```
import os
import pandas as pd
import numpy as np
import datetime as dt
import seaborn as sns
```

```
# 한글폰트 사용 in colab
%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

!apt-get update -qq
!apt-get install fonts-nanum* -qq

path = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'
font_name = fm.FontProperties(fname=path, size=10).get_name()
print(font_name)
plt.rc('font', family=font_name)

fm._rebuild()
mpl.rcParams['axes.unicode_minus'] = False
```

```
NanumBarunGothic
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_rem
```

```
# Python
!python --version
```

```
Python 3.7.12
```

Font - 나눔고딕 인스톨

```
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf

plt.rc('font', family='NanumBarunGothic')
```

```
Reading package lists... Done
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
fonts-nanum is already the newest version (20170925-1).
The following package was automatically installed and is no longer required:
  libnvidia-common-470
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 31 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
fc-cache: succeeded
```

Data 확인

```
# 데이터 확인
df = pd.read_csv('/content/drive/MyDrive/pretest_data.csv')
```

```

df
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

EDA

```
# df['published_date']를 datetime object로 변경
df['published_date'] = pd.to_datetime(df.published_date)

# df['on_trending_date']를 datetime object로 변경
df['on_trending_date'] = pd.to_datetime(df.on_trending_date)

# df['off_trending_date']를 datetime object로 변경
df['off_trending_date'] = pd.to_datetime(df.off_trending_date)

# df['published_date']를 index로 변경
df.set_index(df.published_date, inplace=True)
```

```

# 불필요해진 df['published_date'] column을 제거
df = df.drop('published_date', axis = 1)

# df를 날짜로 정렬
df = df.sort_values(by='published_date')

# df 재확인
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2644 entries, 2021-03-25 to 2021-07-29
Data columns (total 24 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   video_id                   2644 non-null   object
 1   channel_id                 2644 non-null   object
 2   category_name              2644 non-null   object
 3   duration                   2644 non-null   object
 4   tags                       2274 non-null   object
 5   description                2604 non-null   object
 6   on_trending_date           2644 non-null   datetime64[ns]
 7   off_trending_date          2644 non-null   datetime64[ns]
 8   on_rank                    2644 non-null   int64
 9   off_rank                   2644 non-null   int64
 10  on_views                   2644 non-null   int64
 11  off_views                  2644 non-null   int64
 12  on_likes                   2644 non-null   int64
 13  off_likes                  2644 non-null   int64
 14  on_dislikes                2644 non-null   int64
 15  off_dislikes               2644 non-null   int64
 16  on_comments                2644 non-null   int64
 17  off_comments               2644 non-null   int64
 18  on_channel_subscribers     2644 non-null   int64
 19  off_channel_subscribers    2644 non-null   int64
 20  on_channel_total_views     2644 non-null   int64
 21  off_channel_total_views    2644 non-null   int64
 22  on_channel_total_videos    2644 non-null   int64
 23  off_channel_total_videos   2644 non-null   int64
dtypes: datetime64[ns](2), int64(16), object(6)
memory usage: 516.4+ KB
```

```
df.describe()
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
df['month'] = df.index.month
df['week'] = df.index.week
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: weekofyear and week have been deprecat
```

```
march_df = df.loc[df['month']==3]
april_df = df.loc[df['month']==4]
may_df = df.loc[df['month']==5]
june_df = df.loc[df['month']==6]
july_df = df.loc[df['month']==7]
```

Q1. 데이터 타입별 시각화

1. 전체기간 카테고리 -> 채널 -> 비디오 개수
2. 월별 카테고리 -> 채널 -> 비디오 개수
3. 월별 TOP 10 채널 (분류 기준 : 비디오 개수)
4. 주별 TOP 5 채널 (분류 기준 : 비디오 개수)
5. 월별 카테고리별 태그 키워드 순위

1. 전체기간 카테고리 -> 채널 -> 비디오 개수

```
# 전체 기간 카테고리 중 채널 당 비디오 개수
q1_1_df = df.groupby(['category_name', 'channel_id'])['off_channel_total_videos']
```

```
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_1_df.first())))

plt.yticks(fontsize=10)
plt.xticks(xtick_label_position, q1_1_df, fontsize=6, rotation='vertical')

bars = plt.bar(xtick_label_position, q1_1_df.first())
plt.tick_params(axis='x', direction='out', length=3, pad=5, labelsize=6)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_1_df.first()[i], ha='center', fontsize

plt.grid()
plt.title('전체 기간 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

2. 월별 카테고리 -> 채널 -> 비디오 개수

```
# 3월 카테고리 중 채널 당 비디오 개수
q1_2_march_df = march_df.groupby(['category_name', 'channel_id'])['off_channel_total_videos']

# 4월 카테고리 중 채널 당 비디오 개수
q1_2_april_df = april_df.groupby(['category_name', 'channel_id'])['off_channel_total_videos']

# 5월 카테고리 중 채널 당 비디오 개수
q1_2_may_df = may_df.groupby(['category_name', 'channel_id'])['off_channel_total_videos']

# 6월 카테고리 중 채널 당 비디오 개수
[illegible]'])['off_channel_total_videos']

# 7월 카테고리 중 채널 당 비디오 개수
q1_2_july_df = july_df.groupby(['category_name', 'channel_id'])['off_channel_total_videos']
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
# 3월 그래프
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_2_march_df.first())))

plt.yticks(fontsize=10)
plt.xticks(xtick_label_position, q1_2_march_df, fontsize=15, rotation='vertical')

bars = plt.bar(xtick_label_position, q1_2_march_df.first())
plt.tick_params(axis='x', direction='out', length=15, pad=5, labelsize=15)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_2_march_df.first()[i], ha='center', fc
plt.grid()
plt.title('3월 중 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

```
# 4월 그래프
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_2_april_df.first())))

plt.yticks(fontsize=10)
plt.xticks(xtick_label_position, q1_2_april_df, fontsize=10, rotation='vertical')

bars = plt.bar(xtick_label_position, q1_2_april_df.first())
plt.tick_params(axis='x', direction='out', length=15, pad=10, labelsize=10)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_2_april_df.first()[i], ha='center', fc
plt.grid()
plt.title('4월 중 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

```
# 5월 그래프
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_2_may_df.first())))

plt.yticks(fontsize=10)
plt.xticks(xtick_label_position, q1_2_may_df, fontsize=10, rotation='vertical')

bars = plt.bar(xtick_label_position, q1_2_may_df.first())
plt.tick_params(axis='x', direction='out', length=15, pad=10, labelsize=10)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_2_may_df.first()[i], ha='center', font
plt.grid()
plt.title('5월 중 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

```
# 6월 그래프
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_2_june_df.first())))

plt.yticks(fontsize=10)
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
rotation='vertical')

bars = plt.bar(xtick_label_position, q1_2_june_df.first())
plt.tick_params(axis='x', direction='out', length=15, pad=10, labelsize=10)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_2_june_df.first()[i], ha='center', for
plt.grid()
plt.title('6월 중 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

```
# 7월 그래프
fig = plt.figure(figsize=(400, 10))
ax = fig.add_subplot()
xtick_label_position = list(range(len(q1_2_july_df.first())))

plt.yticks(fontsize=10)
plt.xticks(xtick_label_position, q1_2_july_df, fontsize=10, rotation='vertical')

bars = plt.bar(xtick_label_position, q1_2_july_df.first())
```

```
plt.tick_params(axis='x', direction='out', length=15, pad=10, labelsize=10)

for i, b in enumerate(bars):
    ax.text(b.get_x()+b.get_width()*(1/2), b.get_height()+0.1, q1_2_july_df.first()[i], ha='center', for
plt.grid()
plt.title('7월 중 카테고리 중 채널 당 비디오 개수', loc='center', pad=20)
plt.show()
```

3. 월별 TOP 10 채널 (분류 기준 : 비디오 개수)

3월 Top10 Channel

```
march_df['video_counts'] = march_df['off_channel_total_videos'] - march_df['on_channel_total_videos']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  """Entry point for launching an IPython kernel.
```

```
march_top10 = march_df.sort_values(by='video_counts', ascending=False).head(10)['video_counts']
labels = march_df.sort_values(by='video_counts', ascending=False).head(10)['channel_id']
plt.pie(march_top10, autopct='%1.1f%%')
plt.title('3월 TOP 10 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

4월 Top10 Channel

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
s'] - april_df['on_channel_total_videos']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  """Entry point for launching an IPython kernel.
```

```
april_top10 = april_df.sort_values(by='video_counts', ascending=False).head(10)['video_counts']
labels = april_df.sort_values(by='video_counts', ascending=False).head(10)['channel_id']
plt.pie(april_top10, autopct='%1.1f%%')
plt.title('4월 TOP 10 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

5월 Top10 Channel

```
may_df['video_counts'] = may_df['off_channel_total_videos'] - may_df['on_channel_total_videos']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  """Entry point for launching an IPython kernel.
```

```
may_top10 = may_df.sort_values(by='video_counts', ascending=False).head(10)['video_counts']
labels = may_df.sort_values(by='video_counts', ascending=False).head(10)['channel_id']
plt.pie(may_top10, autopct='%1.1f%%')
plt.title('5월 TOP 10 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

6월 Top10 Channel

```
june_df['video_counts'] = june_df['off_channel_total_videos'] - june_df['on_channel_total_videos']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  """Entry point for launching an IPython kernel.
```

```
june_top10 = june_df.sort_values(by='video_counts', ascending=False).head(10)['video_counts']
labels = june_df.sort_values(by='video_counts', ascending=False).head(10)['channel_id']
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

7월 Top10 Channel

```
july_df['video_counts'] = july_df['off_channel_total_videos'] - july_df['on_channel_total_videos']
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  """Entry point for launching an IPython kernel.
```

```python
july_top10 = july_df.sort_values(by='video_counts', ascending=False).head(10)['video_counts']
labels = july_df.sort_values(by='video_counts', ascending=False).head(10)['channel_id']
plt.pie(july_top10, autopct='%1.1f%%')
plt.title('7월 TOP 10 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

4. 주별 TOP 5 채널 (분류 기준 : 비디오 개수)

```python
w12_df = df.loc[df['week']==12]
w13_df = df.loc[df['week']==13]
w14_df = df.loc[df['week']==14]
w15_df = df.loc[df['week']==15]
w16_df = df.loc[df['week']==16]
w17_df = df.loc[df['week']==17]
w18_df = df.loc[df['week']==18]
w19_df = df.loc[df['week']==19]
w20_df = df.loc[df['week']==20]
w21_df = df.loc[df['week']==21]
w22_df = df.loc[df['week']==22]
w23_df = df.loc[df['week']==23]
w24_df = df.loc[df['week']==24]
w25_df = df.loc[df['week']==25]
w26_df = df.loc[df['week']==26]
w27_df = df.loc[df['week']==27]
w28_df = df.loc[df['week']==28]
w29_df = df.loc[df['week']==29]
w30_df = df.loc[df['week']==30]
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```python
w12_df['on_channel_total_views']
w13_df['video_counts'] = w13_df['off_channel_total_views'] - w13_df['on_channel_total_views']
w14_df['video_counts'] = w14_df['off_channel_total_views'] - w14_df['on_channel_total_views']
w15_df['video_counts'] = w15_df['off_channel_total_views'] - w15_df['on_channel_total_views']
w16_df['video_counts'] = w16_df['off_channel_total_views'] - w16_df['on_channel_total_views']
w17_df['video_counts'] = w17_df['off_channel_total_views'] - w17_df['on_channel_total_views']
w18_df['video_counts'] = w18_df['off_channel_total_views'] - w18_df['on_channel_total_views']
w19_df['video_counts'] = w19_df['off_channel_total_views'] - w19_df['on_channel_total_views']
w20_df['video_counts'] = w20_df['off_channel_total_views'] - w20_df['on_channel_total_views']
w21_df['video_counts'] = w21_df['off_channel_total_views'] - w21_df['on_channel_total_views']
w22_df['video_counts'] = w22_df['off_channel_total_views'] - w22_df['on_channel_total_views']
w23_df['video_counts'] = w23_df['off_channel_total_views'] - w23_df['on_channel_total_views']
w24_df['video_counts'] = w24_df['off_channel_total_views'] - w24_df['on_channel_total_views']
w25_df['video_counts'] = w25_df['off_channel_total_views'] - w25_df['on_channel_total_views']
w26_df['video_counts'] = w26_df['off_channel_total_views'] - w26_df['on_channel_total_views']
w27_df['video_counts'] = w27_df['off_channel_total_views'] - w27_df['on_channel_total_views']
w28_df['video_counts'] = w28_df['off_channel_total_views'] - w28_df['on_channel_total_views']
w29_df['video_counts'] = w29_df['off_channel_total_views'] - w29_df['on_channel_total_views']
w30_df['video_counts'] = w30_df['off_channel_total_views'] - w30_df['on_channel_total_views']
```

```
  if __name__ == '__main__':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  if sys.path[0] == '':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  del sys.path[0]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  from ipykernel import kernelapp as app
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
```

```
# 12주 Top5 Channel

                                                 alse).head(5)['video_counts']
                                                 se).head(5)['channel_id']
plt.pie(w12_top5, autopct='%1.1f%%')
plt.title('12주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
# 13주 Top5 Channel

w13_top5 = w13_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
```

```
labels = w13_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w13_top5, autopct='%1.1f%%')
plt.title('13주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 14주 Top5 Channel

w14_top5 = w14_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w14_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w14_top5, autopct='%1.1f%%')
plt.title('14주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 15주 Top5 Channel

w15_top5 = w15_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w15_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w15_top5, autopct='%1.1f%%')
plt.title('15주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
# 16주 Top5 Channel

w16_top5 = w16_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w16_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w16_top5, autopct='%1.1f%%')
plt.title('16주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```python
# 17주 Top5 Channel

w17_top5 = w17_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w17_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w17_top5, autopct='%1.1f%%')
plt.title('17주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```python
# 18주 Top5 Channel

w18_top5 = w18_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w18_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w18_top5, autopct='%1.1f%%')
plt.title('18주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```python
# 19주 Top5 Channel

w12_top5 = w12_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w12_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w12_top5, autopct='%1.1f%%')
plt.title('12주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 20주 Top5 Channel

w20_top5 = w20_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w20_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w20_top5, autopct='%1.1f%%')
plt.title('20주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 21주 Top5 Channel

w21_top5 = w21_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w21_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w21_top5, autopct='%1.1f%%')
plt.title('21주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```

w22_top5 = w22_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w22_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w22_top5, autopct='%1.1f%%')
plt.title('22주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 23주 Top5 Channel

w23_top5 = w23_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w23_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w23_top5, autopct='%1.1f%%')
plt.title('23주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 24주 Top5 Channel

w24_top5 = w24_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w24_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w24_top5, autopct='%1.1f%%')
plt.title('24주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 25주 Top5 Channel

w25_top5 = w25_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w25_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w25_top5, autopct='%1.1f%%')
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
0.95, 0.3))
```

```
# 26주 Top5 Channel

w26_top5 = w26_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w26_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w26_top5, autopct='%1.1f%%')
plt.title('26주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 27주 Top5 Channel

w27_top5 = w27_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w27_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w27_top5, autopct='%1.1f%%')
plt.title('27주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 28주 Top5 Channel

w28_top5 = w28_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w28_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w28_top5, autopct='%1.1f%%')
plt.title('28주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
# 29주 Top5 Channel

w29_top5 = w29_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w29_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w29_top5, autopct='%1.1f%%')
plt.title('29주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

