Cohort Analysis

Cohort Analysis serves as a method in data analysis aimed at understanding the behaviors and attributes of distinct user or customer groups over time.

Methodology

Cohort analysis offers significant value to businesses by providing insights into user behavior with greater granularity and practicality. Below outlines the procedural steps for conducting Cohort Analysis:

1. Define cohorts based on specific attributes or events. For instance, in an e-commerce setting, cohorts may be formed based on the month of a user's initial purchase.
2. Gather pertinent data for analysis purposes.
3. Determine the time intervals to be examined.
4. Organize users into cohorts according to the predetermined attributes or events.
5. Select the key performance indicators (KPIs) for analysis.
6. Compute the chosen metrics for each cohort across the designated timeframes.
7. Develop visualizations to effectively communicate your findings.

In [1]:
```
# Importing necessary libraries

import pandas as pd
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
import plotly
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_dark"
```

In [2]:
```
# Reading CSV file

df = pd.read_csv(".//Dataset//cohorts.csv")
```

In [3]:
```
# Date forma Conversion

df["Date"]=pd.to_datetime(df["Date"],format = "%d/%m/%Y")
```

In [4]:
```
# Checking for null values or missing values

df.isnull().sum()
```

Out[4]:
```
Date              0
New users         0
Returning users   0
Duration Day 1    0
Duration Day 7    0
dtype: int64
```

In [5]:
```
df.describe()
```

Out[5]:

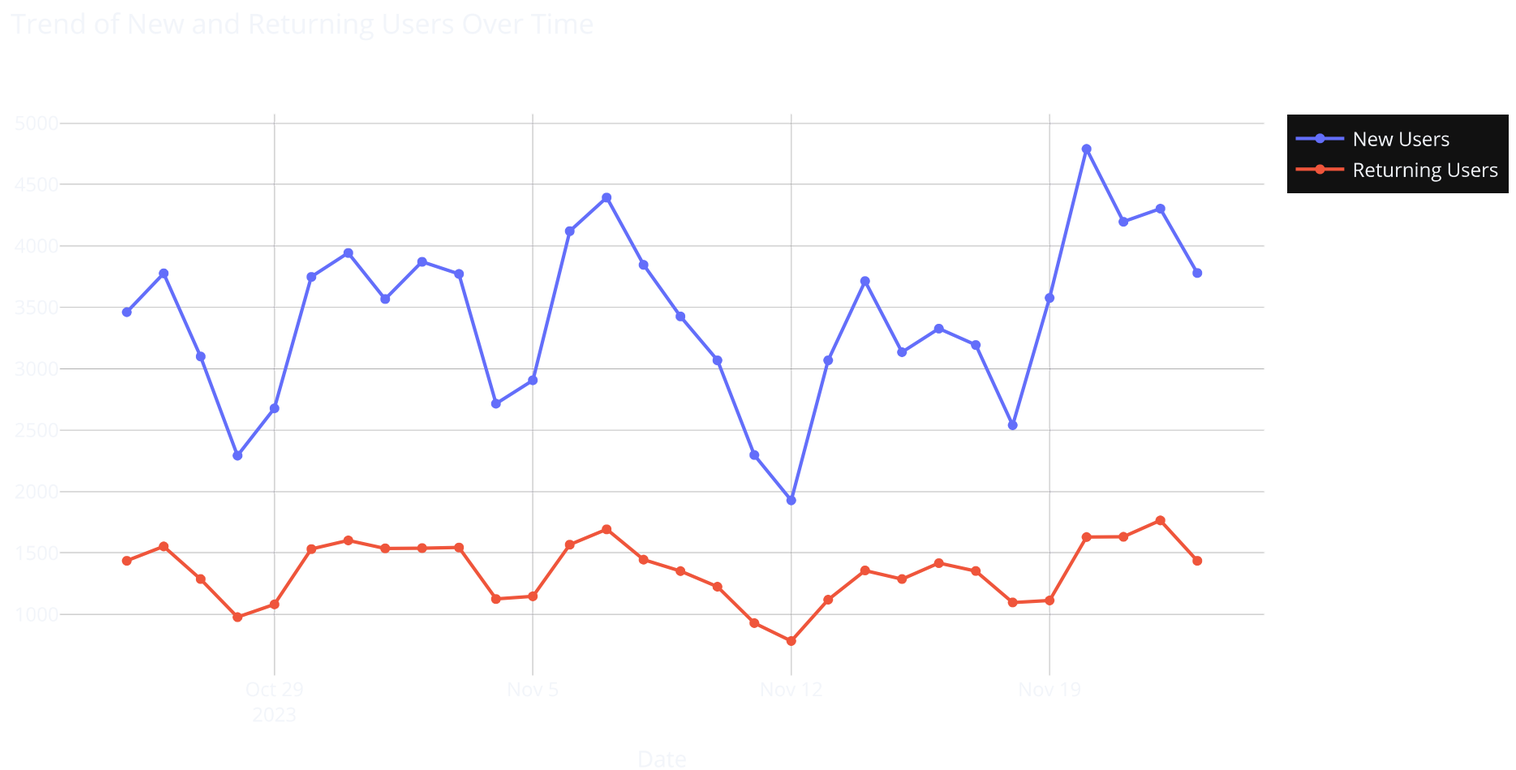
	New users	Returning users	Duration Day 1	Duration Day 7
count	30.000000	30.000000	30.000000	30.000000
mean	3418.166667	1352.866667	208.259594	136.037157
std	677.407486	246.793189	64.730830	96.624319
min	1929.000000	784.000000	59.047619	0.000000
25%	3069.000000	1131.500000	182.974287	68.488971
50%	3514.500000	1388.000000	206.356554	146.381667
75%	3829.500000	1543.750000	230.671046	220.021875
max	4790.000000	1766.000000	445.872340	304.350000

In [6]:
```
df
```

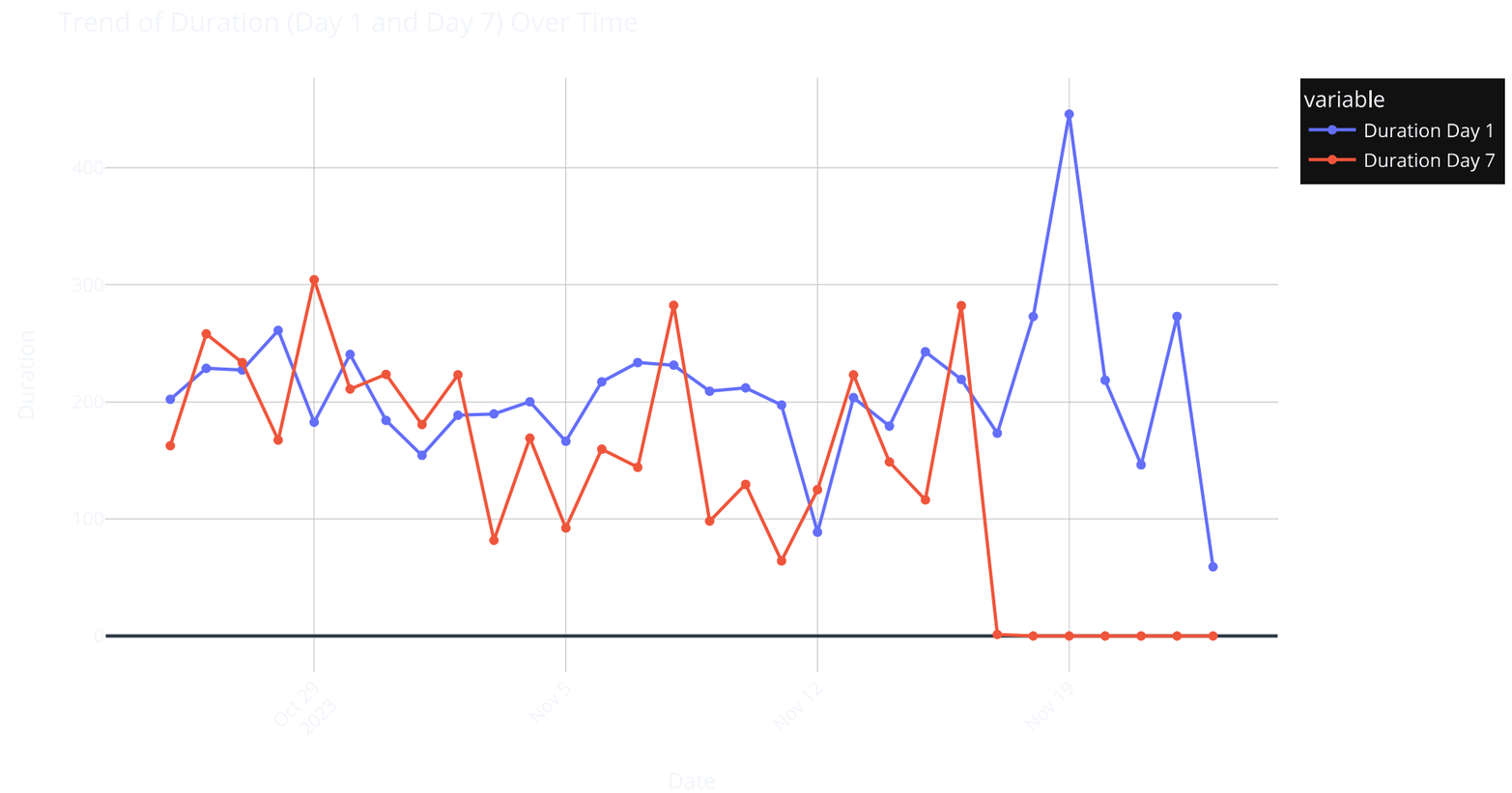
Out[6]:

	Date	New users	Returning users	Duration Day 1	Duration Day 7
0	2023-10-25	3461	1437	202.156977	162.523809
1	2023-10-26	3777	1554	228.631944	258.147059
2	2023-10-27	3100	1288	227.185841	233.550000
3	2023-10-28	2293	978	261.079545	167.357143
4	2023-10-29	2678	1082	182.567568	304.350000
5	2023-10-30	3748	1532	240.543956	210.900000
6	2023-10-31	3943	1603	184.194444	223.463415
7	2023-11-01	3568	1538	154.312925	180.655172
8	2023-11-02	3871	1540	188.531250	223.137931
9	2023-11-03	3772	1545	189.689394	81.705882
10	2023-11-04	2716	1126	200.044643	169.000000
11	2023-11-05	2907	1148	166.305556	92.200000
12	2023-11-06	4121	1568	217.125604	159.545455
13	2023-11-07	4394	1693	233.579235	144.083333
14	2023-11-08	3846	1446	231.350746	282.500000
15	2023-11-09	3426	1353	209.083969	98.097561
16	2023-11-10	3069	1226	211.943182	129.476191
17	2023-11-11	2298	930	197.261905	64.083333
18	2023-11-12	1929	784	88.641026	124.941176
19	2023-11-13	3069	1120	203.629139	223.062500
20	2023-11-14	3714	1358	179.275862	148.680000
21	2023-11-15	3135	1288	242.807692	116.238095
22	2023-11-16	3327	1418	219.187097	282.166667
23	2023-11-17	3194	1354	173.192661	1.250000
24	2023-11-18	2541	1098	272.900000	0.000000
25	2023-11-19	3577	1114	445.872340	0.000000
26	2023-11-20	4790	1630	218.441177	0.000000
27	2023-11-21	4197	1632	146.167488	0.000000
28	2023-11-22	4304	1766	273.037037	0.000000
29	2023-11-23	3780	1437	59.047619	0.000000

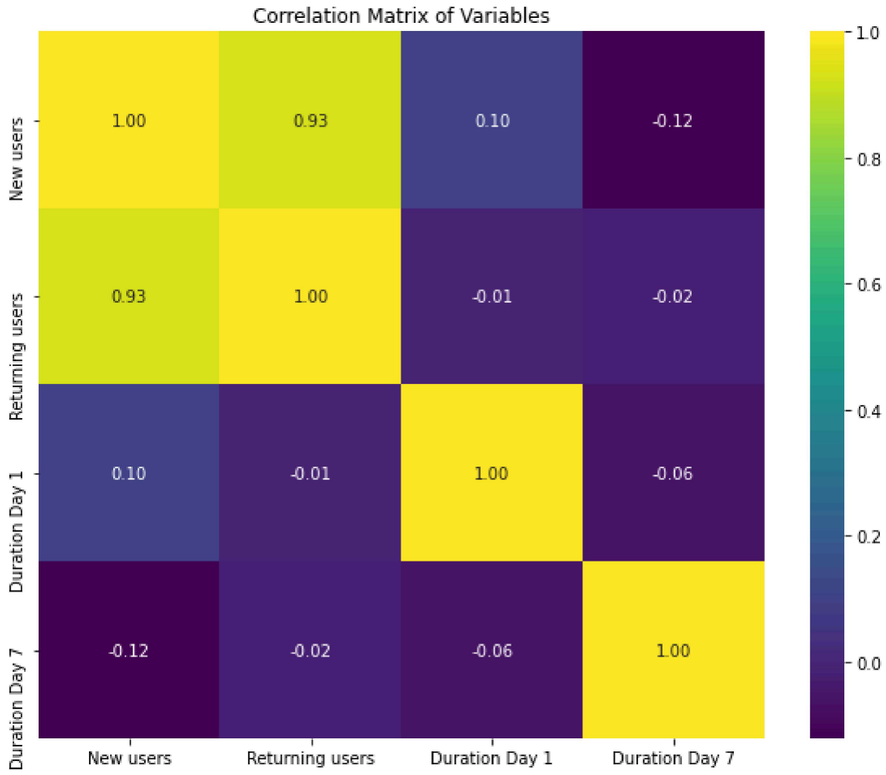
In [7]:
```
# Trend analysis for New and Returning Users

fig= go.Figure()

fig.add_trace(go.Scatter(x=df['Date'], y=df['New users'], mode='lines+markers', name='New Users'))
fig.add_trace(go.Scatter(x=df['Date'], y=df['Returning users'], mode='lines+markers', name='Returning Users'))
fig.update_layout(title='Trend of New and Returning Users Over Time',
                  xaxis_title='Date',
                  yaxis_title='Number of Users')
fig.show()
```



In [8]:
```
# Trend of duration over time

fig = px.line(data_frame=df, x='Date', y=['Duration Day 1', 'Duration Day 7'], markers=True, labels={'value': 'Duration'})
fig.update_layout(title='Trend of Duration (Day 1 and Day 7) Over Time', xaxis_title='Date', yaxis_title='Duration', xaxis=dict(tickangle=-45))
fig.show()
```



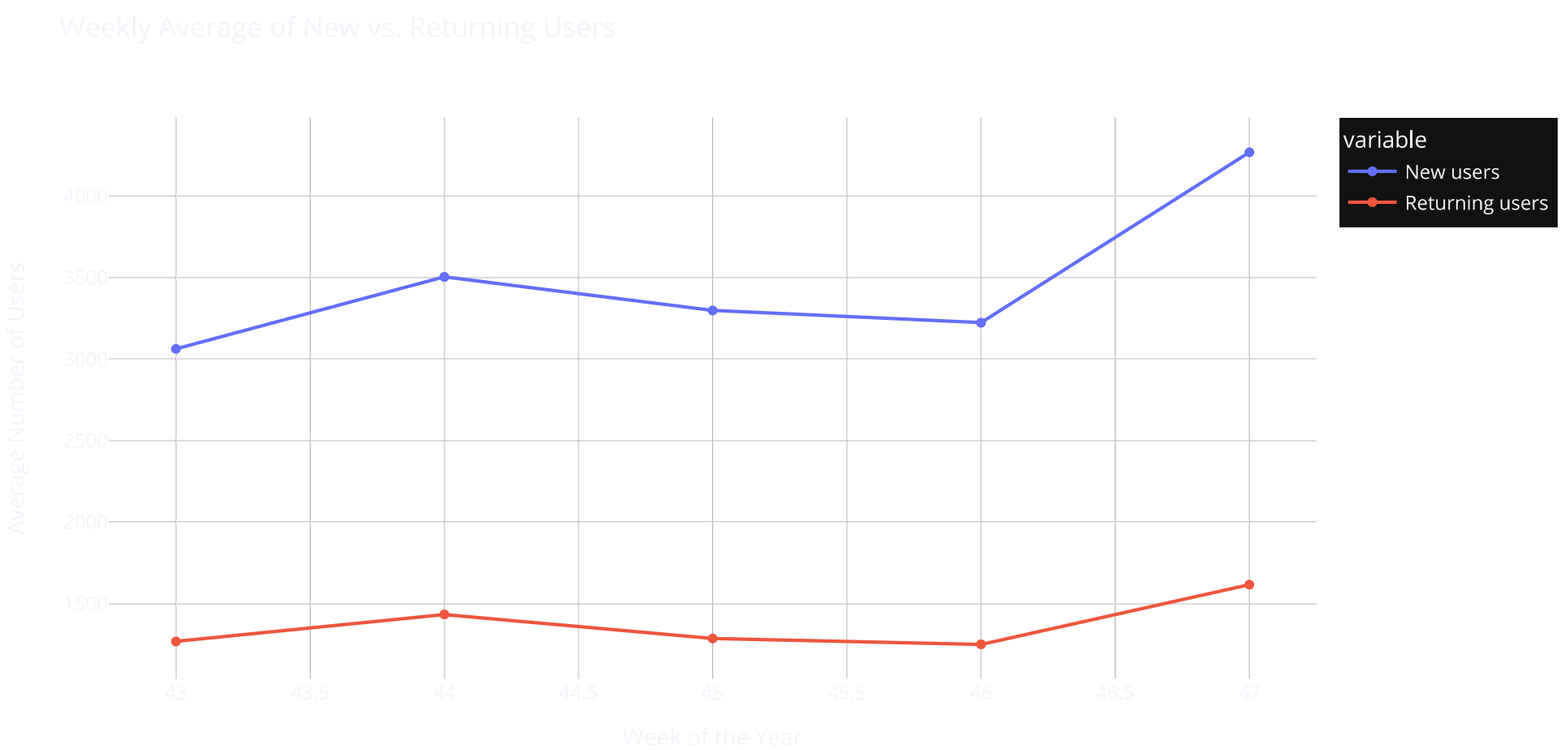
In [9]:
```
# Correlation matrix
correlation_matrix = df.corr()

# Plotting the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f")
plt.title('Correlation Matrix of Variables')
plt.show()
```

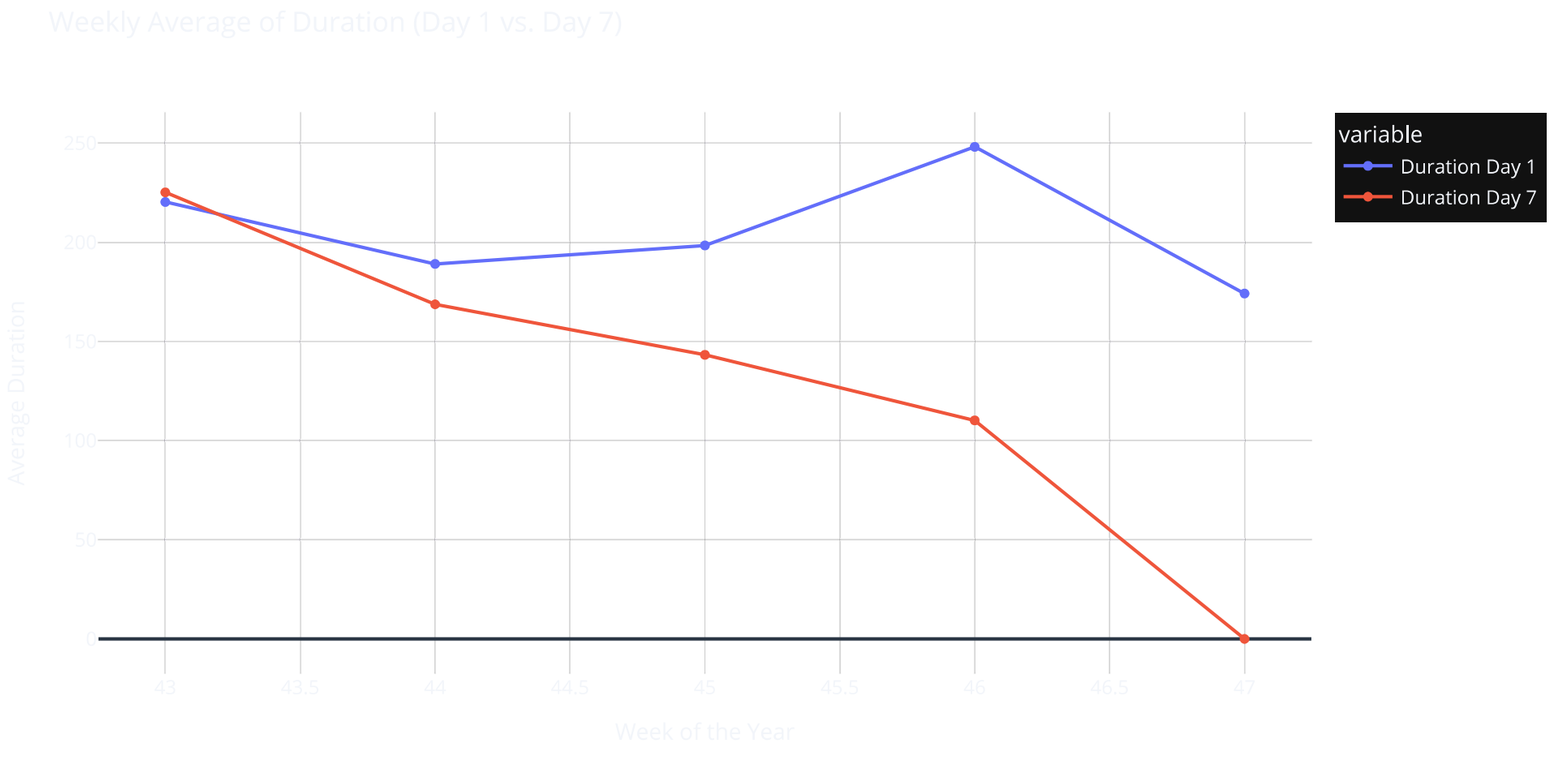


In [10]:
```
# Group the data according to week

df["Week"]= df["Date"].dt.isocalendar().week

# Weekly averages

weekly_averages = df.groupby('Week').agg({
    'New users': 'mean',
    'Returning users': 'mean',
    'Duration Day 1': 'mean',
    'Duration Day 7': 'mean'
}).reset_index()

print(weekly_averages.head())
```

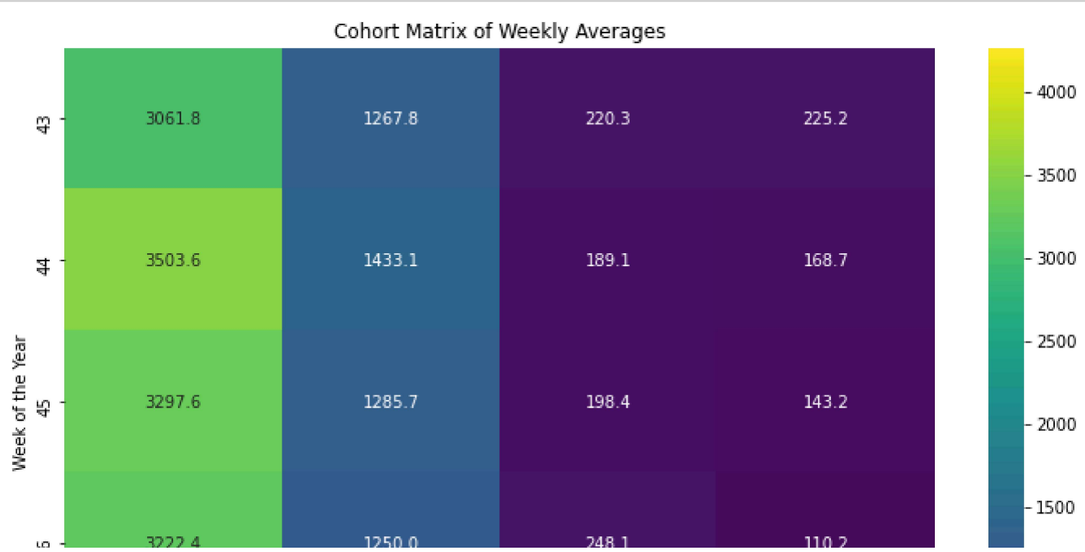
```
   Week    New users  Returning users  Duration Day 1  Duration Day 7
0    43  3061.800000      1267.800000      220.324375      225.185602
1    44  3503.571429      1433.142857      189.088881      168.723200
2    45  3297.571429      1285.714286      198.426524      143.246721
3    46  3222.428571      1250.000000      248.123542      110.199609
4    47  4267.750000      1616.250000      174.173330        0.000000
```

In [11]:
```
fig1 = px.line(weekly_averages, x='Week', y=['New users', 'Returning users'], markers=True,
               labels={'value': 'Average Number of Users'}, title='Weekly Average of New vs. Returning Users')
fig1.update_xaxes(title='Week of the Year')
fig1.update_yaxes(title='Average Number of Users')

fig2 = px.line(weekly_averages, x='Week', y=['Duration Day 1', 'Duration Day 7'], markers=True,
               labels={'value': 'Average Duration'}, title='Weekly Average of Duration (Day 1 vs. Day 7)')
fig2.update_xaxes(title='Week of the Year')
fig2.update_yaxes(title='Average Duration')

fig1.show()
fig2.show()
```




Weekly Average of Duration (Day 1 vs. Day 7)
Average Duration
250
200
150
100
50
0
43
43.5
44
44.5
45
45.5
46
46.5
47
Week of the Year
variable
Duration Day 1
Duration Day 7

In [12]:
```
# Creating a cohort matrix
cohort_matrix = weekly_averages.set_index('Week')

# Plotting the cohort matrix
plt.figure(figsize=(12, 8))

sns.heatmap(cohort_matrix, annot=True, cmap='viridis', fmt=".1f")
plt.title('Cohort Matrix of Weekly Averages')
plt.ylabel('Week of the Year')
plt.show()
```



Conclusion

It is observed that variations in the number of new and returning users on a weekly basis. Notably, there was a notable surge in both categories during Week 47. The average engagement durations on Day 1 and Day 7 differ across the weeks, showing no consistent trend in relation to the influx of new or returning users. This implies that additional factors could be impacting user engagement levels.

In [ ]: