por  $\rho$  e, quando se está trabalhando com amostras, o coeficiente de correlação é indicado pela letra  $r$ .

O coeficiente de correlação ( $r$ ) pode variar de -1,00 a +1,00. Assim,  $r = +1$  indica a presença de uma correlação linear positiva perfeita de forma ascendente sobre as amostras presentes em  $X$  e  $Y$ , enquanto  $r = -1$  indica correlação linear negativa perfeita de forma descendente sobre as amostras presentes em  $X$  e  $Y$ . Um coeficiente de correlação 0 significa que não existe um relacionamento linear entre  $X$  e  $Y$ . Na maior parte dos estudos, o valor de  $r$  (-1,1). Na Figura 1, você pode ver como esses dados estão dispersos.



Em problemas de regressão, alguns analistas de dados precisam lançar mão de estratégias que forneçam a proporção da variação de uma variável que é previsível em relação a outra variável. Para isso, uma métrica bastante utilizada em análise de dados é o coeficiente de determinação  $R^2$ , ou CF. O coeficiente de determinação é a proporção da variabilidade na variável dependente que é previsível a partir da(s) variável(eis) independente(s). Sua fórmula é a seguinte:

$$CF = r^2$$

O coeficiente de determinação é uma saída essencial da análise de regressão. Trata-se de uma medida que permite determinar como alguém pode estar certo ao fazer previsões a partir de determinado modelo/gráfico. O coeficiente de determinação é o quadrado da correlação ( $r$ ), portanto, varia de 0 a 1;  $R^2 = 0$  significa que a variável dependente não pode ser prevista a partir da variável independente, enquanto  $R^2 = 1$  significa que a variável dependente pode ser prevista sem erro da variável independente.  $R^2$  entre 0 e 1 indica até que ponto a variável dependente é previsível.

Por exemplo, seja  $r = 0,83$ , temos  $r^2 = (0,83)^2 = 0,6889$ . Isso quer dizer que 68,89% da variação total da variável dependente são explicados, isto é, são previsíveis. Decorre que 31,11% da variação total da variável dependente permanecem não explicados, isto é, não previsíveis.

## Correlação de dados com Pandas e Numpy

Agora, vamos ver como implementar a correlação linear de Pearson a partir das bibliotecas Pandas e Numpy. Pandas oferece métodos estatísticos para instâncias do tipo Series e DataFrame. Por exemplo, dados dois objetos Series com o mesmo número de itens, você pode chamar `.corr()` em um deles com o outro como o primeiro argumento.

Acompanhe um exemplo inicial utilizando Pandas. Na Figura 2, duas séries foram criadas, e foi feita uma correlação entre as duas séries. Você pode ver que a correlação  $x$  em relação a  $y$  é igual a correlação de  $y$  em relação a  $x$ . Outra opção é especificar a correlação de Pearson no parâmetro da função `corr()`; por exemplo, `x.corr(y, method='pearson')`.

```
import pandas as pd  
x = pd.Series(range(20,30))  
x
```

```
0    20  
1    21  
2    22  
3    23  
4    24  
5    25  
6    26  
7    27  
8    28  
9    29  
dtype: int64
```

```
y = pd.Series([3, 1, 4, 5, 14, 12, 22, 45, 89, 62])  
y
```

```
0    3  
1    1  
2    4  
3    5  
4    14  
5    12  
6    22  
7    45  
8    89  
9    62  
dtype: int64
```

**Figura 2.** Correlação entre duas séries em Pandas.

Numpy tem muitas rotinas estatísticas, incluindo `np.corrcoef()`, que retorna uma matriz de coeficientes de correlação de Pearson. No próximo exemplo, você pode começar importando Numpy e definindo duas matrizes Numpy. Essas são instâncias da classe `ndarray`; chame-as de `x` e `y`. Você usa `np.arange()` para criar uma matriz `X` de números inteiros entre 20 (inclusive) e 30 (exclusivo). Então você usa `np.array()` para criar uma segunda matriz `Y` contendo números inteiros arbitrários.

