Website Data Analysis Project

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

✓ [68] < 10 ms

```
df = pd.read_csv("C:/Users/mohsin/Downloads/data-export (1).csv")
df
```

✓ [69] 59ms

	# ----------	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9
0	Session primary channel group (Default channel...	Date + hour (YYYYMMDDHH)	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count
1	Direct	2024041623	237	300	144	47.526666666666700	0.6075949367088610	4.673333333333330	0.48	1402
2	Organic Social	2024041719	208	267	132	32.09737827715360	0.6346153846153850	4.295880149812730	0.4943820224719100	1147
3	Direct	2024041723	188	233	115	39.93991416309010	0.6117021276595740	4.587982832618030	0.49356223175965700	1069
4	Organic Social	2024041718	187	256	125	32.16015625	0.6684491978609630	4.078125	0.48828125	1044
...	...	...	...	...	...	...	...	...	...	...
3178	Unassigned	2024042806	0	1	0	0	0	2	0	2
3179	Unassigned	2024043005	0	1	0	0	0	2	0	2

```
df.columns = df.iloc[0]
df = df.drop(index=0).reset_index(drop = True)
df.columns = ["channel group" ,"DateHour",  "Users", "Sessions", "Engaged sessions", "Average engagement time per session", "Engaged sessions per user",    "Events per session",  "Engagement rate",  "Event count"]
df
```

✓ [70] 20ms

	channel group	DateHour	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count
0	Direct	2024041623	237	300	144	47.526666666666700	0.6075949367088610	4.673333333333330	0.48	1402
1	Organic Social	2024041719	208	267	132	32.09737827715360	0.6346153846153850	4.295880149812730	0.4943820224719100	1147
2	Direct	2024041723	188	233	115	39.93991416309010	0.6117021276595740	4.587982832618030	0.49356223175965700	1069
3	Organic Social	2024041718	187	256	125	32.16015625	0.6684491978609630	4.078125	0.48828125	1044
4	Organic Social	2024041720	175	221	112	46.918552036199100	0.64	4.529411764705880	0.5067873303167420	1001
...	...	...	...	...	...	...	...	...	...	...
3177	Unassigned	2024042806	0	1	0	0	0	2	0	2
3178	Unassigned	2024043005	0	1	0	0	0	2	0	2
3179	Unassigned	2024043006	0	1	0	0	0	2	0	2
3180	Unassigned	2024050105	0	1	0	0	0	2	0	2
3181	Unassigned	2024050307	0	1	0	0	0	2	0	2

3182 rows × 10 columns

{} Code M↓ Markdown

```
df.info()
```

✓ [73] 18ms

```
Data columns (total 10 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   channel group                       3182 non-null   object
 1   DateHour                            3182 non-null   object
 2   Users                               3182 non-null   object
 3   Sessions                            3182 non-null   object
 4   Engaged sessions                    3182 non-null   object
 5   Average engagement time per session 3182 non-null   object
 6   Engaged sessions per user           3182 non-null   object
 7   Events per session                  3182 non-null   object
```

```
df["DateHour"] = pd.to_datetime(df["DateHour"],format="%Y%M%d%H",errors="coerce")
df
```
✓ [74] 28ms

	channel group	DateHour	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count
0	Direct	2024-01-16 23:04:00	237	300	144	47.526666666666700	0.6075949367088610	4.673333333333330	0.48	1402
1	Organic Social	2024-01-17 19:04:00	208	267	132	32.09737827715360	0.6346153846153850	4.295880149812730	0.4943820224719100	1147
2	Direct	2024-01-17 23:04:00	188	233	115	39.93991416309010	0.6117021276595740	4.587982832618030	0.49356223175965700	1069
3	Organic Social	2024-01-17 18:04:00	187	256	125	32.16015625	0.6684491978609630	4.078125	0.48828125	1044
4	Organic Social	2024-01-17 20:04:00	175	221	112	46.918552036199100	0.64	4.529411764705880	0.5067873303167420	1001
...	...	...	...	...	...	...	...	...	...	...
3177	Unassigned	2024-01-28 06:04:00	0	1	0	0	0	2	0	2
3178	Unassigned	2024-01-30 05:04:00	0	1	0	0	0	2	0	2
3179	Unassigned	2024-01-30 06:04:00	0	1	0	0	0	2	0	2
3180	Unassigned	2024-01-01 05:05:00	0	1	0	0	0	2	0	2
3181	Unassigned	2024-01-03 07:05:00	0	1	0	0	0	2	0	2

3182 rows × 10 columns

```
numeric_cols = df.columns.drop(["channel group", "DateHour"])
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors="coerce")
df["Hour"] = df["DateHour"].dt.hour
df.info()
```
✓ [75] 35ms

```
 0   channel group                          3182 non-null   object
 1   DateHour                               3182 non-null   datetime64[ns]
 2   Users                                  3182 non-null   int64
 3   Sessions                               3182 non-null   int64
 4   Engaged sessions                       3182 non-null   int64
 5   Average engagement time per session    3182 non-null   float64
 6   Engaged sessions per user              3182 non-null   float64
 7   Events per session                     3182 non-null   float64
 8   Engagement rate                        3182 non-null   float64
 9   Event count                            3182 non-null   int64
 10  Hour                                   3182 non-null   int32
dtypes: datetime64[ns](1)  float64(4)  int32(1)  int64(4)  object(1)
```

```
df.head()
```
✓ [76] 20ms

	channel group	DateHour	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count	Hour
0	Direct	2024-01-16 23:04:00	237	300	144	47.526667	0.607595	4.673333	0.480000	1402	23
1	Organic Social	2024-01-17 19:04:00	208	267	132	32.097378	0.634615	4.295880	0.494382	1147	19
2	Direct	2024-01-17 23:04:00	188	233	115	39.939914	0.611702	4.587983	0.493562	1069	23
3	Organic Social	2024-01-17 18:04:00	187	256	125	32.160156	0.668449	4.078125	0.488281	1044	18
4	Organic Social	2024-01-17 20:04:00	175	221	112	46.918552	0.640000	4.529412	0.506787	1001	20

```
df.describe()
```
✓ [77] 48ms

	DateHour	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count	Hour
count	3182	3182.000000	3182.000000	3182.000000	3182.000000	3182.000000	3182.000000	3182.000000	3182.000000	3182.000000
mean	2024-01-16 19:30:57.548711680	41.935889	51.192646	28.325581	66.644581	0.606450	4.675969	0.503396	242.272470	11.807040
min	2024-01-01 00:05:00	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000
25%	2024-01-10 02:04:00	20.000000	24.000000	13.000000	32.103034	0.561404	3.750000	0.442902	103.000000	6.000000
50%	2024-01-17 01:04:00	42.000000	51.000000	27.000000	49.020202	0.666667	4.410256	0.545455	226.000000	12.000000
75%	2024-01-24 00:04:00	60.000000	71.000000	41.000000	71.487069	0.750000	5.217690	0.633333	339.000000	18.000000
max	2024-01-30 23:04:00	237.000000	300.000000	144.000000	4525.000000	2.000000	56.000000	1.000000	1402.000000	23.000000
std	NaN	29.582258	36.919962	20.650569	127.200659	0.264023	2.795228	0.228206	184.440313	6.886686

Visualization Part

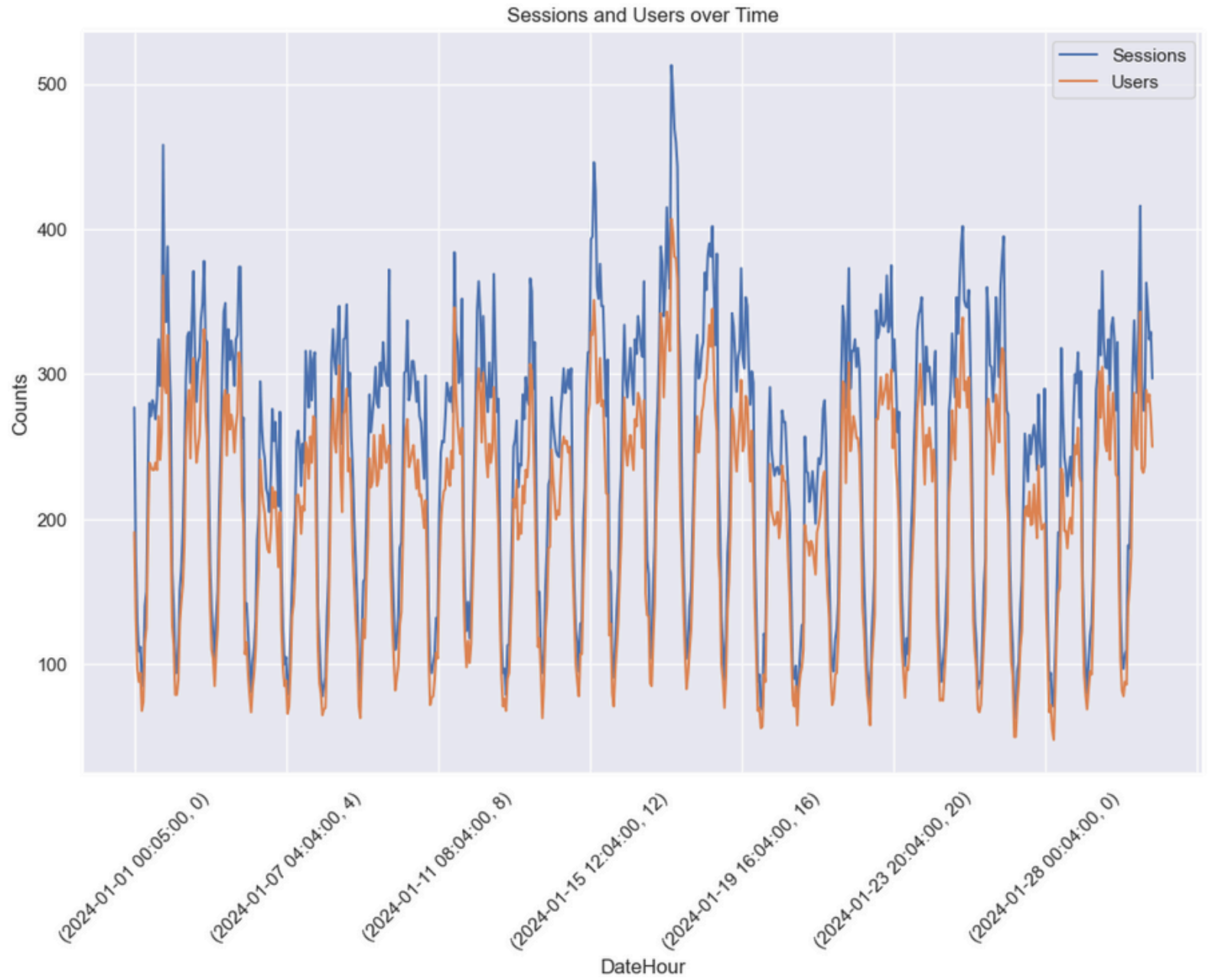
```

```

	channel group	DateHour	Users	Sessions	Engaged sessions	Average engagement time per session	Engaged sessions per user	Events per session	Engagement rate	Event count	Hour
0	Direct	2024-01-16 23:04:00	237	300	144	47.526667	0.607595	4.673333	0.480000	1402	23
1	Organic Social	2024-01-17 19:04:00	208	267	132	32.097378	0.634615	4.295880	0.494382	1147	19
2	Direct	2024-01-17 23:04:00	188	233	115	39.939914	0.611702	4.587983	0.493562	1069	23
3	Organic Social	2024-01-17 18:04:00	187	256	125	32.160156	0.668449	4.078125	0.488281	1044	18
4	Organic Social	2024-01-17 20:04:00	175	221	112	46.918552	0.640000	4.529412	0.506787	1001	20

Sessions and user over time

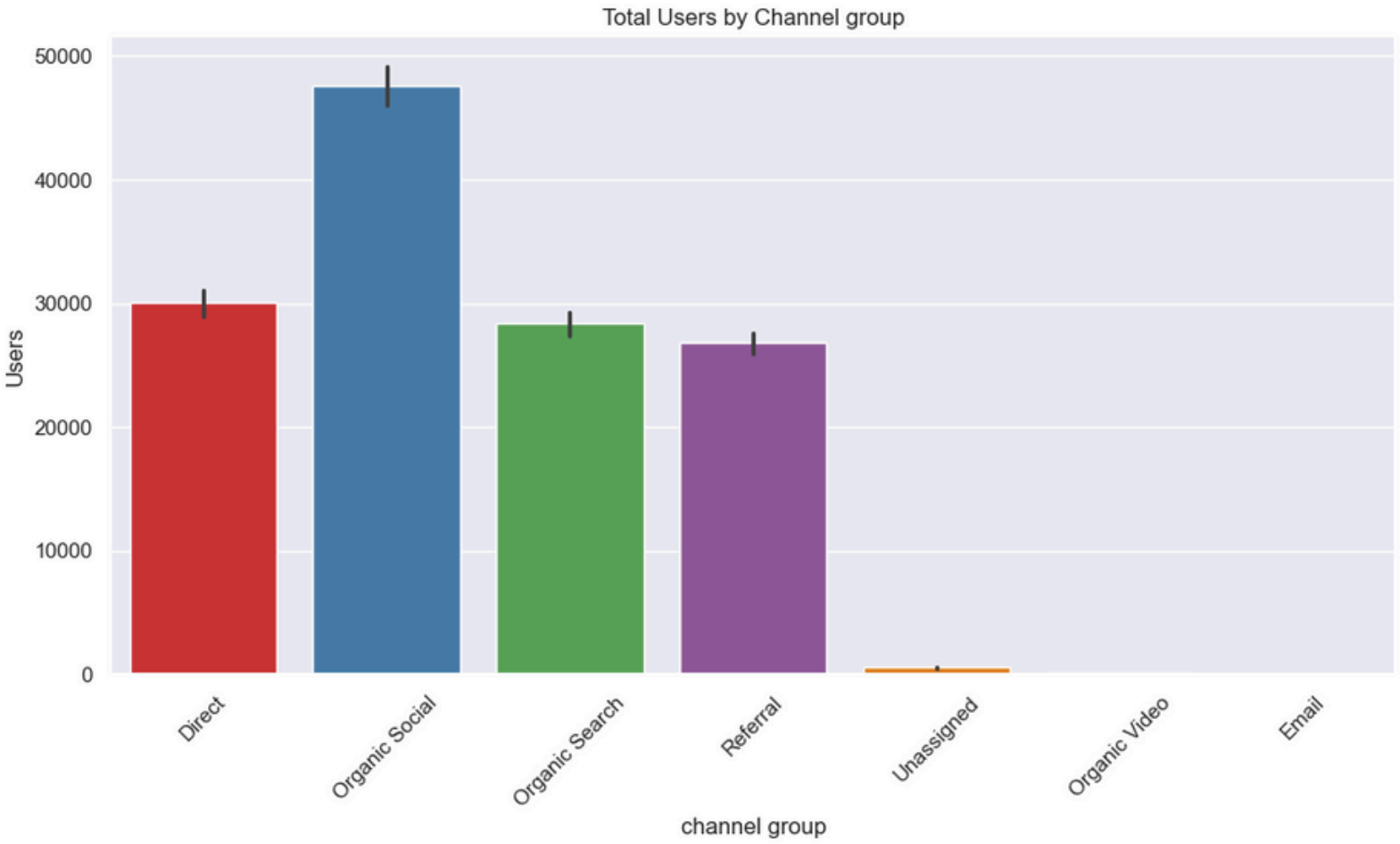
```
plt.figure(figsize = (12,8))
sns.set(style="darkgrid")
ddd = df.groupby(["DateHour","Hour"])[["Sessions","Users"]].sum().plot(ax = plt.gca())
plt.title("Sessions and Users over Time")
plt.xlabel("DateHour")
plt.ylabel("Counts")
plt.xticks(rotation = 45)
plt.show()
```
✓ [92] 200ms



total Users by Chennals

```
plt.figure(figsize = (12,6))
sns.barplot(data = df, x = df["channel group"],y = df["Users"] ,estimator =np.sum,palette = "Set1")
plt.xticks(rotation = 45)
plt.title("Total Users by Channel group")
plt.show()
```

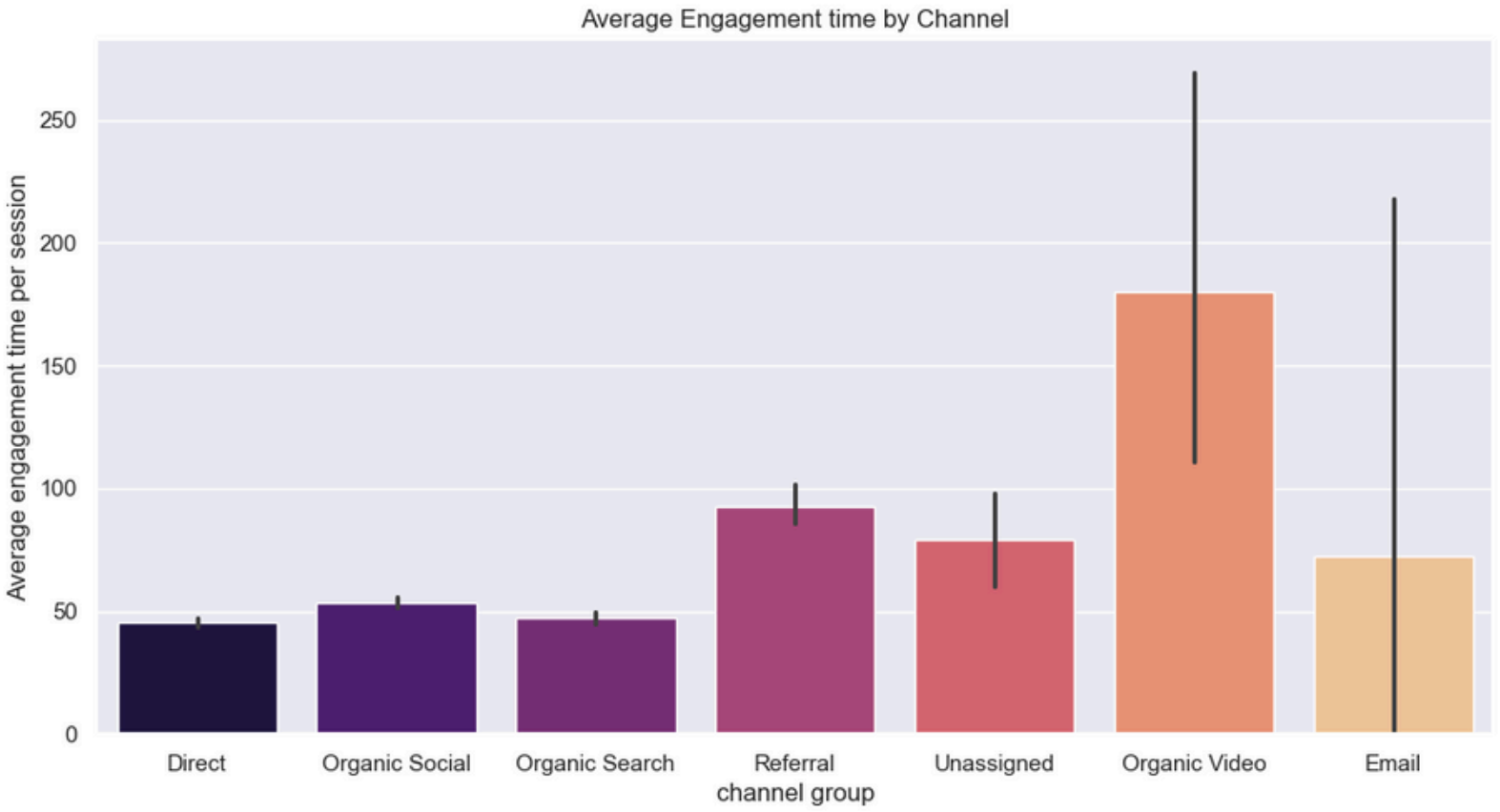
✓ [80] 276ms



Average Engagement time by Channel

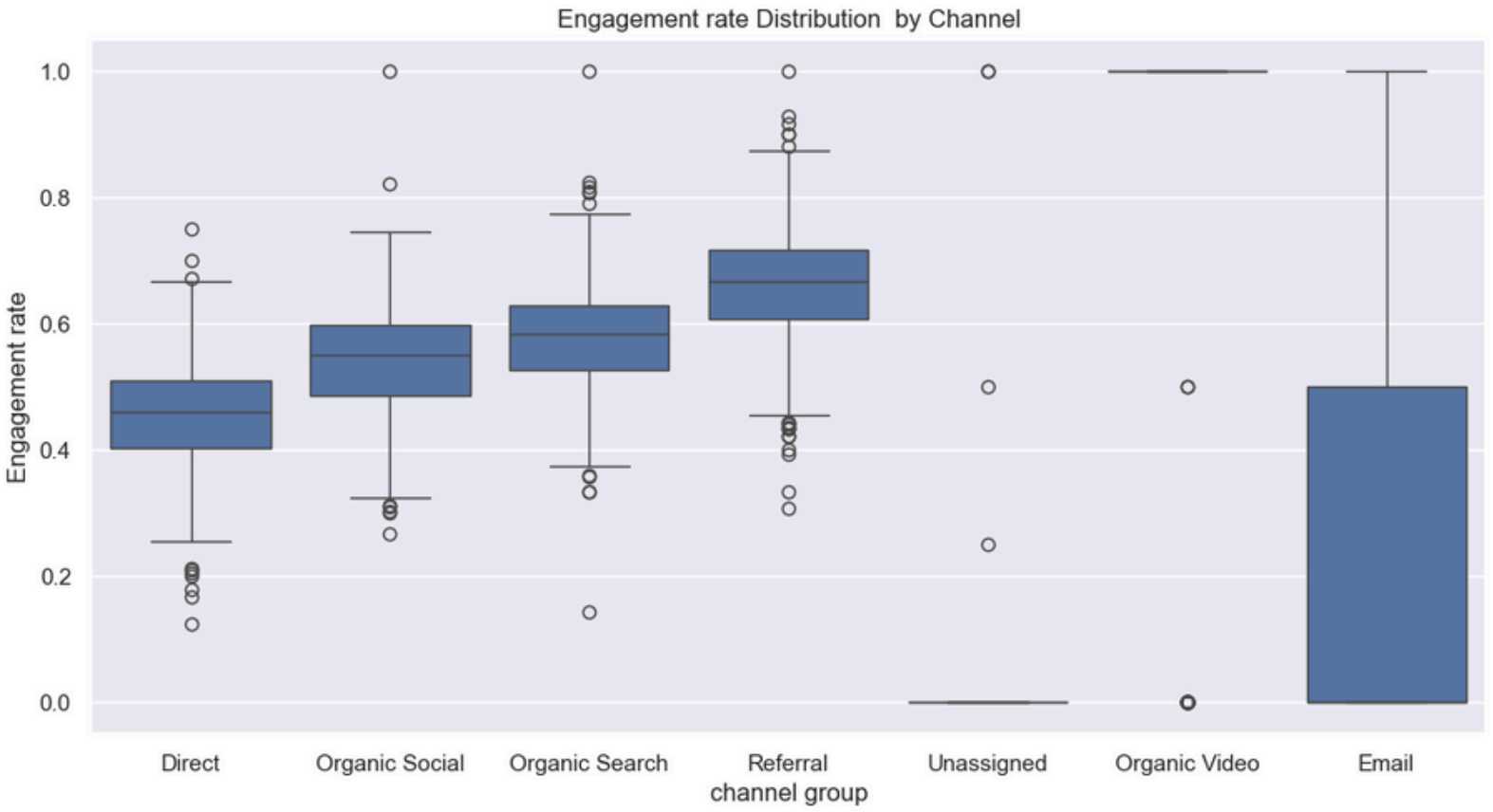
```
plt.figure(figsize = (12,6))
sns.barplot(data = df, x = df["channel group"],y = df["Average engagement time per session"],palette = "magma")
plt.title("Average Engagement time by Channel")
plt.show()
```

✓ [81] 279ms



Engagement rate Distribution by Channel

```
plt.figure(figsize = (12,6))
sns.boxplot(data = df, x = df["channel group"],y = df["Engagement rate"])
plt.title("Engagement rate Distribution  by Channel")
plt.show()
✓ [82] 146ms
```

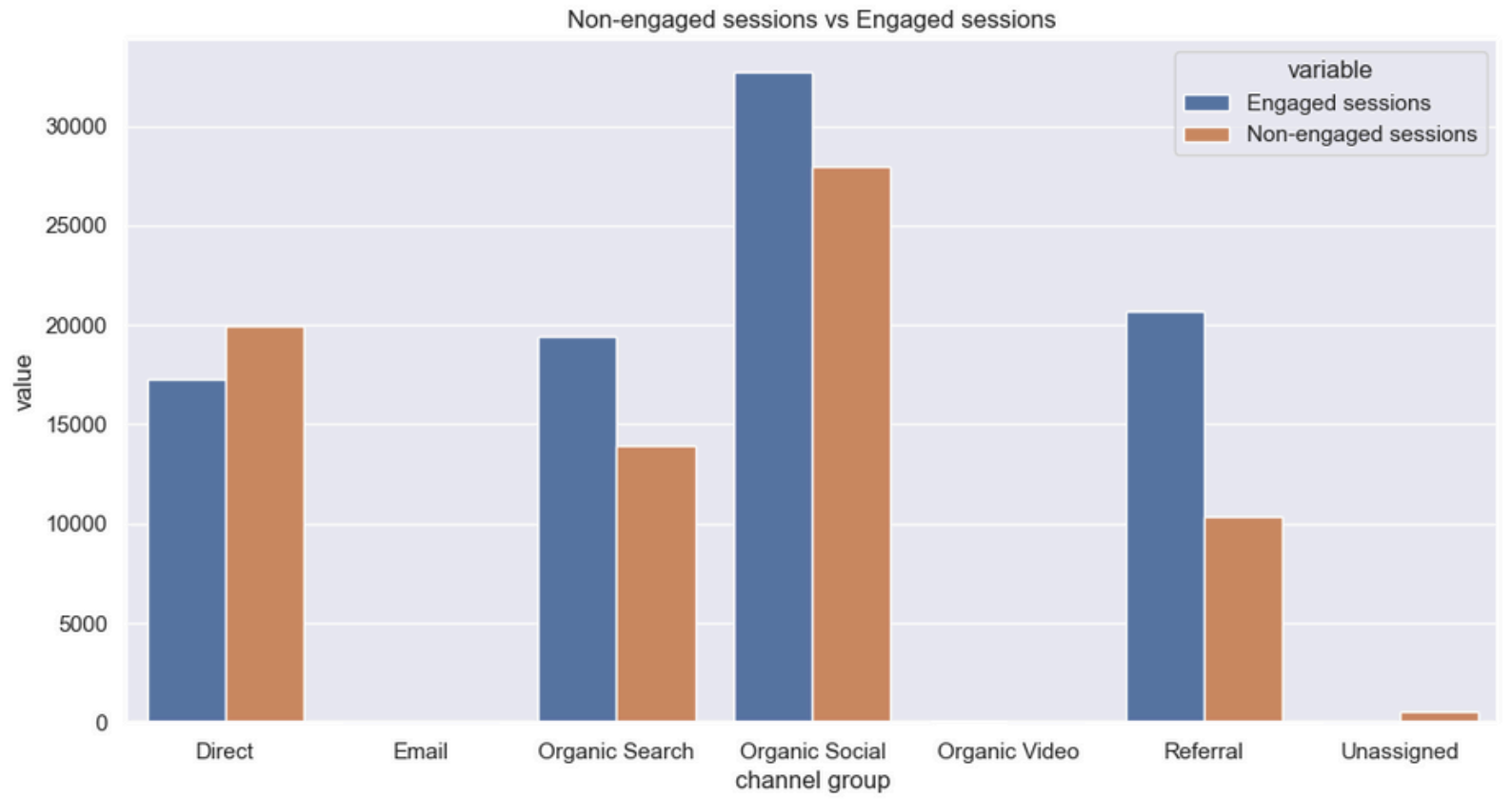


Engaged VS NON Engaged Sessions

```
session_df = df.groupby(["channel group"])[["Sessions","Engaged sessions"]].sum().reset_index()
session_df["Non-engaged sessions"] = session_df["Sessions"] - session_df["Engaged sessions"]
session_df_melted = session_df.melt(id_vars = ["channel group"],value_vars = ["Engaged sessions","Non-engaged sessions",])

plt.figure(figsize = (12,6))
sns.barplot(data = session_df_melted, x = "channel group",y = "value",hue="variable")
plt.title("Non-engaged sessions vs Engaged sessions")
plt.show()
```

✓ [83] 199ms

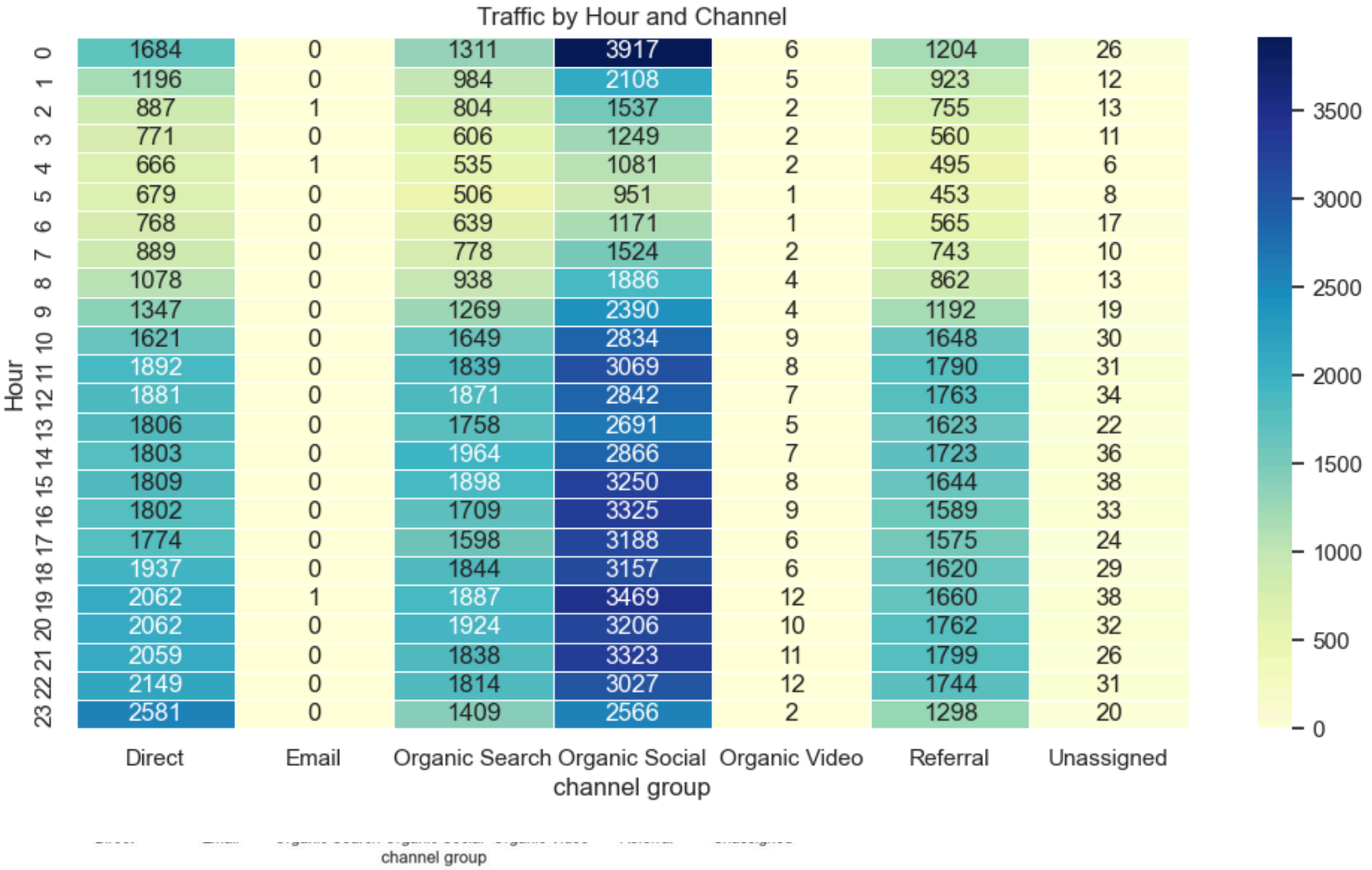


Traffic by Hour and Channel

```
heatmap_data = df.groupby(["Hour","channel group"])["Sessions"].sum().unstack().fillna(0)
heatmap_data

plt.figure(figsize = (12,6))
sns.heatmap(data = heatmap_data,annot = True,fmt = '.0f',linewidths = .5,cmap = "YlGnBu")
plt.title("Traffic by Hour and Channel")
plt.xlabel("channel group")
plt.ylabel("Hour")
plt.show()
```

✓ [84] 353ms

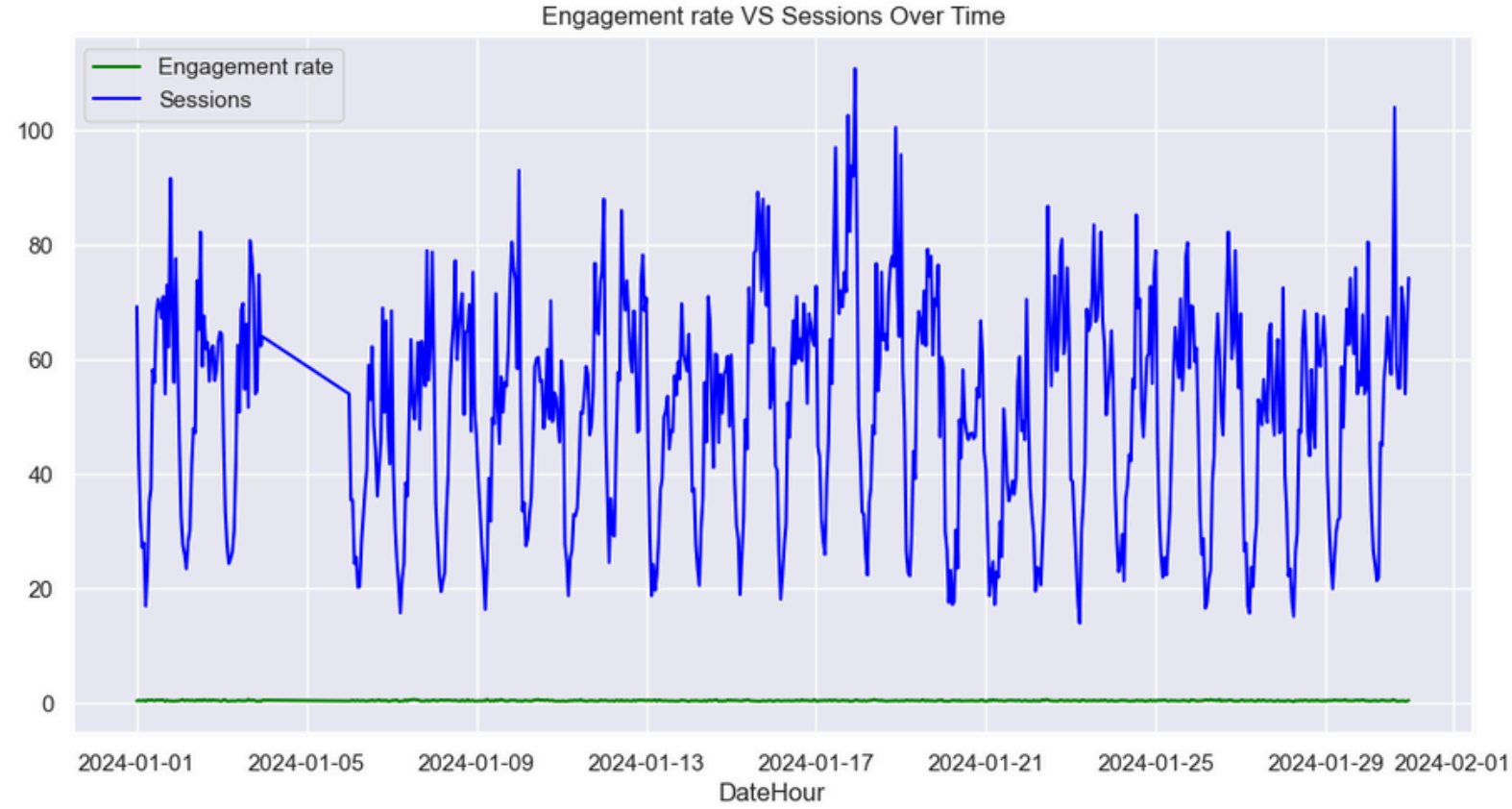


channel group

Engagement Rate VS Sessions Over Time

```
df_plot = df.groupby(["DateHour"])[["Engagement rate","Sessions"]].mean().reset_index()
df_plot
plt.figure(figsize = (12,6))
plt.plot(df_plot["DateHour"],df_plot["Engagement rate"],label = "Engagement rate",color = "green")
plt.plot(df_plot["DateHour"],df_plot["Sessions"],label = "Sessions",color = "blue")
plt.title("Engagement rate VS Sessions Over Time")
plt.legend()
plt.xlabel("DateHour")
plt.grid(True)
plt.show()
```

✓ [85] 143ms



Project by Mohsin Saifi

Thank you