Fireball NASA

In [2]:

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

# import requests

# URL = "https://ssd-api.jpl.nasa.gov/fireball.api"
# response = requests.get(URL).json()

# with open ("data.json", "w", encoding="utf-8") as f:
#     f.write(json.dumps(response))

with open("data.json", "r", encoding="utf-8") as f:
    extracted = json.loads(f.read())
```

Basic cleanup

In [3]:

```
data = extracted["data"]
df = pd.DataFrame(data)
df.columns = extracted["fields"]

df.drop_duplicates()
# Convert numerical values to float
df = df.astype({'energy': 'float', 'alt': 'float', 'impact-e': 'float', 'vel': '
df = df.rename(columns={"date": "date", "energy": "energy", "alt": "altitude", "
df.dropna(how='all')
df.insert(1, "time", pd.to_datetime(df["date"]).dt.time)
df["date"] = pd.to_datetime(df["date"]).dt.date
df["year"] = pd.to_datetime(df["date"]).dt.year
df["coordinates"] = "(" + df["lat"] + "°" + df["lat-dir"] + ","+ df["lon"] + "°"
df.drop(columns=["lat", "lat-dir", "lon", "lon-dir"], inplace=True)

# Set energy in GJ instead of 10**10
df["energy"] = df["energy"] * 10
```

In [4]:

```
display("Data types", df.dtypes)
display("Data summary", df.describe())
```

```
'Data types'
date              object
time              object
energy           float64
impact_energy    float64
altitude         float64
velocity         float64
year               int32
coordinates       object
dtype: object
'Data summary'
```

	energy	impact_energy	altitude	velocity	year
count	1022.000000	1022.000000	562.000000	329.000000	1022.000000
mean	710.892368	1.170409	36.646441	18.595441	2010.334638
std	11835.514644	14.019468	10.793371	6.153648	8.752532
min	20.000000	0.073000	14.000000	9.800000	1988.000000
25%	31.250000	0.110000	29.600000	14.400000	2003.000000
50%	60.000000	0.190000	35.200000	17.200000	2010.000000
75%	143.000000	0.420000	40.700000	21.100000	2018.000000
max	375000.000000	440.000000	74.300000	49.000000	2025.000000

Columns

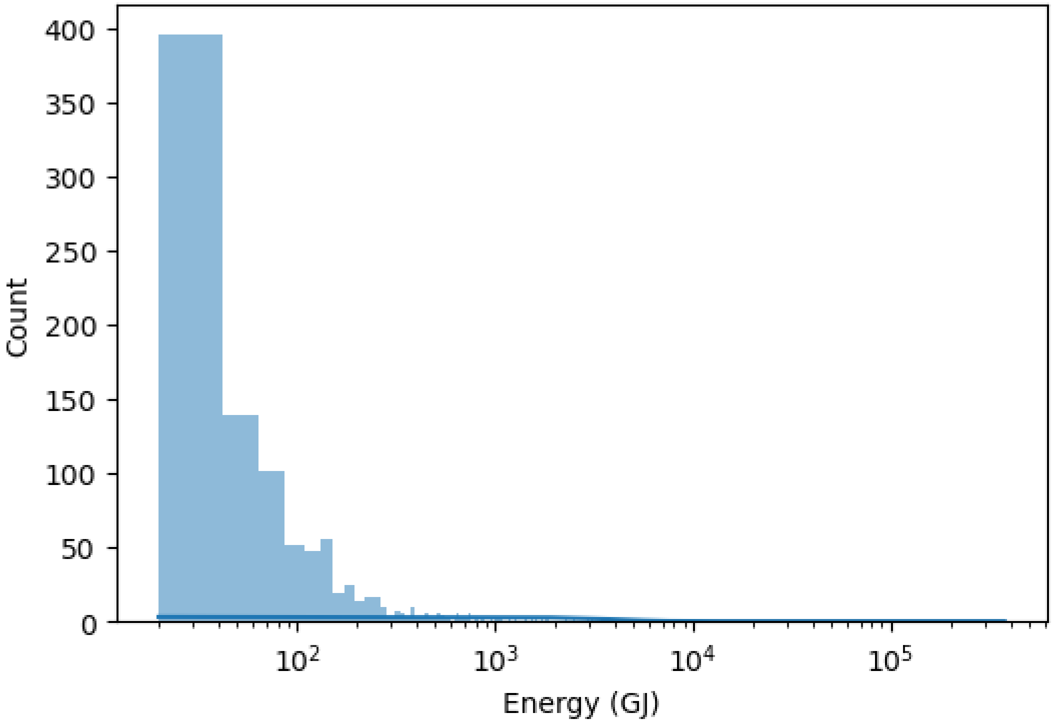
- date: YYYY-MM-DD
- year: YYYY
- time: HH:MM:SS
- energy: Gigajoule (1e+9j)
- impact_energy: kiloton
- altitude: km
- velocity: km/s

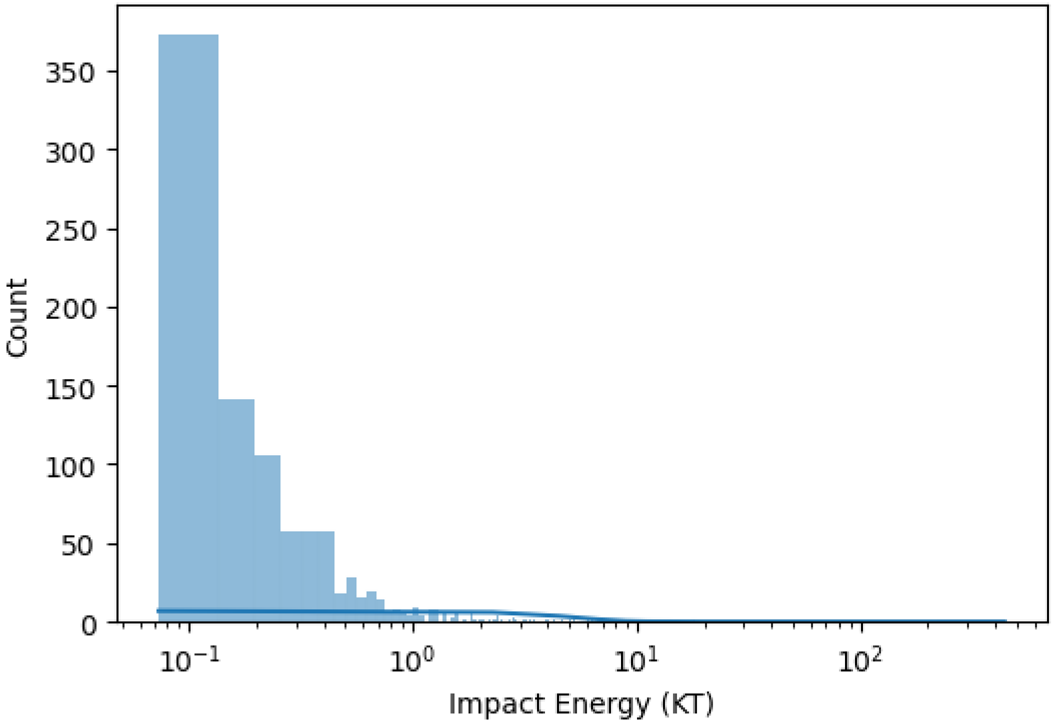
Distributions

Energy (GJ) & Impact Energy (KT) - (Log scale)

In [5]:

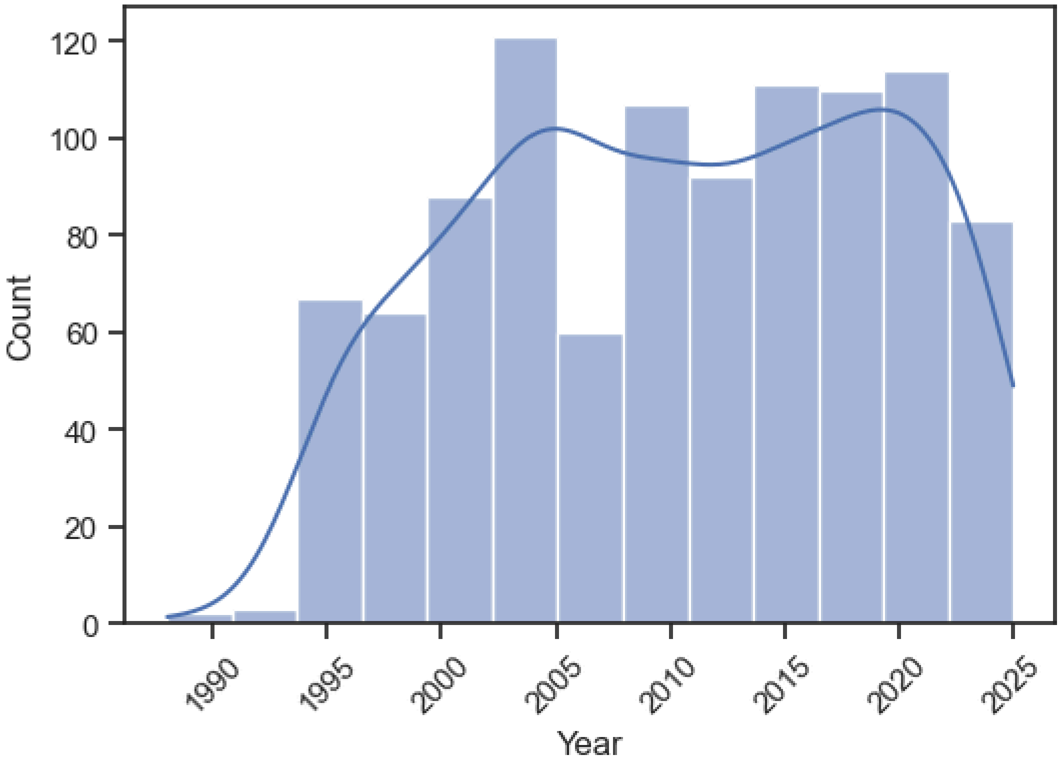
```python
columns = ["energy", "impact_energy"]
labels = ["Energy (GJ)", "Impact Energy (KT)"]

for col, label in zip(columns, labels):
    plt.figure(figsize=[6, 4])
    plt.xlabel(label)
    sns.histplot(x=df[col], kde=True)
    plt.xscale("log")
    plt.show()
```

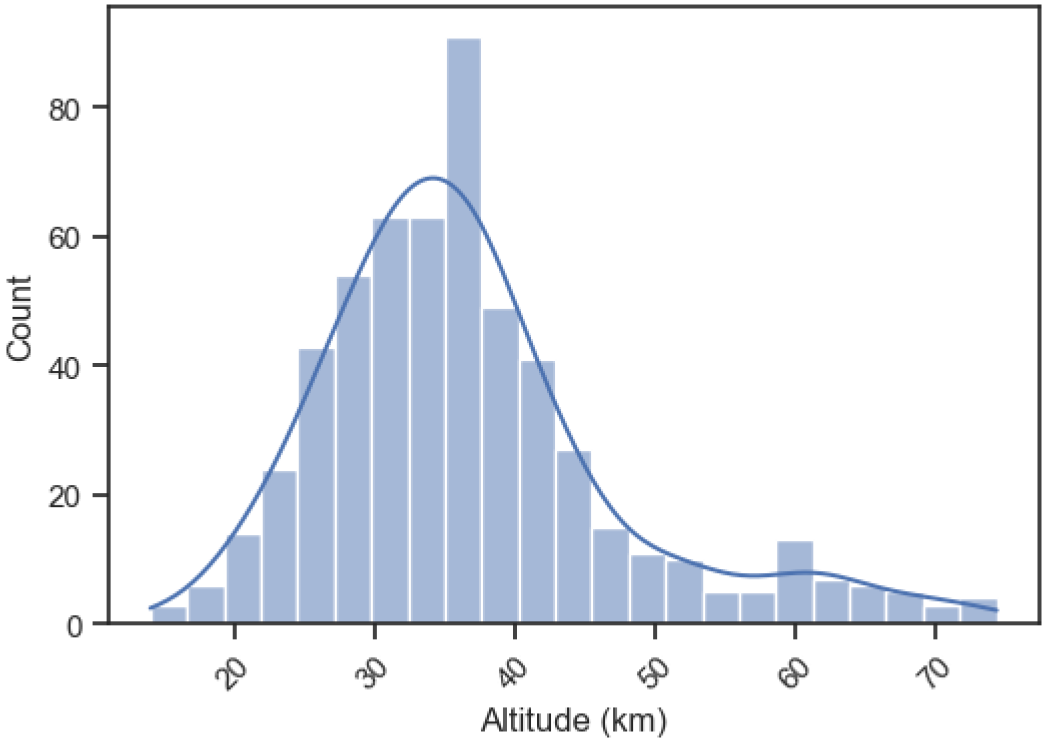


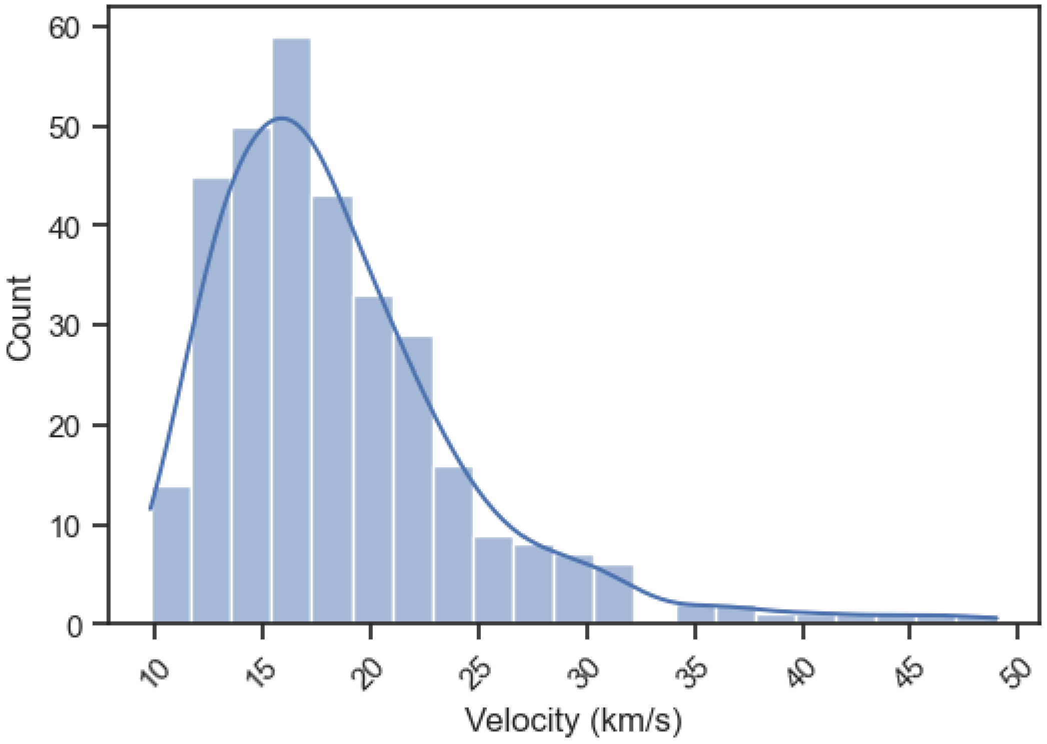


Year, Velocity (km/s), Altitude (km)

In [6]:
```
columns = ["year", "altitude", "velocity"]
labels = ["Year", "Altitude (km)", "Velocity (km/s)"]
sns.set_theme(style="ticks")
for col, label in zip(columns, labels):
    plt.figure(figsize=[6, 4])
    plt.xlabel(label)
    plt.xticks(rotation=45)
    sns.histplot(x=df[col], kde=True)
    plt.show()
```


Count
0
20
40
60
80
100
120
1990
1995
2000
2005
2010
2015
2020
2025
Year


Count
0
20
40
60
80
20
30
40
50
60
70
Altitude (km)



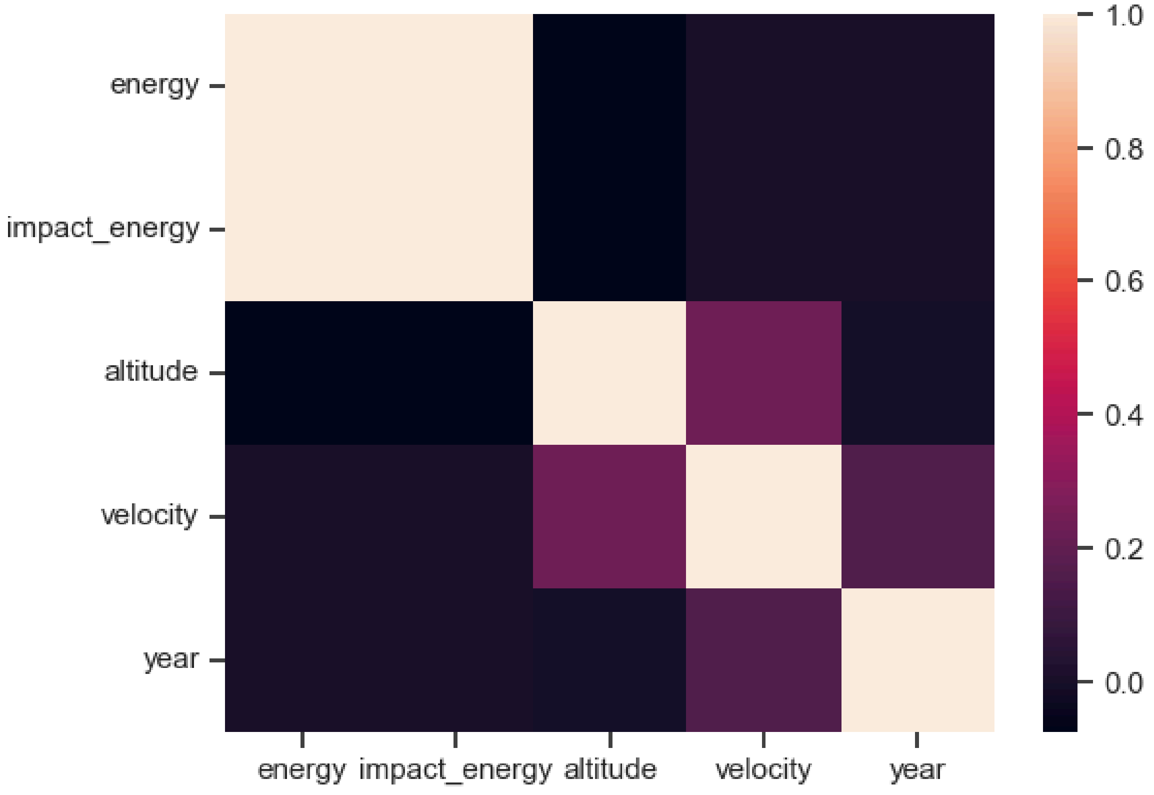
Correlation

In [7]:

```
sns.heatmap(data=df[df.describe().columns].corr())
```

Out[7]:

```
<Axes: >
```


energy
impact_energy
altitude
velocity
year
energy
impact_energy
altitude
velocity
year