In order to provide insights into the relationship between file size and download behavior, we create a line chart of maximum and minimum request sizes for each agent. The data was first filtered to exclude agents with more than 10,000 downloads and files larger than 16 MB. This was done to eliminate extreme outliers that could distort the overall trends in the data. In order to more clearly display the trends, the values were rounded to the nearest integer. Finally, the agents were sorted by minimum value and then by maximum value to enable comparison of the request sizes for each agent.

In [1]:

```
import pandas as pd
import numpy as np
from scipy import stats
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from tqdm.notebook import tqdm
import re
pd.set_option('max_columns', None)
```

In [2]:

```
df = pd.read_csv('data.csv', index_col=0)
df.shape
```

Out[2]:

```
(6643221, 4)
```

In [3]:

```
df_temp = df[df['bytes_returned'] > 16*pow(1024,2)]
exclude_cid = set(df_temp['cid'].unique())
len(exclude_cid)
```

Out[3]:

```
3321
```

In [4]:

```
df_temp = df[['agent','timestamp']].groupby(['agent']).count()
df_temp = df_temp.rename(columns={"timestamp": "count"})
df_temp = df_temp[df_temp['count']>10000]
df_temp = df_temp.reset_index()
exclude_agent = set(df_temp['agent'].unique())
len(exclude_agent)
```

Out[4]:

```
73
```

In [5]:

```
df1 = df[(~df['agent'].isin(exclude_agent)) & (~df['cid'].isin(exclude_cid))]
df1.shape
```

Out[5]:

```
(4008852, 4)
```

In [6]:

```
df1.shape[0]/df.shape[0]
```

Out[6]:

```
0.6034500432847258
```

In [7]:

```
df1 = df1[['agent','bytes_returned']]
df1['bytes_returned'] = df1['bytes_returned']/pow(1024,2)
```

In [8]:

```
def q10(x):
    return x.quantile(0.1)

def q90(x):
    return x.quantile(0.9)

df2 = df1.groupby(['agent']).agg(['min','median','max','mean'])
df2.columns = df2.columns.get_level_values(1)
df2 = df2.round(0).astype(int)
df2 = df2.reset_index()
df2.head()
```

Out[8]:

	agent	min	median	max	mean
0	AVProMobileVideo/6.1.7.39280 (Linux;Android 10...	6	6	6	6
1	AccompanyBot	0	0	0	0
2	ActionExtension/3 CFNetwork/1220.1 Darwin/20.3.0	0	0	0	0
3	AirPlay/2.0 (App/30.172.0) MFi_AirPlay_Device ...	0	1	3	1
4	AirPlay/2.0 (App/30.172.0) MFi_AirPlay_Device ...	0	1	4	1

In [9]:

```
# df2[['max','min','mean']] = df2[['max','min','mean']].astype(int)
df2['gap'] = df2['max'] - df2['min']
df2 = df2.sort_values(by=['min','max'])

df2 = df2.reset_index()
df2 = df2.drop(['index'],axis=1)
df2 = df2.reset_index()
df2['idx_percentage'] = df2['index']/df2.shape[0]
df2.head()
```

Out[9]:

	index	agent	min	median	max	mean	gap	idx_percentage
0	0	AccompanyBot	0	0	0	0	0	0.000000
1	1	ActionExtension/3 CFNetwork/1220.1 Darwin/20.3.0	0	0	0	0	0	0.000047
2	2	Aloha/8 CFNetwork/1240.0.4 Darwin/20.6.0	0	0	0	0	0	0.000095
3	3	AlphaWallet/417 CFNetwork/1240.0.4 Darwin/20.6.0	0	0	0	0	0	0.000142
4	4	AlphaWallet/417 CFNetwork/1327.0.4 Darwin/21.3.0	0	0	0	0	0	0.000190

In [10]:

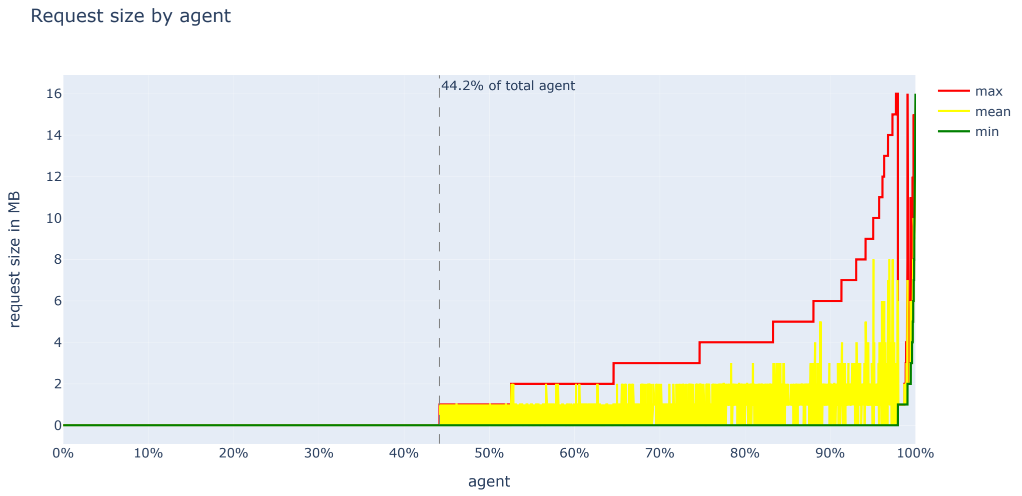
```
df_temp = df2[(df2['min']==0) & (df2['gap']==0)]
p1 = df_temp.shape[0]/df2.shape[0]
p1
```

Out[10]:

```
0.4414653822426802
```

In [11]:

```
fig = go.Figure()

# Create and style traces
fig.add_trace(go.Scatter(x=df2['idx_percentage'], y=df2['max'], name='max', line=dict(color='red')))
fig.add_trace(go.Scatter(x=df2['idx_percentage'], y=df2['mean'], name='mean', line=dict(color='yellow')))
fig.add_trace(go.Scatter(x=df2['idx_percentage'], y=df2['min'], name='min', line=dict(color='green')))

# Edit the layout
fig.update_layout(title='Request size by agent',
                   xaxis_title='agent',
                   yaxis_title='request size in MB')

fig.add_vline(x=p1, line_width=1, line_dash="dash", line_color="grey",
annotation_text="44.2% of total agent", annotation_position="top right")

fig.update_xaxes(tickformat = ',.0%')

# fig.update_xaxes(visible=True, showticklabels=False)
# fig.update_yaxes(visible=True, showticklabels=True)

fig.show()
```


Request size by agent
44.2% of total agent
max
mean
min
request size in MB
agent