Depois de ter duas matrizes do mesmo comprimento, você pode chamar `np.corrcoef()` com ambas as matrizes como argumentos, como mostra a Figura 3.

```
import numpy as np
x = np.arange(20, 30)
y = np.array([3, 1, 4, 5, 14, 12, 22, 45, 89, 62])
r = np.corrcoef(x, y)
r
```

```
array([[1.          , 0.85903518],
       [0.85903518, 1.        ]])
```

```
r[0,1]
```

```
0.859035180995684
```

```
r[1,0]
```

```
0.8590351809956841
```

**Figura 3.** Correlação entre duas matrizes em Numpy.

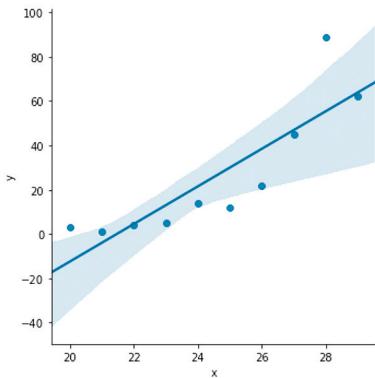
## Visualização de correlações com Seaborn

O primeiro passo no estudo da relação entre duas variáveis contínuas é desenhar um gráfico de dispersão das variáveis para verificar a linearidade. O coeficiente de correlação não deve ser calculado se o relacionamento não for linear. Apenas para fins de correlação, não importa realmente em qual eixo as variáveis são plotadas. No entanto, convencionalmente, a variável independente é plotada no eixo x (horizontalmente) e a variável dependente é plotada no eixo y (verticalmente).

A partir de agora, os códigos escritos em Numpy e Pandas serão combinados para facilitar a visualização de diferentes gráficos. Como já mencionado, primeiramente você vai ver, na Figura 4, um gráfico de dispersão para verificar como os dados de  $X$  e  $Y$  do exemplo anterior estão dispersos.

```
z = np.column_stack((x,y)) # uma matriz de dados composta pelos valores das variáveis x  
e y  
dataset = pd.DataFrame(z,columns=["x", "y"]) # um dataframe pandas é criado com os valores da matriz z  
  
sns.lmplot(x="x",y="y",data=dataset)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f5bc42d7190>
```



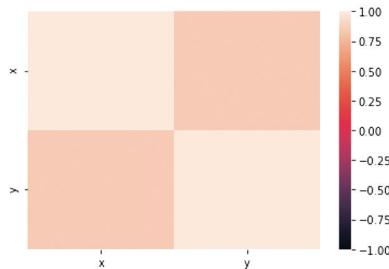
**Figura 4.** Gráfico de dispersão dos dados de  $X$  e  $Y$  do exemplo.

Perceba que alguns dados estão próximos, ou seja, observando de perto, há uma sombra convergindo no centro, onde há um pedaço dos nossos dados. Esse ponto convergente é, na verdade, a média estatística que indica o nível de variabilidade em um respectivo nível de pontos.

Outra forma de visualizar o nível de correlação é por meio de mapas de calor (*heatmaps*). A partir dessas amostras iniciais neste estudo (em  $X$  e  $Y$ ), você pode ver um exemplo desse gráfico na Figura 5. Nas próximas seções do capítulo, esse último gráfico de visualização será bastante usado para outros cenários de correlação.

```
import seaborn as sns
dataset = pd.DataFrame(z,columns=["x", "y"]) # um datafram pandas é criado com os valores da matriz z
sns.heatmap(dataset.corr(),vmin=-1, vmax=1) # vmin e vmax definem o intervalo da escala de valores de correlação
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f390174ccf8>



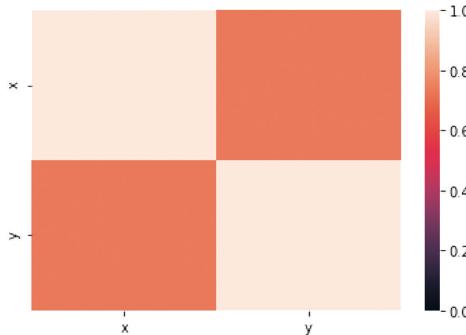
**Figura 5.** Visualização do nível de correlação com mapa de calor (*heatmap*).

Na Figura 6, você pode ver um exemplo de uso da correlação  $R^2$  com o gráfico *heatmap*. Esse resultado mostra que a variável dependente  $y$  está mais correlacionada com a variável independente  $x$ .

