CHEAT SHEET

Limpeza e tratamentos de dados
por **Escola de Data Science**

alura

Para o nosso DataFrame, vamos chamá-lo df.

Explorando os dados

```
df.head(n)  #Imprime as n primeiras linhas de um DataFrame
df['x'].value_counts()  #Conta o número de linhas com cada valor exclusivo da variável "w"
df.shape  #Tupla com o número de linhas e colunas do DataFrame "df"
df.describe()   #Estatísticas descritivas básicas para cada coluna do DataFrame "df"
df.quantile()  #Retorna valores no quantil fornecido sobre o eixo solicitado
df.info()  #Retorna informações sobre o DataFrame, como Index, Dtype e detalhes do uso de memória
```

Lidando com dados duplicados

```
df.duplicated().sum()   #Verifica se há dados duplicados
df[df.duplicated()]  #Filtra quais são as amostras duplicadas do conjunto de dados
df.drop_duplicates()  #Remove as amostras duplicadas
```

Lidando com dados nulos

```
df.fillna(valor)   #Substitue todos os que dados NA/nulos por valor
df.dropna()  #Remove as amostras que contenham valores ausentes (NA/null) em qualquer coluna do DataFrame
```

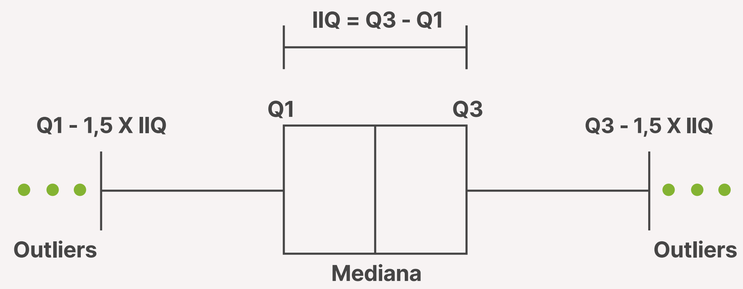
Selecionando os dados

```
df[col]  #Retorna a coluna (col) selecionada como uma Series
df[['col1', 'col2']]  #Seleciona as colunas 'col1' e 'col2' do DataFrame e retorna um novo
                        DataFrame contendo apenas essas colunas
df.loc[:, 'col1': 'col4']  #Seleciona todas as linhas e colunas entre col1 e col4 (inclusive)
df.loc[df['x'] == 5, ['col1', 'col4']]   #Seleciona todas as linhas em que a coluna 'x' é igual
                                           a 5 e somente as colunas 'col1' e 'col4'
df.loc[:,df.isnull().any()]  #Retorna valores no quantil fornecido sobre o eixo solicitado
df.info()  #Seleciona todas as colunas do DataFrame que contém pelo menos um valor nulo.
```

Lidando com outliers

```
#Plota o boxplot para a coluna age do DataFrame

import seaborn as sns
sns.boxplot(x=df['age'])
```




```
#Filtra as amostras candidatas a outliers

Q1 = df[coluna].quantile(.25)
Q3 = df[coluna].quantile(.75)
IIQ = Q3-Q1
limite_inferior = Q1 - 1.5*IIQ
limite_superior = Q3 + 1.5*IIQ
outliers_index = (df[coluna] < limite_inferior) | (df[coluna] > limite_superior)
df[outliers_index]
```

Lidando com features categóricas

```
df.replace() #Substitui valores em um DataFrame por outros valores especificados
pd.get_dummies(df, dtype=int)  #Aplica a técnica de One Hot Encoder (dummy) no DataFrame "df"
```

Tam
M
G
P
M

→

G	M	P
0	1	0
1	0	0
0	0	1
0	1	0