In [0]:

```
#importando as bibliotecas
import pandas as pd  #bibioteca responsável para o tratamento e limpeza dos dados
import numpy as np #biblioteca utilizada para o tratamento eficiente de dados numéricos
import datetime  #biblioteca utilizada para trabalhar com datas
from matplotlib import pyplot as plt  #plotar os gráficos
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns #plot de gráficos
```

In [65]:

```
#importando os dados para o google colab
from google.colab import files
uploaded = files.upload()
```

Choose File No file selected

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving aircrafts.csv to aircrafts (2).csv
```

In [63]:

```
#importando os dados para o google colab
from google.colab import files
uploaded = files.upload()
```

Choose File No file selected

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving occurrences.csv to occurrences (1).csv
```

In [66]:

```
# Carregando dataframe aeronaves
df_aeronaves = pd.read_csv("aircrafts.csv", encoding='latin-1')
df_aeronaves.head()
```

Out[66]:

	Unnamed: 0	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	ta
0	0	4	45602	PPGXE	241	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	
1	1	40	53551	PPGSZ	160	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	
2	2	118	43721	PTCMT	1232	AIRPLANE	BEECH AIRCRAFT	95-B55	PISTON	2.0	
3	3	130	35556	PTEQI	3992	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-721C	PISTON	1.0	
4	4	191	32579	PPVMM	4365	AIRPLANE	BOEING COMPANY	737-241	JET	2.0	

In [67]:

```
# Carregando dataframe ocorrências
df_occurrences = pd.read_csv("occurrences.csv", encoding='latin-1')
df_occurrences.head()
```

Out[67]:

	Unnamed: 0	occurrence_id	classification	type of occurrence	localization	fu	country	aerodrome	occurrence_day	time	under_inves
0	0	47965	ACCIDENT	ENGINE FAILURE DURING THE FLIGHT	ARIQUEMES	RO	BRAZIL	SJOG	2013-05-05	11:00:00	UN
1	1	50313	SERIOUS INCIDENT	LANDING WITHOUT LANDING GEAR	CACOAL	RO	BRAZIL	SSKW	2013-11-25	12:32:00	
2	2	34078	ACCIDENT	LOSS OF CONTROL ON THE GROUND	CEREJEIRAS	RO	BRAZIL	****	2008-08-07	15:10:00	
3	3	44988	ACCIDENT	SLOW LANDING	AMAJARI	RR	BRAZIL	****	2011-08-11	17:00:00	
4	4	38855	ACCIDENT	LOSS OF CONTROL IN THE AIR	ACEGUÁ	RS	BRAZIL	****	2009-12-28	17:30:00	

In [68]:

```
#Fazendo um Join com os dois dataframes
df_acidentes  = df_aeronaves.merge(df_occurrences, on="occurrence_id", how='left')
df_acidentes.head()
```

Out[68]:

	Unnamed: 0_x	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	ta
0	0	4	45602	PPGXE	241	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	
1	1	40	53551	PPGSZ	160	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	
2	2	118	43721	PTCMT	1232	AIRPLANE	BEECH AIRCRAFT	95-B55	PISTON	2.0	
3	3	130	35556	PTEQI	3992	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-721C	PISTON	1.0	
4	4	191	32579	PPVMM	4365	AIRPLANE	BOEING COMPANY	737-241	JET	2.0	

In [69]:

```
# Removendo Colunas desnecessárias
colunas_retiradas = ['Unnamed: 0_x', 'Unnamed: 0_y', 'extraction_day_y', 'Unnamed: 0_x'] #lista
que contém as colunas a serem retiradas
df_acidentes.drop(columns=colunas_retiradas,axis=1,inplace=True)
df_acidentes.head()
```

Out[69]:

	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	takeoff_max_v
0	4	45602	PPGXE	241	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	
1	40	53551	PPGSZ	160	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	56-C	PISTON	1.0	

	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	takeoff_max_v
2	118	43721	PTCMT	1232	AIRPLANE	BEECH AIRCRAFT	95-B55	PISTON	2.0	
3	130	35556	PTEQI	3992	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-721C	PISTON	1.0	
4	191	32579	PPVMM	4365	AIRPLANE	BOEING COMPANY	737-241	JET	2.0	1

In [70]:

```
# Verificando Informações do dataframe
df_acidentes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2043 entries, 0 to 2042
Data columns (total 39 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   aircraft_id               2043 non-null   int64
 1   occurrence_id             2043 non-null   int64
 2   registration              2043 non-null   object
 3   operator_id               2043 non-null   int64
 4   equipment                 2042 non-null   object
 5   manufacturer              2043 non-null   object
 6   model                     2043 non-null   object
 7   engine_type               2043 non-null   object
 8   engines_amount            2034 non-null   float64
 9   takeoff_max_weight (Lbs)  2043 non-null   int64
 10  seatings_amount           2025 non-null   float64
 11  year_manufacture          2039 non-null   float64
 12  registration_country      2031 non-null   object
 13  registration_category     2043 non-null   object
 14  registration_aviation     2043 non-null   object
 15  origin_flight             2043 non-null   object
 16  destination_flight        2043 non-null   object
 17  operation_phase           2042 non-null   object
 18  type_operation            2043 non-null   object
 19  damage_level              2043 non-null   object
 20  fatalities_amount         355 non-null    float64
 21  extraction_day_x          2043 non-null   object
 22  classification            2043 non-null   object
 23  type of occurrence        2043 non-null   object
 24  localization              2043 non-null   object
 25  fu                        2043 non-null   object
 26  country                   2043 non-null   object
 27  aerodrome                 2040 non-null   object
 28  occurrence_day            2043 non-null   object
 29  time                      2043 non-null   object
 30  under_investigation       2043 non-null   object
 31  investigating_command     2043 non-null   object
 32  investigation_status      1834 non-null   object
 33  report_number             436 non-null    object
 34  published_report          1001 non-null   float64
 35  publication_day           1001 non-null   object
 36  recommendation_amount     2043 non-null   int64
 37  aircrafts_involved        2043 non-null   int64
 38  takeoff                   256 non-null    float64
dtypes: float64(6), int64(6), object(27)
memory usage: 638.4+ KB
```

In [71]:

```
#verificar informações dos dados do nosso dataset
df_acidentes.describe()
```

Out[71]:

	aircraft_id	occurrence_id	operator_id	engines_amount	takeoff_max_weight (Lbs)	seatings_amount	year_manufacture	fatalities_a

	aircraft_id	occurrence_id	operator_id	engines_amount	takeoff_max_weight (Lbs)	seatings_amount	year_manufacture	fatalities_
count	2043.000000	2043.000000	2043.000000	2034.000000	2043.000000	2025.000000	2039.000000	355
mean	12300.670093	43961.869799	3156.447871	1.244346	11750.045032	8.928889	1902.494850	3
std	7654.268691	7857.658738	1645.351104	0.483653	48511.565643	26.922299	402.024605	13
min	4.000000	25799.000000	13.000000	0.000000	0.000000	0.000000	0.000000	1
25%	9061.000000	38839.500000	1821.000000	1.000000	1860.000000	2.000000	1975.000000	1
50%	11267.000000	45564.000000	3992.000000	1.000000	3600.000000	4.000000	1986.000000	2
75%	13601.500000	50353.500000	3992.000000	2.000000	4750.000000	6.000000	1999.000000	2
max	39147.000000	65312.000000	6270.000000	4.000000	630499.000000	301.000000	2015.000000	199

In [72]:

```
# Verificando Ocorrências que possuem mais de um registro,  ou seja um acidente envolvendo mais de
1 avião.
df_acidentes['occurrence_id'].value_counts()
```

Out[72]:

```
38419    4
52288    2
43869    2
50801    2
52539    2
        ..
45753    1
43706    1
45755    1
45757    1
45056    1
Name: occurrence_id, Length: 2027, dtype: int64
```

In [73]:

```
# Verificando um exemplo de ocorrência com mais de um registro, ou seja um acidente envolvendo mai
s de 1 avião.
# Neste caso abaixo temos o acidente que aconteceu em Congonhas no dia 26/08/2009
df_acidentes[df_acidentes['occurrence_id'] == 38419]
```

Out[73]:

	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	takeoff_
882	10406	38419	PRSJE	3992	AIRPLANE	HAWKER BEECHCRAFT	C90GTI	TURBOPROP	2.0	
883	10407	38419	PRONE	3992	AIRPLANE	LEARJET	45	JET	2.0	
884	10408	38419	PTWVG	1657	AIRPLANE	RAYTHEON AIRCRAFT	HAWKER 800XP	JET	2.0	
1998	34248	38419	N413HB	3992	AIRPLANE	HAWKER BEECHCRAFT	HAWKER 4000	JET	2.0	

In [0]:

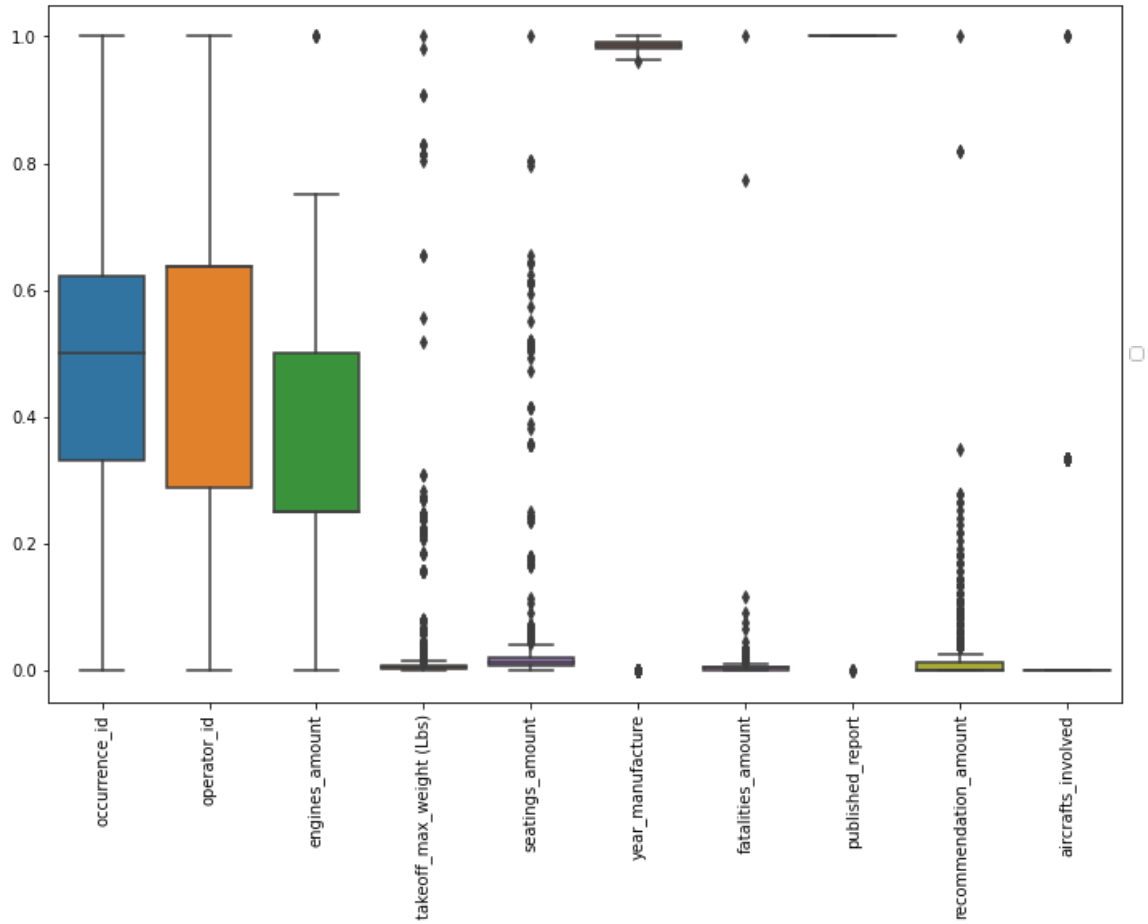
```
# Criando um dataframe para utilizarmos como analise no Boxplot
df_boxplot = df_acidentes
```

In [75]:

```
# Preparar um dataset para analisar os seus outliers

colunas_retiradas = ['registration', 'equipment', 'manufacturer', 'model', 'engine_type', 'registration
```

```
colunas_retiradas = ['registration','equipment','manufacturer','model','engine_type','registration_
country','registration_category','registration_aviation'
,'origin_flight','destination_flight','operation_phase','type_operation','damage_level','extraction
_day_x','classification','type of occurrence','localization','fu'
,'country',
'aerodrome','occurrence_day','time','under_investigation','investigating_command','investigation_st
atus','report_number','publication_day'] #lista que contém as colunas a serem retiradas
df_boxplot = df_boxplot.drop(columns=colunas_retiradas,axis=1)
df_boxplot.values.reshape((-1, 1))


# Ajustando a Escala para exibir o boxplot de análise de outliers
min_max_scaler = MinMaxScaler()
df_scaled_boxplot = pd.DataFrame(min_max_scaler.fit_transform(df_boxplot.iloc[:, 1:-1]))
df_scaled_boxplot.columns = df_boxplot.columns[1:-1]

plt.figure(figsize=(12,8))
fig = sns.boxplot(data=df_scaled_boxplot)
for item in fig.get_xticklabels():
    item.set_rotation(90)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```

```
No handles with labels found to put in legend.
```

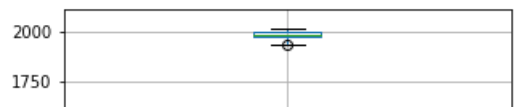


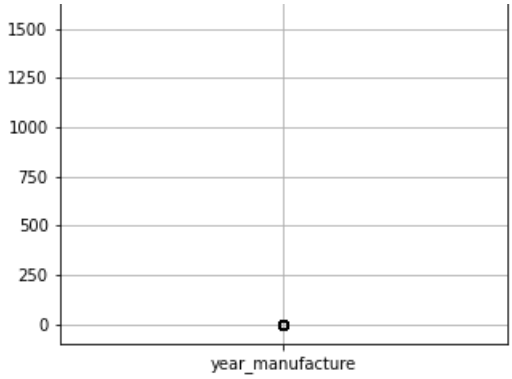
In [76]:

```
#identificando os outliers
plt.figure(figsize=(5,5))
df_acidentes.boxplot(['year_manufacture'])
```

Out[76]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1630339ef0>
```





In [77]:

```
#identificando os outliers
df_acidentes[df_acidentes['year_manufacture']==0]
```

Out[77]:

	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	takeoff_m
97	1629	41569	PTGRE	120	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-201A	PISTON	1.0	
139	2398	32378	PTGEE	2388	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-200A	PISTON	1.0	
196	3352	45594	PTGQD	1552	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-201	PISTON	1.0	
369	6283	51437	PPZZZ	3992	AIRPLANE	FABRICACAO PROPRIA	RANS S-10-AVIAO	PISTON	1.0	
388	6691	42365	PTNTR	3992	AIRPLANE	NEIVA INDUSTRIA AERONAUTICA	EMB-710C	PISTON	1.0	
...	...	...	...	...	...	...	...	...	...	...
2023	34635	51597	CXJYN	5932	AIRPLANE	CESSNA AIRCRAFT	402B	PISTON	2.0	
2024	34636	52539	RA0976G	3992	AIRSHIP	MAULE AIRCRAFT	***	PISTON	1.0	
2025	34637	52539	SPBBP	3992	AIRSHIP	MAULE AIRCRAFT	***	PISTON	1.0	
2028	34643	53193	PTXXX5	3992	AIRPLANE	***	***	UNKNOWN	NaN	
2042	39147	28437	DGOMM	3992	AIRPLANE	PIPER AIRCRAFT	PA34	PISTON	2.0	

87 rows × 39 columns

In [78]:

```
#identificando os outliers
df_acidentes[df_acidentes['equipment']=='UNKNOWN']
```

Out[78]:

	aircraft_id	occurrence_id	registration	operator_id	equipment	manufacturer	model	engine_type	engines_amount	takeoff_max
1931	29955	53341	PTZPD	3992	UNKNOWN	***	***	UNKNOWN	0.0	
1944	31111	53150	PUDSJ	3992	UNKNOWN	***	***	UNKNOWN	0.0	
1954	31810	53474	PUIPF	3992	UNKNOWN	***	***	UNKNOWN	0.0	
1984	33745	53484	PUTOF	3992	UNKNOWN	***	***	UNKNOWN	0.0	
2041	38941	60879	ZPTVU	3992	UNKNOWN	WZQ-OKECIE	PZL 106 KRUK	UNKNOWN	NaN	

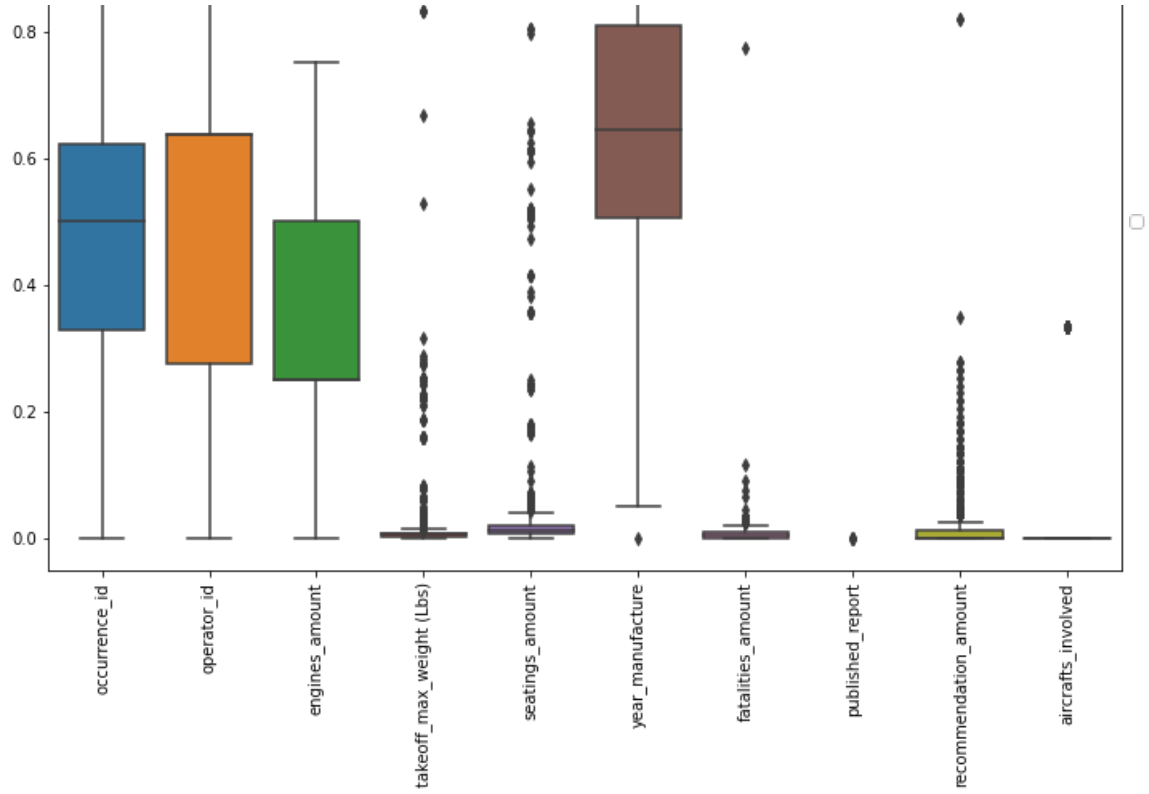
In [0]:

```
# Remover dados que consideramos como Outliers

# Aeronaves sem ANO de Fabricação
df_remove = df_acidentes.loc[df_acidentes['year_manufacture']==0]
#df_bike.loc[df_bike["dteday"].isnull()]
df_acidentes = df_acidentes.drop(df_remove.index)

# Tipos de Aeronaves não definidas
df_remove = df_acidentes.loc[df_acidentes['equipment']=='UNKNOWN']
#df_bike.loc[df_bike["dteday"].isnull()]
df_acidentes = df_acidentes.drop(df_remove.index)
# df_bike.shape
```

In [80]:

```
# Verificando novamente como ficou nosso gráfico de Outliers, e está Ok!!

df_boxplot = df_acidentes
colunas_retiradas = ['registration','equipment','manufacturer','model','engine_type','registration_
country','registration_category','registration_aviation'
,'origin_flight','destination_flight','operation_phase','type_operation','damage_level','extraction
_day_x','classification','type of occurrence','localization','fu'
,'country',
'aerodrome','occurrence_day','time','under_investigation','investigating_command','investigation_st
atus','report_number','publication_day'] #lista que contém as colunas a serem retiradas
df_boxplot = df_boxplot.drop(columns=colunas_retiradas,axis=1)
df_boxplot.values.reshape((-1, 1))


# Ajustando a Escala para exibir o boxplot de análise de outliers
min_max_scaler = MinMaxScaler()
df_scaled_boxplot = pd.DataFrame(min_max_scaler.fit_transform(df_boxplot.iloc[:, 1:-1]))
df_scaled_boxplot.columns = df_boxplot.columns[1:-1]

plt.figure(figsize=(12,8))
fig = sns.boxplot(data=df_scaled_boxplot)
for item in fig.get_xticklabels():
    item.set_rotation(90)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```

```
No handles with labels found to put in legend.
```

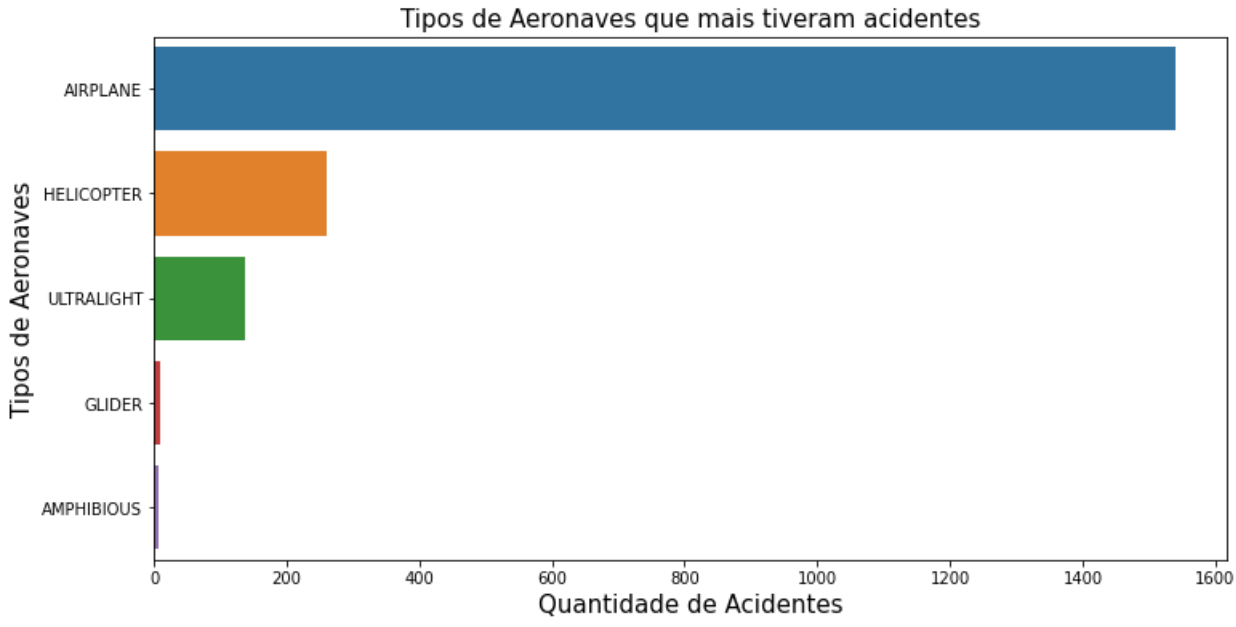


In [81]:

```
# Tipos de Aeronaves que mais tiveram acidentes

fig = plt.figure(figsize=(12,6))
fig = sns.countplot(y='equipment', data=df_acidentes, order = df_acidentes['equipment'].value_count
s().index)
fig.set_title('Tipos de Aeronaves que mais tiveram acidentes',fontsize=15)
fig.set_ylabel('Tipos de Aeronaves',fontsize=15)
fig.set_xlabel('Quantidade de Acidentes',fontsize=15)
```

Out[81]:

```
Text(0.5, 0, 'Quantidade de Acidentes')
```



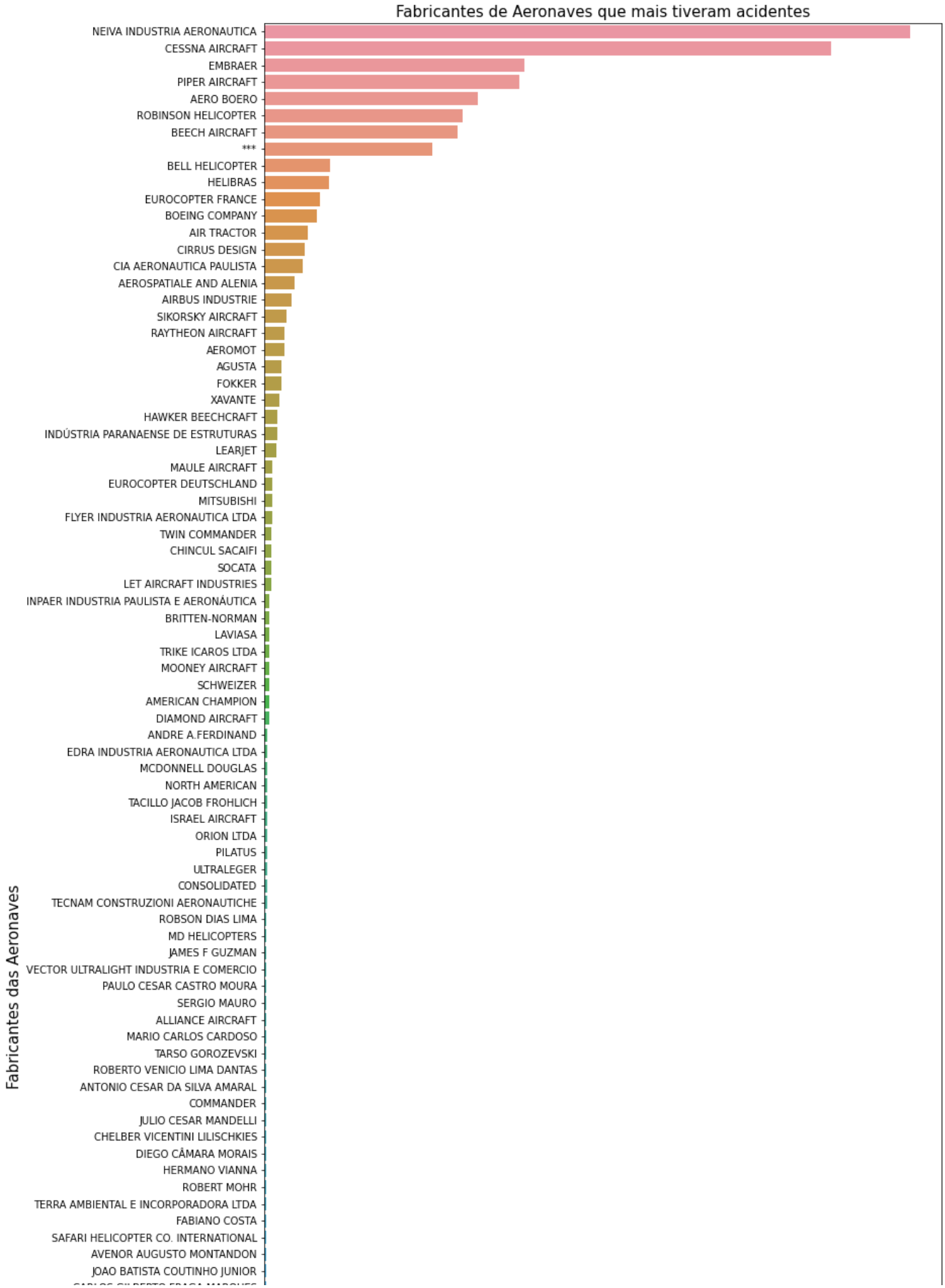
In [82]:

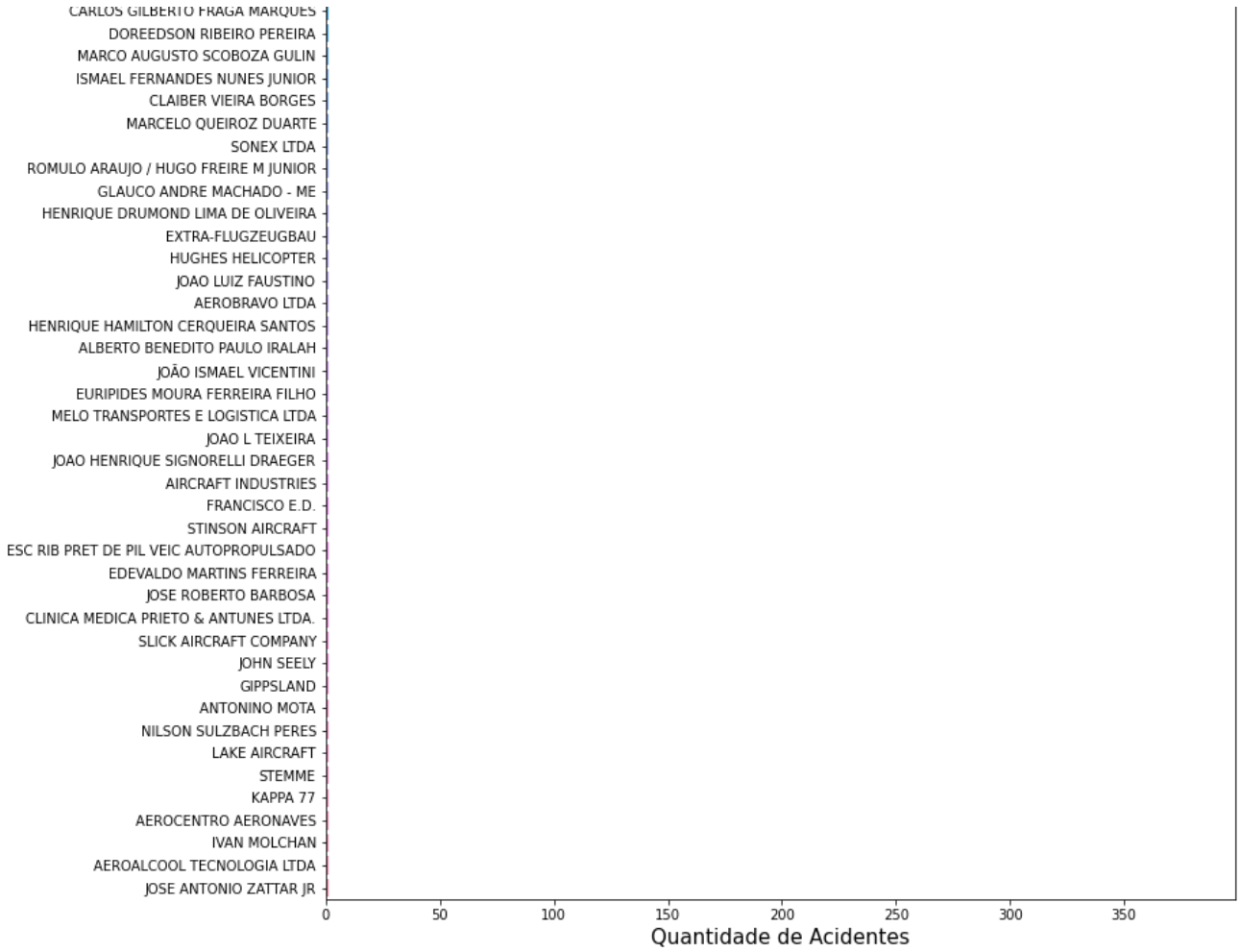
```
# Fabricantes de Aviões que mais tiveram acidentes

fig = plt.figure(figsize=(12,35))
```

```
fig = sns.countplot(y = 'manufacturer', data = df_acidentes, order = df_acidentes['manufacturer'].value_counts().index)
fig.set_title('Fabricantes de Aeronaves que mais tiveram acidentes',fontsize=15)
fig.set_ylabel('Fabricantes das Aeronaves',fontsize=15)
fig.set_xlabel('Quantidade de Acidentes',fontsize=15)
```

Out[82]:

```
Text(0.5, 0, 'Quantidade de Acidentes')
```





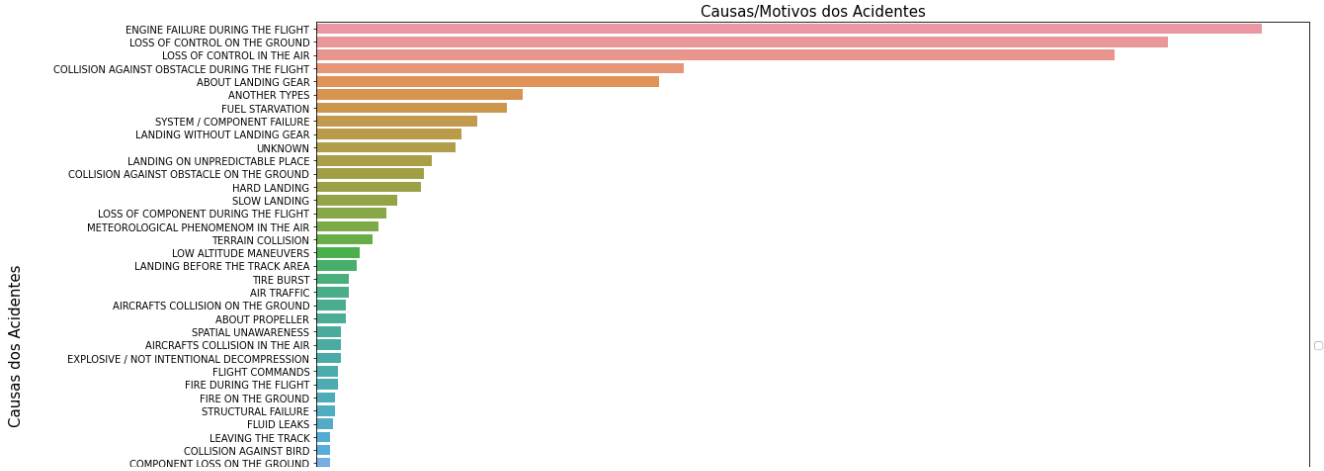
In [83]:

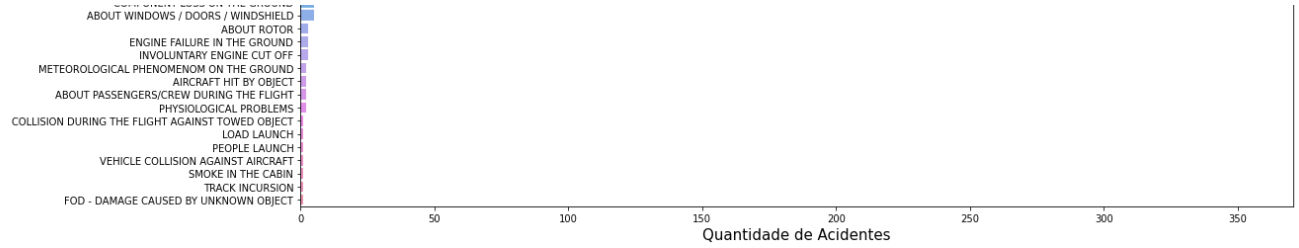
```
# Causa dos Acidentes

fig = plt.figure(figsize=(18,12))
fig = sns.countplot(y = 'type of occurrence', data = df_acidentes, order = df_acidentes['type of occurrence'].value_counts().index)
for item in fig.get_xticklabels():
    item.set_rotation(0)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
fig.set_title('Causas/Motivos dos Acidentes',fontsize=15)
fig.set_ylabel('Causas dos Acidentes',fontsize=15)
fig.set_xlabel('Quantidade de Acidentes',fontsize=15)
```

```
No handles with labels found to put in legend.
```

Out[83]:

```
Text(0.5, 0, 'Quantidade de Acidentes')
```





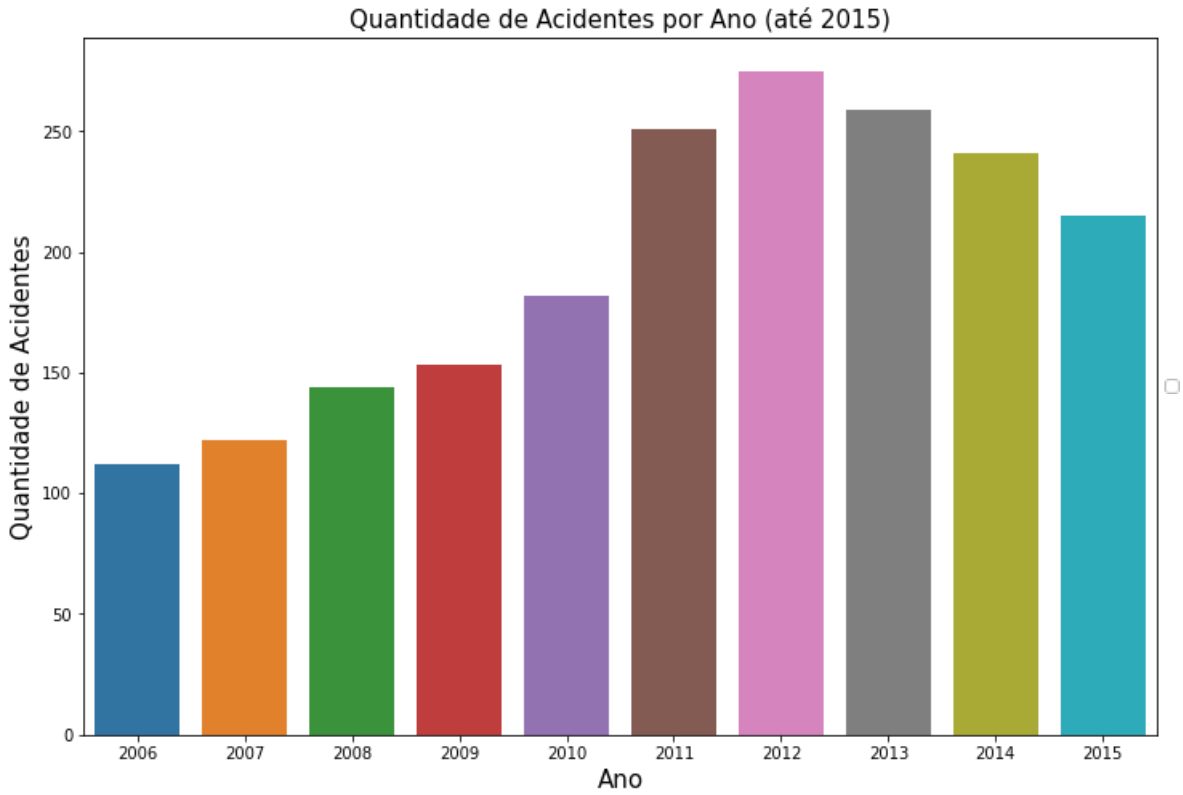
In [84]:

```
# Quantidade de Acidentes por Ano
import datetime

df_acidentes['occurrence_day'] = pd.to_datetime(df_acidentes.occurrence_day)
year = df_acidentes['occurrence_day'].dt.year
order = df_acidentes['occurrence_day'].dt.year.value_counts().index
order = order.sort_values(ascending=True)

fig = plt.figure(figsize=(12,8))
fig = sns.countplot(x = year, data = df_acidentes, order = order)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
fig.set_title('Quantidade de Acidentes por Ano (até 2015)',fontsize=15)
fig.set_ylabel('Quantidade de Acidentes',fontsize=15)
fig.set_xlabel('Ano',fontsize=15)
```

```
No handles with labels found to put in legend.
```

Out[84]:

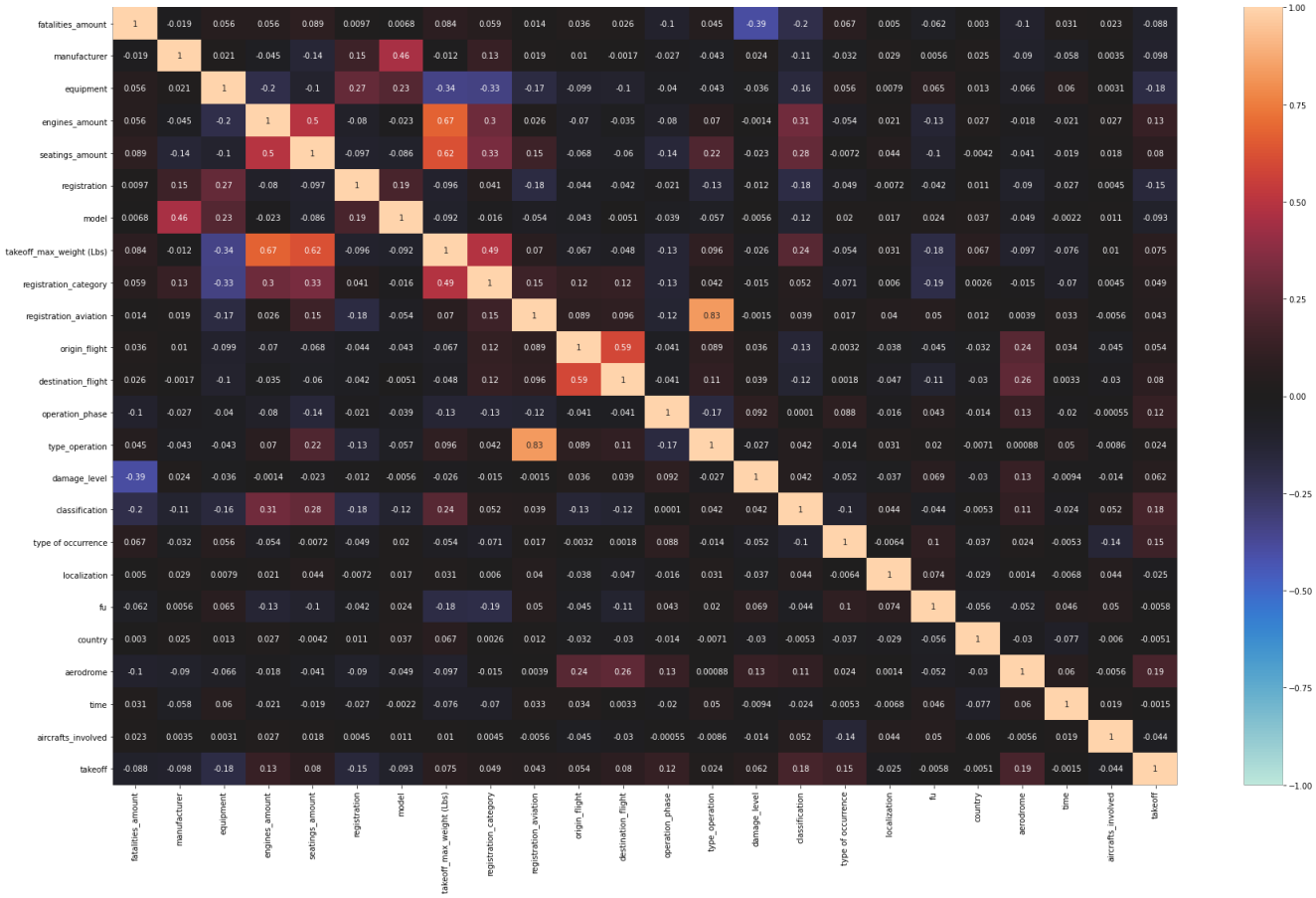
```
Text(0.5, 0, 'Ano')
```



In [85]:

```
# Criando nosso dataframe para análise da matriz de correlação
df_matriz_correlacao = df_acidentes

# Definindo as colunas categoricas
categoricas=["fatalities_amount",  "manufacturer", "equipment","engines_amount","seatings_amount","
registration", "model","takeoff_max_weight (Lbs)",
                                "registration_category","registration_aviation", "origin_flight", '
estination_flight", "operation_phase", "type_operation", "damage_level",
```

```
                                "classification", "type of occurrence", "localization", "fu", "cou
try", "aerodrome", "time", "aircrafts_involved", "takeoff"]

for coluna in categoricas:
  df_matriz_correlacao[coluna]=pd.Categorical(df_matriz_correlacao[coluna]).codes

# Matriz de Correlação

df_matriz_correlacao = df_copia[["fatalities_amount",  "manufacturer", "equipment","engines_amount"
,"seatings_amount","registration", "model","takeoff_max_weight (Lbs)",
                                 "registration_category","registration_aviation", "origin_flight", '
estination_flight", "operation_phase", "type_operation", "damage_level",
                                 "classification", "type of occurrence", "localization", "fu", "cou
try", "aerodrome", "time", "aircrafts_involved", "takeoff"]]


#realizando o plot da matriz de correlação
plt.figure(figsize=(30, 18))
matriz_de_correlação = df_matriz_correlacao.corr()  #construindo a matriz de correlação
sns.heatmap(matriz_de_correlação, annot=True,vmin=-1, vmax=1,center= 0)  #plotando a matriz de corr
elação com o seaborn
plt.show()



# Considerações sobre nossa matriz de correlação
# 1: Temos uma correlação FORTE para TYPE OPERATION x REGISTRATION AVIATION       0.83
# 2: Temos uma correlação MODERADA para ENGINES AMOUNT x TAKEOFF MAX WEIGHT (LSB) 0.67
# 3: Temos uma correlação MODERADA para SEATINGS AMOUNT x TAKEOFF MAX WEIGHT (LSB)0.62
# 4: Temos uma correlação MODERADA para ENGINES AMOUNT x SEATINGS AMOUNT          0.5
# 5: Temos uma correlação FRACA para MANUFACTURER  x MODEL                        0.46
# 6: Temos uma correlação FRACA para FATALITIES AMOUNT  x DAMAGE LEVEL            0.39
# 7: Temos uma correlação FRACA para SEATINGS AMOUNT  x REGISTRATION CATEGORY     0.33

# Temos mais correlações mais que consideramos despresiveis para nossa análise.
```



In [0]:

```
# A principio paramos nossa análise por aqui e seguiremos para outro dataset
```