```
r_squared = dataset.corr()**2
sns.heatmap(r_squared,vmin=0, vmax=1)
```

Out[14]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fb0af1354a8>



**Figura 6.** Visualização do uso da correlação  $R^2$  com mapa de calor (*heatmap*).



### Fique atento

Dependendo do horário do dia ou da dificuldade visual do respectivo analista dessas imagens, é importante utilizar paletas com estilo de cores diferentes com gráfico *heatmap*. O parâmetro “cmap” contém alguns valores de cores para paletas diferentes, como “YIGnBu”, “Blues”, “BuPu” e “Greens”.

## 2 Principais correlações de bases de dados

Em geral, as bases de dados são compostas por diferentes tipos de variáveis (como variáveis categóricas, discretas ou contínuas). Em situações nas quais é necessário determinar como duas ou mais variáveis categóricas estão associadas em um estudo, a **tabulação cruzada** é uma das abordagens iniciais a serem consideradas (DASS, 2010).

A tabulação cruzada é um método para analisar quantitativamente o relacionamento entre muitas variáveis. A tabulação cruzada agrupa variáveis para entender a correlação entre diferentes variáveis. Também mostra como as correlações mudam de um agrupamento de variáveis para outro. Essa técnica é aplicada em análise estatística para encontrar padrões, tendências e probabilidades nos conjuntos de dados.

Em uma tabulação cruzada, criamos uma tabela semelhante à de uma distribuição de frequência, mas mesclamos as contagens de diferentes valores de duas ou mais variáveis. A tabela de tabulação cruzada pode estar entre duas variáveis (tabulação cruzada bivariada) e três variáveis (tabulação cruzada de três variáveis). De modo geral, a tabulação cruzada é útil para obtermos informações sobre como os valores de duas variáveis estão relacionadas, qual classificação cruzada é mais selecionada pelos entrevistados e como essas classificações cruzadas são diferentes umas das outras.

## Exemplo prático de tabulação cruzada com Pandas e Seaborn

A partir de agora, você vai ver os conceitos de tabulação cruzada a partir da coleta de bases de dados. No Pandas, a função `crosstab` cruzada cria uma tabela cruzada. Aqui, você vai analisar correlações de diferentes marcas de carro. Veja a seguir um código para o carregamento de um conjunto de dados de automóveis do UCI Machine Learning Repository. Neste exemplo, a base de dados foi resumida para um número limitado de seis marcas de carro, para facilitar sua compreensão; entretanto, a base de dados original é mais extensa que isso.

```
import pandas as pd
import seaborn as sns
# Define o nome das colunas, pois os dados não contém (baseado nos atributos do site da UCI)
cabecalho = ["symboling", "normalized_losses", "make", "fuel_type", "aspiration",
             "num_doors", "body_style", "drive_wheels", "engine_location",
             "wheel_base", "length", "width", "height", "curb_weight",
             "engine_type", "num_cylinders", "engine_size", "fuel_system",
             "bore", "stroke", "compression_ratio", "horsepower", "peak_rpm",
             "city_mpg", "highway_mpg", "price"]

# Ler o arquivo CSV e converte os valores faltantes (?) para NaN
df_base_de_dados = pd.read_csv("http://mlr.cs.umass.edu/ml/machine-learning-databases/autos/imports-85.data",
                                header=None, names=cabecalho, na_values="?")

# Define a lista da marcas de carro
marcas_de_carro = ["toyota", "nissan", "honda", "mitsubishi", "subaru",
                    "volkswagen"]

# Cria uma cópia de dados só com 8 marcas de carros famosos.
df = df_base_de_dados[df_base_de_dados.make.isin(marcas_de_carro)].copy()
```

Na Figura 7, você pode ver a função `crosstab` em ação e visualizar quantos estilos de carro cada marca de carro fabricou. A função `crosstab` pode operar em matrizes Numpy, séries ou colunas em um DataFrame Pandas. Neste exemplo, `df.make` representa as linhas da função `crosstab` e `df.body_style` as colunas da função `crosstab`. Note, por exemplo, que a Volkswagen fabrica 9 sedãs (*sedan*) e 1 conversível (*convertible*).

```
pd.crosstab(df.make, df.body_style)
```

	body_style	convertible	hardtop	hatchback	sedan	wagon
make						
<b>honda</b>	0	0		7	5	1
<b>mitsubishi</b>	0	0		9	4	0
<b>nissan</b>	0	1		5	9	3
<b>subaru</b>	0	0		3	5	4
<b>toyota</b>	1	3		14	10	4
<b>volkswagen</b>	1	0		1	9	1

**Figura 7.** Utilização da função crosstab.

A análise com a biblioteca Pandas fica bem mais fácil, principalmente para criar soluções alternativas de tabelas. No entanto, há outro caso comum de sumarização de dados, utilizado quando você pretende normalizar os dados em cada combinação. Isso pode ser feito usando o parâmetro `normalize`, como você pode ver na Figura 8.

```
pd.crosstab(df.make, df.body_style, normalize=True)
```

	body_style	convertible	hardtop	hatchback	sedan	wagon
make						
<b>honda</b>	0.00	0.00		0.07	0.05	0.01
<b>mitsubishi</b>	0.00	0.00		0.09	0.04	0.00
<b>nissan</b>	0.00	0.01		0.05	0.09	0.03
<b>subaru</b>	0.00	0.00		0.03	0.05	0.04
<b>toyota</b>	0.01	0.03		0.14	0.10	0.04
<b>volkswagen</b>	0.01	0.00		0.01	0.09	0.01

**Figura 8.** Utilização do parâmetro `normalize`.

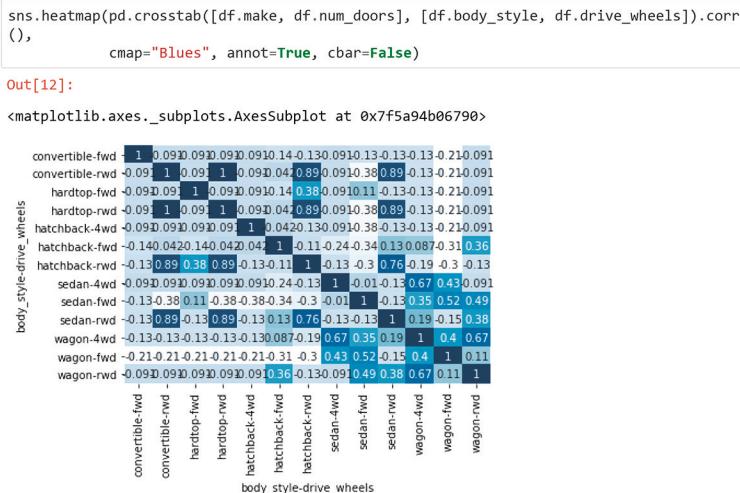
Um dos recursos mais úteis da tabela de referência cruzada é que você pode aplicar o agrupamento (*group by*) em várias colunas do DataFrame Pandas. Por exemplo, se você quer ver como os dados são distribuídos por tração dianteira (*fwd*) e tração traseira (*rwd*), pode incluir a coluna *drive\_wheels* na lista de colunas válidas no segundo argumento para *crosstab*, como mostra a Figura 9.

```
pd.crosstab([df.make, df.num_doors], [df.body_style, df.drive_wheels],
            rownames=['Auto Manufacturer', 'Doors'],
            colnames=['Body Style', 'Drive Type'],
            dropna=False)
```

		Body Style	convertible	hardtop	hatchback	sedan	wagon				
	Drive Type	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd
Auto Manufacturer	Doors										
honda	four	0	0	0	0	0	0	0	0	4	0
	two	0	0	0	0	0	0	7	0	0	0
mitsubishi	four	0	0	0	0	0	0	0	0	4	0
	two	0	0	0	0	0	0	9	0	0	0
nissan	four	0	0	0	0	0	0	1	0	0	0
	two	0	0	0	0	1	0	0	1	4	0
subaru	four	0	0	0	0	0	0	0	0	2	3
	two	0	0	0	0	0	0	1	2	0	0
toyota	four	0	0	0	0	0	0	6	0	0	7
	two	0	0	1	0	0	3	0	2	6	0
volkswagen	four	0	0	0	0	0	0	0	0	7	0
	two	0	1	0	0	0	0	1	0	0	2

**Figura 9.** Inclusão da coluna *drive\_wheels* na lista de colunas válidas no segundo argumento para *crosstab*.

Para o exemplo final, você vai reunir todos os dados usando uma tabela de referência cruzada. Depois vai aplicar uma técnica de correlação entre os dados usando um gráfico *heatmap*. O resultado da Figura 10 mostra uma correlação envolvendo o número de portas (*num\_doors*) e o tipo de carro (*body\_style*) diante o tipo de tração (*Drive Type*).



**Figura 10.** Utilização de tabela de referência cruzada e correlação de dados com *heatmap*.

Outra forma de lidar com dados categóricos é usando a medida Cramér's V. Essa medida está entre 0 e 1 e indica quão fortemente duas variáveis categóricas estão associadas (onde 0 significa que não há associação e 1 é associação completa). Diferentemente da correlação de Pearson, não há valores negativos, pois não existe uma associação negativa. Sua fórmula é descrita a seguir:

$$\chi^2 = \sum_{i,j} \frac{\left( n_{i,j} - \frac{n_i \cdot n_j}{n} \right)^2}{\frac{n_i \cdot n_j}{n}}$$

$$\Phi_c = \sqrt{\frac{\chi^2}{N(k-1)}}$$

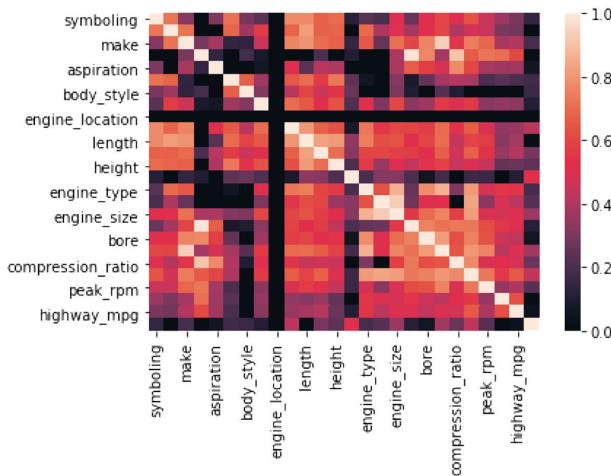
onde:

- $\Phi_c$ : denota Cramér's V.
- $\chi^2$ : estatística qui-quadrada.
- $n_{i,j}$ : número de vezes de duas variáveis ( $X_i, Y_j$ ).
- $N$ : tamanho da amostra envolvida no teste  $\chi^2$ .
- $k$ : menor número de categorias de cada variável.

Em nível de programação em Python, a métrica de Cramér's V é descrita a seguir com a função `cramers_v`, sendo  $x$  e  $y$  duas variáveis de entrada. Você pode observar que a função `crosstab` está presente. Essa função vai permitir a formação da matriz de correlação entre as duas variáveis.

