

Nome: Marcus Vinicius Bueno Nunes

Contato: (16)99278-0444

Desafio: "Ocorrências Aeronáuticas na Aviação Civil Brasileira".

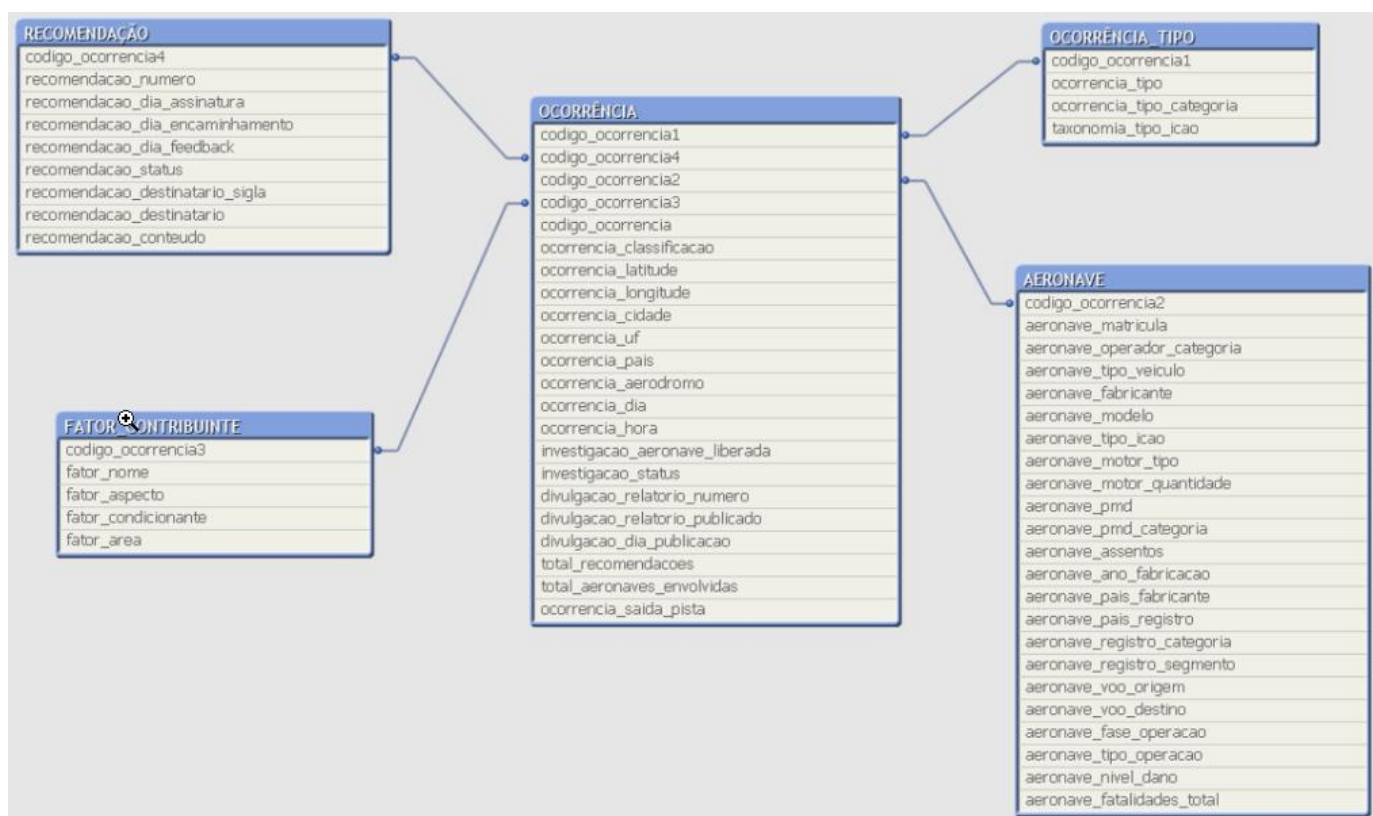
O desafio consiste na exploração e análise de dados referentes a ocorrências na aviação civil brasileira, a fim de serem extraídos *insights* e suposições acerca dos dados.

Os processos utilizados devem ser explicados, com o intuito de avaliar o raciocínio do participante para que possa ser compreendido e avaliado.

O desafio é composto por 5 bases de dados diferentes, quais sejam:

1. OCORRÊNCIA.csv
2. OCORRÊNCIA_TIPO.csv
3. AERONAVE.csv
4. FATOR_CONTRIBUINTE.csv
5. RECOMENDAÇÃO.csv

As bases se relacionam através da base de dados OCORRÊNCIA.csv, conforme o diagrama abaixo:



A análise foi desenvolvida em cima dos arquivos: OCORRENCIA.csv, OCORRENCIA_TIPO.csv, AERONAVE.csv.

O desafio foi feito utilizando a *IDE Jupyter notebook* e a linguagem de programação *Python*.

Importação das bibliotecas necessárias para a análise de dados

```
import pandas as pd
```

```
import numpy as np
```

```
import datetime
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
from matplotlib.pyplot import xticks
```

Acima foi feito a importação das bibliotecas necessárias no projeto:

pandas: utilizado para a criação e manipulação de *dataframes*.

numpy: utilizado para a manipulação de arrays e matrizes multidimensionais.

datetime: utilizado para a manipulação e tratamento de datas e horários

matplotlib e seaborn: utilizado para a plotagem de gráficos.

#CONFIGURAÇÕES DOS GRÁFICOS

```
plt.rcParams['figure.figsize'] = (20.0, 08.0)
```

```
plt.rcParams['font.family'] = "serif"
```

```
sns.color_palette("Blues", as_cmap=True)
```

```
sns.set(style="white")
```

Nesta seção de configuração dos gráficos, defini apenas algumas configurações de tamanho, fonte e cor.

Carregamento da base de dados

Foi feito o carregamento dos dados contidos nos arquivos csv para dentro de *dataframes* da biblioteca pandas:

```
base_aeronave = pd.read_csv('aeronave_2010_2020.csv', delimiter=';')
```

```
base_ocorrenciia = pd.read_csv('ocorrenciia_2010_2020.csv', delimiter=';')
```

```
base_ocorrenciia_tipo = pd.read_csv('ocorrenciia_tipo_2010_2020.csv', delimiter=';')
```

No *dataframe* *base_aeronave* constam informações a respeito das aeronaves envolvidas nas ocorrências.

No *dataframe* *base_ocorrenciia* constam detalhes a respeito das ocorrências em si.

No *dataframe* *base_ocorrenciia_tipo* constam informações a respeito dos tipos das ocorrências.

Abaixo selecionei campos e considereei serem bons candidatos a conter em nesta análise.

Seleção dos campos pertinentes a análise - BASE AERONAVE

```
base_aeronave = base_aeronave[['codigo_ocorrenciia2', 'aeronave_tipo_veiculo',  
'aeronave_fabricante', 'aeronave_modelo',  
'aeronave_motor_tipo', 'aeronave_motor_quantidade', 'aeronave_ano_fabricacao',  
'aeronave_pais_fabricante', 'aeronave_assentos',  
'aeronave_registro_segmento', 'aeronave_voo_origem', 'aeronave_voo_destino',  
'aeronave_pmd_categoria', 'aeronave_fase_operacao', 'aeronave_tipo_operacao',  
'aeronave_nivel_dano', 'aeronave_fatalidades_total']]
```

Seleção dos campos pertinentes a análise - BASE OCORRENCIA

```
base_ocorrenciia = base_ocorrenciia[['codigo_ocorrenciia1', 'codigo_ocorrenciia2', 'codigo_ocorrenciia3',  
'codigo_ocorrenciia4', 'ocorrenciia_classificacao', 'ocorrenciia_uf',  
'ocorrenciia_dia', 'ocorrenciia_hora', 'total_aeronaves_envolvidas',  
'ocorrenciia_saida_pista']]
```

Seleção dos campos pertinentes a análise - BASE OCORRENCIA TIPO

```
base_ocorrenciia_tipo = base_ocorrenciia_tipo[['codigo_ocorrenciia1', 'ocorrenciia_tipo',  
'ocorrenciia_tipo_categoria', 'taxonomia_tipo_icao']]
```

Tratamento dos dados – BASE AERONAVE

Selecionando somente veículos iguais a: avião, helicóptero e ultraleve

```
base_aeronave = base_aeronave.query('aeronave_tipo_veiculo == "AVIÃO" or aeronave_tipo_veiculo == "HELICÓPTERO" or aeronave_tipo_veiculo == "ULTRALEVE")
```

Considerados somente os registros com valores iguais a 'AVIÃO', 'HELICÓPTERO' e 'ULTRALEVE', pois os outros registros somavam uma quantidade irrelevante para a nossa análise.

```
# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '***' - CAMPO: aeronave_motor_tipo
for index, row in base_aeronave.iterrows():
    if row['aeronave_motor_tipo'] == '***':
        base_aeronave.at[index, 'aeronave_motor_tipo'] = 'INDEFINIDO'
```

Em todos os campos onde os valores constam como '***' foi atribuído o valor 'INDEFINIDO' a eles.

```
# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '***' - CAMPO: aeronave_motor_quantidade
for index, row in base_aeronave.iterrows():
    if row['aeronave_motor_quantidade'] == '***':
        base_aeronave.at[index, 'aeronave_motor_quantidade'] = 'INDEFINIDO'
```

```
# Eliminando valores que sejam iguais a 0 - CAMPO: aeronave_ano_fabricacao
base_aeronave = base_aeronave.query('aeronave_ano_fabricacao > 0')
base_aeronave.aeronave_ano_fabricacao = base_aeronave['aeronave_ano_fabricacao'].astype(int)
```

```
# agrupando valores diferentes de Brasil como tipo 'OUTROS'
for index, row in base_aeronave.iterrows():
    if row['aeronave_pais_fabricante'] != 'BRASIL':
        base_aeronave.at[index, 'aeronave_pais_fabricante'] = 'OUTROS'
```

```
# Eliminando valores nulos: CAMPO: aeronave_assentos
base_aeronave = base_aeronave[base_aeronave['aeronave_assentos'].notnull()]
```

Os registros onde o campo 'aeronave_assentos' constavam como nulo, foram removidos. Por se tratar de um valor discreto preferi removê-lo, por receio de atribuir valores erroneamente, descaracterizando as análises. No caso de variáveis contínuas, poderia ter sido atribuído o valor da mediana, ou média dependendo do caso.

```
# Categorizando a capacidade de passageiros das aeronaves
for index, row in base_aeronave.iterrows():
    if row['aeronave_assentos'] < 20:
        base_aeronave.at[index, 'aeronave_capacidade_passageiros'] = 'BAIXA_CAPACIDADE'
    elif row['aeronave_assentos'] >= 20 and row['aeronave_assentos'] < 100:
        base_aeronave.at[index, 'aeronave_capacidade_passageiros'] = 'MEDIA_CAPACIDADE'
    elif row['aeronave_assentos'] >= 100:
        base_aeronave.at[index, 'aeronave_capacidade_passageiros'] = 'GRANDE_CAPACIDADE'
```

Acima foi feita uma categorização dos dados, a fim de facilitar a análise, pois, tínhamos muitos valores variados.

```
# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '***' - CAMPO: aeronave_registro_segmento
for index, row in base_aeronave.iterrows():
    if row['aeronave_registro_segmento'] == '***':
        base_aeronave.at[index, 'aeronave_registro_segmento'] = 'INDEFINIDO'
```

```
# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '***' - CAMPO: aeronave_voo_origem
```

```

for index, row in base_aeronave.iterrows():
    if row['aeronave_voo_origem'] == '****':
        base_aeronave.at[index, 'aeronave_voo_origem'] = 'INDEFINIDO'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: aeronave_voo_destino
for index, row in base_aeronave.iterrows():
    if row['aeronave_voo_destino'] == '****':
        base_aeronave.at[index, 'aeronave_voo_destino'] = 'INDEFINIDO'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: aeronave_fase_operacao
for index, row in base_aeronave.iterrows():
    if row['aeronave_fase_operacao'] == '****':
        base_aeronave.at[index, 'aeronave_fase_operacao'] = 'INDETERMINADA'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: aeronave_tipo_operacao
for index, row in base_aeronave.iterrows():
    if row['aeronave_tipo_operacao'] == '****':
        base_aeronave.at[index, 'aeronave_tipo_operacao'] = 'INDEFINIDO'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: aeronave_nivel_dano
for index, row in base_aeronave.iterrows():
    if row['aeronave_nivel_dano'] == '****':
        base_aeronave.at[index, 'aeronave_nivel_dano'] = 'INDEFINIDO'

```

Tratamento dos dados – BASE FATOR CONTRIBUINTE

```

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: fator_aspecto
for index, row in base_fator_contribuinte.iterrows():
    if row['fator_aspecto'] == '****':
        base_fator_contribuinte.at[index, 'fator_aspecto'] = 'INDEFINIDO'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: fator_condicionante
for index, row in base_fator_contribuinte.iterrows():
    if row['fator_condicionante'] == '****':
        base_fator_contribuinte.at[index, 'fator_condicionante'] = 'INDEFINIDO'

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: fator_area
for index, row in base_fator_contribuinte.iterrows():
    if row['fator_area'] == '****':
        base_fator_contribuinte.at[index, 'fator_area'] = 'INDEFINIDO'

```

Tratamento dos dados – BASE OCORRÊNCIA

```

# Atribuindo o valor 'INDEFINIDO' onde os valores são iguais a '****' - CAMPO: ocorrencia_uf
for index, row in base_ocorrencia.iterrows():
    if row['ocorrencia_uf'] == '****':
        base_ocorrencia.at[index, 'ocorrencia_uf'] = 'INDEFINIDO'

# Eliminando valores nulos CAMPO: ocorrencia_dia
base_ocorrencia = base_ocorrencia[base_ocorrencia['ocorrencia_dia'].notnull()]

```

Acima eliminei um único registro, onde o valor do campo 'ocorrencia_dia' constava como nulo.

```
# Extraindo o ano da data da ocorrência CAMPO: ocorrencia_dia
base_ocorrencia['ocorrencia_dia'] = pd.to_datetime(base_ocorrencia['ocorrencia_dia'],
format='%d/%m/%Y')
base_ocorrencia['ocorrencia_dia'] = base_ocorrencia.ocorrencia_dia.dt.year
```

Acima extrai somente o ano da data, para que possa ser feito um gráfico de linha, a fim de analisar se as ocorrências aéreas cresceram ou diminuíram ao longo dos anos.

Criando a base *acidentes_ocorridos_por_ano* a partir da quantidade de acidentes/incidentes ocorridos no ano e o ano

```
acidentes_ocorridos_por_ano = pd.DataFrame({'Quantidade de acidentes ocorridos':[552, 576, 647, 654,
567, 471, 403, 432, 444, 496, 510]},
index=[2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020])
```

Acima criei outro *dataframe*, contendo todos os anos existentes na base, e o total de ocorrências existentes em cada ano.

```
# Eliminando valores nulos CAMPO: ocorrencia_hora
base_ocorrencia = base_ocorrencia[base_ocorrencia['ocorrencia_hora'].notnull()]
```

Acima eliminei um único registro, onde o valor do campo 'ocorrencia_hora' constava como nulo.

```
# Extraindo a hora do horário da ocorrência CAMPO: ocorrencia_hora
base_ocorrencia['ocorrencia_hora'] = pd.to_datetime(base_ocorrencia['ocorrencia_hora'],
format='%H:%M:%S')
base_ocorrencia['ocorrencia_hora'] = base_ocorrencia.ocorrencia_hora.dt.hour
```

Acima foi extraído somente a hora do campo 'ocorrencia_hora', para que pudesse ser analisado se há uma maior incidência de ocorrências de acordo com o período do dia.

```
# Classificando em período do dia de acordo com o horário da ocorrência do acidente/incidente
for index, row in base_ocorrencia.iterrows():
    if row['ocorrencia_hora'] >= 0 and row['ocorrencia_hora'] <= 6:
        base_ocorrencia.at[index, 'periodo_ocorrencia'] = 'MADRUGADA'
    elif row['ocorrencia_hora'] > 6 and row['ocorrencia_hora'] < 12:
        base_ocorrencia.at[index, 'periodo_ocorrencia'] = 'MANHA'
    elif row['ocorrencia_hora'] >= 12 and row['ocorrencia_hora'] < 18:
        base_ocorrencia.at[index, 'periodo_ocorrencia'] = 'TARDE'
    elif row['ocorrencia_hora'] >= 18 and row['ocorrencia_hora'] <= 23:
        base_ocorrencia.at[index, 'periodo_ocorrencia'] = 'NOITE'
```

Acima fiz uma categorização dos registros, de acordo com a hora do dia em que eles ocorreram, veja:

MADRUGADA: evento ocorrido entre a 0 hora e 6 horas.

MANHÃ: evento ocorrido entre as 7 horas e 11 horas.

TARDE: evento ocorrido entre as 12 horas e 17 horas.

NOITE: evento ocorrido entre as 19 horas e 23 horas.

Tratamento dos dados – BASE TIPO OCORRÊNCIA

Não houve necessidade de fazer nenhum tipo de tratamento nos dados.

Junção entre as bases de dados

```
# Join entre as bases base_ocorrenciia e base_aeronave, gerando a base_completa
base_completa = pd.merge(base_ocorrenciia, base_aeronave, on='codigo_ocorrenciia2', how='inner')
base_completa = pd.merge(base_completa, base_ocorrenciia_tipo, on='codigo_ocorrenciia1', how='inner')
```

Acima fiz a junção entre as bases de dados, gerando uma nova base chamada 'base_completa':

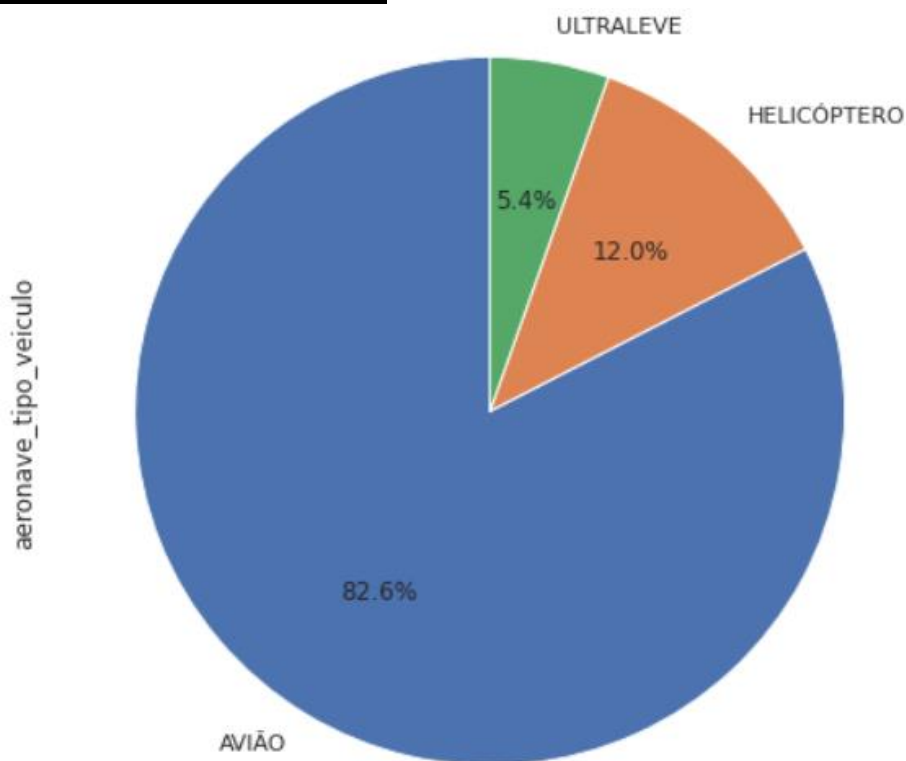
1º passo: Foram unidas as bases 'base_ocorrenciia' e 'base_aeronave' através do campo 'codigo_ocorrenciia2', considerando somente os registros correspondentes dos dois lados da base.

2º passo: Foram unidas as bases 'base_completa' e 'base_ocorrenciia_tipo' através do campo 'codigo_ocorrenciia_tipo'

Análise exploratória

```
base_aeronave["aeronave_tipo_veiculo"].value_counts().plot.pie(figsize=(20,8),
labels=["AERONAVE", "HELICÓPTERO", "ULTRALEVE"],
autopct='%1.1f%%',
startangle=90).set_title("% DE OCORRÊNCIAS EM NOSSA BASE", weight='bold', size=18);
```

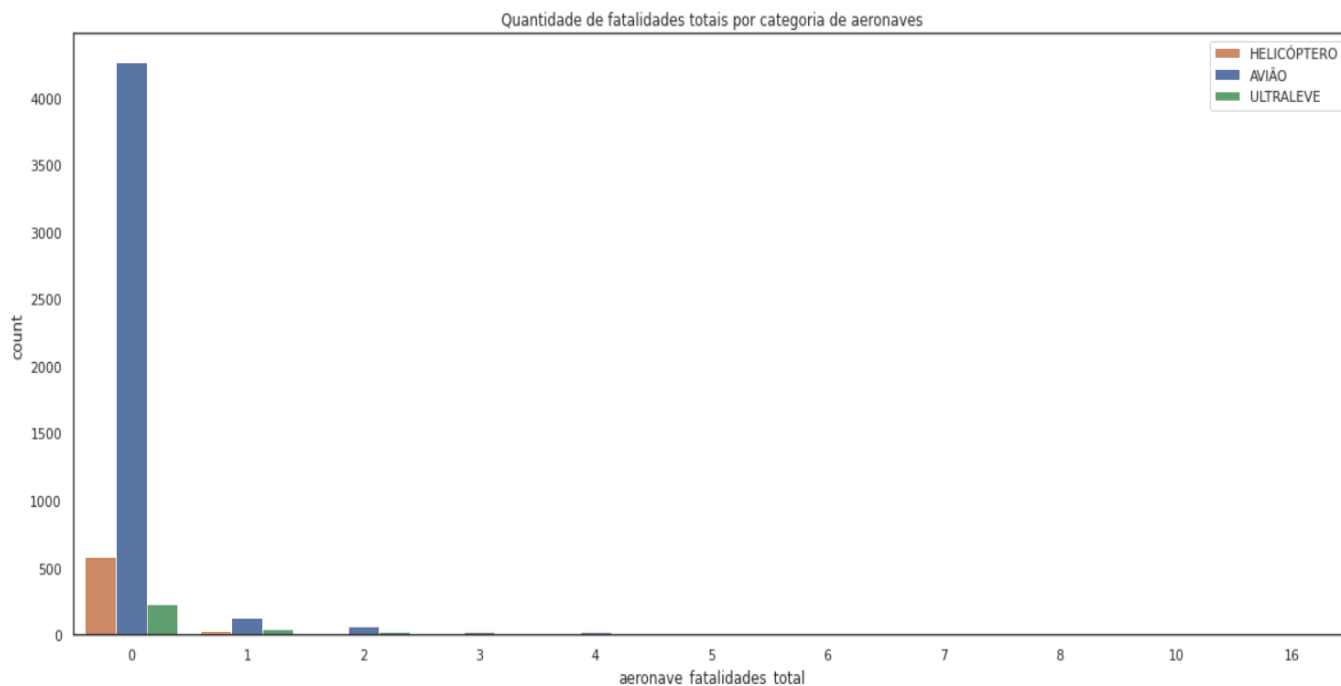
Percentual de ocorrências na base de dados



Em nossa base de dados podemos ver que a maioria das ocorrências são referentes a aviões.

Tipo Veículo X Fatalidades totais

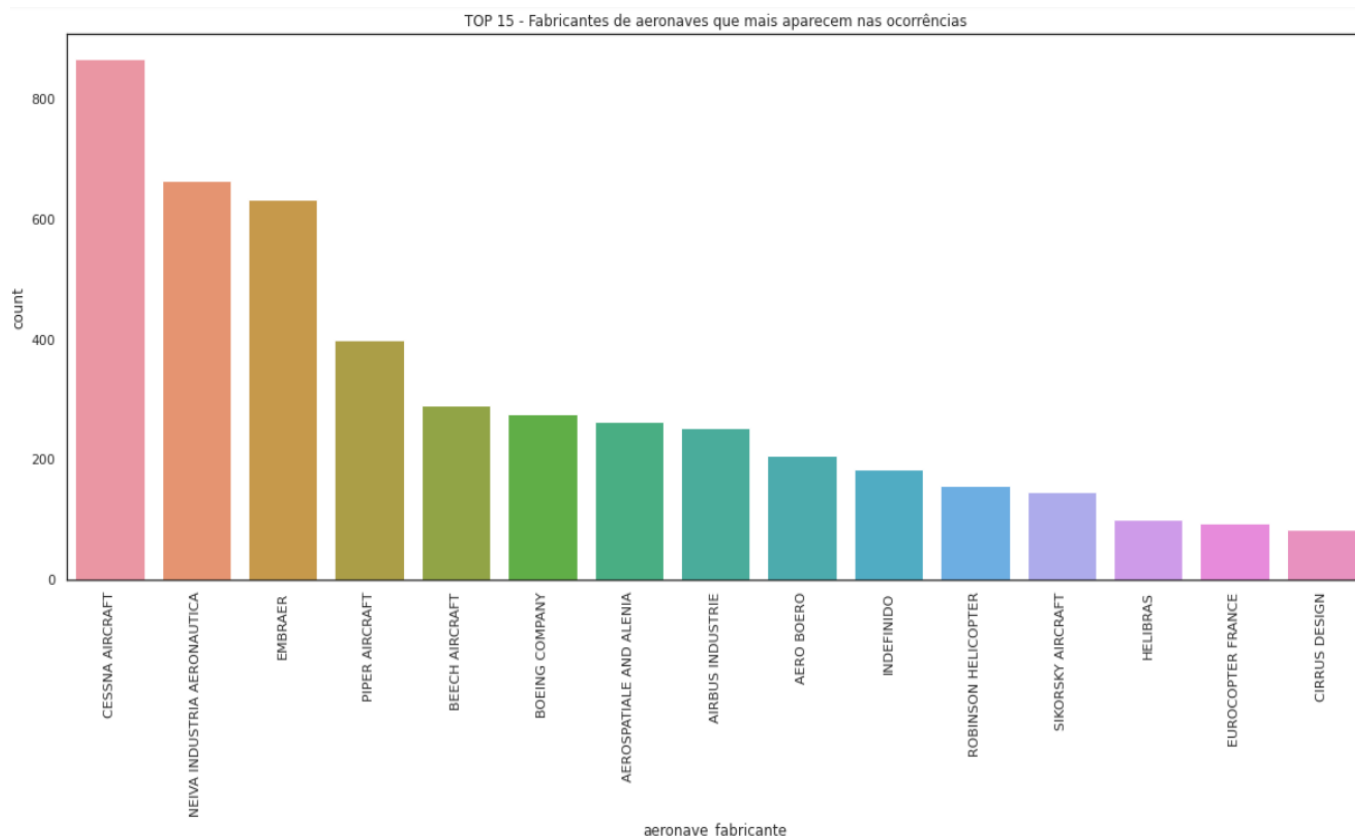
```
sns.countplot(hue=base_completa.aeronave_tipo_veiculo, x=base_completa.aeronave_fatalidades_total)
plt.legend(loc='upper right')
plt.title('Quantidade de fatalidades totais por categoria de aeronaves');
```



Como podemos ver, apesar de aviões terem um número muito maior de ocorrências, isto não se reflete em quantidade de fatalidades.

Fabricante X Fatalidades totais

```
sns.countplot(x=base_completa.aeronave_fabricante,
order=base_completa.aeronave_fabricante.value_counts().iloc[:15].index), xticks(rotation=90)
plt.title('TOP 15 - Fabricantes de aeronaves que mais aparecem nas ocorrências');
```

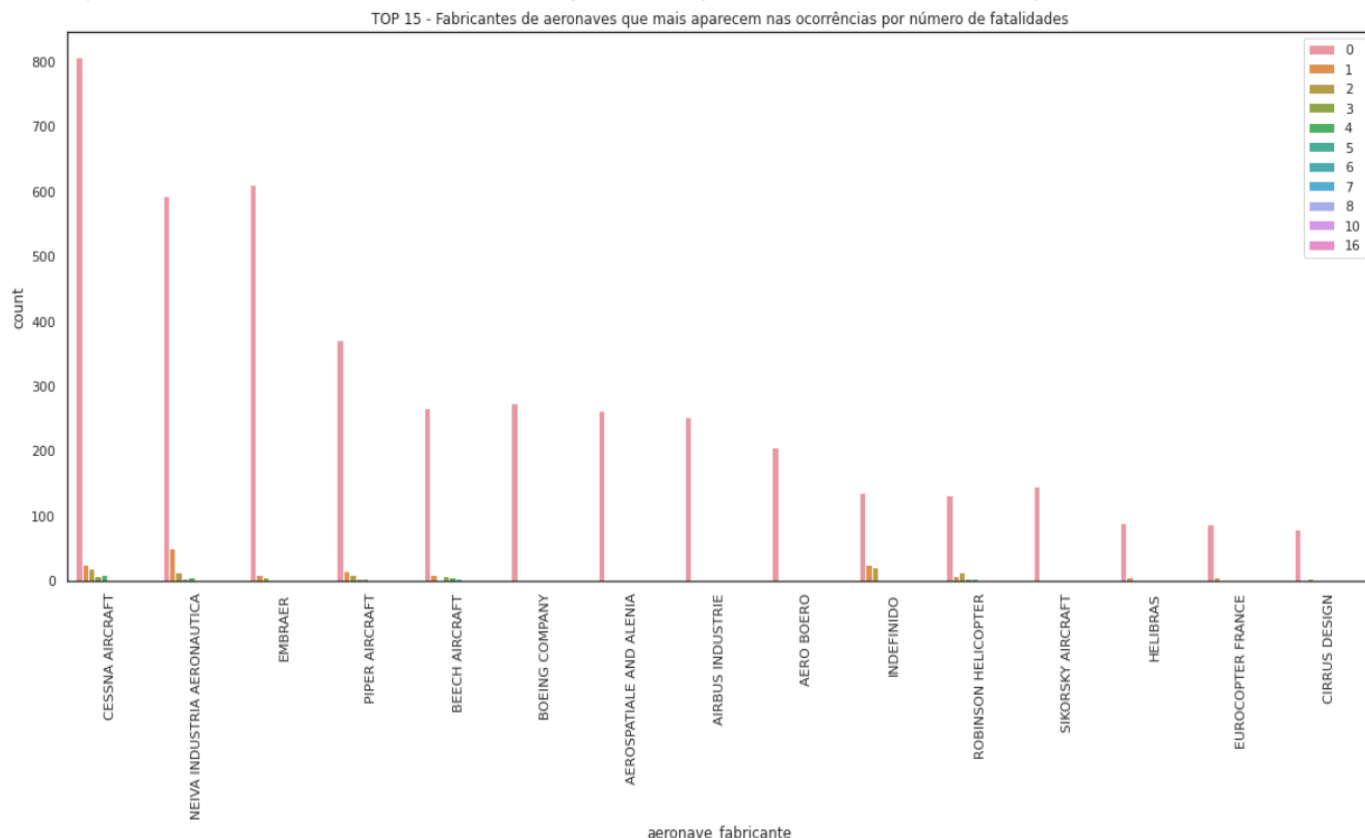


Vamos ver em relação as fatalidades o que os dados nos dizem:

```
sns.countplot(x=base_completa.aeronave_fabricante, hue=base_completa.aeronave_fatalidades_total,
order=base_completa.aeronave_fabricante.value_counts().iloc[:15].index), xticks(rotation=90)
```

```
plt.legend(loc='upper right')
```

```
plt.title('TOP 15 - Fabricantes de aeronaves que mais aparecem nas ocorrências por número de fatalidades');
```



Embora a Fabricante NEIVA INDUSTRIA AERONAUTICA tenha quase 200 ocorrências a menos que a CESSNA AIRCRAFT, podemos perceber que o número de fatalidades envolvendo ao menos uma vida é mais que o dobro da 1ª colocada.

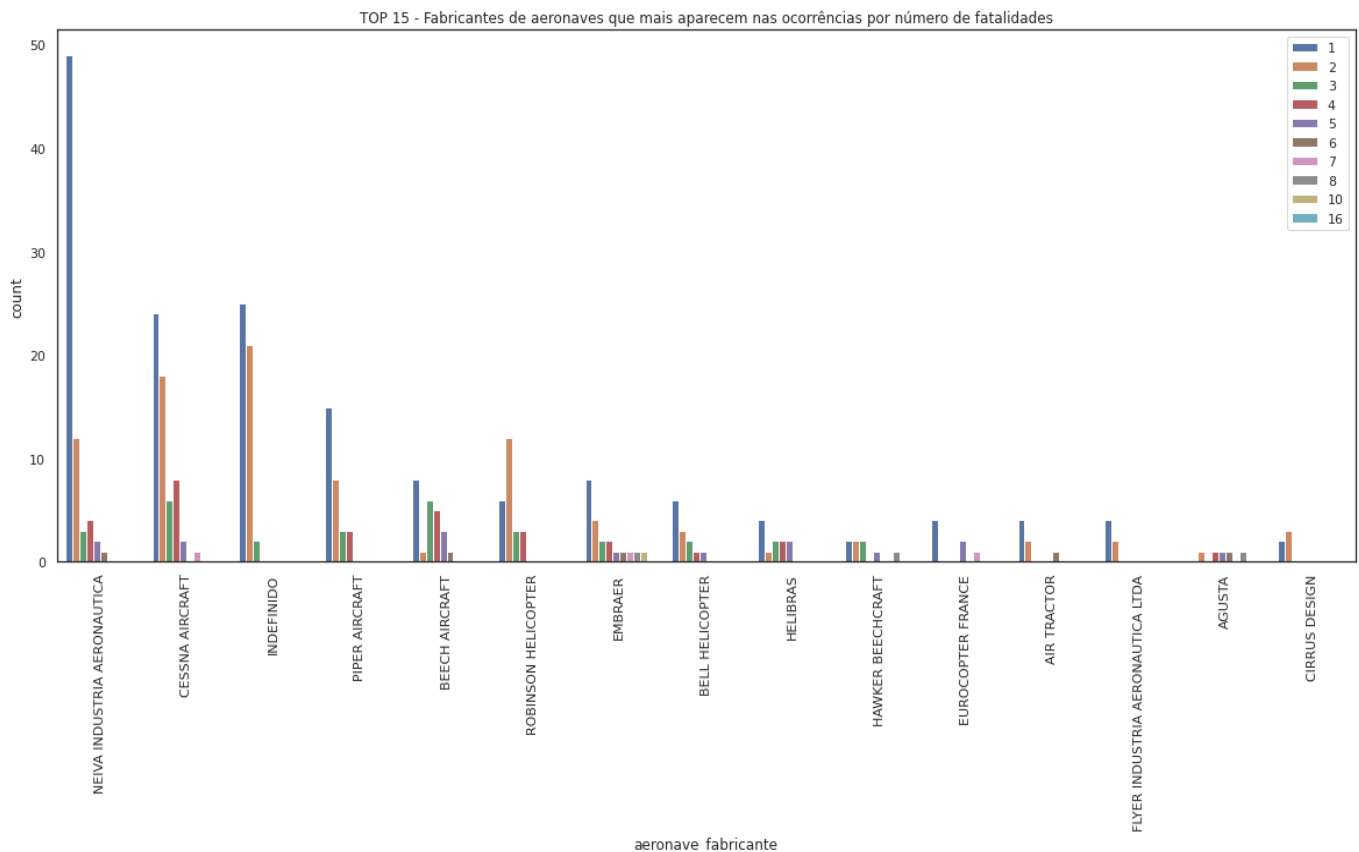
Vejamos agora as fabricantes, considerando somente os acidentes/incidentes com a fatalidade de ao menos uma pessoa:

```
base_fatalidade_maior_que_zero = base_completa.query('aeronave_fatalidades_total > 0')
```

```
sns.countplot(x=base_fatalidade_maior_que_zero.aeronave_fabricante,
hue=base_fatalidade_maior_que_zero.aeronave_fatalidades_total,
order=base_fatalidade_maior_que_zero.aeronave_fabricante.value_counts().iloc[:15].index,xticks(rotation=90))
```

```
plt.legend(loc='upper right')
```

```
plt.title('TOP 15 - Fabricantes de aeronaves que mais aparecem nas ocorrências por número de fatalidades');
```

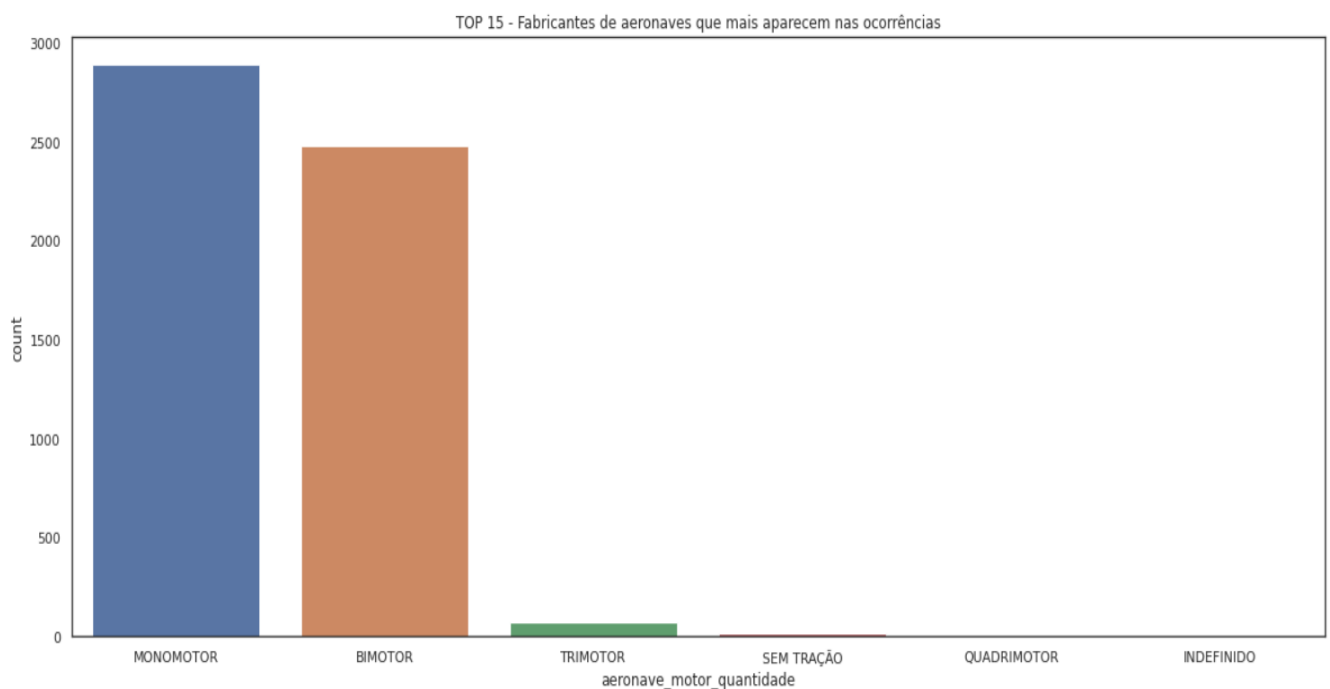



Como pressupus, apesar da fabricante NEIVA INDUSTRIA AERONAUTICA não ser a líder em ocorrências, ela é líder em acidentes/incidentes fatais envolvendo uma vida.

Ainda assim, se desconsiderarmos os registros indefinidos, a CESSNA AIRCRAFT lidera os acidentes/incidentes fatais entre duas e quatro vidas.

Quantidade de Motores x Fatalidades totais

```
sns.countplot(x=base_completa.aeronave_motor_quantidade,
order=base_completa.aeronave_motor_quantidade.value_counts().index)#,
xticks(rotation=90)
plt.title('TOP 15 - Fabricantes de aeronaves que mais aparecem nas ocorrências');
```



Podemos observar que a base se resume basicamente a aeronaves com um e dois motores apenas, juntas totalizando 98% dos registros.

MONOMOTOR: 53%

BIMOTOR: 45%

TRIMOTOR: 1,3%

SEM TRAÇÃO: menor que 1%

QUADRIMOTOR: menor que 1%

INDEFINIDO: menor que 1%

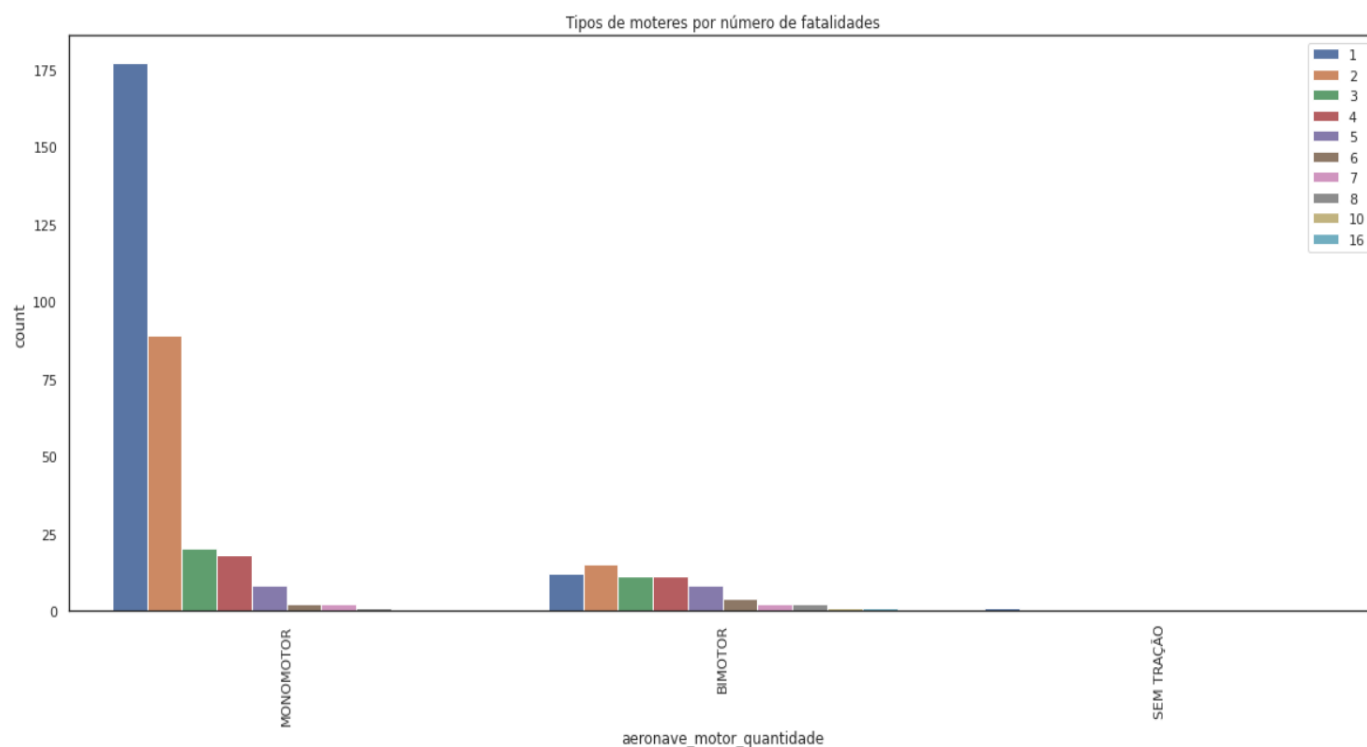
Vejamos a relação entre fatalidades e quantidades de motores nas aeronaves.

```
sns.countplot(x=base_fatalidade_maior_que_zero.aeronave_motor_quantidade,  
hue=base_fatalidade_maior_que_zero.aeronave_fatalidades_total,
```

```
order=base_fatalidade_maior_que_zero.aeronave_motor_quantidade.value_counts().index,xticks(rotation  
=90)
```

```
plt.legend(loc='upper right')
```

```
plt.title('Tipos de moteres por número de fatalidades');
```



Três coisas podemos observar aqui:

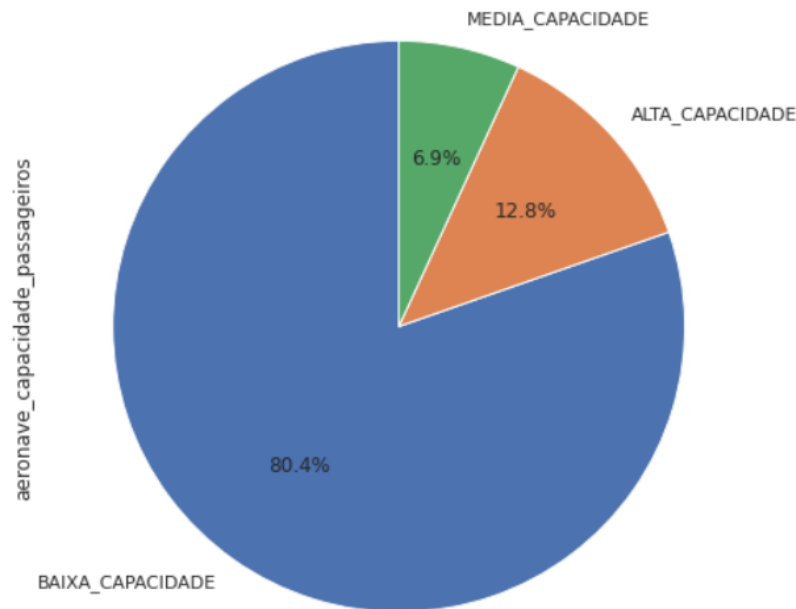
1. Não constam acidentes/incidentes fatais em nossa base envolvendo aeronaves trimotor e quadrimotor.
2. No caso de aviões bimotores, o número de fatalidades com mais de duas vidas é superior ao número de fatalidades com uma vida apenas.
3. A quantidade de vidas perdidas nos acidentes/incidentes com aviões bimotores é indiscutivelmente menor, considerando o número total de ocorrências com fatalidades ou não.

Isso pode ser explicado talvez por a aeronave não ficar desassistida na falha de um motor, apesar de não ser a condição ideal de voo, ainda há um motor para ser utilizado.

Capacidade de passageiros da aeronave x Fatalidades totais

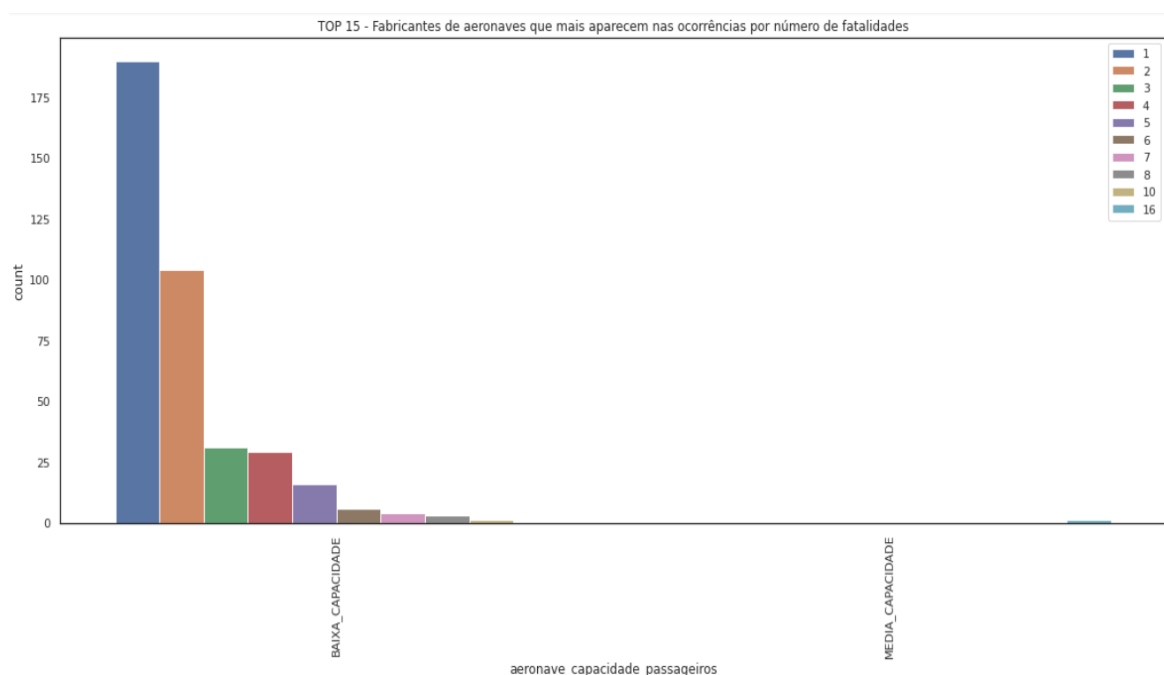
```
base_aeronave["aeronave_capacidade_passageiros"].value_counts().plot.pie(figsize=(20,8),
labels=["BAIXA_CAPACIDADE", "ALTA_CAPACIDADE", "MEDIA_CAPACIDADE"],
autopct='%1.1f%%',
startangle=90).set_title("% DE OCORRÊNCIAS EM NOSSA BASE", weight='bold', size=18);
```

Percentual de ocorrências na base de dados



A base é majoritariamente composta por aeronaves que comportam até 19 passageiros. Seguido de aeronaves de grande capacidade de passageiros (100 passageiros ou mais). E por último, aeronaves com média capacidade (20 a 99 passageiros).

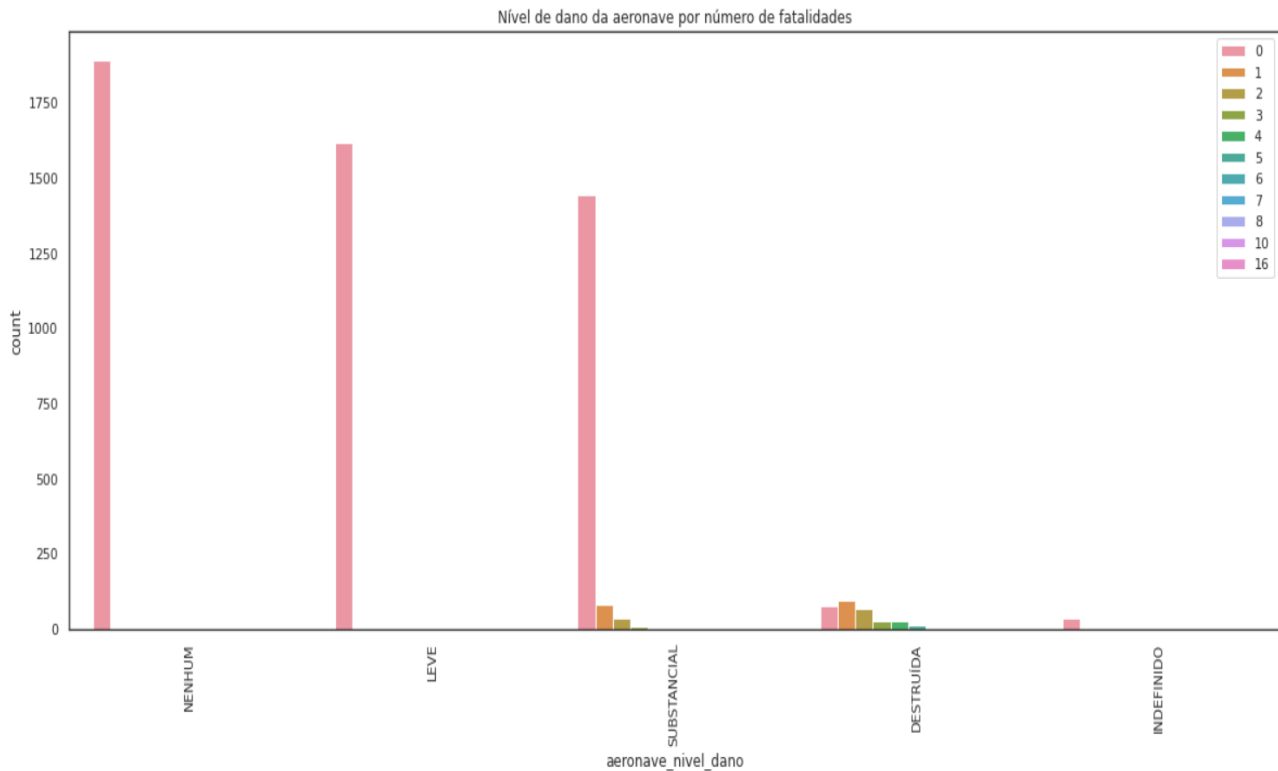
```
sns.countplot(x=base_fatalidade_maior_que_zero.aeronave_capacidade_passageiros,
hue=base_fatalidade_maior_que_zero.aeronave_fatalidades_total,
order=base_fatalidade_maior_que_zero.aeronave_capacidade_passageiros.value_counts().index,xticks(rotation=90)
plt.legend(loc='upper right')
```



Podemos ver que as aeronaves com grandes capacidades de passageiros felizmente não aparecem, quando olhamos somente para incidentes onde houve a perda de uma vida ao menos.

Aeronaves de média capacidade aparecem somente uma vez, mas com um acidente/incidente com 16 vidas perdidas.

O fato de aeronaves de grande capacidade não aparecerem, talvez se dê ao fato de serem grandes e tecnológicos aviões, que seguem rigorosas normas e exigências de segurança para poderem operar transportando uma grande quantidade de passageiros.



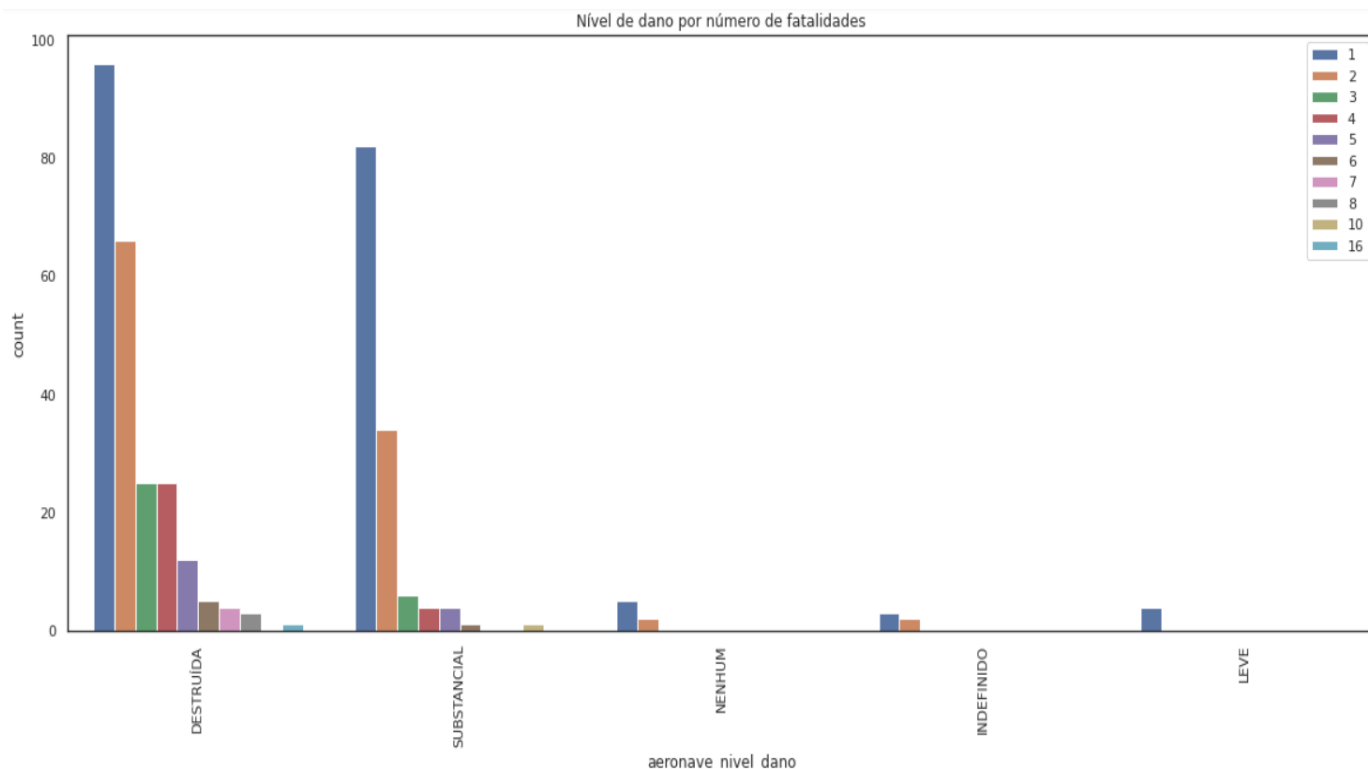
Nível de dano da aeronave x Fatalidades totais

```
sns.countplot(x=base_completa.aeronave_nivel_dano, hue=base_completa.aeronave_fatalidades_total,
order=base_completa.aeronave_nivel_dano.value_counts().index,xticks(rotation=90)
plt.legend(loc='upper right')
plt.title('Nível de dano da aeronave por número de fatalidades');
```

Podemos ver que pequenos incidentes com as aeronaves aparentemente não causam fatalidades.

Vamos olhar agora somente para os incidentes com fatalidades, para ver se a nossa suspeita está correta...

```
sns.countplot(x=base_fatalidade_maior_que_zero.aeronave_nivel_dano,
hue=base_fatalidade_maior_que_zero.aeronave_fatalidades_total,
order=base_fatalidade_maior_que_zero.aeronave_nivel_dano.value_counts().index,xticks(rotation=90)
plt.legend(loc='upper right')
plt.title('Nível de dano por número de fatalidades');
```



Podemos ver que quando ocorrem danos substanciais, também há uma quantidade relevante de fatalidades, menor que quando a aeronave é totalmente destruída, mas ainda sim considerável.

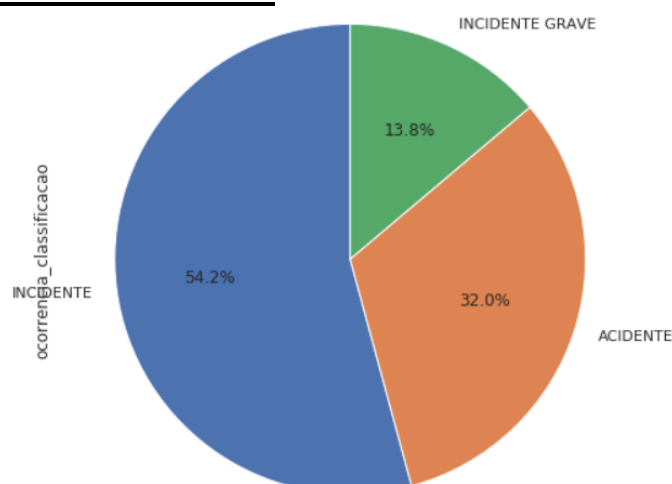
Vemos também casos curiosos, onde não houve danos a aeronave, mas houveram fatalidades. Pode ser um caso de erro nos dados, ou algo neste sentido.

Gravidade das ocorrências

Segundo o site <http://imedita.com.br/acidente-incidente-diferenca/>, eis a diferença entre acidente e incidente: a grande diferença entre **acidente** e **incidente** é o dano. No incidente o fato inesperado e potencialmente perigoso acontece, mas graças a alguma circunstância favorável ele não causa danos a ninguém. Já no acidente o dano acontece e há perdas significativas.

```
base_completa["ocorrencia_classificacao"].value_counts().plot.pie(figsize=(20,8),
labels=["INCIDENTE", "ACIDENTE", "INCIDENTE GRAVE"],
autopct='%1.1f%%',
startangle=90).set_title("% DE OCORRÊNCIAS EM NOSSA BASE", weight='bold', size=18);
```

Percentual de ocorrências na base de dados

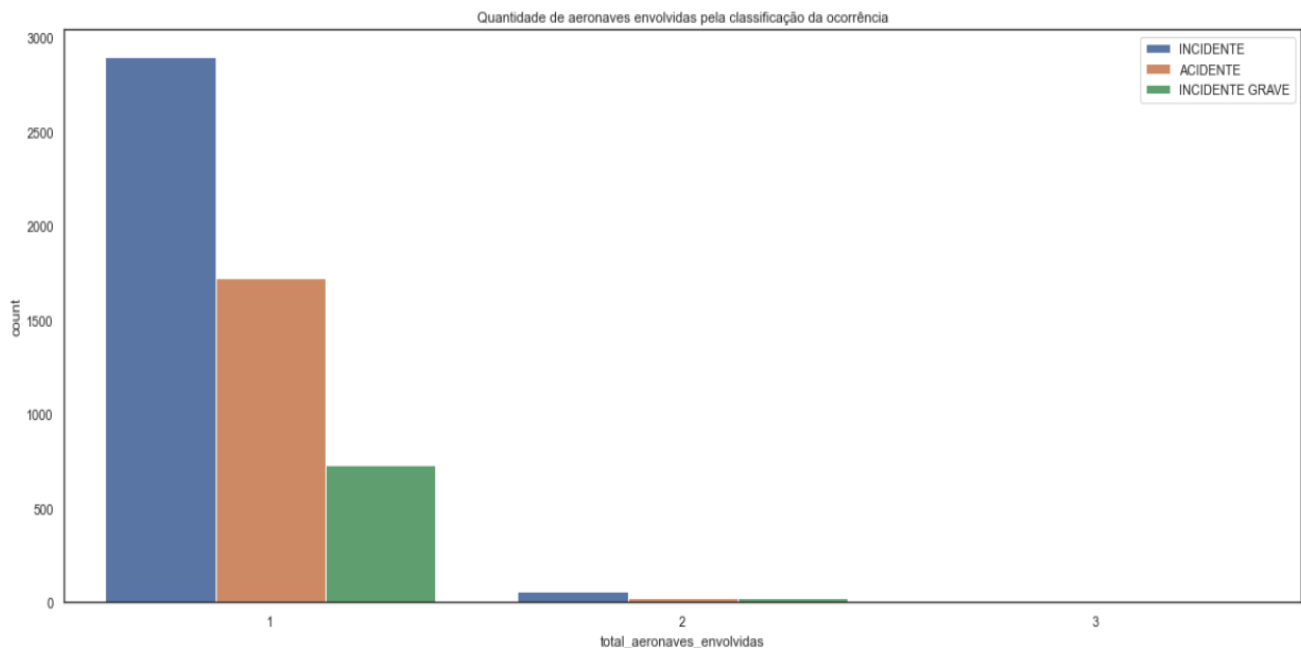


Incidentes contemplam a maioria das ocorrências, com 54,2% do total, seguido de Acidentes, com 32% e Incidente Graves, com 13,8%.

Vamos ver a classificação das ocorrências, considerando o número de aviões envolvidos.

Número de aeronaves envolvidas x Classificação da ocorrência

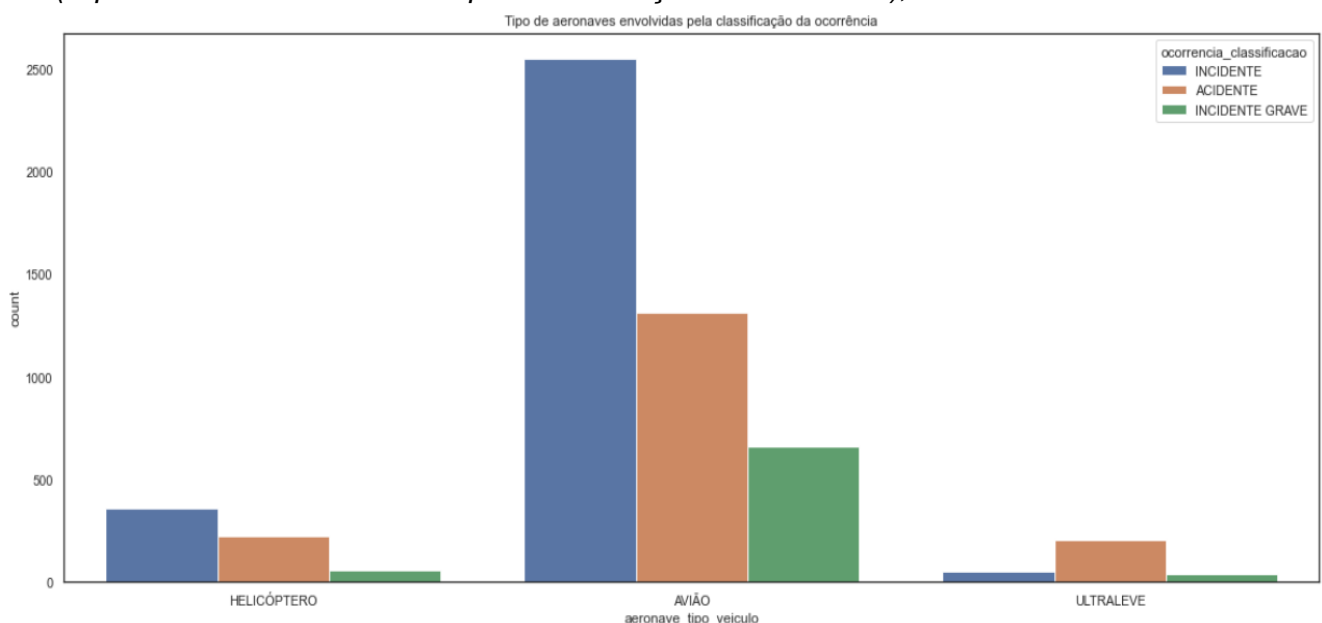
```
sns.countplot(base_completa.total_aeronaves_envolvidas, hue=base_completa.occurencia_classificacao)  
plt.legend(loc='upper right')  
plt.title('Quantidade de aeronaves envolvidas pela classificação da ocorrência');
```



Felizmente incidentes graves são menos frequentes dentre as ocorrências. E as ocorrências envolvendo mais de uma aeronave em sua maioria não são graves.

Tipo de veículo da aeronave x Classificação da ocorrência

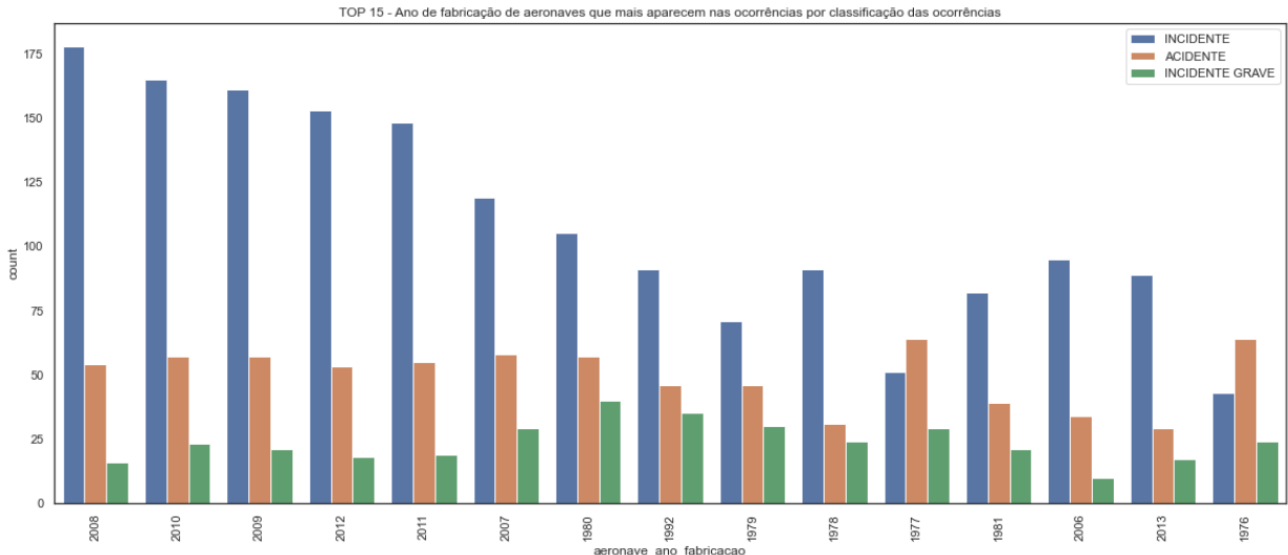
```
sns.countplot(base_completa.aeronave_tipo_veiculo, hue=base_completa.occurencia_classificacao)  
plt.title('Tipo de aeronaves envolvidas pela classificação da ocorrência');
```



O único tipo de aeronave que não segue a tendência da base completa é o ultraleve, com mais acidentes que Incidentes e Incidentes Graves.

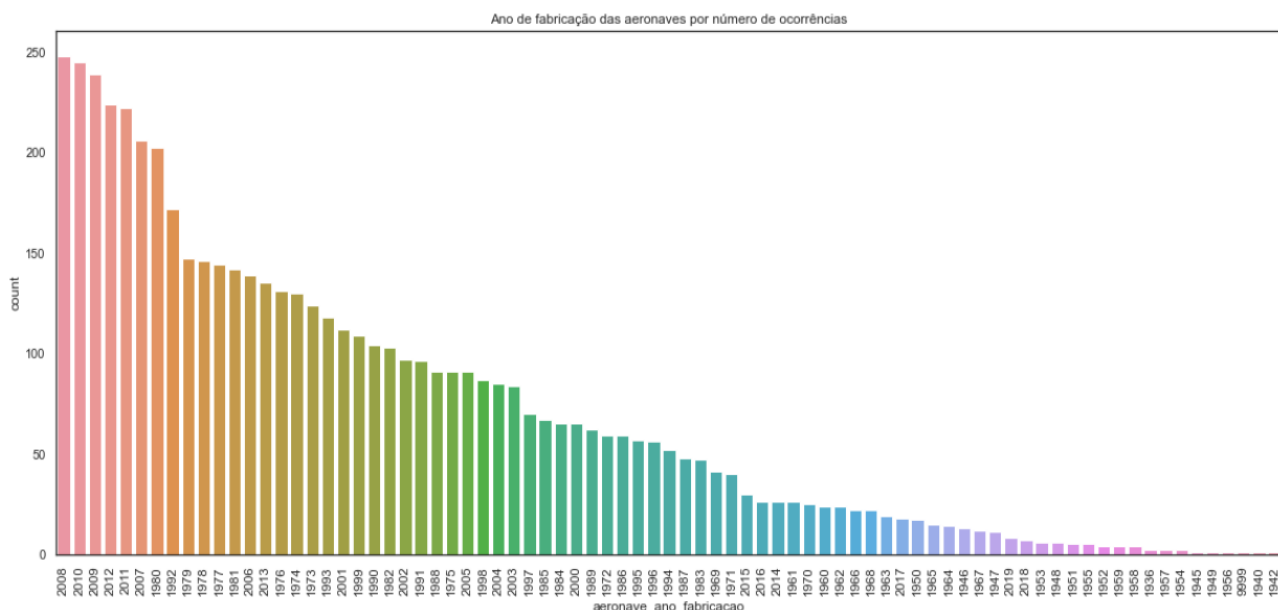
Ano de fabricação da aeronave x Classificação da ocorrência

```
sns.countplot(x=base_completa.aeronave_ano_fabricacao, hue=base_completa.ocorrencia_classificacao,
order=base_completa.aeronave_ano_fabricacao.value_counts().iloc[:15].index,xticks(rotation=90)
plt.legend(loc='upper right')
plt.title('TOP 15 - Ano de fabricação de aeronaves que mais aparecem nas ocorrências por classificação das
ocorrências');
```



Não consegui apontar uma relação clara do ano de fabricação da aeronave com o tipo de ocorrências. A princípio pensamos que o fato de as aeronaves serem mais antigas, pudesse ocasionar maiores problemas devido a desgaste, manutenção e outros aspectos referentes ao tempo.

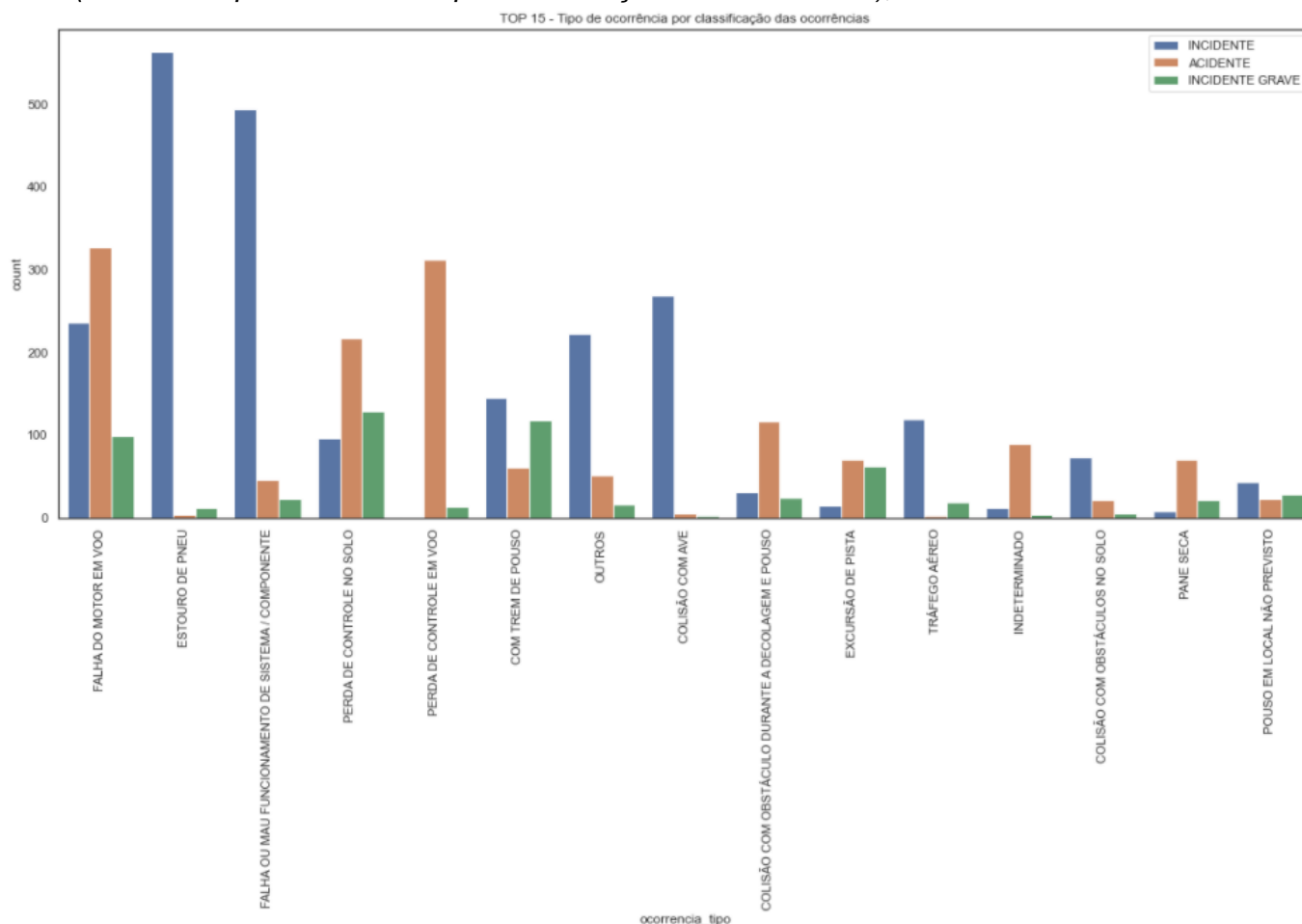
```
sns.countplot(base_completa.aeronave_ano_fabricacao,
order=base_completa.aeronave_ano_fabricacao.value_counts().index,xticks(rotation=90)
plt.title('Ano de fabricação das aeronaves por número de ocorrências');
```



Podemos verificar que, as maiores quantidades de ocorrências acontecem com as aeronaves com ano de fabricação entre 2007 e 2011. Ao contrário do que se poderia pensar, os aviões mais novos são os que mais podem ocasionar ocorrências.

Tipo da ocorrência x Classificação da ocorrência

```
sns.countplot(x=base_completa.occurencia_tipo, hue=base_completa.occurencia_classificacao,
order=base_completa.occurencia_tipo.value_counts().iloc[:15].index,xticks(rotation=90)
plt.legend(loc='upper right')
plt.title('TOP 15 - Tipo de ocorrência por classificação das ocorrências');
```



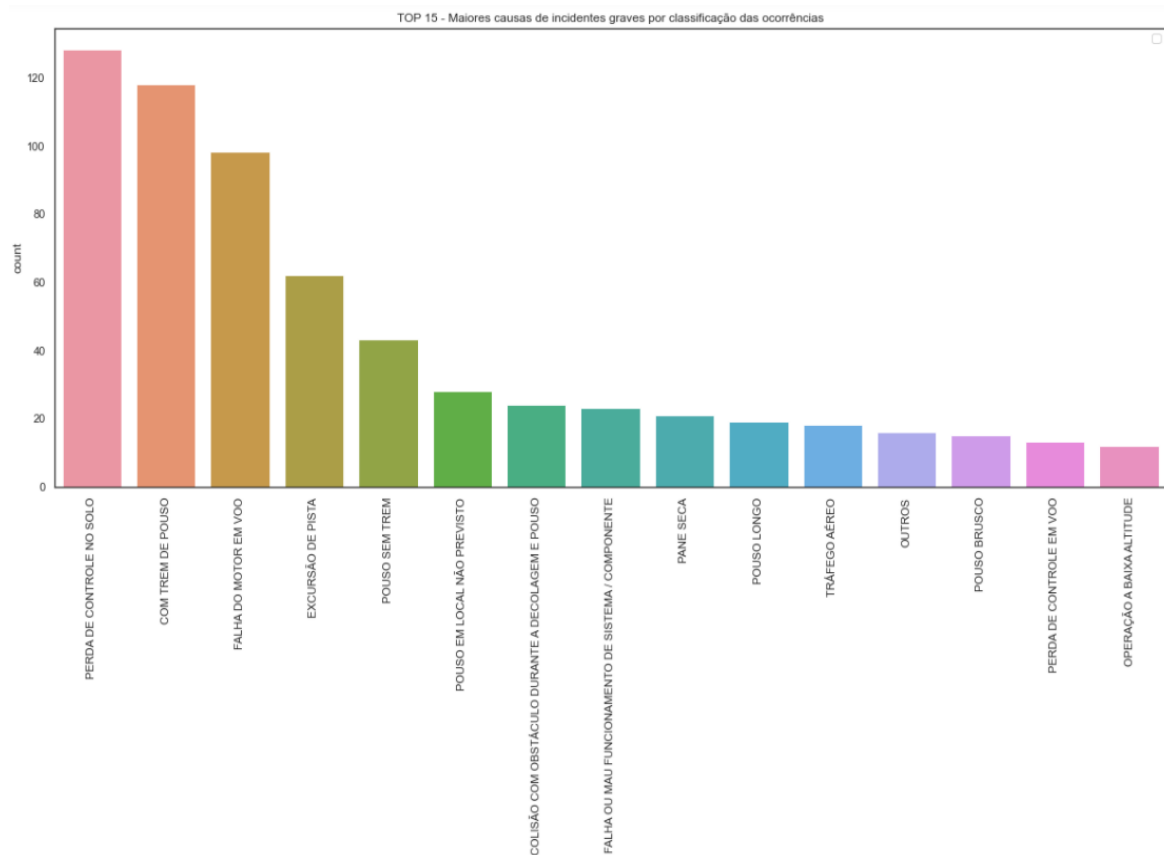
Segundo o gráfico, os **maiores causadores de incidentes são**: estouro de pneu, falha ou mau funcionamento de sistema/componente e colisão com aves.

Maiores causadores de acidentes: falha do motor em voo e perda de controle em voo estão praticamente empatados e em terceiro, perda de controle em solo.

Maiores causadores de incidentes graves: perda de controle em solo, problemas com o trem de pouso e falha do motor em voo.

Vamos olhar com mais detalhes as causas de incidentes graves:

```
base_incidentes_graves = base_completa.query('occurencia_classificacao == "INCIDENTE GRAVE"')
sns.countplot(x=base_incidentes_graves.occurencia_tipo,
order=base_incidentes_graves.occurencia_tipo.value_counts().iloc[:15].index,xticks(rotation=90)
plt.legend(loc='upper right')
plt.title('TOP 15 - Maiores causas de incidentes graves por classificação das ocorrências');
```

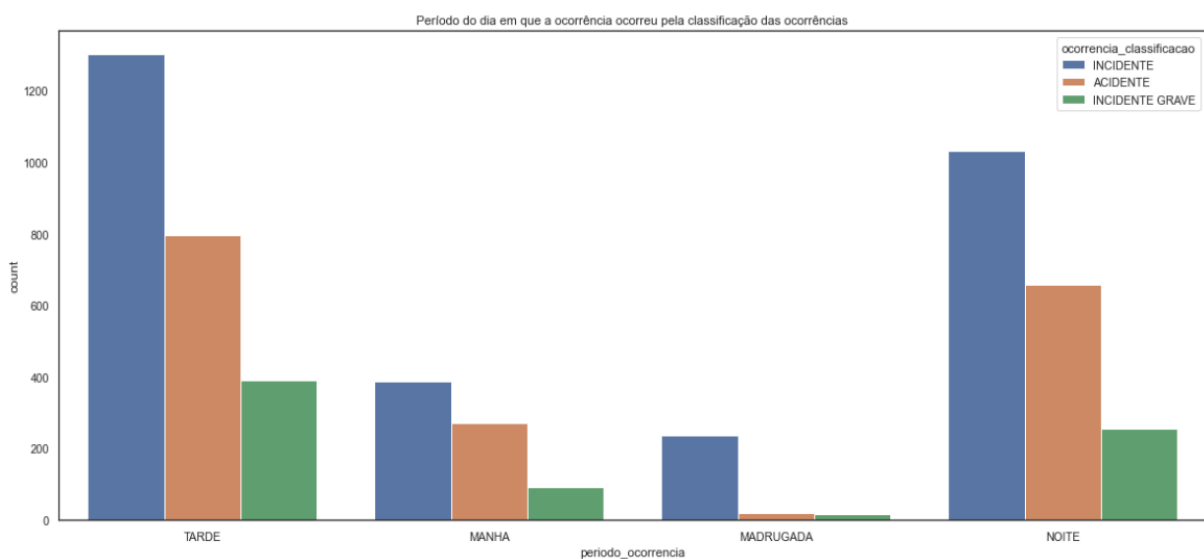



Podemos observar que entre as 7 maiores causas de incidentes graves, 6 estão relacionadas a pouso e decolagem!

Considerando estas informações, vejamos se a má visibilidade da pista ou algum outro fator envolvendo uma menor claridade do dia pode contribuir com este cenário:

Período do dia da ocorrência x Classificação da ocorrência

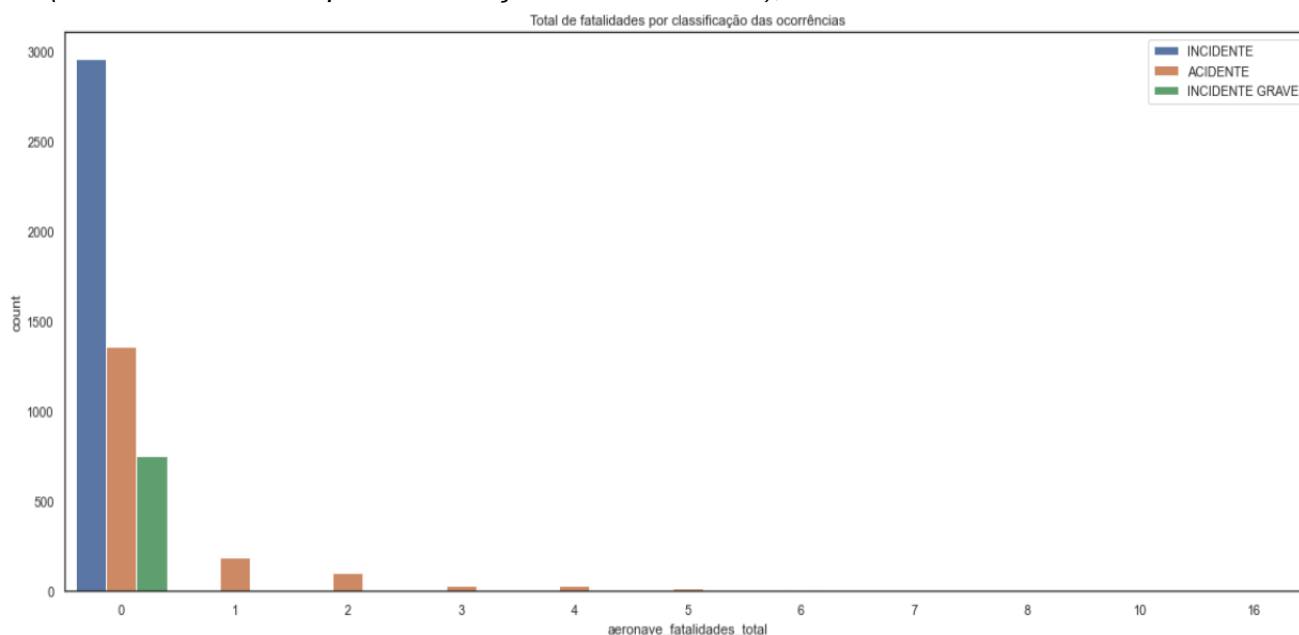
```
sns.countplot(base_completa.periodo_ocorrenci, hue=base_completa.ocorrenci_classificacao)
plt.title('Período do dia em que a ocorrência ocorreu pela classificação das ocorrências');
```



Apesar dos números das ocorrências serem altos no período da noite, não podemos afirmar que ele influencie na incidência das ocorrências, considerando que o período em que há mais ocorrências é durante a tarde.

Número de fatalidades x Classificação da ocorrência

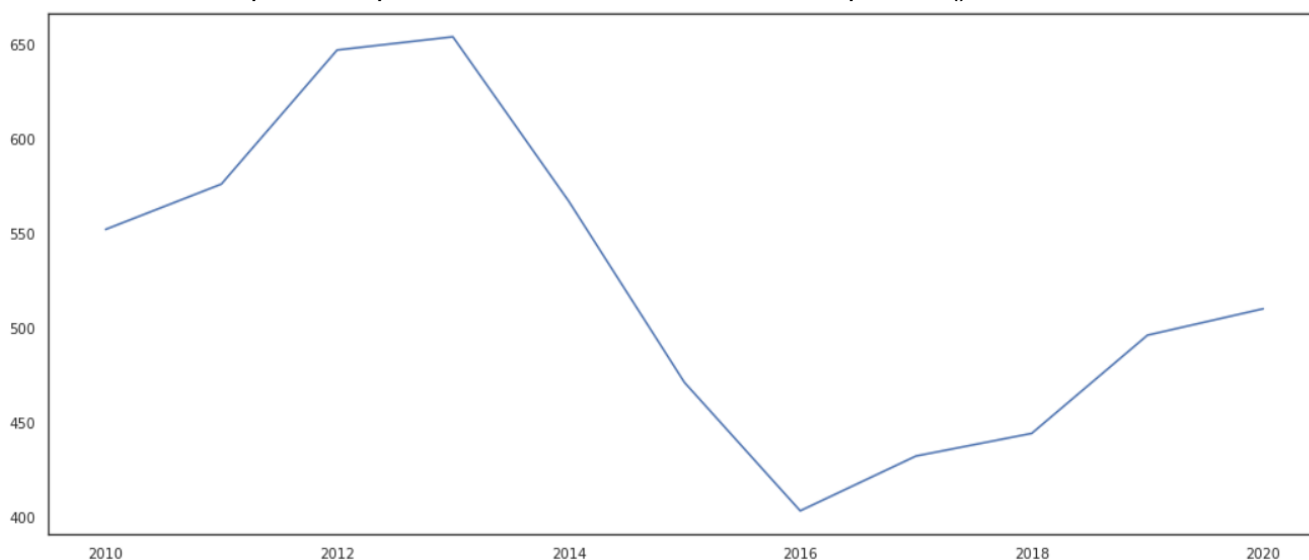
```
sns.countplot(base_completa.aeronave_fatalidades_total, hue=base_completa.occurencia_classificacao)
plt.legend(loc='upper right')
plt.title('Total de fatalidades por classificação das ocorrências');
```



Podemos ver que os grandes causadores de fatalidades são os acidentes.

Quantidade de acidentes ocorridos no ano

```
plt.figure(figsize=(18, 7))
acidentes_ocorridos_por_ano.quantidade_de_acidentes_ocorridos.plot.line();
```



Podemos ver que a partir de 2012 ocorreu uma forte queda nas ocorrências de incidentes/acidentes de aeronaves no Brasil, atingindo o seu menor valor em 2016.

Em 2017 essas ocorrências voltaram a crescer, mas ainda assim, abaixo dos valores registrados no início da década passada.

Novas medidas de segurança, protocolos, procedimentos/recomendações podem ter sido implementados para que estes números tenham diminuído nos últimos anos.