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
pd.set_option('max_columns', None)
```

In [2]:
```
df = pd.read_csv('data.csv', index_col=0)
df.shape
```

Out[2]:
```
(6643221, 4)
```

Average request size per minute

In [3]:
```
# create time series array
idx = pd.to_datetime(df['timestamp'])
arr = df['bytes_returned'].astype(int).array
s = pd.Series(arr, index=idx)
# downsample into 1 minute bins
s = s.resample('1T').mean()

# create dataframe
df1 = pd.DataFrame(s)
df1 = df1.reset_index()
df1 = df1.rename(columns={0: 'request_size'})
df1['request_size'] = df1['request_size']/pow(1024,2)
df1.head()
```

Out[3]:

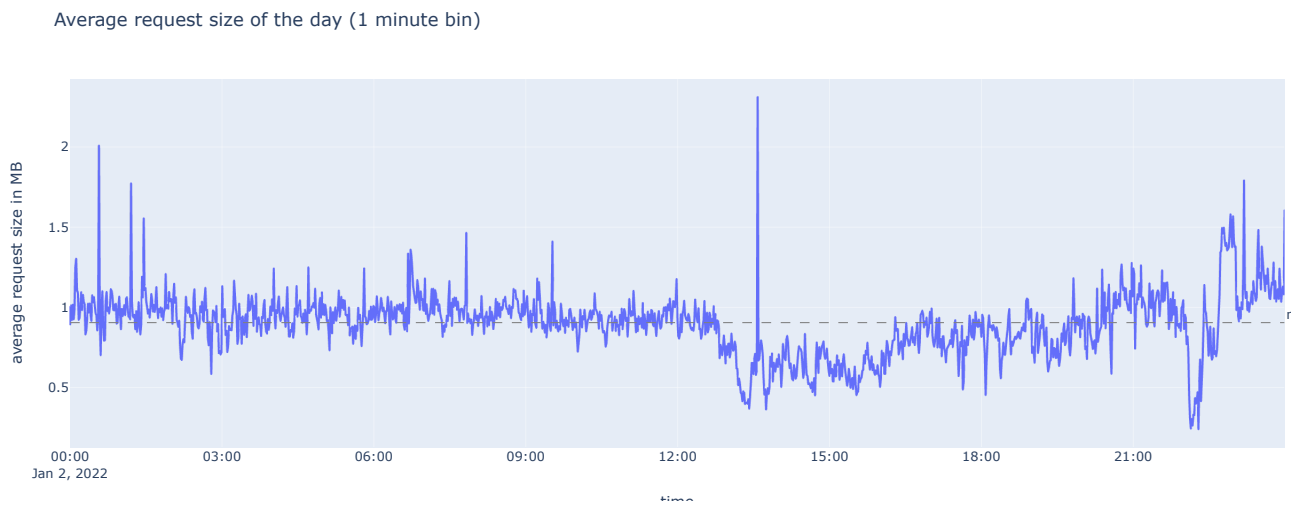
	timestamp	request_size
0	2022-01-02 00:00:00+00:00	1.002515
1	2022-01-02 00:01:00+00:00	0.896166
2	2022-01-02 00:02:00+00:00	1.013101
3	2022-01-02 00:03:00+00:00	0.922915
4	2022-01-02 00:04:00+00:00	1.017610

In [4]:
```
fig = px.line(df1, x='timestamp', y="request_size", title='Average request size of the day (1 minute bin)')

average = df['bytes_returned'].mean()/pow(1024,2)
fig.add_hline(y=average, line_width=1, line_dash="dash", line_color="grey",
              annotation_text="mean<br>{0:.1f}".format(average),
              annotation_position="right")

fig.update_xaxes(title="time")
fig.update_yaxes(title="average request size in MB")

fig.show()
```



Number of requests per minute

In [5]:
```
# create time series array
idx = pd.to_datetime(df['timestamp'])
arr = df['bytes_returned'].astype(int).array
s = pd.Series(arr, index=idx)
# downsample into 1 minute bins
s = s.resample('1T').count()

# create dataframe
df2 = pd.DataFrame(s)
df2 = df2.reset_index()
df2 = df2.rename(columns={0: 'request_count'})
df2.head()
```

Out[5]:

	timestamp	request_count
0	2022-01-02 00:00:00+00:00	1485
1	2022-01-02 00:01:00+00:00	4389
2	2022-01-02 00:02:00+00:00	4398
3	2022-01-02 00:03:00+00:00	4768

	timestamp	request_count
4	2022-01-02 00:04:00+00:00	4880

In [6]:
```
df['bytes_returned'].count()
```

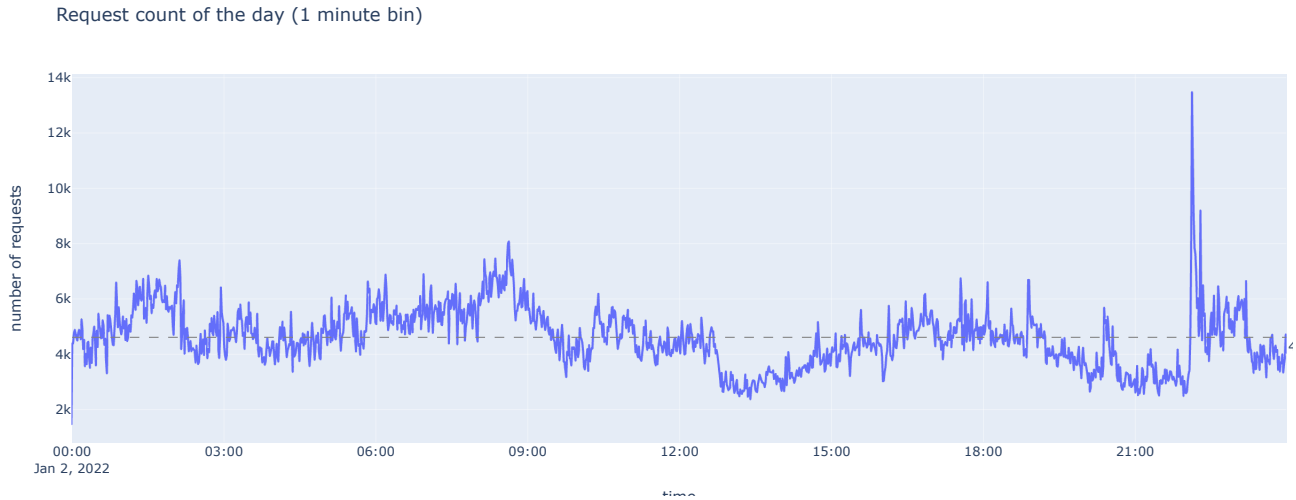
Out[6]:
```
6643221
```

In [7]:
```
fig = px.line(df2, x='timestamp', y="request_count", title='Request count of the day (1 minute bin)')

average = df2['request_count'].mean()
fig.add_hline(y=average, line_width=1, line_dash="dash", line_color="grey",
              annotation_text="mean<br>{0:.1f}".format(average),
              annotation_position="right")

fig.update_xaxes(title="time")
fig.update_yaxes(title="number of requests")

fig.show()
```



Traffic

In [8]:
```
# create time series array
idx = pd.to_datetime(df['timestamp'])
arr = df['bytes_returned'].astype(int).array
s = pd.Series(arr, index=idx)
# downsample into 1 minute bins
s = s.resample('1T').sum()

# create dataframe
df3 = pd.DataFrame(s)
df3 = df3.reset_index()
df3 = df3.rename(columns={0: 'request_size'})
# request size in GB
df3['request_size'] = df3['request_size']/pow(1024,3)

# calculate cumulative sum
df3['request_size_cumulative'] = df3['request_size'].cumsum()
df3['request_size_cumulative'] = df3['request_size_cumulative']/1024
# calculate percentage
total = df3.iloc[-1]['request_size_cumulative']
df3['percentage'] = df3['request_size_cumulative'] / total
df3.head()
```

Out[8]:

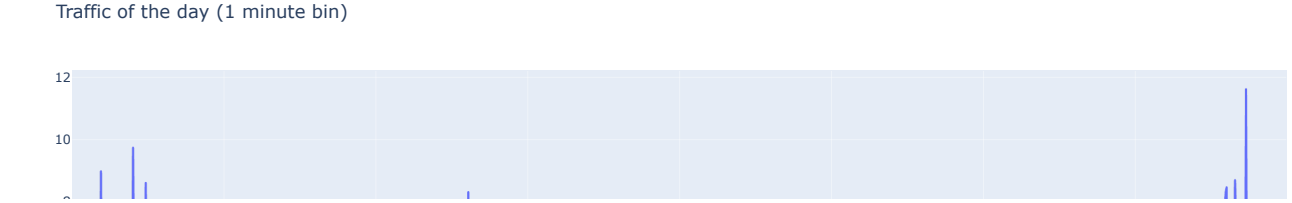
	timestamp	request_size	request_size_cumulative	percentage
0	2022-01-02 00:00:00+00:00	1.453842	0.001420	0.000248
1	2022-01-02 00:01:00+00:00	3.841085	0.005171	0.000902
2	2022-01-02 00:02:00+00:00	4.351190	0.009420	0.001643
3	2022-01-02 00:03:00+00:00	4.297321	0.013617	0.002375
4	2022-01-02 00:04:00+00:00	4.849550	0.018353	0.003201

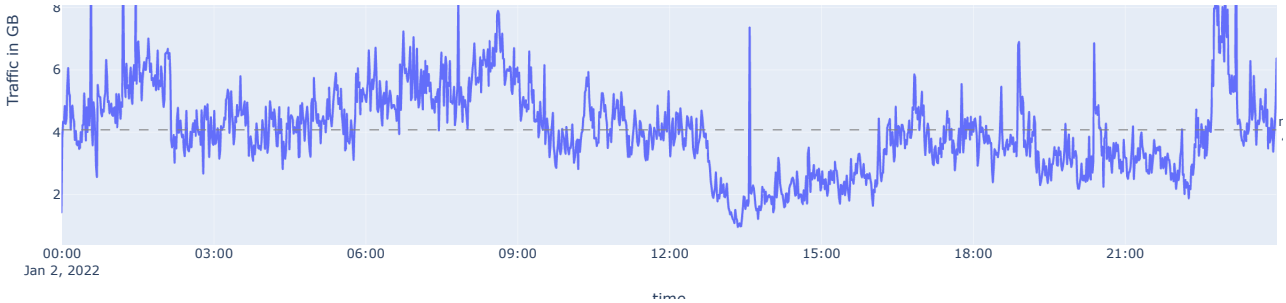
In [9]:
```
fig = px.line(df3, x='timestamp', y="request_size", title='Traffic of the day (1 minute bin)')

average = df3['request_size'].mean()
fig.add_hline(y=average, line_width=1, line_dash="dash", line_color="grey",
              annotation_text="mean<br>{0:.2f}".format(average),
              annotation_position="right")

fig.update_xaxes(title="time")
fig.update_yaxes(title="Traffic in GB")

fig.show()
```





In [10]:

```
fig = px.line(df3, x='timestamp', y="request_size_cumulative", title='Cumulative traffic of the day')

fig.update_xaxes(title="time")
fig.update_yaxes(title="cumulative traffic in TB")

fig.show()
```