```
import scipy.stats as ss
def cramers_v(x, y):
    confusion_matrix = pd.crosstab(x,y)
    chi2 = ss.chi2_contingency(confusion_matrix)[0]
    n = confusion_matrix.sum().sum()
    phi2 = chi2/n
    r,k = confusion_matrix.shape
    phi2corr = max(0, phi2-((k-1)*(r-1))/(n-1))
    rcorr = r-((r-1)**2)/(n-1)
    kcorr = k-((k-1)**2)/(n-1)
    return np.sqrt(phi2corr/min((kcorr-1),(rcorr-1)))
```

Aplicando a base de dados de exemplo, você pode ver o resultado da matriz de correlação com a métrica de Cramér's V nesse código (considerando as adaptações na geração da matriz de todos os valores). Conservando os nomes das colunas descritos na variável de programação `cabeçalho` (que você viu no primeiro código desta seção), a Figura 11 mostra a visualização da correlação. Neste cenário, os valores presentes na base de dados são todos considerados como categóricos.



**Figura 11.** Visualização da correlação com a métrica de Cramér's V.

### 3 Multicolinearidade em conjuntos de dados

Em análise de dados, há cenários nos quais duas ou mais variáveis independentes são altamente correlacionadas entre si, especialmente em modelos de aprendizado de máquina voltados para problemas de regressão (LAROSE, C.; LAROSE, D., 2019; MONTGOMERY; PECK; VINING, 2012). A **multicolinearidade** é definida como a existência de um alto grau de correlação entre as variáveis independentes (GRAPENTINE, 1997). Isso significa que uma variável independente pode ser prevista a partir de outra variável independente em um modelo de regressão.

A multicolinearidade se encaixa como um problema na construção de modelos de regressão, devido à dificuldade em distinguirmos as relações individuais das variáveis independentes na variável dependente. Por exemplo, considere a seguinte equação linear:

$$Y = a + b \cdot x_1 + c \cdot x_2$$

O coeficiente  $a$  é o aumento em  $Y$  para um aumento de unidade em  $x_1$ , mantendo  $x_2$  constante. Entretanto, como  $x_1$  e  $x_2$  são altamente correlacionados, as alterações em  $x_1$  também causariam alterações em  $x_2$ , e não poderíamos ver seu efeito individual em  $Y$ .

Alguns fatores implicam na multicolinearidade, acompanhe:

- quando a coleta de dados foi mal projetada ou gerou dados insuficientes;
- quando são criadas novas variáveis que têm um alto grau de dependência com outras variáveis;
- quando variáveis iguais são incluídas no conjunto de dados;
- quando são criadas variáveis fictícias, nas quais uma variável poderia resolver um problema; por exemplo: cria-se uma variável booleana *sexo masculino* e outra *sexo feminino* quando uma só variável chamada *sexo* poderia resolver o problema.

## Como identificar a multicolinearidade

De forma geral, há duas técnicas para identificar a multicolinearidade: utilizando a matriz de correlação e o fator de inflação da variância (VIF, *variance inflation factor*). A partir da matriz de correlação é possível verificar a existência de dependência linear entre um par de variáveis. Além disso, ela é capaz de analisar a existência de multicolinearidade, verificando se algum par de variáveis apresenta correlação alta. Nesse cenário, o cálculo dos autovalores da matriz de correlação pode conduzir o nível de relacionamento entre as variáveis. Por exemplo, um autovalor pequeno em relação aos demais aponta um mau condicionamento da matriz.

O fator de inflação da variância (VIF) é outra estratégia interessante (e é a principal adotada neste capítulo). VIF representa o incremento da variância sob a presença da multicolinearidade (YOO *et al.*, 2014). O cálculo do VIF é dado pela seguinte equação:

$$VIF = \frac{1}{1 - R^2}$$

Sendo  $R^2$  o coeficiente de determinação que já vimos anteriormente. Portanto, quanto mais próximo o valor de  $R^2$  ao valor 1, maior o valor de VIF e, consequentemente, maior a multicolinearidade com a variável independente específica. Ou seja, se  $VIF \leq 1$ , então as variáveis não são correlacionadas;  $1 < VIF \leq 5$  indica que as variáveis são moderadamente correlacionadas; e  $VIF > 5$  indica que as variáveis são altamente correlacionadas.

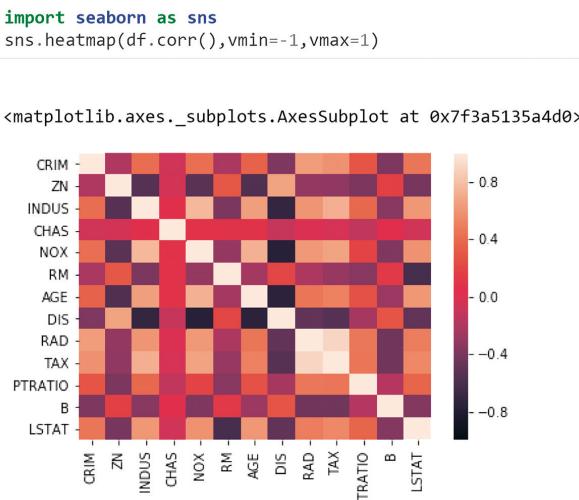
## Exemplos práticos de multicolinearidade

Para este estudo, você vai precisar carregar uma base de dados de preços de casas em Boston, Estados Unidos. Acompanhe o código a seguir.

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import warnings
from pandas import DataFrame, Series
from sklearn.datasets import load_boston

boston = load_boston()
X = boston["data"]
Y = boston["target"]
names = list(boston["feature_names"])
df = pd.DataFrame(X, columns=names)
```

Agora, para identificar se existe a multicolinearidade, você deve aplicar a primeira técnica, a matriz de correlação, como mostra a Figura 12. Observe que algumas variáveis têm um alto grau de correlação como *RM* e *LSTAT*.



**Figura 12.** Identificação de multicolinearidade com matriz de correlação.

Acompanhe agora a análise dos autovalores. Observe que o valor 6.12684883+0.j indica a possibilidade de existir variáveis com um alto grau de correlação.

```
from scipy import linalg as la
eigvals, eigvecs = la.eig(df.corr())
print(eigvals)

[ 6.12684883+0.j  1.43327512+0.j  1.24261667+0.j  0.85757511+0.j
 0.83481594+0.j  0.65740718+0.j  0.53535609+0.j  0.39609731+0.j
 0.06350926+0.j  0.27694333+0.j  0.16930298+0.j  0.18601437+0.j
 0.22023782+0.j]
```

Agora, para obter um nível de observação de forma ampla, veja a implementação da técnica de VIF na Figura 13. Perceba que *PTRATIO* e *TAX* têm um alto valor de VIF, o que significa que podem ser previstas por outras variáveis independentes no conjunto de dados.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

def calc_vif(X):
    # Calculo do VIF
    vif = pd.DataFrame()
    vif["variaveis"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

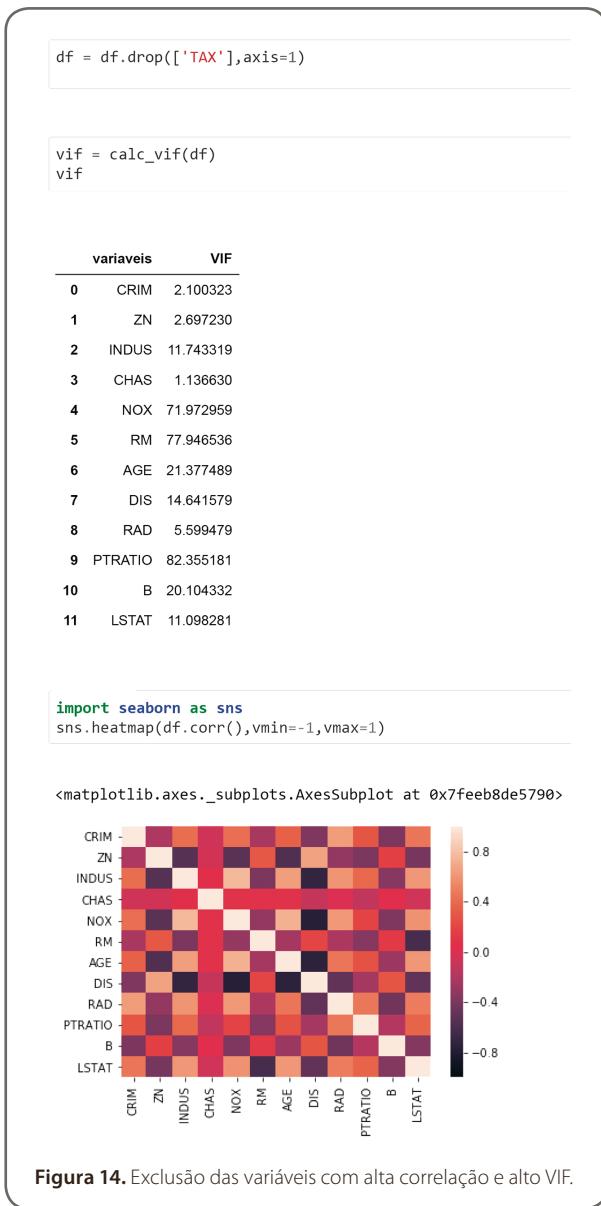
    return(vif)

vif = calc_vif(df)
vif
```

	variaveis	VIF
0	CRIM	2.100373
1	ZN	2.844013
2	INDUS	14.485758
3	CHAS	1.152952
4	NOX	73.894947
5	RM	77.948283
6	AGE	21.386850
7	DIS	14.699652
8	RAD	15.167725
9	TAX	61.227274
10	PTRATIO	85.029547
11	B	20.104943
12	LSTAT	11.102025

Figura 13. Implementação de VIF na base de dados de exemplo.

Para solucionar este problema, é importante que você delete variáveis que tenham alta correlação e alto VIF. No nosso exemplo, a variável *TAX* está presente nos dois cenários, então, deve ser removida, como mostra a Figura 14.



Você pode observar uma melhora no grau de correlações. Essa técnica pode ser aplicada em mais variáveis, caso você queira encontrar um conjunto de variáveis independentes. Isso se torna válido quando há interesse em fazer seleção de melhores atributos para modelos preditivos, por exemplo.

## Conclusão

Em alguns cenários, os analistas de dados precisam de estratégias matemáticas para conhecer o relacionamento entre os dados. Várias ferramentas estatísticas são importantes na observação e visualização de dados, entre elas Pandas, Numpy e Seaborn.

Neste capítulo, você estudou a análise de correlação entre os dados. A correlação é um dos conceitos estatísticos que mede a relação linear entre duas variáveis de interesse. Este estudo foi focado, inicialmente, na correlação de Pearson para variáveis numéricas. Depois foram vistas também as correlações usando tabelas cruzadas para lidar com correlações entre variáveis categóricas. Por fim foi apresentado o conceito de multicolinearidade e como tratar essa característica na base de dados.



## Referências

- DASS, M. *Cross-tabulation*. New Jersey: American Cancer Society; Wiley, 2010. (Wiley International Encyclopedia of Marketing).
- GRAPENTINE, T. Managing multicollinearity. *Marketing Research*, v. 9, n. 3, p. 10–21, 1997.
- LAROSE, C. D.; LAROSE, D. T. *Data science using Python and R*. New Jersey: Wiley, 2019.
- MONTGOMERY, D. C.; PECK, E. A.; VINING, G. *Introduction to linear regression analysis*. 5. ed. New Jersey: Wiley, 2012.
- MORETTIN, P. A.; BUSSAB, W. O. *Estatística básica*. 6. ed. São Paulo: Saraiva Educação, 2017.
- YOO, W. et al. A study of effects of multicollinearity in the multivariable analysis. *International Journal of Applied Science and Technology*, v. 4, n. 5, p. 9–19, 2014. Disponível em: [http://www.ijastnet.com/journals/Vol\\_4\\_No\\_5\\_October\\_2014/2.pdf](http://www.ijastnet.com/journals/Vol_4_No_5_October_2014/2.pdf). Acesso em: 10 jul. 2020.

## Leitura recomendada

- CHEN, P. Y.; POPOVICH, P. M. *Correlation: parametric and nonparametric measures*. New York: Sage, 2011. (Quantitative Applications in the Social Sciences, v. 139).



### Fique atento

Os *links* para *sites da web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS