Análise de dados

Associação entre variáveis categóricas

Carregamento do dataset

Para esta explicação, utilizaremos a base de dados do RMS Titanic, disponível na biblioteca Seaborn.

Segue a importação das bibliotecas e o carregamento da base de dados:

```
import seaborn as sns
import pandas as pd
import numpy as np
import statsmodels.api as sm

✓ 0.0s

# Carregue o conjunto de dados do Titanic
df = sns.load_dataset('titanic')

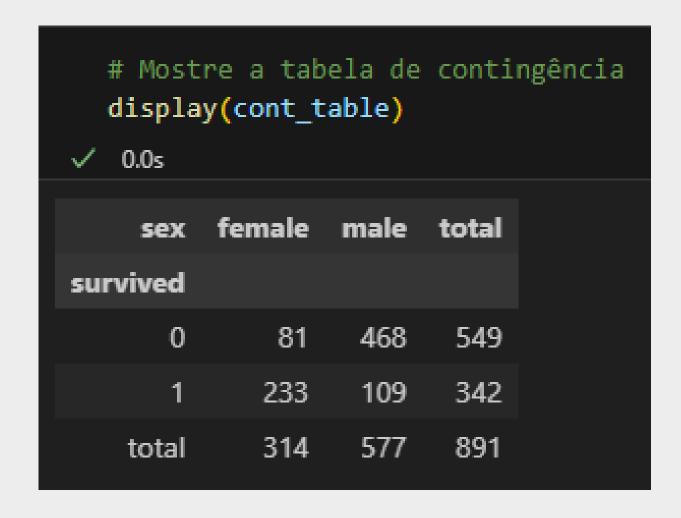
✓ 0.0s
```

Quando queremos analisar o comportamento de duas variáveis X1 e X2, podemos recorrer a uma tabela de contingência. Na tabela de contingência, cada elemento representa a frequência observada das realizações simultâneas de X1 e X2.

Em Python, podemos usar a função *crosstab* da biblioteca pandas para criar uma tabela de contingência. Neste exemplo, criaremos uma tabela de contingência para análise das variáveis "survived" e "sex".

```
# Crie a tabela de contingência
cont_table = pd.crosstab(
    df['survived'],
    df['sex'],
    margins=True,
    margins_name="total"
)
```

A tabela de contingência criada é mostrada na proxima página.



Esta tabela nos motstra algumas informações interessantes:

- Dos 891 passageiros, 342 sobreviveram.
- Dos 549 mortos, 81 eram do sexo feminino;
- Dos 342 sobreviventes, 233 eram do sexo feminino.

Estes são apenas alguns exemplos de informações. A leitura

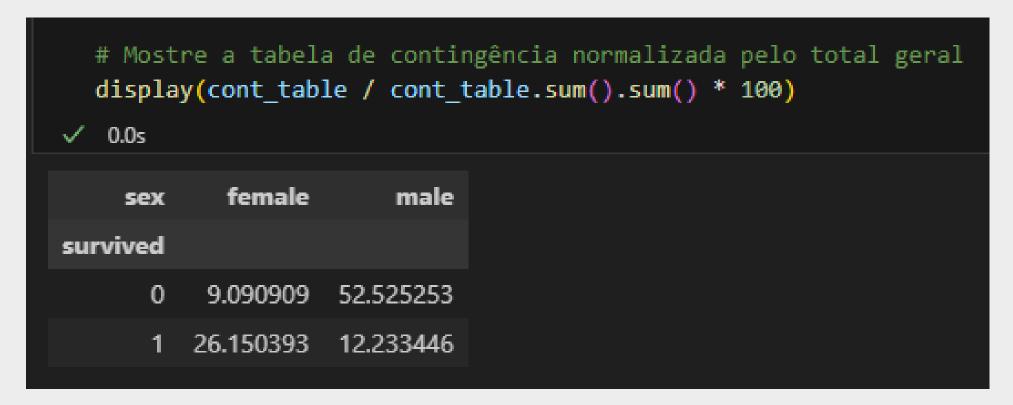
foi feita no sentido das linhas, mas também poderia ser feita no sentido das colunas.

Agora removeremos a coluna e linha de total para as próximas operações que serão realizadas com a tabela. Também poderiamos não ter passado os parâmetros margins (default=False) e margins_name na função crosstab.

```
# Remova coluna e linha de total
cont_table.drop("total", axis=1, inplace=True)
cont_table.drop("total", axis=0, inplace=True)
✓ 0.0s
```

Pode ser muito mais interessante visualizar a tabela de contingência em porcentagem em relação ao total geral ou em relação ao total das linhas/colunas.

Abaixo mostro uma forma de apresentar a tabela de contingência em porcentagem em relação ao total geral.



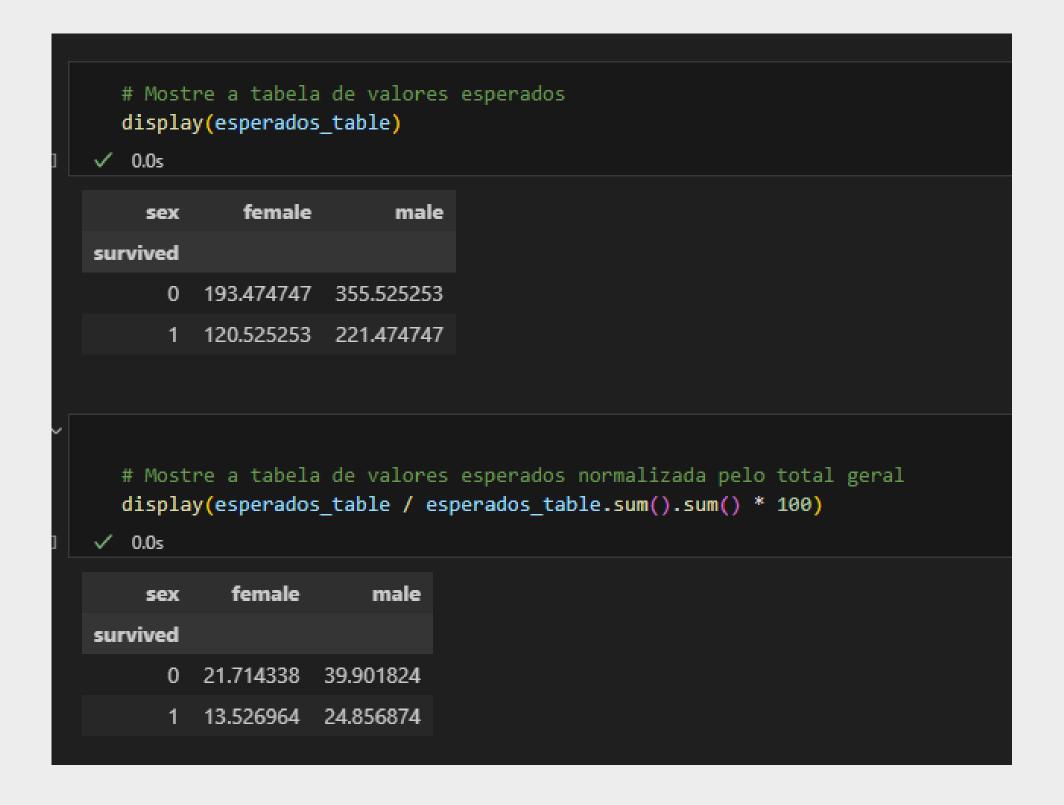
Também podemos calcular uma tabela com os valores esperados caso não houvesse dependência entre as variáveis analisadas. Cada elemento da tabela de valores esperados é calculado pela equação:

$$E_{ij} = \frac{(\text{Total da linha } i) \times (\text{total da coluna } j)}{\text{Total geral}}$$

Valores esperados

A tabela de valores esperados pode ser criada pelo produto externo entre os vetores contendo as somas das colunas por linha e as somas das linhas por coluna, respectivamente, divido pelo total geral. Em Python, podemos usar a função *outer* da biblioteca numpy conforme segue:

Valores esperados



Resíduos

Com a tabela de contingência (valores observados) e a tabela de valores esperados, podemos calcular uma tabela de resíduos, onde cada elemento da tabela é calculado pela equação:

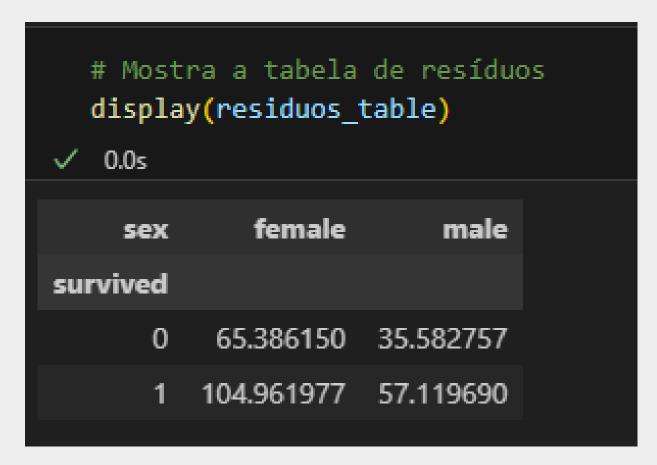
$$r_{ij} = \frac{(\text{observado}_{ij} - \text{esperado}_{ij})^2}{esperado_{ij}}$$

Em Python, a tabela de resíduos pode ser calculada da seguinte forma:

```
# Calcule a tabela de resíduos
residuos_table = (cont_table - esperados_table)**2 / esperados_table
✓ 0.0s
```

A seguir é mostrada a tabela de resíduos, que será utilizada para calcular a medida de associação entre as variáveis analisadas.

Qui-quadrado



O qui-quadrado (χ^2) mede o afastamento global entre as variáveis categóricas analisadas, sendo calculado pela soma dos resíduos.

A seguir veremos o cálculo do qui-quadrado pela soma dos resíduos da nossa tabela e pela biblioteca statsmodels, que oferece classes e funções de vários modelos estatísticos.

qui-quadrado

```
# Calcule o chi-quadrado
chi2 = residuos_table.sum().sum()
print("Qui-quadrado calculado:", chi2)

# Calcule o chi-quadrado pelo statsmodels
table = sm.stats.Table(cont_table)
chi2_sm = table.test_nominal_association().statistic
print("Qui-quadrado statsmodels:", chi2_sm)

✓ 0.0s

Qui-quadrado calculado: 263.05057407065567
Qui-quadrado statsmodels: 263.0505740706556
```

Como podemos ver, nosso qui-quadrado, calculado passo a passo, é basicamente o mesmo calculado através da biblioteca statsmodels. O mesmo é válido para variáveis com mais de duas categorias. Sugiro testar o passo a passo com as variáveis "survived" e "pclass" (categorias 1, 2 e 3), por exemplo.

Espero que tenham gostado!