

Titanic: Estudo de Caso Voltado à Dados

Fatores que influenciaram na sobrevivência de
passageiros e tripulantes

Thiago Panini

PROJETO DE CONCLUSÃO DE CURSO - UDACITY, A UNIVERSIDADE DO VALE DO SILÍCIO

[HTTPS://GITHUB.COM/THIAGOPANINI/DATA-SCIENCE-COURSES/TREE/MASTER/UDACITY](https://github.com/thiagopanini/data-science-courses/tree/master/udacity)

Trabalho realizado para conclusão do Programa Nanodegree: Fundamentos de Data Science I, com o intuito de aplicar conceitos de estatística descritiva para verificar os atributos diretamente relacionados com a taxa de sobrevivência de passageiros e tripulantes à bordo do Titanic.

Thiago Henrique Gomes Panini - Outubro 2018



Contents

1	Introdução	5
1.1	Motivação	5
1.2	Objetivo	6
1.3	Contextualização	6
2	Análise Exploratória	7
2.1	Conhecendo os Dados	7
2.1.1	Apresentação	7
2.1.2	Tipos Primitivos	8
2.1.3	Dados Faltantes	9
2.1.4	Dados Duplicados	10
2.1.5	Dados Únicos	11
2.1.6	Referências	12
2.2	Preparando os Dados	12
2.2.1	Modificando Índice	12
2.2.2	Preenchendo Dados Nulos	13
2.2.3	Eliminando Colunas	18
2.2.4	Salvando Novo Dataset	18
3	Exploração Gráfica	19
3.1	Taxa de Sobrevidentes	19
3.2	Taxa de Sobrevivência por Gênero	21
3.3	Taxa de Sobrevivência por Classe Econômica	25

3.4	Taxa de Sobrevivência por Faixa Etária	29
3.5	Taxa de Sobrevivência por Porto de Embarque	36
4	Conclusão	40
4.1	Referências	41



1. Introdução

1.1 Motivação

O mundo moderno é um ambiente metamórfico de constantes mudanças que envolvem, principalmente, áreas relacionadas a tecnologia. Um dos maiores movimentos computacionais dos últimos tempos, considera a manipulação de grandes conjuntos de dados como forma de aprimorar processos e negócios, tratando e resolvendo problemas outrora esquecidos devido a complexidade envolvida.

A tomada de decisão baseada em *insights* retirados destes conjuntos de dados é um papel atribuído ao Cientista de Dados, profissional que, a partir de uma massa de informações, estruturadas ou não estruturadas, é responsável por averiguar, analisar, tratar, preparar, organizar, apresentar e realizar uma série de operações que visam encontrar soluções para os desafios propostos.

Dentro deste contexto, dados são gerados a todo momento, sempre em quantidades gigantescas e velocidades astronômicas. Este princípio, conhecido como *Big Data*, é o verdadeiro combustível para o surgimento de novos problemas de negócio que, cada vez mais, exigem soluções inovadoras e revolucionárias, desde aplicações de estatísticas descritivas até implementações de algoritmos de *Machine Learning* para análises preditivas.

Há uma série de conceitos envolvendo o universo de Ciência de Dados e, apesar do destaque principal dado às habilidades estatísticas e de programação, é essencial ter uma visão holística do problema de negócio alvo da análise. Cada situação traz consigo novos desafios munidos de uma flexibilidade única, proporcionando diferentes possibilidades de alcançar os resultados esperados por diferentes caminhos. O dinamismo resultante é o grande responsável por transformar tarefas, antes maçantes, em episódios apaixonantes e arrebatadores.

O projeto em questão contempla a análise e preparação de dados através de um fato histórico: será disponibilizada uma base contendo informações de parte dos passageiros e tripulantes do *Titanic*, navio britânico lançado em sua viagem inaugural em abril de 1912 e sendo mundialmente conhecido pelo seu naufrágio.

A grande motivação para a realização deste projeto é exemplificada pelos benefícios trazidos pelo procedimento conhecido como *Data Wrangling*, ou seja, o processo de transformar e mapear dados brutos em outro formato, com a intenção de torná-lo mais apropriado e valioso para uma

variedade de propósitos. O tratamento de dados de passageiros e tripulantes do navio Titanic não só contribui para um notório aprendizado, como também reforça características analíticas e promove discussões a respeito de um fato marcante de conhecimento mundial.

1.2 Objetivo

O objetivo deste projeto é avaliar, analisar, entender, estudar, preparar e aplicar um conjunto de estatísticas descritivas para retirada de conclusões em uma base de dados que transcreve características de passageiros e tripulantes do Titanic afim de verificar, em primeira instância, quais os fatores que influenciaram diretamente na probabilidade de sobrevivência dos presentes na viagem.

Utilizando a linguagem Python e as ferramentas do pacote Anaconda, espera-se:

- Aplicar todos os passos envolvidos em um processo de análise de dados típico;
- Realizar perguntas que podem ser respondidas por um conjunto de dados e, em seguida, respondê-las;
- Investigar problemas em um conjunto de dados;
- Adquirir prática em comunicar os resultados da análise;
- Utilizar operações vetorizadas no *NumPy* e *Pandas* para aprimorar o código;
- Trabalhar com Series e DataFrame do *Pandas*, que permitem acessar os dados de forma mais conveniente;
- Utilizar o *Matplotlib* para produzir gráficos mostrando as descobertas e os resultados obtidos.

1.3 Contextualização

A busca por conhecimento na área de Ciência de Dados, por si só, já é uma aventura extremamente interessante e recompensadora. Direcionar esta busca através de cursos qualificados é ainda mais gratificante. O Programa Nanodegree, oferecido pela Udacity - A Universidade do Vale do Silício, reúne conceitos avançados e modernos, proporcionando ensinamentos dinâmicos e eficientes à alunos com pouca, nenhuma ou muita vivência no assunto. Através do curso, é possível aplicar o conhecimento teórico adquirido em situações práticas, como projetos e desafios com temas universais e de presença cotidiana.

Para a conclusão do curso **Fundamentos de Data Science I**, foi proposto um projeto final envolvendo a análise de uma base contendo dados de 891 dos 2.224 passageiros e tripulantes a bordo do Titanic. Com essas informações, deve-se realizar questionamentos prévios para serem respondidos utilizando as ferramentas do Python acima mencionadas.

O naufrágio do navio Titanic foi um dos mais marcantes acontecimentos da história. Em 15 de abril de 1912, durante sua viagem inaugural, o Titanic afundou depois de colidir com um iceberg, somando 1.502 vítimas entre passageiros e tripulantes. Esta tragédia chocou a comunidade internacional e levou a melhores normas de segurança para os navios.



2. Análise Exploratória

2.1 Conhecendo os Dados

2.1.1 Apresentação

Com a definição do objeto principal da análise, deve-se, neste momento, iniciar os passos visando o manuseio da base de dados a ser utilizada. Utilizando os recursos do Pandas, biblioteca desenvolvida em Python para o tratamento de bases de dados oriundas das mais diversas fontes e nos mais variados formatos, é realizado o primeiro contato com o Dataset Titanic.

```
import pandas as pd  
df = pd.read_csv('C:/Users/thiagoPanini/Downloads/datasets/titanic-data-6.csv')  
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Allen, Mr. William Henry	male	35.0	1	0	113803	53.1000	C123	S
4	5	0	3			35.0	0	0	373450	8.0500	NaN	S

Figure 2.1: Primeiras cinco linhas com dados de passageiros e tripulantes

Com isso, cria-se familiaridade com o Dataset, visualizando alguns de seus registros, bem como parte de seu conteúdo. Para estreitar ainda mais essa relação, é extremamente importante conhecer a fundo os atributos (ou *features*) do conjunto de dados.

Abaixo, é possível visualizar uma explanação sobre cada um dos campos a serem trabalhados neste projeto:

- **PassengerId** - Coluna criada para identificação dos registros;
- **Survived** - Indica se o passageiro sobreviveu ou não ao acidente (0=Não, 1=Sim);
- **PClass** - Classe Econômica referente ao ticket comprado pelo passageiro;

- **Sex** - Gênero do passageiro;
- **Age** - Idade do passageiro;
- **SibSp** - Número de irmãos ou cônjuges a bordo com o passageiro;
- **Parch** - Número de pais ou crianças a bordo com o passageiro;
- **Ticket** - Número do ticket do passageiro;
- **Fare** - Tarifa paga pelo passageiro;
- **Cabin** - Número da cabine do passageiro;
- **Emarked** - Porto no qual o passageiro embarcou (C=Cherbourg, Q=Queenstown, S=Southampton).

Neste ponto da análise, o ambiente encontra-se preparado e os dados prontos para receberem uma exploração mais concisa. Visando um maior controle nas possíveis modificações futuras, o último passo antecedente à investigação da base contempla a comunicação das dimensões do DataFrame lido pelo Pandas, ou seja, a quantidade de linhas e colunas presentes.

```
print(f'O Dataset possui {df.shape[0]} linhas e {df.shape[1]} colunas.')
O Dataset possui 891 linhas e 12 colunas.
```

Figure 2.2: Contagem de linhas e colunas do Dataset

2.1.2 Tipos Primitivos

As informações reunidas sobre os dados, até o presente momento, são suficientes para aplicar uma abordagem mais incisiva no que diz respeito a preparação da base. A verificação dos tipos primitivos de cada uma das colunas se faz extremamente necessária pois, assim, é possível diagnosticar alguns problemas transcritos, por exemplo, por atributos numéricos salvos como objetos do tipo texto ou dados do tipo inteiro que estão registrados como ponto flutuante.

df.dtypes	
PassengerId	int64
Survived	int64
Pclass	int64
Name	object
Sex	object
Age	float64
SibSp	int64
Parch	int64
Ticket	object
Fare	float64
Cabin	object
Embarked	object
dtype:	object

Figure 2.3: Tipos primitivos do conjunto de dados

Lembrando que objetos *DataFrame* indicam *object* como sendo, na verdade, objetos do tipo *string*. Nesta etapa, o conhecimento sobre os dados e o entendimento de seus atributos são pontos

cruciais para a obtenção de um diagnóstico adequado sobre possíveis alterações em seus tipos primitivos. Caso haja alguma dúvida com relação aos tipos de dados presentes na linguagem Python, a imagem abaixo servirá de auxílio.

Class	Description	Immutable?
<code>bool</code>	Boolean value	✓
<code>int</code>	integer (arbitrary magnitude)	✓
<code>float</code>	floating-point number	✓
<code>list</code>	mutable sequence of objects	
<code>tuple</code>	immutable sequence of objects	✓
<code>str</code>	character string	✓
<code>set</code>	unordered set of distinct objects	
<code>frozenset</code>	immutable form of set class	✓
<code>dict</code>	associative mapping (aka dictionary)	

Figure 2.4: A linguagem Python e seus tipos primitivos

Considerando a análise em questão e os resultados obtidos com o código empregado, conclui-se que não há a necessidade de nenhum tipo de transformação neste momento, visto que cada um dos tipos primitivos encontrados corresponde ao que se espera de sua respectiva coluna.

2.1.3 Dados Faltantes

Um outro ponto de notável importância no processo de exploração se dá pela presença de dados faltantes, ou *Missing Data*, de acordo com a literatura. As decisões tomadas nessa etapa refletem diretamente nos resultados posteriores, sendo estes obtidos através de estatística descritiva ou algoritmos de *Machine Learning*.

Em Python, é possível aplicar diferentes metodologias para visualizar a presença de dados faltantes. Empregando algumas destas estruturas no Dataset alvo da análise, foi possível coletar os seguintes resultados:

```
# Há dados faltantes?
df.isnull().values.any()

True

# Contabilizando
df.isnull().values.sum()
```

689

Figure 2.5: Dados faltantes no Dataset Titanic

Há uma série de abordagens diferentes para o tratamento de dados faltantes. Preenchimento com a média, mediana, moda, aplicação de regressão ou até mesmo a exclusão da referida instância são os principais métodos considerados. Dessa forma, não existe singularidade ou regra padrão para tratar dados faltantes, mas sim um julgamento pontual para definir, em cada aplicação, o melhor método de acordo com os objetivos a serem atingidos.

A figura 2.5 indica que há dados faltantes no Dataset do Titanic, contabilizando, ao todo, 689 instâncias. Dessa forma, é preciso investigar com mais detalhes as origens destes dados faltantes e, posteriormente, avaliar a melhor decisão para o prosseguimento do projeto, sempre de acordo com o que se deseja alcançar. A figura abaixo proverá maiores detalhes sobre estes dados faltantes.

```
# Infos adicionais
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived        891 non-null int64
Pclass          891 non-null int64
Name            891 non-null object
Sex             891 non-null object
Age             714 non-null float64
SibSp           891 non-null int64
Parch           891 non-null int64
Ticket          891 non-null object
Fare            891 non-null float64
Cabin           204 non-null object
Embarked        889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Figure 2.6: Informações adicionais sobre dados faltantes no Dataset

Através desta análise detalhada, percebe-se que a coluna *Cabin* é a mais afetada por dados faltantes, dado que, dos 891 dados de entrada esperados por cada coluna, apenas 204 estão realmente preenchidos no referido atributo. Avaliando a situação e levando em conta o intuito do projeto, é plausível definir pela exclusão do atributo *Cabin*, uma vez que os efeitos desta coluna, para os objetivos da análise, não provocam alterações nos resultados.

Por outro lado, o atributo *Age* que, por sua vez, define a idade dos passageiros presentes no Titanic, possui grande significância e merece maior atenção. Visando a simplificação do projeto, para sanar o problema de dados faltantes nesta coluna será realizado um procedimento de preenchimento com a média, apesar da ciência de que, possivelmente, esta não seja a melhor abordagem. Tal operação será realizada na sessão de tratamento de dados.

Por fim, porém não menos perceptível, a análise do atributo *Embarked*, cuja descrição define o porto no qual cada passageiro embarcou no navio e que, de acordo com a figura 2.7 possui 3 dados faltantes, poderá ser vista em detalhes na sessão de preenchimento de dados faltantes.

2.1.4 Dados Duplicados

Complementando o conjunto de funções e métodos aplicados com o objetivo de prover um maior conhecimento sobre os dados, encontra-se aquelas que identificam a presença de dados duplicados.

Na grande maioria das vezes, dados duplicados são indesejados, pois refletem instâncias repetidas de um registro. Por outro lado, como já visto na sessão de dados faltantes, o processo de preparação dos dados é algo artesanal que requer análises e discussões a todo instante, sempre visando o objetivo

a ser alcançado.

Dessa forma, é possível dizer que há casos onde dados duplicados realmente identificam instâncias idênticas de um registro. Um exemplo de uma situação análoga pode ser encontrado em Datasets contendo informações químicas de vinhos, uma vez que é perfeitamente razoável identificar dois ou mais vinhos com as mesmas propriedades.

A figura abaixo exemplifica a linha de código utilizada para verificar a presença de dados duplicados no Dataset Titanic.

```
# Dados duplicados
df.duplicated().any()
False
```

Figure 2.7: Verificação de dados duplicados no Dataset Titanic

De acordo com o resultado da função aplicada ao conjunto de dados, nota-se que não há dados duplicados e, portanto, não será necessário empregar nenhuma outra função para tratar tal situação.

2.1.5 Dados Únicos

Fechando o ciclo focado na compreensão inicial dos dados, tem-se uma visão útil sobre as instâncias únicas de cada uma das colunas do Dataset em questão. Com essa informação, é verificada a veracidade das informações prévias fornecidas sobre os dados e adiantam-se ideias sobre plotagens gráficas futuras.

No Dataset Titanic, as entradas únicas de cada uma das colunas podem ser visualizadas através da figura abaixo:

```
# Valores únicos em cada coluna
df.nunique()

PassengerId    891
Survived        2
Pclass          3
Name            891
Sex             2
Age             88
SibSp           7
Parch           7
Ticket          681
Fare            248
Cabin          147
Embarked        3
dtype: int64
```

Figure 2.8: Quantidade de entradas únicas em cada coluna

Analizando alguns casos onde há baixa quantidade de entradas únicas, como ocorrido em *Survived*, *Pclass*, *Sex* e *Embarked*, por exemplo, atesta-se a veracidade da descrição dos dados divulgada no início deste trabalho.

Isto pois, os dois valores únicos indicados para a coluna *Survived* são referentes ao numeral 0, indicando as vítimas, e ao numeral 1, indicando os sobreviventes. O mesmo número de entradas é visto em *Sex* que, por sua vez, traz os gêneros masculino e feminino. Em *Pclass*, as três entradas únicas referem-se, respectivamente, às classes Alta, Média e Baixa. Por fim, o atributo *Embarked* traz os três diferentes portos nos quais os passageiros poderiam ter embarcado para a referida viagem.

2.1.6 Referências

 Para a devida abordagem dos conteúdos dessa sessão, foram utilizados princípios, métodos, funções e diversas outras ferramentas do conhecido PyData Stack, ou seja, conjunto de bibliotecas em Python que auxiliam na Ciência de Dados como um todo.

- Kaggle Titanic
<https://translate.google.com/#en/pt/sibling>
- What is Data Munging?
<https://www.edq.com/uk/glossary/data-munging/>
- Data Wrangling
https://en.wikipedia.org/wiki/Data_wrangling

2.2 Preparando os Dados

Após a devida apresentação, familiarização e a exposição introdutória ao Dataset contendo dados dos passageiros e tripulantes do Titanic, foram mapeados os principais pontos a serem revistos ou alterados.

2.2.1 Modificando Índice

O primeiro ponto a ser destacado gira em torno da coluna *PassengerId*. A abertura de arquivos em formato .csv em Python é feita através da biblioteca Pandas que, por sua vez, oferece uma função específica com parâmetros bem definidos. Um destes argumentos (não obrigatórios), pode designar uma coluna específica do Dataset como índice do DataFrame gerado. Para entender melhor a motivação deste tópico, é importante visualizar novamente as primeiras linhas do conjunto de dados.

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Nan	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2, 3101282	7.9250	Nan	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Nan	S

Figure 2.9: Head do conjunto de dados

Como o arquivo foi aberto sem a especificação de parâmetros adicionais, o Pandas gera, por padrão, uma coluna numérica que atua como índice do DataFrame. Entretanto, como pode ser visto na própria figura acima, a coluna *PassengerId* claramente poderia atuar como índice, visto que seu conteúdo se resume a números inteiros, incrementais e não repetitivos, funcionando de forma a identificar cada registro de maneira única.

Há basicamente duas formas de atribuir uma coluna ao índice de um DataFrame: a primeira contempla a reabertura do arquivo alterando um parâmetro específico. A segunda forma, por sua vez, pode ser aplicada através de uma linha de código, sem a necessidade de carregar novamente o Dataset, conforme ilustrado abaixo:

```
df.set_index('PassengerId', inplace=True)
df.head()
```

	Survived	Pclass		Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId												
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599	71.2833	C85	C	
3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	
4	1	1	Allen, Mr. William Henry	male	35.0	1	0	113803	53.1000	C123	S	
5	0	3			35.0	0	0	373450	8.0500	NaN	S	

Figure 2.10: Alterando índice do conjunto de dados

2.2.2 Preenchendo Dados Nulos

Conforme visto na sessão 2.1.3, a figura 2.7 demonstra a quantidade de dados preenchidos em cada coluna do conjunto de dados Titanic. Algumas delas são de grande importância para as análises futuras. Uma visão análoga e objetiva pode ser identificada abaixo para visualizar a real quantidade de dados faltantes por atributo:

```
# Dados nulos
df.isnull().sum()

PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin          687
Embarked        2
dtype: int64
```

Figure 2.11: Quantidade de dados faltantes por atributo

O atributo *Age*, por exemplo, identifica a idade de cada um dos passageiros e possui 177 entradas nulas. Para trata-las, há diferentes abordagens. Visando uma maior objetividade e simplicidade do projeto e, tendo em vista o escopo deste trabalho, os dados faltantes desta coluna serão preenchidos com a média resultante. Em outras palavras, calcula-se a média de todas as idades presentes, atribuindo assim o valor resultante às lacunas onde não há preenchimento. Antes disso, é importante visualizar algumas estatísticas do atributo que representa as idades.

```

# Características do atributo Age
print(f'Média de idades: {df["Age"].mean():.1f} anos.')
print(f'Mediana de idades: {df["Age"].median():.2f} anos.')
print(f'Maior idade encontrada: {df["Age"].max()} anos.')
print(f'Menor idade encontrada: {df["Age"].min()} anos.')

Média de idades: 29.7 anos.
Mediana de idades: 28.00 anos.
Maior idade encontrada: 80.0 anos.
Menor idade encontrada: 0.42 anos.

df['Age'].describe()

```

count	714.000000
mean	29.699118
std	14.526497
min	0.420000
25%	20.125000
50%	28.000000
75%	38.000000
max	80.000000
Name:	Age, dtype: float64

Figure 2.12: Estatísticas sobre o atributo "Age"

Dessa forma, o procedimento para preenchimento dos dados faltantes com a média se dá através da sequência de códigos abaixo:

```

# Preenchendo dados com a média
age_mean = df['Age'].mean()
df['Age'].fillna(age_mean, inplace=True)

# Verificando resultado
print(f'Há dados faltantes na coluna Age? {df["Age"].isnull().any()}')


Há dados faltantes na coluna Age? False

# Visão geral
df.isnull().sum()

```

Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2
	dtype: int64

Figure 2.13: Preenchimento e nova verificação de dados faltantes na coluna "Age"

Com o procedimento já realizado dentro do atributo *Age*, é preciso investigar as duas instâncias de dados faltantes no atributo *Embarked*. Com o código descrito pela figura abaixo, é possível visualizar os detalhes sobre esses dois registros.

```
# Visualizando detalhes sobre dados faltantes em Embarked
df[df['Embarked'].isnull()].head()
```

	Survived	Pclass		Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId												
62	1	1		Icard, Miss. Amelie	female	38.0	0	0	113572	80.0	B28	NaN
830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	0	113572	80.0	B28	NaN

Figure 2.14: Registros com dados faltantes na coluna Embarked

Analizando a figura 2.14 acima, é possível visualizar alguns pontos interessantes sobre os dois registros de local de embarque desconhecido:

- Ambos são de classe alta (Pclass=1);
- Ambos são do gênero feminino (Sex=female);
- Ambos possuem o mesmo número de Ticket (Ticket=113572);
- Ambos pagaram a mesma quantia no ticket (Fare=80.00);
- Ambos se instalaram na mesma cabine (Cabin=B28).

Portanto, para assumir o local onde as duas passageiras embarcaram, é necessário investigar, entre os atributos semelhantes, alguma relação com o local de embarque.

1. Quantos registros possuem Ticket número 113572 ou se instalaram na cabine B28? Onde estes embarcaram?

```
# Dedução pelo Ticket?
print(f"Pessoas com o ticket 113572: {((df['Ticket'] == '113572').sum())}")
print(f"Pessoas na cabine B28: {((df['Cabin'] == 'B28').sum())}")

Pessoas com o ticket 113572: 2
Pessoas na cabine B28: 2
```

Figure 2.15: Investigando atributos Ticket e Cabine

Com o resultado dos métodos aplicados, nota-se a presença de apenas dois registros, ou seja, os mesmos registros cujos dados de embarque são desconhecidos, impossibilitando assim deduzir o local de embarque através dos atributos *Ticket* e *Cabin*.

2. O valor pago no ticket (Fare=80) pode ser um sinal que permita identificar o local de embarque dos passageiros?

```
# Pessoas que pagaram 80 no ticket
f_80 = df.query('Fare == 80')['Name']
f_80

PassengerId
62           Icard, Miss. Amelie
830    Stone, Mrs. George Nelson (Martha Evelyn)
Name: Name, dtype: object
```

Figure 2.16: Passageiros que pagaram exatamente 80.00 no ticket

A figura 2.16 acima evidencia que as únicas duas pessoas que pagaram exatamente 80.00 no ticket são aquelas alvos da investigação e que, portanto, será preciso aplicar uma outra metodologia para captar uma maior massa de análise.

```
# Dedução pelo atributo Fare? Proporção de pessoas que pagaram 80 +/- 30
limit_above = 30
limit_below = 30
target = 80
near_fare = df.query('Fare < @target+@limit_above & Fare > @target-@limit_below')
near_fare = near_fare.groupby('Embarked').count()['Name']
emb_totals = df.groupby('Embarked').count()['Name']
proportions = 100 * near_fare / totals

for state in near_fare.index:
    if state == 'C':
        estado = 'Cherbourg'
    elif state == 'Q':
        estado = 'Queenstown'
    else:
        estado = 'Southampton'
    print(f"Embarcaram em {estado} e pagaram entre 50.00 e 110.00: {proportions[state]:.2f}%")
```

Embarcaram em Cherbourg e pagaram entre 50.00 e 110.00: 21.43%
Embarcaram em Queenstown e pagaram entre 50.00 e 110.00: 2.60%
Embarcaram em Southampton e pagaram entre 50.00 e 110.00: 11.02%

Figure 2.17: Locais de embarque de pessoas que pagaram entre 50.00 e 110.00 no ticket

Aplicando um *range* no valor do ticket, foi possível identificar, através da figura 2.17, que a maior proporção de pessoas que cujo ticket custou entre 50.00 e 110.00 embarcaram em Cherbourg. Trata-se de apenas um indício sobre o possível local de embarque dos passageiros de ids 62 e 830.

Como apenas duas pessoas embarcaram em Queenstown e satisfazem o referido critério, é possível inferir que esta cidade está fora de cogitação.

3. A classe dos passageiros (Pclass=1) pode definir o local de embarque?

```
# Proporção de Classe 1 por porto de embarque

class_embarked = df.groupby(['Pclass', 'Embarked']).count()['Name']
states_index = df['Embarked'].value_counts().index
prop_class = 100 * class_embarked / emb_totals

for state in states_index:
    if state == 'C':
        estado = 'Cherbourg'
    elif state == 'Q':
        estado = 'Queenstown'
    else:
        estado = 'Southampton'
print(f'Embarcaram em {estado} e pertencem à Classe 1: {prop_class[1][state]:.2f}%')

Embarcaram em Southampton e pertencem à Classe 1: 19.72%
Embarcaram em Cherbourg e pertencem à Classe 1: 50.60%
Embarcaram em Queenstown e pertencem à Classe 1: 2.60%
```

Figure 2.18: Locais de embarque de pessoas da Classe 1

Mais uma vez, o resultado da investigação através da classe dos passageiros indicou que a maior parte pertencente à Classe 1 embarcou em Cherbourg.

Neste ponto, é preciso ter clareza em admitir que não há um método preciso para identificar a origem dos dados faltantes, apesar da estatística indicar uma maior probabilidade de seguir caminhos assertivos.

Todavia, de acordo com as investigações realizadas, supõe-se que as duas passageiras cujo local de embarque é indefinido, partiram para o Titanic de Cherbourg, sendo este o valor preenchido nas lacunas.

```
# Preenchendo dados faltantes em Embarked
print(f'Dados faltantes antes da inserção: {df["Embarked"].isnull().sum()}')
df['Embarked'].fillna('C', inplace=True)
print(f'Dados faltantes após a inserção: {df["Embarked"].isnull().sum()}')

Dados faltantes antes da inserção: 2
Dados faltantes após a inserção: 0

# Verificando alterações
df.isnull().sum()

Survived      0
Pclass        0
Name          0
Sex           0
Age           0
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin        687
Embarked      0
dtype: int64
```

Figure 2.19: Inserindo e verificando dados faltantes na coluna Embarked

2.2.3 Eliminando Colunas

A análise prévia realizada na sessão 2.1.3 sobre Dados Faltantes indicou que a coluna *Cabin*, possuindo 687 dados faltantes dos 891 possíveis, é inefetiva para os objetivos desta análise. Seus valores, refletidos pela indicação das cabines nas quais se estabeleceram os passageiros e tripulantes, não apresentam relevância considerável e, portanto, decidiu-se pela sua exclusão do Dataset.

```
# Excluindo coluna Cabin
df.drop('Cabin', axis=1, inplace=True)
df.head()
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599 STON/O2. 3101282	71.2833 7.9250	C S
3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	113803	53.1000	S
4	1	1	Allen, Mr. William Henry	male	35.0	1	0	373450	8.0500	S
5	0	3								

Figure 2.20: Eliminando coluna Cabin e verificando resultado

2.2.4 Salvando Novo Dataset

Com as diversas alterações realizadas no conjunto de dados, faz-se necessário o registro de um novo arquivo para que seja possível resgatar o trabalho de maneira rápida e eficiente. Isto é feito através da linha de código abaixo:

```
# Salvando novo Dataset
df.to_csv('C:/Users/thiagoPanini/Downloads/datasets/titanic-data-6-edited.csv')
```

Figure 2.21: Salvando alterações em um novo Dataset

3. Exploração Gráfica

Em um processo de análise de dados, é muito comum atribuir conclusões e retiradas de insights por meio de visualizações e painéis gráficos. Se confeccionadas de maneira adequada, tais visões são capazes de mostrar, de forma clara e precisa, a realidade dos dados, seja através de comparações, distribuições, correlações, ou quaisquer outros parâmetros.

Para dar continuidade neste capítulo, é preciso realizar a importação das bibliotecas responsáveis pela construção das plotagens, sendo elas:

- Módulo pyplot da biblioteca Matplotlib - Plotagens gráficas
- Seaborn - Plotagens gráficas e melhorias no design
- NumPy - Suporte para plotagens

3.1 Taxa de Sobreviventes

Como o objetivo do trabalho gira em torno do número de sobrevidentes e como este está relacionado a outros atributos, é importante visualizar, antes de mais nada, qual a taxa de sobrevivência dada pelas informações contidas no Dataset. Para cumprir este papel, foi criado o gráfico abaixo:

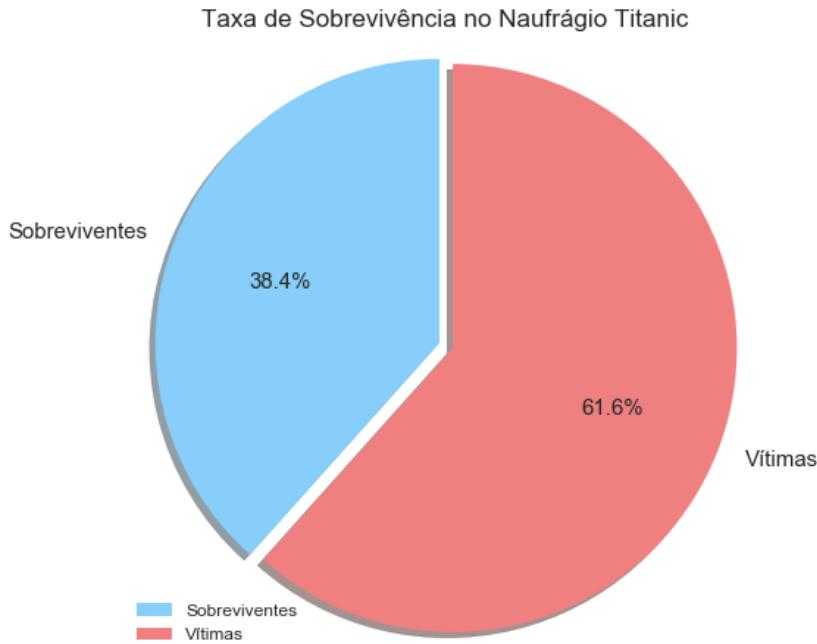


Figure 3.1: Taxa de Sobrevivência

A partir do gráfico acima, é possível identificar que, dos 891 passageiros e tripulantes, 38.4% sobreviveram e 61.6% foram vítimas. Tais números estão relativamente próximos dos índices históricos propostos pelos 2.224 presentes no navio, uma vez que estes, segundo o site *Wikipedia*, apresentam taxas de 32% de sobreviventes e 68% de vítimas. Em outras palavras, pode-se dizer que os registros presentes neste Dataset estão coerentes com a realidade.

Abaixo, segue o código utilizado para a plotagem da figura 3.2:

```

# Número de sobreviventes e vítimas
tx_surv = df.groupby('Survived').count()['Sex'][1]
tx_vict = df.groupby('Survived').count()['Sex'][0]

# Taxa de Sobreviventes
labels = ['Sobreviventes', 'Vítimas']
sizes = [tx_surv, tx_vict]
explode = (0.05, 0)
colors = ['lightskyblue', 'lightcoral']
fig, ax = plt.subplots(figsize=(6.5, 6.5))
wedges, texts, autotexts = ax.pie(sizes, labels=labels,
                                   startangle=90, shadow=True, explode=explode,
                                   autopct='%.1f%%', colors=colors)
ax.set_title('Taxa de Sobrevivência no Naufrágio Titanic', fontsize=17)
ax.axis('equal')
plt.legend(fontsize=12, loc='lower left')
plt.tight_layout()
plt.setp(autotexts, size=15)
plt.setp(texts, size=15)
plt.show()

```

Figure 3.2: Código para plotagem - Taxa de Sobrevivência

3.2 Taxa de Sobrevida por Gênero

Continuando a investigação sobre os sobreviventes do naufrágio, faz-se oportuna a visualização dos índices de sobrevida de acordo com o gênero dos passageiros. As investigações realizadas nos dados permitiram criar a tabela abaixo:

Gênero	Instâncias	Taxa de Sobrevida
Masculino	577	18.89%
Feminino	314	74.20%

Table 3.1: Taxa de Sobrevida por Gênero

Graficamente, estas informações estão dispostas de acordo com a figura 3.3 abaixo:

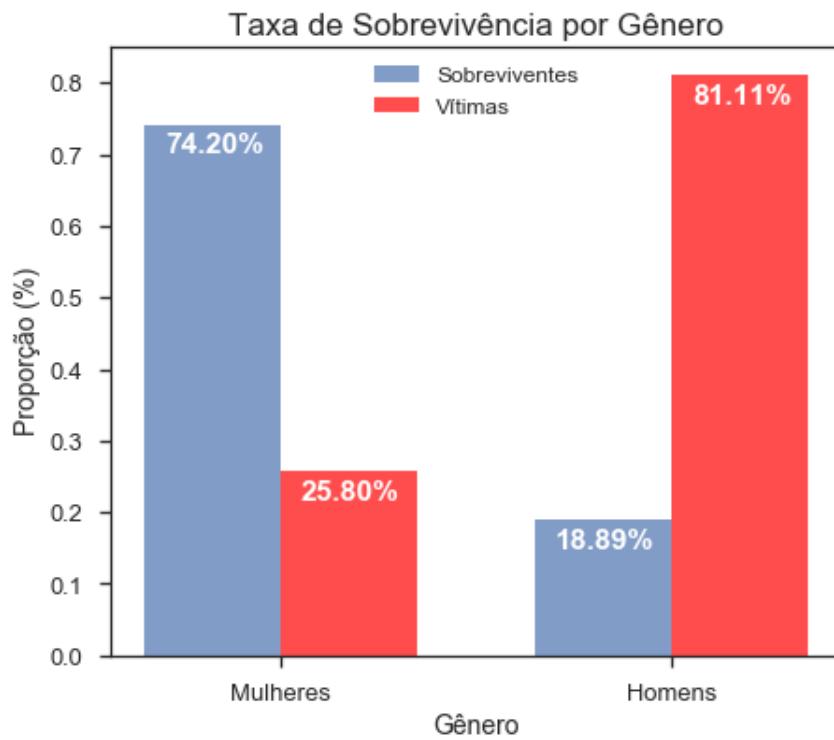


Figure 3.3: Probabilidade de Sobrevida por Gênero

O gráfico acima indica que, da totalidade de instâncias do gênero feminino, 74,20% mulheres sobreviveram e 25,80% foram vítimas. Analogamente, considerando a quantidade total de homens, apenas 18,89% sobreviveram e 81,11% foram vítimas. Apesar de indicar que a probabilidade de uma mulher ter sobrevivido ao naufrágio foi maior, é importante ressaltar que a análise acima foi baseada na quantidade total por gênero, o que pode deturpar o julgamento, visto que a quantidade de mulheres presentes no navio era relativamente menor que a quantidade de homens.

Para sanar este impasse, é possível considerar, na análise, não apenas a quantidade total de instâncias por gênero, mas também a totalidade de sobreviventes e vítimas, ressaltando a porcentagem dos gêneros presentes em cada um dos grupos.

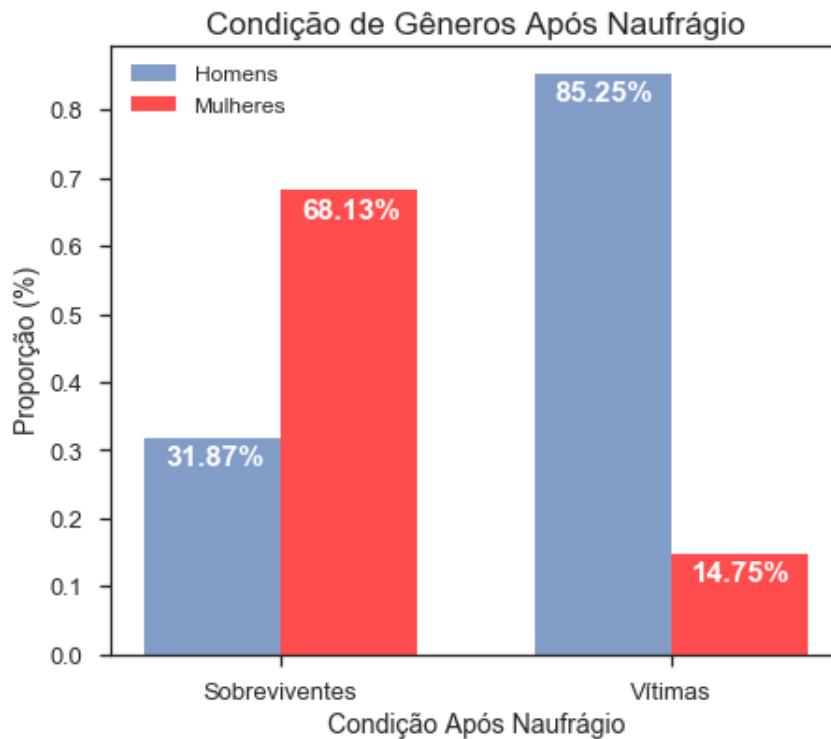


Figure 3.4: Proporção de Sobreviventes e Vítimas por Gênero

Assim, é possível dizer que, da quantidade total de sobreviventes, 68,13% eram mulheres e 31,87% eram homens. De modo equivalente, da totalidade de vítimas, 85,25% eram do gênero masculino e apenas 14,75% do gênero feminino. A probabilidade de uma mulher ter sobrevivido ao naufrágio foi bem maior que a de um homem. Alguns argumentos que possivelmente explicariam este resultado:

- Aplicação da regra "Mulheres e crianças primeiro" - prioridade no resgate;
- Maior quantidade de homens presentes;
- Peso máximo em botes salva-vidas pode ter favorecido o gênero feminino;

O código responsável por gerar o gráfico da figura 3.3 possui a sintaxe descrita pela figura abaixo:

```

# Totalização por gênero
total_gender = df.groupby('Sex').count()['Name']
# Taxas de sobrevida por gênero
gender_rate = df.groupby(['Sex', 'Survived']).count()['Name']
proportions = gender_rate / total_gender
prop_f = proportions['female']
prop_m = proportions['male']
# Proporção de sobreviventes e vítimas
survivors = [prop_f[1], prop_m[1]]
victims = [prop_f[0], prop_m[0]]

# Plotagem gráfica
ind = np.arange(len(survivors))
width = .35
sns.set_style('ticks')
sns.set_context('talk')
fig, ax = plt.subplots(figsize=(7, 6))
m_bar = plt.bar(ind, survivors, width, alpha=.7, label='Sobreviventes')
f_bar = plt.bar(ind+width, victims, width, color='r', alpha=.7, label='Vítimas')
plt.xlabel('Gênero', fontsize=14)
plt.ylabel('Proporção (%)', fontsize=14)
plt.title('Taxa de Sobrevida por Gênero', fontsize=17)
locations = ind + width / 2
labels = ['Mulheres', 'Homens']
plt.xticks(locations, labels, fontsize=13)
plt.legend(fontsize=12)
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.2%}'.format(height), (p.get_x() + .15 * width, p.get_y() + height - 0.04),
               color='w', weight='bold')
plt.show()

```

Figure 3.5: Código - Taxa de Sobrevida por Gênero

Adicionalmente, é interessante citar que as informações de sobrevida por gênero também poderiam constar no gráfico de pizza da figura 3.2, dinamizando ainda mais a visualização.

Para tal, foi criado o gráfico da figura 3.6 contendo informações conjuntas entre sobreviventes e vítimas (fatias externas) por gêneros (fatias internas). A porcentagem das fatias internas representam suas respectivas proporções de acordo com grupo macro da análise, ou seja, a contagem real de sobreviventes e vítimas.

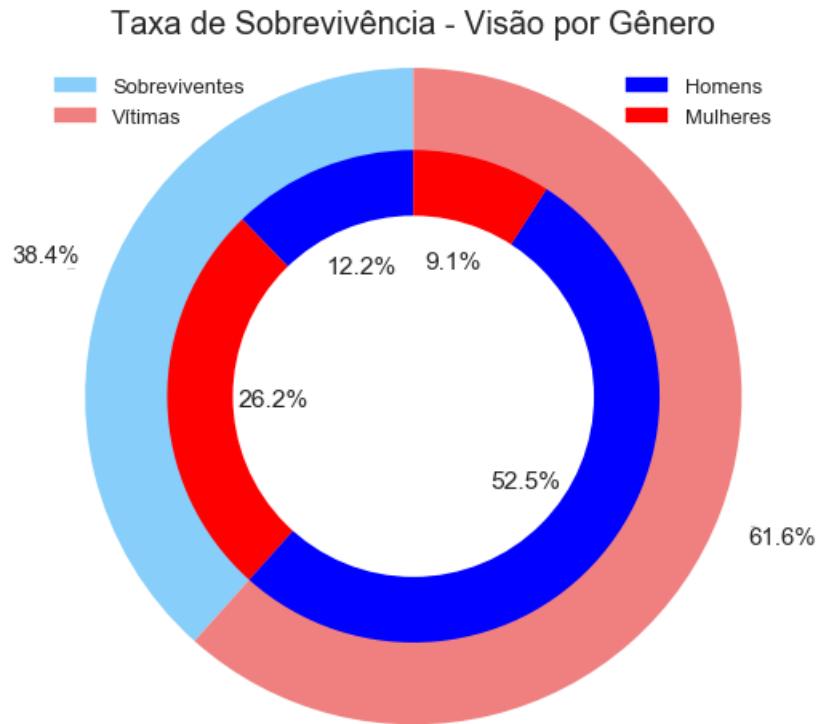


Figure 3.6: Visão Geral de Sobrevidentes por Gênero

Desse modo, é possível perceber que a fatia interna vermelha, referente ao gênero feminino, representa a maior parcela da fatia externa de sobrevidentes. Em outras palavras, é válido dizer que 26,2% de todos os passageiros e tripulantes podem ser classificados como sobrevidentes do gênero feminino. Analogamente, é possível concluir, através da análise da fatia referente às vitimas, que 52,5% do total de passageiros encontram-se no grupo de vítimas do gênero masculino.

Para facilitar a análise dos atributos relacionados a probabilidade de sobrevivência ao naufrágio do navio Titanic, a tabela abaixo será criada para ser alimentada conforme a formalização de conclusões. A presente sessão analisou a probabilidade de sobrevivência de acordo com o gênero e, como mencionado acima, o gênero feminino se mostrou predominante neste critério.

Table 3.2: Respostas das Investigações

Atributo	Maior Prob. Sobrevidência
Gênero	Feminino

3.3 Taxa de Sobrevida por Classe Econômica

Um outro fator que, segundo análise prévia, pode apresentar considerável influência nos índices de sobrevida, é a Classe Econômica *Pclass* de cada um dos passageiros e tripulantes.

Sabe-se que o atributo *Pclass* apresenta três possíveis instâncias:

- '1' - Passageiros e tripulantes da Classe Alta
- '2' - Passageiros e tripulantes da Classe Média
- '3' - Passageiros e tripulantes da Classe Baixa

A apuração no conjunto de dados permitiu visualizar os números expostos na tabela e gráfico abaixo:

Classe	Total Passageiros	Taxa de Sobrevida
Alta	216	63%
Média	314	47%
Baixa	491	24%

Table 3.3: Taxa de Sobrevida por Classe Econômica

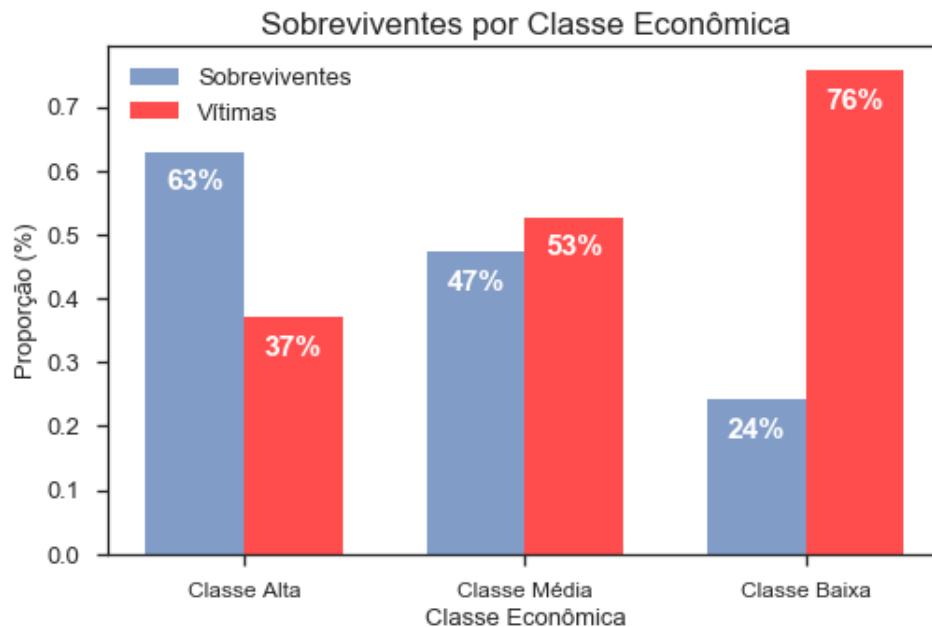


Figure 3.7: Proporção de Sobreviventes por Classe Econômica

Interpretando as informações obtidas acima, chega-se a conclusão de que passageiros de classe econômica alta possuíam maiores chances de sobrevida, seguidos da classe média e baixa que, por sua vez, apresentou os maiores índices de vítimas do naufrágio, totalizando 76%.

Alguns possíveis fatores que explicariam tal resultado:

- Passageiros de Classe Alta possivelmente tiveram prioridade para evacuação nos botes salva-vidas;
- Passageiros de Classe Alta possivelmente tiveram acesso facilitado durante a evacuação;
- Mesmo após a evacuação, passageiros de Classe Alta possivelmente tiveram melhores condições de proteção ao frio em alto mar;
- Por possivelmente possuírem melhores condições financeiras, os passageiros de Classe Alta poderiam ter habilidades de nado que se fizeram úteis após o naufrágio;

O código para plotagem do gráfico acima segue conforme a figura abaixo:

```

surv_counts = df.groupby(['Survived', 'Pclass']).count()['Name']
class_total = df.groupby('Pclass').count()['Name']
s_prop = surv_counts[1] / class_total
v_prop = surv_counts[0] / class_total

ind = np.arange(len(s_prop))
width = .35
sns.set_style('ticks')
sns.set_context('talk')
fig, ax = plt.subplots(figsize=(8, 5))
s_bar = plt.bar(ind, s_prop, width, alpha=.7, label='Sobreviventes')
v_bar = plt.bar(ind+width, v_prop, width, color='r', alpha=.7, label='Vítimas')
plt.ylabel('Proporção (%)', fontsize=13)
plt.xlabel('Classe Econômica', fontsize=13)
plt.title('Sobreviventes por Classe Econômica', fontsize=17)
locations = ind + width / 2
labels = ['Classe Alta', 'Classe Média', 'Classe Baixa']
plt.xticks(locations, labels, fontsize=12)
plt.legend()
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0%}'.format(height), (p.get_x() + .20*width, p.get_y() + height - 0.06),
                color='w', weight='bold')
plt.show()

```

Figure 3.8: Código - Proporção de Sobreviventes por Classe Econômica

Assim como avaliado na análise de sobreviventes por gênero, a quantidade total em cada uma das classes pode interferir no julgamento sobre suas respectivas probabilidades de sobrevivência. Para tal, o gráfico abaixo foca apenas a visão de sobreviventes separados por classe econômica.

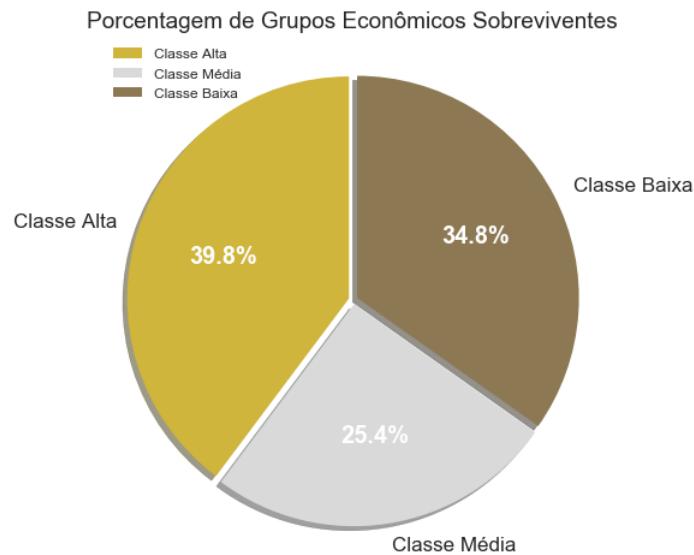


Figure 3.9: Grupo de Sobrevidentes Dividido por Classe Econômica

Antes de concluir alguns fatos referentes aos resultados acima mostrados, é importante discorrer sobre a possibilidade de englobar as informações de Classe Econômica e Gênero em um único gráfico, da mesma forma como evidenciado pela figura 3.6.

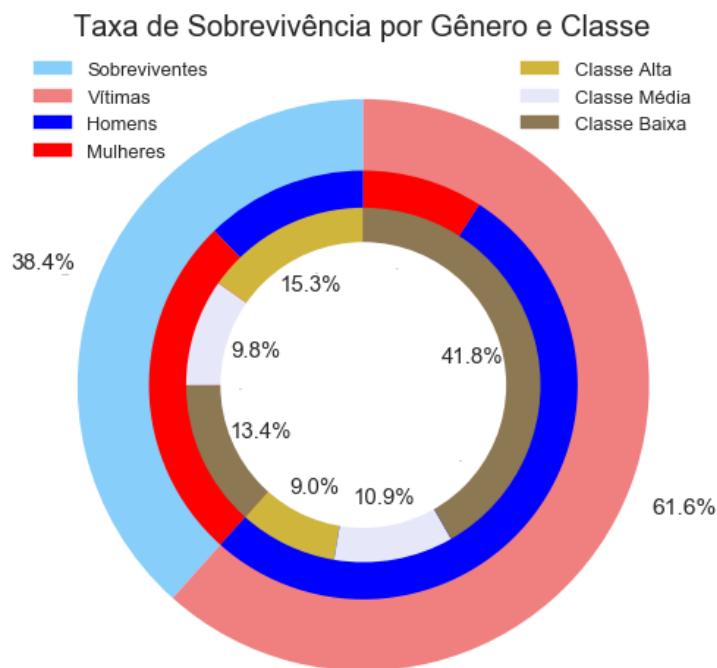


Figure 3.10: Sobrevidentes por Gênero e Classe Econômica

A partir da figura 3.11, ao avaliar a fatia externa referente às condições após o naufrágio e, adentrando alguns níveis, é possível verificar as proporções de cada grupo econômico nas cores Ouro, Prata e Bronze, representando, respectivamente, as classes Alta, Média e Baixa. Visualizando a parcela externa que representa as vítimas, conclui-se que a maior porcentagem está representada pela cor Bronze, ou seja, pela classe Baixa.

Analizando a parcela de sobreviventes, o gráfico 3.9 indica que 39,8% pertencem à classe Alta, 34,8% à classe Média e, por fim, 25,4% à classe Baixa. Apesar da similaridade entre os números, é imprescindível considerar a diferença entre a quantidade de passageiros em cada uma das classes pois, sabendo da predominância de passageiros da classe Baixa com relação às classes Média e Alta, sendo esta última com o menor número de passageiros presentes, é factível concluir que a probabilidade de um passageiro da classe Alta sobreviver era relativamente maior do que um passageiro da classe Baixa.

O gráfico da figura 3.9 foi gerado através do código ilustrado pela figura abaixo:

```
# Distribuição das Classes Econômicas no grupo de Sobreviventes

surv_class = df.groupby(['Survived', 'Pclass']).count()['Name'][1]
surv_class

labels = ['Classe Alta', 'Classe Média', 'Classe Baixa']
colors = ['#CFB53B', '#d9d9d9', '#8C7853']
explode = (0.02, 0.02, 0.02)
fig, ax = plt.subplots(figsize=(7, 7))
wedges, texts, autotexts = ax.pie(surv_class, labels=labels,
                                   startangle=90, shadow=True, explode=explode,
                                   autopct='%.1f%%', colors=colors)
ax.set_title('Procentagem de Grupos Econômicos Sobreviventes', fontsize=19)
ax.axis('equal')
plt.legend(fontsize=12, loc='upper left')
plt.tight_layout()
plt.setp(autotexts, size=20, weight='bold', color='w')
plt.setp(texts, size=17)
plt.show()
```

Figure 3.11: Código - Sobreviventes por Classe Econômica

Continuando com a filosofia adotada na sessão anterior, abaixo encontra-se a tabela atualizada com os resultados obtidos, contemplando agora as informações sobre a influência da Classe Econômica.

Table 3.4: Respostas das Investigações

Atributo	Maior Prob. Sobrevidêcia
Gênero	Feminino
Classe	Alta

3.4 Taxa de Sobrevida por Faixa Etária

Uma outra avaliação que poderia ser feita no conjunto de dados do Titanic, diz respeito a coluna *Age*. A ideia por trás da análise é verificar se houve influência da faixa etária na chance de sobrevida dos passageiros.

Entretanto, como o atributo *Age* do Dataset oferece dados numéricos de idades dos passageiros, será necessário realizar um agrupamento dos dados de acordo com o especificado abaixo:

- Crianças - Passageiros com idade menor que 21 anos;
- Adultos - Passageiros com idade entre 21 e 50 anos;
- Idosos - Passageiros com 50 anos de idade ou mais.

Algumas observações no Dataset permitiram a criação da tabela abaixo que evidencia dados estatísticos e úteis sobre o atributo *Age*.

Atributo	Idade
Contagem	891
Média	29,70
Desv.P	13,00
Mínimo	00,42
25%	22,00
50%	29,70
75%	35,00
Máximo	80,00

Table 3.5: Dados Estatísticos do Atributo Idade

Através de um histograma e, utilizando as ferramentas do **seaborn**, a visão sobre a distribuição de idades no Dataset é apresentada em dois formatos diferentes: o primeiro considerando um gráfico simples de distribuição e, o segundo, mostrando uma distribuição não paramétrica trazida pela função *kdeplot*. Tais gráficos poderão ser vistos logo a seguir na figura 3.13.

A importância das distribuições, paramétricas e não paramétricas, neste contexto, é atribuída à definição dos valores a serem estabelecidos na separação dos grupos etários, uma vez que, com os gráficos acima, é possível identificar as diferentes concentrações de idades e como estas poderiam influenciar nas análises posteriores.

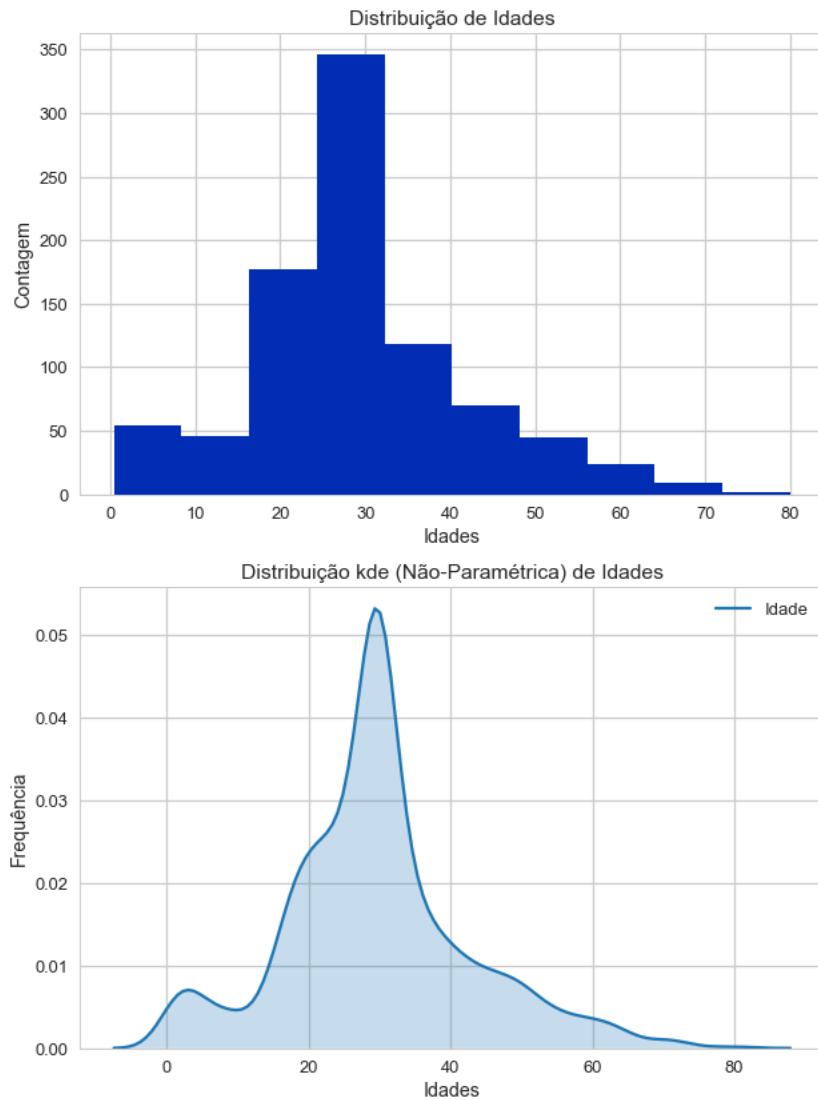


Figure 3.12: Diferentes Distribuições de Idade

A análise que se faz após a interpretação dos gráficos acima, permite identificar uma maior concentração de passageiros e tripulantes com idades entre 25 e 30 anos. Um fato, talvez curioso, diz respeito a quantidade relativamente considerável de passageiros com menos de 20 anos de idade, dado o contexto histórico e o gigantesco evento do lançamento do Titanic ao mar.

Para a criação dos gráficos da figura 3.13, foram utilizadas as seguintes linhas de código:

```
# Distribuição de Idades
fig, ax = plt.subplots(1, 2, figsize=(10, 14))
sns.set_context('talk')
sns.set_style('whitegrid')
# Primeiro plot = histograma
plt.subplot(211)
sns.distplot(df['Age'], kde=False, bins=10,
             hist_kws={"histtype": "step", "linewidth": 3,
                        "alpha": 1, "color": "#002db3"})
plt.title('Distribuição de Idades')
plt.xlabel('Idades')
plt.ylabel('Contagem')
# Segundo plot = kde
plt.subplot(212)
sns.kdeplot(df['Age'], shade=True, label='Idade')
plt.title('Distribuição kde (Não-Paramétrica) de Idades')
plt.xlabel('Idades')
plt.ylabel('Frequência')

plt.show()
```

Figure 3.13: Código para Plotagem de Distribuições de Idade

A separação de idades em grupos se deu através das linhas de código mostradas pela figura abaixo, onde foi necessário criar uma nova coluna de nome *AgeRange* para receber os novos itens gerados:

```
# Definindo ranges de idade
bin_edges = [df['Age'].describe()['min'], 21,
             55, df['Age'].describe()['max']]

# Definindo Labels
bin_names = ['Criança', 'Adulto', 'Idoso']

# Criando nova coluna com registros classificados por idade
df['AgeRange'] = pd.cut(df['Age'], bin_edges, labels=bin_names)
```

Figure 3.14: Separando registros por grupos de faixa etária

Para verificar as alterações, é possível imprimir as primeiras linhas do Dataset:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	AgeRange
1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	S	Adulto
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	female	38.000000	1	0	PC 17599	71.2833	C	Adulto
3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2 3101282	7.9250	S	Adulto
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	S	Adulto
5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	S	Adulto
6	0	3	Moran, Mr. James	male	29.699118	0	0	330877	8.4583	Q	Adulto
7	0	1	McCarthy, Mr. Timothy J	male	54.000000	0	0	17463	51.8625	S	Adulto
8	0	3	Palsson, Master. Gosta Leonard	male	2.000000	3	1	349909	21.0750	S	Criança
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000	0	2	347742	11.1333	S	Adulto
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000	1	0	237736	30.0708	C	Criança
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.000000	1	1	PP 9549	16.7000	S	Criança
12	1	1	Bonnell, Miss. Elizabeth	female	58.000000	0	0	113783	26.5500	S	Idoso
13	0	3	Saundercock, Mr. William Henry	male	20.000000	0	0	A/5. 2151	8.0500	S	Criança
14	0	3	Andersson, Mr. Anders Johan	male	39.000000	1	5	347082	31.2750	S	Adulto
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.000000	0	0	350406	7.8542	S	Criança

Figure 3.15: Verificando novo atributo do Dataset

A ideia é comparar os valores contidos no atributo *Age* com a classificação obtida no atributo *AgeRange* e, como o Dataset possui muitas colunas, a aplicação de um filtro para imprimir apenas alguns atributos é extremamente válida:

```
# Filtrando para melhor visualização
df.loc[:, ['Name', 'Survived', 'Age', 'AgeRange']].head(15)
```

PassengerId	Name	Survived	Age	AgeRange
1	Braund, Mr. Owen Harris	0	22.000000	Adulto
2	Cumings, Mrs. John Bradley (Florence Briggs Th... Th...	1	38.000000	Adulto
3	Heikkinen, Miss. Laina	1	26.000000	Adulto
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.000000	Adulto
5	Allen, Mr. William Henry	0	35.000000	Adulto
6	Moran, Mr. James	0	29.699118	Adulto
7	McCarthy, Mr. Timothy J	0	54.000000	Adulto
8	Palsson, Master. Gosta Leonard	0	2.000000	Criança
9	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	1	27.000000	Adulto
10	Nasser, Mrs. Nicholas (Adele Achem)	1	14.000000	Criança
11	Sandstrom, Miss. Marguerite Rut	1	4.000000	Criança
12	Bonnell, Miss. Elizabeth	1	58.000000	Idoso
13	Saundercock, Mr. William Henry	0	20.000000	Criança
14	Andersson, Mr. Anders Johan	0	39.000000	Adulto
15	Vestrom, Miss. Hulda Amanda Adolfina	0	14.000000	Criança

Figure 3.16: Filtrando colunas para melhor visualização do novo atributo

Apesar da classificação ter aparentemente surtido efeito, foi encontrada uma inconsistência no registro de id número 804 que, por sua vez, representa o passageiro com a menor idade entre todos.

# Verificando inconsistência			
df_age_null = df[df['AgeRange'].isnull()]			
df_age_null.loc[:, ['Name', 'Survived', 'Age', 'AgeRange']]			
PassengerId			
804 Thomas, Master. Assad Alexander 1 0.42 NaN			

Figure 3.17: Valor NaN em AgeRange

Provavelmente, esta inconsistência foi gerada na aplicação da função `cut()` do Pandas, mais especificamente em algum de seus parâmetros contidos na variável `bin_edges`. Para saná-la, basta aplicar pontualmente a classificação adequada.

# Classificando manualmente e verificando alterações			
df['AgeRange'][804] = 'Criança'			
print(f'Há dados nulos na coluna "AgeRange"? {df.isnull().values.any()}')			
print(f'Qual a classificação do passageiro 804? {df["AgeRange"][804]}')			
Há dados nulos na coluna "AgeRange"? False			
Qual a classificação do passageiro 804? Criança			

Figure 3.18: Classificando Registro Manualmente

Após esta última verificação, é possível afirmar que a função `cut()` do Pandas surtiu o efeito esperado, separando os registros em grupos de acordo com a idade. Analisando a figura 3.16, nota-se que os passageiros de id 7, 12 e 13 são exemplos bem sucedidos da classificação aplicada, visto que encontram-se em situações limiares que poderiam ocasionar erros em caso de incoerência no código utilizado.

Antes de realizar plotagens gráficas, é importante quantizar as instâncias obtidas em cada um dos grupos.

Grupo Etário	Total Passageiros	Sobreviventes
Crianças	204	43%
Adultos	647	38%
Idosos	40	30%

Table 3.6: Dados sobre Sobrevida de cada Grupo Etário

Com isso, é possível visualizar graficamente os efeitos da faixa etária na sobrevida dos indivíduos.

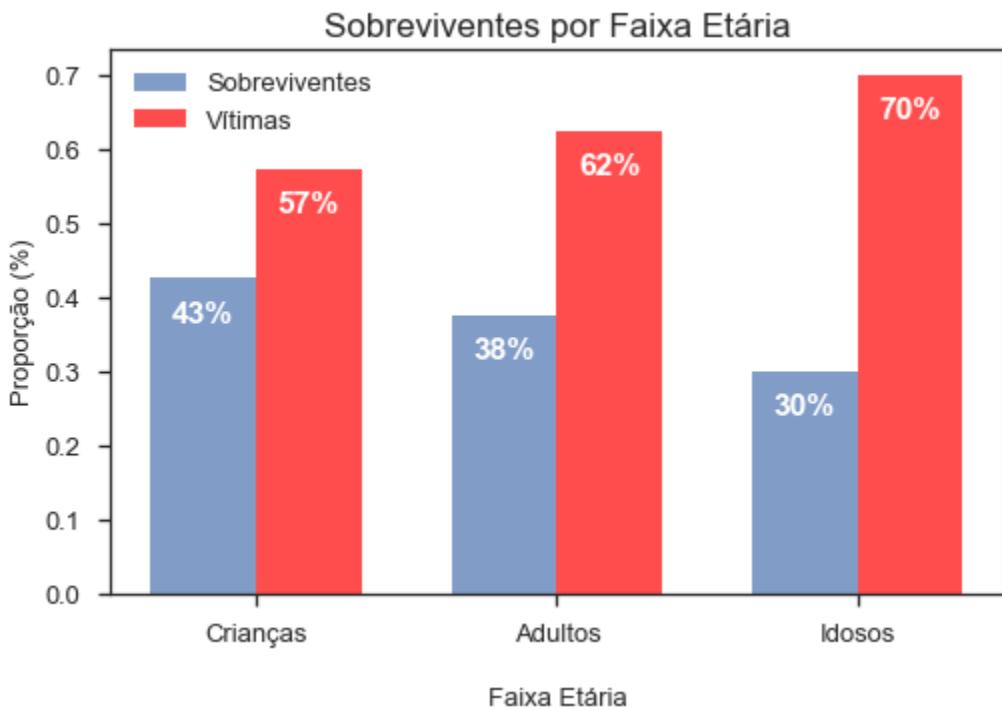


Figure 3.19: Taxa de Sobrevivência por Faixa Etária

O gráfico 3.19 indica que o grupo etário classificado como "Crianças" obteve maiores chances de sobrevivência. Dentre alguns pontos que podem ser destacados com o objetivo de entender tais resultados, encontram-se:

- Prioridade de resgate foi dada às crianças;
- Idosos poderiam ter encontrado dificuldades de sobrevivência dados os imensos obstáculos presentes;
- A baixa resistência física dos Idosos pode ter contribuído com a grande porcentagem de vítimas;

Uma outra visão que pode ser apresentada leva em consideração a porcentagem de cada grupo etário dentro da totalidade de sobreviventes.

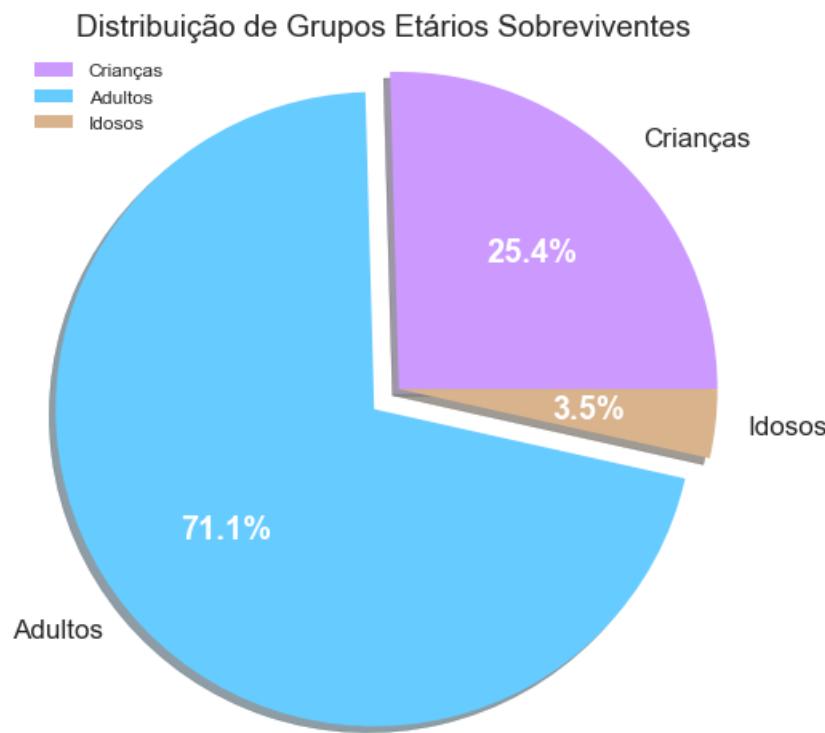


Figure 3.20: Sobrevidentes por Grupo Etário

Apesar do gráfico acima indicar uma maior parcela de adultos no grupo de sobrevidentes, é importante ressaltar a grande diferença de instâncias presentes neste grupo com relação aos demais. No total, foram contabilizados 647 adultos contra 204 crianças e apenas 40 idosos. Este número contribui diretamente para que, entre os sobrevidentes, 71,1% ossem adultos e, por outro lado, apenas 38% dos adultos terem sobrevidido.

Os códigos para plotagens dos gráficos 3.19 e 3.20 são semelhantes aos códigos já apresentados em plotagens relacionadas.

Como de costume, abaixo estão as informações atualizadas sobre a probabilidade de sobrevida de acordo com as conclusões tiradas até o momento:

Table 3.7: Respostas das Investigações

Atributo	Maior Prob. Sobrevida
Gênero	Feminino
Classe	Alta
Faixa Etária	Crianças

3.5 Taxa de Sobrevida por Porto de Embarque

Um fator que possivelmente influenciou na probabilidade de sobrevida dos passageiros e tripulantes pode ser traduzido pelo atributo *Embarked* que, por sua vez, diz respeito ao porto de embarque dos passageiros.

Como averiguado durante as sessões de entendimento dos dados, os valores contidos na coluna *Embarked* possuem os seguintes significados:

- 'Q' - Porto de Queenstown / Nova Zelândia
- 'S' - Porto de Southampton / Inglaterra
- 'C' - Porto de Cherbourg / França

A suspeita é de que haja, entre os passageiros que embarcaram em cada um dos portos acima descritos, significantes diferenças em atributos chave, como Classe Econômica, Idade, Ticket Médio, entre outros.

De acordo com investigações prévias realizadas, a tabela abaixo descreve alguns números importantes a respeito dessas diferenças.

Porto / Atributos	Queenstown	Southampton	Cherbourg
Total Passageiros	77	644	170
Idade Média	29.11 anos	29.48 anos	30.79 anos
Ticket Médio	\$13.27	\$27.08	\$60.19
Classe Alta (%)	2.6%	19.7%	51.2%
Classe Média (%)	3.9%	25.5%	10.0%
Classe Baixa (%)	93.6%	54.8%	38.8%
Homens (%)	53.24%	68.50%	55.88%
Mulheres (%)	46.76%	31.50%	44.12%
Crianças (%)	15.58%	24.70%	19.41%
Adultos (%)	80.52%	71.43%	73.53%
Idosos (%)	3.90%	3.97%	7.06%
Prob. Sobrevida (%)	38.96%	33.90%	55.35%

Table 3.8: Dados sobre passageiros de acordo com o porto de embarque

Avaliando os dados contidos na tabela 3.8, é possível perceber, logo de cara, uma maior probabilidade de sobrevida entre os passageiros que embarcaram em Cherbourg. Na ciência de tal informação, nota-se também que há diferenciações importantes em outros atributos, como por exemplo, a grande concentração de passageiros de Classe Alta que embarcaram em Cherbourg. Ademais, a menor porcentagem de passageiros de Classe Baixa embarcaram neste mesmo porto.

Como esperado as conclusões reteiradas da análise acima indicaram que o Ticket Médio pago por passageiros que embarcaram em Cherbourg é consideravelmente maior, o que confirma o fato de passageiros de Classe Alta terem pago um valor maior no Ticket.

A função que retorna todos os parâmetros estatísticos de cada um dos portos de embarque pode ser visualizada através da figura abaixo:

```

def stats_bay(bay):
    bay = bay.strip().upper()

    contagem = df.groupby('Embarked').count()['Name']
    medias = df.groupby('Embarked').mean()
    p_class_emb = df.groupby(['Embarked', 'Pclass']).count()['Name']
    prop_class = 100 * p_class_emb[bay] / contagem[bay]
    gender_emb = df.groupby(['Embarked', 'Sex']).count()['Name']
    prop_gender = 100 * gender_emb[bay] / contagem[bay]
    agerange_emb = df.groupby(['Embarked', 'AgeRange']).count()['Name']
    prop_agerange = 100 * agerange_emb[bay] / contagem[bay]
    surv_emb = df.groupby(['Embarked', 'Survived']).count()['Name']
    prop_surv = 100 * surv_emb[bay] / contagem[bay]

    if bay == 'C':
        bay_name = 'Cherbourg'
    elif bay == 'S':
        bay_name = 'Southampton'
    else:
        bay_name = 'Queenstown'

    print(f'--- DADOS DE {bay_name.upper()} ---')
    print(f'Total de passageiros com embarque em {bay_name}: {contagem[bay]} pessoas.')
    print(f'Idade média dos passageiros de {bay_name}: {medias["Age"][bay]:.2f} anos.')
    print(f'Ticket médio dos passageiros de {bay_name}: ${medias["Fare"][bay]:.2f}')
    print('--- CLASSE ECONÔMICA ---')
    print(f'Porcentagem de Classe Alta que embarcaram em {bay_name}: {prop_class[1]:.2f}%')
    print(f'Porcentagem de Classe Média que embarcaram em {bay_name}: {prop_class[2]:.2f}%')
    print(f'Porcentagem de Classe Baixa que embarcaram em {bay_name}: {prop_class[3]:.2f}%')
    print('--- GÊNERO ---')
    print(f'Porcentagem de Homens que embarcaram em {bay_name}: {prop_gender["male"]:.2f}%')
    print(f'Porcentagem de Mulheres que embarcaram em {bay_name}: {prop_gender["female"]:.2f}%')
    print('--- FAIXA ETÁRIA ---')
    print(f'Porcentagem de Crianças que embarcaram em {bay_name}: {prop_agerange["Criança"]:.2f}%')
    print(f'Porcentagem de Adultos que embarcaram em {bay_name}: {prop_agerange["Adulto"]:.2f}%')
    print(f'Porcentagem de Idosos que embarcaram em {bay_name}: {prop_agerange["Idoso"]:.2f}%')
    print('--- PROBABILIDADE DE SOBREVIVÊNCIA ---')
    print(f'Probabilidade de sobrevida para embarcantes em {bay_name}: {prop_surv[1]:.2f}%')

```

Figure 3.21: Código para gerar tabela com dados relacionados aos portos

Apesar de conter uma massiva análise estatística, a tabela 3.8 possui dados em excesso, o que pode dificultar a visualização e o entendimento por parte dos interessados.

De todos os itens listados sobre os passageiros, apenas alguns atributos podem ser elegíveis como influenciadores diretos na probabilidade de sobrevida. Para analisar os resultados de maneira mais dinâmica, faz-se necessária a criação de plotagens gráficas como visto nas figuras 3.22 e 3.23. Assim, é possível visualizar com clareza a proporção de sobreviventes por porto de embarque.

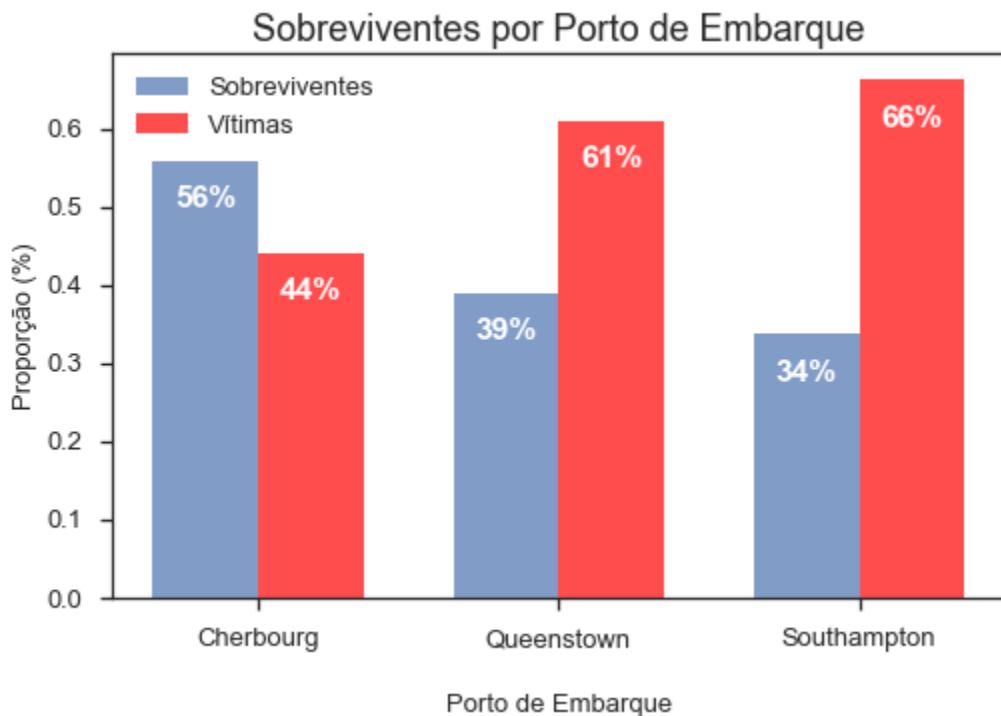


Figure 3.22: Proporção de Sobreviventes por Porto de Embarque

Confirmado os resultados da análise bruta, a proporção entre sobreviventes e vítimas foi mais favorável aos passageiros que embarcaram em Cherbourg, França.

Alguns pontos que explicariam os resultados obtidos:

- Maior presença de passageiros de Classe Alta em Cherbourg;
- Passageiros que embarcaram em Cherbourg pagaram um valor alto no Ticket (indicativo de Classe Econômica Alta);
- Passageiros de Classe Alta provavelmente teriam condições privilegiadas no navio, como localização dos quartos, acesso às rotas de emergência ou à equipamentos de segurança, entre outros;
- Pouca porcentagem de Jovens e Crianças que embarcaram em Queenstown pode ter contribuído para a possível baixa prioridade de resgate;
- Grande presença de passageiros de Classe Baixa em Queenstown influenciaram diretamente na proporção de vítimas.

Para corroborar os tópicos levantados, é importante visualizar um segundo gráfico, sendo este composto pela divisão das Classes Econômicas em cada um dos portos de embarque.

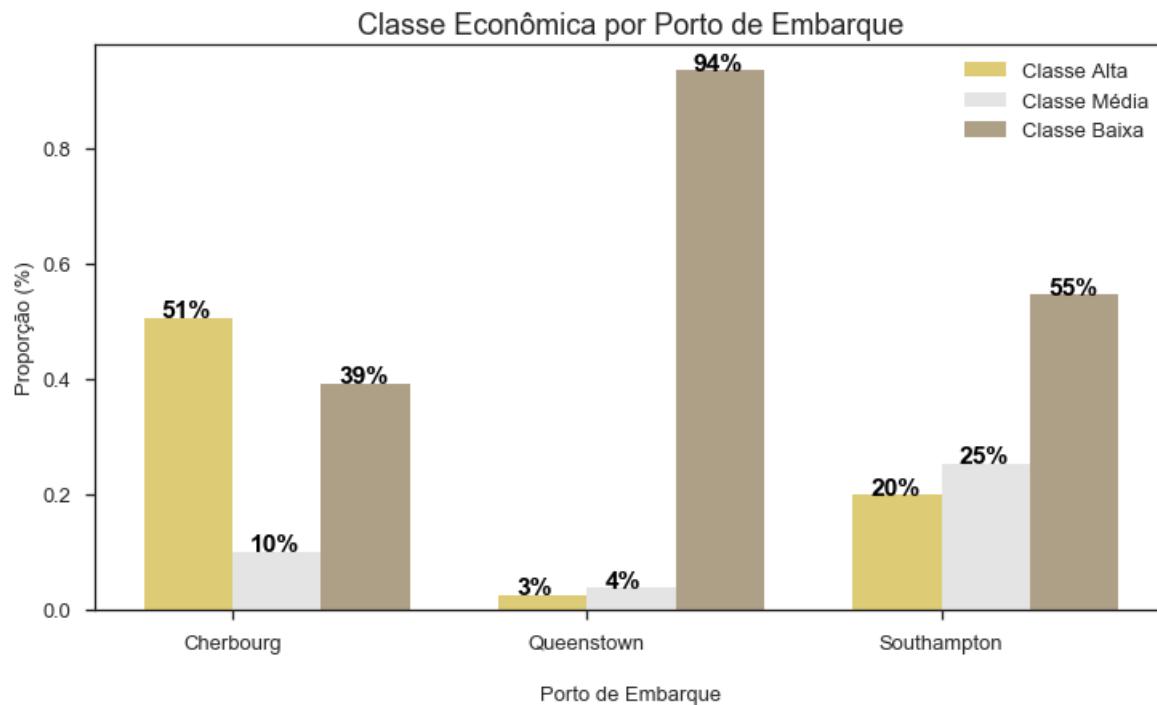


Figure 3.23: Proporção de Classe Econômica por Porto de Embarque

De fato, a divisão de Classes Econômicas por Porto de Embarque pode ser vista de forma mais clara com o gráfico 3.23. Em Cherbourg houve uma grande concentração de passageiros de Classe Alta. Por outro lado, em Queenstown, a presença de passageiros de Classe Baixa foi quase total.

Assim, finalizando a análise do Dataset Titanic através da aplicação de conceitos de estatística descritiva para definir os atributos que influenciaram na chance de sobrevida de passageiros e tripulantes, é apresentada a tabela abaixo com um resumo sobre as conclusões obtidas.

Table 3.9: Respostas das Investigações

Atributo	Maior Prob. Sobrevida
Gênero	Feminino
Classe	Alta
Faixa Etária	Crianças
Porto de Embarque	Cherbourg



4. Conclusão

A análise de dados permite retirar informações e elevar o nível de conclusões a um patamar antes inimaginável. O naufrágio do Titanic se mostrou um excelente instrumento de análise para evidenciar, na prática, uma pequena parcela do poder oferecido pelo ferramental da ciência de dados.

Através da união entre conceitos estatísticos, probabilísticos, computacionais e de entendimento do caso estudado, foi possível concluir que as maiores chances de sobrevivência estavam em nichos específicos de passageiros que contemplavam, por exemplo, o gênero feminino, a classe econômica alta, as menores faixas etárias e até mesmo o porto de embarque.

Em cada um dos respectivos tópicos, foi possível discorrer sobre os principais motivos que poderiam servir como justificativas para os resultados obtidos. O levantamento de hipóteses, como a possível dificuldade encontrada por idosos ou a possível prioridade à passageiros de classes econômicas mais altas, por exemplo, somente foi possível com uma investigação direcionada na base de dados oferecida, aplicando conceitos de limpeza, preparação, transformação e análise.

As plotagens gráficas e tabelas presentes nesse relatório foram desenvolvidas visando a simplicidade e coerência dos fatos apresentados, sempre procurando transmitir o impacto adequado causado pelo tratamento puro dos dados realizado nos bastidores do código.

Por fim, e não menos importante, as conclusões obtidas com relação as chances de sobrevivência em cada um dos tópicos estudados ocasionaram não somente um maior entendimento sobre as ferramentas de análise de dados, mas contribuíram também para um enriquecimento histórico-pessoal, uma vez que os insights obtidos com a presente análise poderão ser comunicados à qualquer pessoa que já tenha ouvido falar sobre o navio Titanic.

4.1 Referências



Documentação Oficial:

- Pandas: <https://pandas.pydata.org/>
- NumPy: <http://www.numpy.org/>
- Matplotlib: <https://matplotlib.org/>
- Seaborn: <https://seaborn.pydata.org/>

Dados Históricos:

- Naufrágio Titanic: https://pt.wikipedia.org/wiki/RMS_Titanic
- Conjunto de dados: <https://www.kaggle.com/c/titanic>

Suporte Técnico:

- Stack Overflow: <https://stackoverflow.com/>
- Udacity: <https://br.udacity.com/>
- KDnuggets: <https://www.kdnuggets.com/2017/06/7-steps-mastering-data-preparation-python.html>
- Slack: <https://slack.com/>