

Senac

**Fecomércio
Sesc**

Data Science – Princípios e Técnicas

Maio
2024



Data Science

Onde me encontrar:

<https://www.linkedin.com/in/marco-mialaret-junior/>

e

<https://github.com/MatmJr>

Gráficos

Data Science



Existe outro gráfico que pode ajudar na visualização da correlação. O scatter plot (ou gráfico de dispersão) é utilizado para observar e mostrar a relação entre duas variáveis numéricas, permitindo aos analistas e cientistas de dados identificar correlações, tendências, agrupamentos, outliers e padrões nos dados de forma intuitiva.

Data Science

```
from pandas.plotting import scatter_matrix  
import matplotlib.pyplot as plt
```

```
# Criar a scatter matrix
```

```
axes = scatter_matrix(data_encoded[cols], alpha=0.5,  
figsize=(12, 12), diagonal='hist')
```

```
# Exibir o plot
```

```
plt.show()
```

Data Science



Olhando para as correlações:

Gráfico de barras mostrando a taxa de sobrevivência por gênero. Este gráfico pode destacar como a sobrevivência difere entre os gêneros.

Data Science

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.barplot(x='Sex', y='Survived', data=data)
plt.title('Taxa de Sobrevivência por Sexo')
plt.show()
```

Data Science

Gráfico de barras mostrando a taxa de sobrevivência por classe de passageiro. Isso pode mostrar como a classe afeta a sobrevivência.

Data Science

```
sns.barplot(x='Pclass', y='Survived', data = data)
plt.title('Taxa de Sobrevivência por Classe de Passageiro')
plt.show()
```

Data Science

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

grouped = data.groupby(['Pclass',
                        'Sex'])['Survived'].sum().reset_index()
```

```
sns.barplot(x='Pclass', y='Survived', hue='Sex', data=grouped)
plt.show()
```

Data Science

Gráfico mostrando a relação entre tarifa e classe de passageiro. Este gráfico pode ilustrar como a tarifa se relaciona inversamente com a classe.

Data Science

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.stripplot(x='Pclass', y='Fare', data=data, jitter=True,
size=5, linewidth=1)
plt.title('Tarifa por Classe de Passageiro')
plt.xlabel('Classe de Passageiro')
plt.ylabel('Tarifa')
plt.show()
```

Data Science

Boxplot mostrando a distribuição de idades em cada classe de passageiro. Útil para ver a tendência de idades em diferentes classes.

Data Science

```
sns.boxplot(x='Pclass', y='Age', data=data)  
plt.title('Distribuição de Idade por Classe de Passageiro')  
plt.show()
```

Data Science

A distribuição dos sobreviventes por idade:

```
import seaborn as sns
import matplotlib.pyplot as plt

sobreviventes = data[data['Survived'] == 1]

sns.histplot(data=sobreviventes, x='Age', hue='Pclass',
             element='step')
plt.show()
```

Data Science

```
sns.kdeplot(data=sobreviventes, x='Age', hue='Pclass', fill=True)  
plt.title('Distribuição de Densidade de Idades dos Sobreviventes  
por Classe')  
plt.xlabel('Idade')  
plt.ylabel('Densidade')  
plt.show()
```


O Plotly

Data Science



Um dashboard é uma ferramenta de visualização de dados que consolida e apresenta informações de maneira visual, interativa e fácil de entender, frequentemente usada para monitorar, analisar e comunicar insights importantes sobre conjuntos de dados.

Data Science

Na análise de dados, dashboards são essenciais para transformar dados brutos em informações acionáveis, permitindo aos usuários entender tendências, padrões e desempenhos em tempo real.

Data Science

`!pip install dash plotly`

Data Science

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import pandas as pd
```

Data Science

```
# Aplicação Dash
app = dash.Dash(__name__)

# Layout
app.layout = html.Div(children=[
    html.H1(children='Titanic Data Dashboard', style={'textAlign': 'center'}),
```

Data Science

```
dcc.Graph(  
    id='survival-sex',  
    figure=px.bar(data, x='Sex', y='Survived', color='Sex', barmode='group',  
                  labels={'Survived': 'Survival Rate', 'Sex': 'Sex'})  
),  
  
dcc.Graph(  
    id='class-fare',  
    figure=px.box(data, x='Pclass', y='Fare', color='Pclass',  
                  labels={'Fare': 'Fare', 'Pclass': 'Passenger Class'})  
)  
])
```

Data Science

```
# Executar o servidor
if __name__ == '__main__':
    app.run_server(debug=True)
```


Dúvidas?



Telephone:

E-mail:

