Flood videos from city cameras - Dataset Exploratory Data Analysis

In [13]:

```
import pandas as pd, matplotlib.pyplot as plt, seaborn as sns; sns.set()
```

Reload data

In [14]:

```
data_path = '../Dados'
control_path = f'{data_path}/Controle de vídeos/videos_control_19-04.csv'

control = pd.read_csv(control_path)
control['timestamp'] = pd.to_datetime(control['timestamp'])

control.columns
```

Out[14]:

```
Index(['blob_name', 'blob_size', 'bucket_name', 'file_name', 'code',
       'n_folders', 'timestamp', 'folder_structure'],
      dtype='object')
```

Exploratory data analsysis

In [15]:

```
def event_info(blob_name):
    "returns event type and event id given `blob_name` matching the folder structure `{source}/{type}/{event}/{c
ode}`."
    info = blob_name.split('/')
    return '/'.join(info[:2]), info[2]
```

Cameras already recorded

In [16]:

```
print('Cameras recorded:', control['code'].nunique())
```

```
Cameras recorded: 2426
```

Folder structure count

In [17]:

```
control[['n_folders', 'folder_structure']].value_counts().sort_index()
```

Out[17]:

```
n_folders  folder_structure
3          {type}/{code}                                21182
4          polygons/{type}/{code}                          69
           {type}/{subtype}/{code}                        476
5          polygons/{type}/{polygon}/{code}             11187
           {source}/{type}/{event}/{code}                9914
6          polygons/{source}/{type}/{polygon}/{code}      462
Name: count, dtype: int64
```

Total events per type count

In [18]:

```
folder_structure_msk = control['folder_structure'] =='{source}/{type}/{event}/{code}'
event_cnt = control[folder_structure_msk]['blob_name'].apply(event_info).tolist()
event_cnt = pd.DataFrame(event_cnt, columns=['event_type', 'event_id'])

event_type_cnt = event_cnt.drop_duplicates()['event_type'].value_counts()

print(f'\n{event_type_cnt}\n')
```

```
event_type
waze/flood            568
comando/bolsão         47
comando/alagamento      7
comando/lâmina          5
rivers/manual           2
Name: count, dtype: int64
```

Videos per camera

In [19]:

```
print(f'Videos per camera:\n\n{control["code"].value_counts()}')
```

```
Videos per camera:

code
1487    529
278     526
3323    438
1635    254
1636    251
       ... 
195       1
1944      1
1943      1
1931      1
563       1
Name: count, Length: 2426, dtype: int64
```

Mega Bytes (MB) per câmera

In [20]:

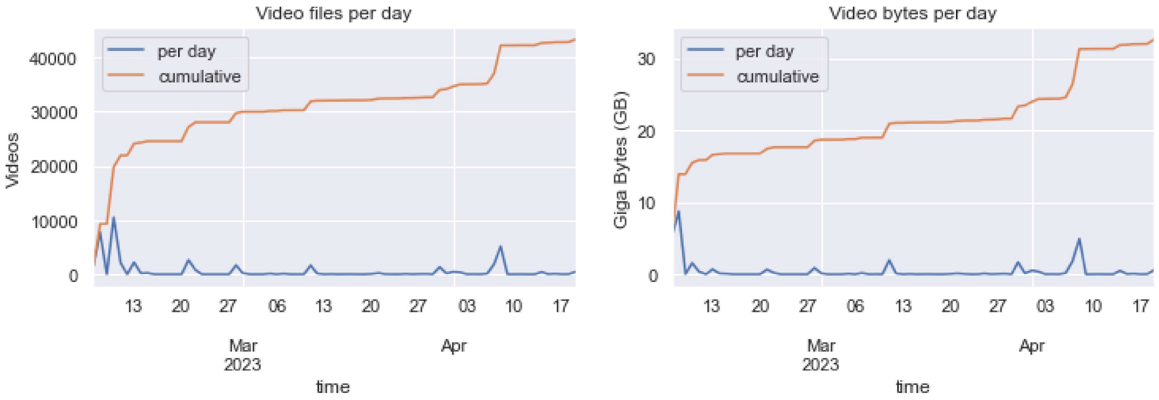
```
print(f'Total MB per câmera:\n\n{control.groupby("code")["blob_size"].sum().sort_values(ascending=False) / 1e
6}')
```

```
Total MB per câmera:

code
278     996.349552
1487    492.780141
482     406.056572
3323    380.590463
267     375.663969
           ...    
2272      0.016630
146       0.016618
2271      0.016520
422       0.016491
1025      0.015996
Name: blob_size, Length: 2426, dtype: float64
```

Total videos and bytes per day and cumulative

In [21]:

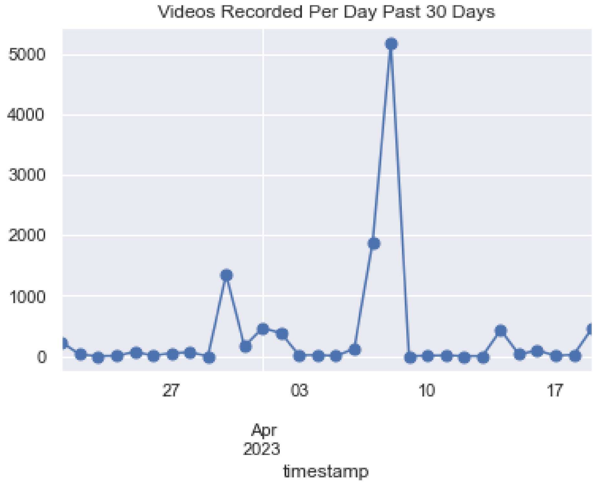
```
time_cnt = control.groupby('timestamp').count()['blob_name']
time_size_cnt = control.groupby('timestamp')['blob_size'].sum()

fig, axs = plt.subplots(1, 2, figsize=(12, 3))

time_cnt.resample('1D').sum().plot(ax=axs[0])
time_cnt.resample('1D').sum().cumsum().plot(ax=axs[0])
axs[0].set(title='Video files per day', ylabel='Videos', xlabel='time')
axs[0].legend(['per day', 'cumulative'])

(time_size_cnt.resample('1D').sum() / 1e9).plot(ax=axs[1])
(time_size_cnt.resample('1D').sum().cumsum() / 1e9).plot(ax=axs[1])
axs[1].set(title='Video bytes per day', ylabel='Giga Bytes (GB)', xlabel='time')
axs[1].legend(['per day', 'cumulative'])
plt.show()
```



Recently recorded · Past 30 days

In [22]:

```
days = 30

ax = time_cnt.resample('1D').sum().tail(days).plot(marker='o', ms=7, title='Videos Recorded Per Day Past 30 Day
s')
```



Build videos datetime index

In [23]:

```
ts = control.set_index('timestamp') # datetime index
```

Events per day

Extract event videos dataset

In [24]:

```
folder_structure_msk = ts['folder_structure'] =='{source}/{type}/{event}/{code}'

ts_events = ts[folder_structure_msk]

ts_events = pd.DataFrame(
    ts_events['blob_name'].map(event_info).tolist(),
    index=ts_events.index, columns=['event_type', 'event_id']
)

ts_events.head()
```

Out[24]:

	event_type	event_id
timestamp		
2023-03-30 19:55:06	comando/alagamento	93898
2023-03-30 20:00:09	comando/alagamento	93898
2023-03-30 20:05:10	comando/alagamento	93898
2023-03-30 20:10:13	comando/alagamento	93898
2023-03-30 20:15:12	comando/alagamento	93898

Events per type per day count

In [25]:

```
freq = '1D'

event_type_cnt = []
for event_type in ts_events['event_type'].unique():
    event_type_cnt.append(ts_events[ts_events['event_type']==event_type].resample(freq).nunique()['event_id'].re
name(event_type))

event_type_cnt = pd.concat(event_type_cnt, axis=1).fillna(0.0).drop('rivers/manual', axis=1)

event_type_cnt.head()
```

Out[25]:

	comando/alagamento	comando/bolsão	comando/lâmina	waze/flood
timestamp				
2023-03-05	0.0	0.0	0.0	3
2023-03-06	0.0	0.0	0.0	0
2023-03-07	0.0	2.0	0.0	12
2023-03-08	0.0	0.0	0.0	0
2023-03-09	0.0	0.0	0.0	2

Events' videos per type per day count

In [26]:

```
freq = '1D'

event_videos_type_cnt = []
for event_type in ts_events['event_type'].unique():
    event_videos_type_cnt.append(ts_events[ts_events['event_type']==event_type].resample(freq).count()['event_id'].rename(event_type))

event_videos_type_cnt = pd.concat(event_videos_type_cnt, axis=1).fillna(0.0).drop('rivers/manual', axis=1)

event_videos_type_cnt.head()
```

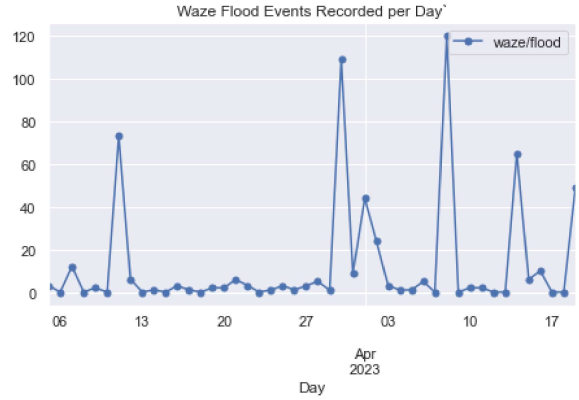
Out[26]:

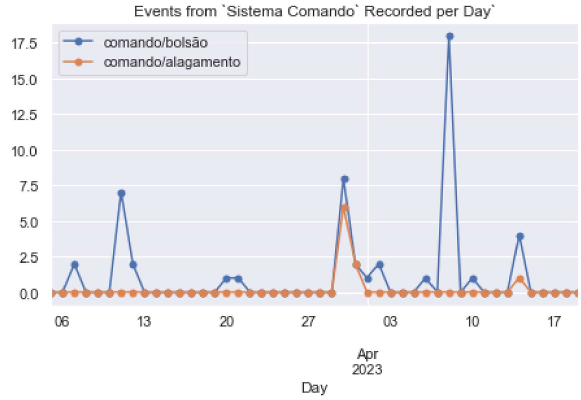
	comando/alagamento	**comando/bolsão**	**comando/lâmina**	**waze/flood**
timestamp				
2023-03-05	0.0	0.0	0.0	7
2023-03-06	0.0	0.0	0.0	0
2023-03-07	0.0	20.0	0.0	108
2023-03-08	0.0	0.0	0.0	0
2023-03-09	0.0	0.0	0.0	10

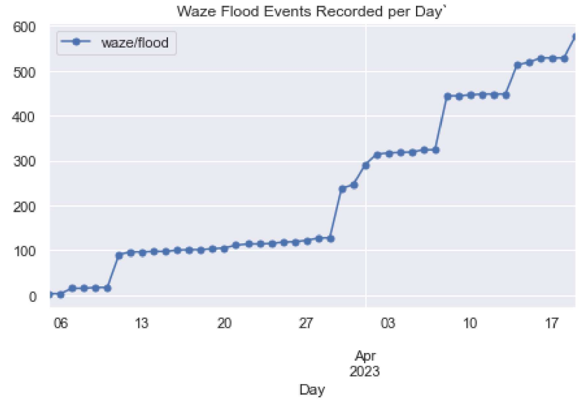
Events and videos per type per day · Line Chart

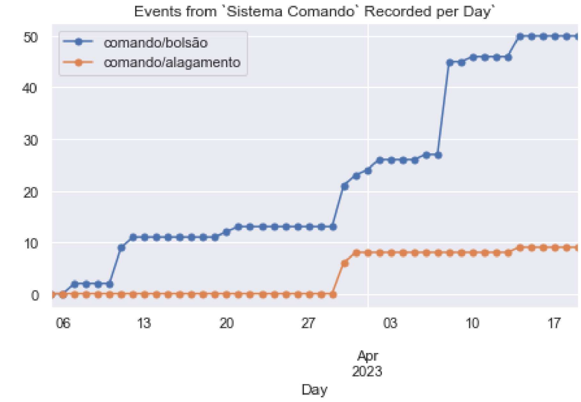
In [27]:

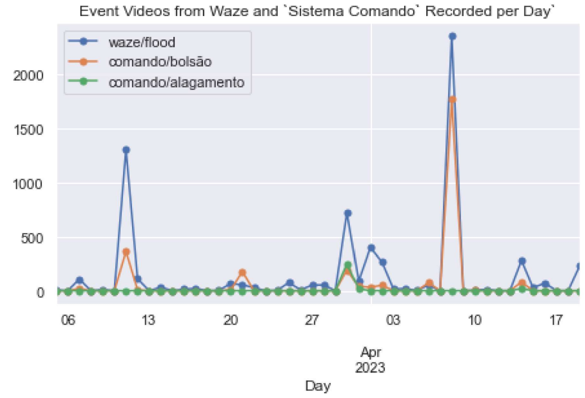
```
fig, axs = plt.subplots(1, 2, figsize=(16, 4))
event_type_cnt[['waze/flood']].plot(
    ax=axs[0], marker='o', ms=5, xlabel='Day',
    title='Waze Flood Events Recorded per Day`'
)
event_type_cnt[['comando/bolsão', 'comando/alagamento']].plot(
    ax=axs[1], marker='o', ms=5, xlabel='Day',
    title='Events from `Sistema Comando` Recorded per Day`'
)
plt.show()

fig, axs = plt.subplots(1, 2, figsize=(16, 4))
event_type_cnt[['waze/flood']].cumsum().plot(
    ax=axs[0], marker='o', ms=5, xlabel='Day',
    title='Waze Flood Events Recorded per Day`'
)
event_type_cnt[['comando/bolsão', 'comando/alagamento']].cumsum().plot(
    ax=axs[1], marker='o', ms=5, xlabel='Day',
    title='Events from `Sistema Comando` Recorded per Day`'
)
plt.show()

fig, axs = plt.subplots(1, 2, figsize=(16, 4))
event_videos_type_cnt[['waze/flood', 'comando/bolsão', 'comando/alagamento']].plot(
    ax=axs[0], marker='o', ms=5, xlabel='Day',
    title='Event Videos from Waze and `Sistema Comando` Recorded per Day`'
)
event_videos_type_cnt[['waze/flood', 'comando/bolsão', 'comando/alagamento']].cumsum().plot(
    ax=axs[1], marker='o', ms=5, xlabel='Day',
    title='Event Videos from Waze and `Sistema Comando` Recorded per Day`'
)
plt.show()
```

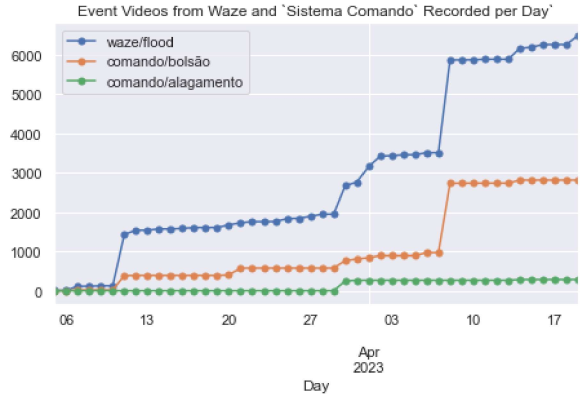












Events and videos count per period

In [29]:

```
def event_info(blob_name):
    "returns event type and event id given `blob_name` matching the folder structure `{source}/{type}/{event}/{c
ode}`."
    info = blob_name.split('/')
    return ['/'.join(info[:2]), info[2]]
```

2023-04-07 and 2023-04-08

Events count

In [30]:

```
ts_cut = ts[(ts.index >= '2023-04-07') & (ts.index < '2023-04-09')]

folder_structure_msk = ts_cut['folder_structure'] =='{source}/{type}/{event}/{code}'

event_cnt = ts_cut.loc[folder_structure_msk]['blob_name'].apply(event_info).tolist()
event_cnt = pd.DataFrame(event_cnt, columns=['event_type', 'event_id'])

video_cnt = event_cnt['event_type'].value_counts().rename('video_count')
event_cnt = event_cnt.drop_duplicates()['event_type'].value_counts().rename('event_count')

event_cnts = pd.concat([event_cnt, video_cnt], axis=1)

print(f'\n{event_cnts}\n')
```

```
                event_count  video_count
event_type
waze/flood              120         2353
comando/bolsão           18         1764
comando/lâmina            2          126
rivers/manual             2          177
```

2023-03-30 and 2023-03-31

Events count

In [31]:

```
ts_cut = ts[(ts.index >= '2023-03-30') & (ts.index <= '2023-04-01')]
folder_structure_msk = ts_cut['folder_structure'] =='{source}/{type}/{event}/{code}'

event_cnt = ts_cut[folder_structure_msk]['blob_name'].apply(event_info).tolist()
event_cnt = pd.DataFrame(event_cnt, columns=['event_type', 'event_id'])

video_cnt = event_cnt['event_type'].value_counts().rename('video_count')
event_cnt = event_cnt.drop_duplicates()['event_type'].value_counts().rename('event_count')

event_cnts = pd.concat([event_cnt, video_cnt], axis=1)

print(f'\n{event_cnts}\n')
```

```
                    event_count  video_count
event_type
waze/flood                  114          810
comando/bolsão                9          230
comando/alagamento            6          266
```

2023-04-18 and 2023-04-19

Events count

In [32]:

```
ts_cut = ts[(ts.index >= '2023-04-19') & (ts.index <= '2023-04-20')]
folder_structure_msk = ts_cut['folder_structure'] =='{source}/{type}/{event}/{code}'

event_cnt = ts_cut[folder_structure_msk]['blob_name'].apply(event_info).tolist()
event_cnt = pd.DataFrame(event_cnt, columns=['event_type', 'event_id'])

video_cnt = event_cnt['event_type'].value_counts().rename('video_count')
event_cnt = event_cnt.drop_duplicates()['event_type'].value_counts().rename('event_count')

event_cnts = pd.concat([event_cnt, video_cnt], axis=1)

print(f'\n{event_cnts}\n')
```

```
            event_count  video_count
event_type
waze/flood           49          229
```