```
# 30주 Top5 Channel

w30_top5 = w30_df.sort_values(by='video_counts', ascending=False).head(5)['video_counts']
labels = w30_df.sort_values(by='video_counts', ascending=False).head(5)['channel_id']
plt.pie(w30_top5, autopct='%1.1f%%')
plt.title('30주 TOP 5 Channel')
plt.legend(labels=labels, loc='lower left', bbox_to_anchor=(0.95, 0.3))
```

5. 월별 카테고리별 태그 키워드 순위

Q2. 새로운 지표 개발 및 상관관계 확인

각각의 비디오는 시청자의 호응도(engagement)를 판단할 수 있는 객관적인 지표들이 있음.

example) views, likes, dislikes, ... etc

1. 비디오를 인기 동영상 기준에 부합하도록 분류할 수 있는 새로운 지표를 개발
2. 새로운 지표를 사용하여, engagement와 어떤 상관관계가 있는지 설명

```
# 지표 1: 인기 동영상 선정 이후 구독자 증가량
# 선정 이유 : 인기 동영상 선정 이후 구독자 수의 증가는 영상에 만족한 사람들이 많다는 것을 의미한다고 가정

df['subscribers'] = df['off_channel_subscribers'] - df['on_channel_subscribers']
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

```
# 선정 이유 : 순위의 차이가 적다는 것은 최후 순위 이후 급격하게 차트에서 나가게 된 것을 의미한다고 가정
# 이는 engagement가 높았다면 발생하지 않았을거라 생각
df['rank'] = df['off_rank'] - df['on_rank']
```

```
df.corr()
```

otal_views	on_channel_total_videos	off_channel_total_videos	month	week	new_indicator	subscribers	rank
-0.173227	-0.098883	-0.099022	-0.009760	-0.015894	-0.061146	-0.154043	-0.684709
-0.192692	-0.137106	-0.137702	-0.034237	-0.059432	-0.015347	-0.060670	0.433112
0.496053	0.013871	0.014238	0.006841	0.011585	0.773296	0.648629	0.081233
0.426483	0.003197	0.003445	0.003362	0.006840	0.877438	0.638164	0.096178
0.516788	-0.023047	-0.022897	0.009068	0.014369	0.687309	0.696246	0.049553
0.454456	-0.019231	-0.019109	0.009646	0.013174	0.806886	0.720204	0.053754
0.461163	-0.013693	-0.013433	-0.003016	-0.000335	0.311661	0.476542	0.108477
0.414769	-0.014753	-0.014574	0.003780	0.005246	0.728992	0.634871	0.107048
0.344553	-0.007968	-0.007885	-0.000256	-0.001474	0.954962	0.585103	0.045428
0.321359	-0.007619	-0.007544	-0.001294	-0.002328	0.976012	0.568074	0.046298
0.785450	0.048609	0.049013	-0.003302	0.003644	0.323295	0.528810	0.023828
0.784577	0.048270	0.048717	-0.003273	0.003648	0.325890	0.534427	0.024522
0.999989	0.234382	0.235140	-0.004269	0.003771	0.245418	0.302576	0.016116
1.000000	0.234057	0.234846	-0.004269	0.003770	0.248979	0.305887	0.016831
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	0.101443
0.016831	-0.011575	-0.011907	-0.017318	-0.031077	0.047092	0.101443	1.000000

```
'''
인기 동영상 선정 이후 구독자 증가량과 꽤 가장 높은 양의 상관관계를 갖는 지표는 'off_likes'이다.
동영상 업로드 후 구독자 수의 증가량은 인기 동영상에서 사라지기 전 기록된 좋아요수와 0.720204의 상관관계를 갖고 있다.

인기 동영상의 최초 순위와 최후 순위의 격차는 영상의 'video_id'와 연관이 있다고 판단이 된다.(음의 상관관계)
하지만, 구체적으로 어떠한 영향을 끼치는지 판단할 수 없다.
'''
```

자동으로 저장할 수 없습니다. 이 파일은 원격으로 또는 다른 탭에서 업데이트되었습니다. 차이 보기

✓ 0초 오후 9:29에 완료됨