```
[3]
import pandas as pd
```
Python

```
[4]
utube=pd.read_csv(r"D:\DATA SCIENCE\DATASET\youtube_ad_revenue.csv")
```
Python

```
[5]
utube
```
Python

	video_id	date	views	likes	comments	watch_time_minutes	video_length_minutes	subscribers	category	device	country	ad_revenue_usd
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	2.862137	228086	Entertainment	TV	IN	203.178237
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	23.738069	736015	Gaming	Tablet	CA	140.880508
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	26.200634	240534	Education	TV	CA	360.134008
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	11.770340	434482	Entertainment	Mobile	UK	224.638261
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	6.635854	42030	Education	Mobile	CA	165.514388
...	...	...	...	...	...	...	...	...	...	...	...	...
122395	vid_2902	2024-12-14 10:50:40.993199	9853	1673.0	147.0	42075.704885	25.490195	210818	Education	Tablet	US	280.986396

UNDERSTAND THE DATA

```
utube.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122400 entries, 0 to 122399
Data columns (total 12 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   video_id              122400 non-null  object 
 1   date                  122400 non-null  object 
 2   views                 122400 non-null  int64  
 3   likes                 116283 non-null  float64
 4   comments              116288 non-null  float64
 5   watch_time_minutes    116295 non-null  float64
 6   video_length_minutes  122400 non-null  float64
 7   subscribers           122400 non-null  int64  
 8   category              122400 non-null  object 
 9   device                122400 non-null  object 
 10  country               122400 non-null  object 
 11  ad_revenue_usd        122400 non-null  float64
dtypes: float64(5), int64(2), object(5)
memory usage: 11.2+ MB
```

```
utube.describe()
```

	views	likes	comments	watch_time_minutes	video_length_minutes	subscribers	ad_revenue_usd
count	122400.000000	116283.000000	116288.000000	116295.000000	122400.000000	122400.000000	122400.000000
mean	9999.856283	1099.633618	274.396636	37543.827721	16.014165	502191.719902	252.727210
std	99.881260	519.424089	129.741739	12987.724246	8.083790	288397.470103	61.957052
min	9521.000000	195.000000	48.000000	14659.105562	2.000142	1005.000000	126.590603
25%	9933.000000	650.000000	162.000000	26366.320569	9.004695	252507.500000	199.902018
50%	10000.000000	1103.000000	274.000000	37531.990337	16.005906	503465.500000	252.749699
75%	10067.000000	1547.000000	387.000000	48777.782090	23.021260	752192.000000	305.597518
max	10468.000000	2061.000000	515.000000	61557.670089	29.999799	999997.000000	382.768254

```
utube.nunique()
```

```
video_id                5000
date                     365
views                    736
likes                   1855
comments                 466
watch time minutes     114000
```

```
utube.duplicated().sum()
```

[9] Python

```
np.int64(2400)
```

```
utube.isnull().sum()
```

[10] Python

```
video_id                 0
date                     0
views                    0
likes                 6117
comments              6112
watch_time_minutes    6105
video_length_minutes     0
subscribers              0
category                 0
device                   0
country                  0
ad_revenue_usd           0
dtype: int64
```

```
utube.columns
```

[11] Python

```
Index(['video_id', 'date', 'views', 'likes', 'comments', 'watch_time_minutes',
       'video_length_minutes', 'subscribers', 'category', 'device', 'country',
       'ad_revenue_usd'],
```

VISHUALIZING THE DATA

```
numerical_columns=['views', 'likes', 'comments', 'watch_time_minutes',
       'video_length_minutes', 'subscribers', 'ad_revenue_usd']
```

[12] Python

```
print(utube[numerical_columns].describe())
```

[13] Python

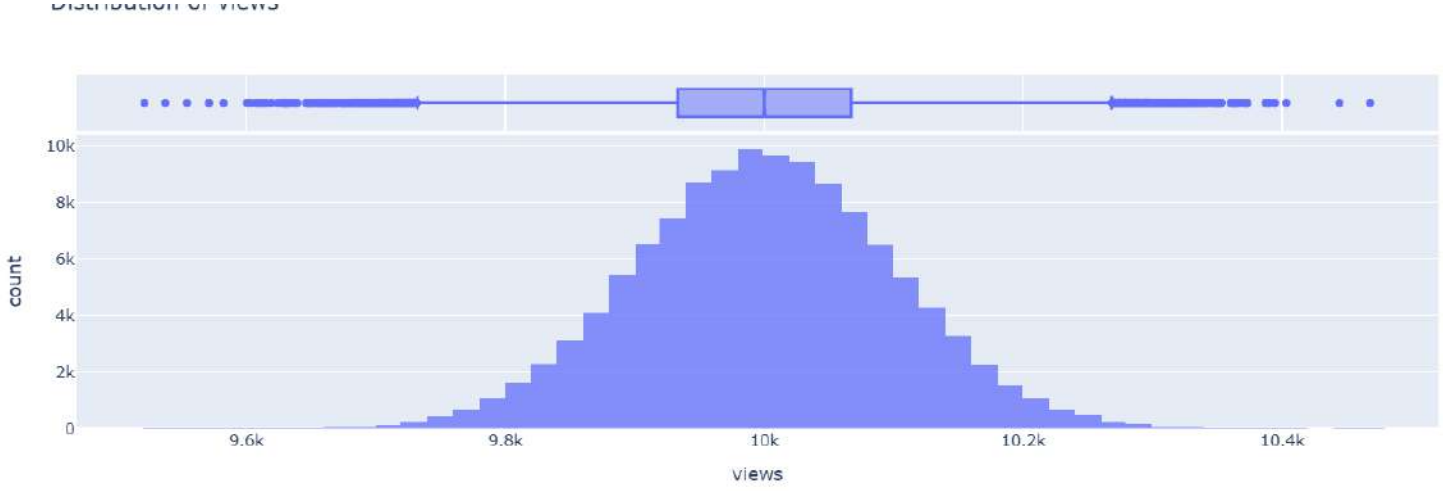
```
...                views          likes       comments  watch_time_minutes  \
count  122400.000000  116283.000000  116288.000000       116295.000000
mean     9999.856283    1099.633618     274.396636        37543.827721
std        99.881260     519.424089     129.741739        12987.724246
min      9521.000000     195.000000      48.000000        14659.105562
25%      9933.000000     650.000000     162.000000        26366.320569
50%     10000.000000    1103.000000     274.000000        37531.990337
75%     10067.000000    1547.000000     387.000000        48777.782090
max     10468.000000    2061.000000     515.000000        61557.670089

       video_length_minutes    subscribers  ad_revenue_usd
count         122400.000000  122400.000000   122400.000000
mean              16.014165  502191.719902      252.727210
std                8.083790  288397.470103       61.957052
min                2.000142    1005.000000      126.590603
25%                9.004695  252507.500000      199.902018
50%               16.005906  503465.500000      252.749699
```

```python
import matplotlib.pyplot as plt
import seaborn as sns
```

Python

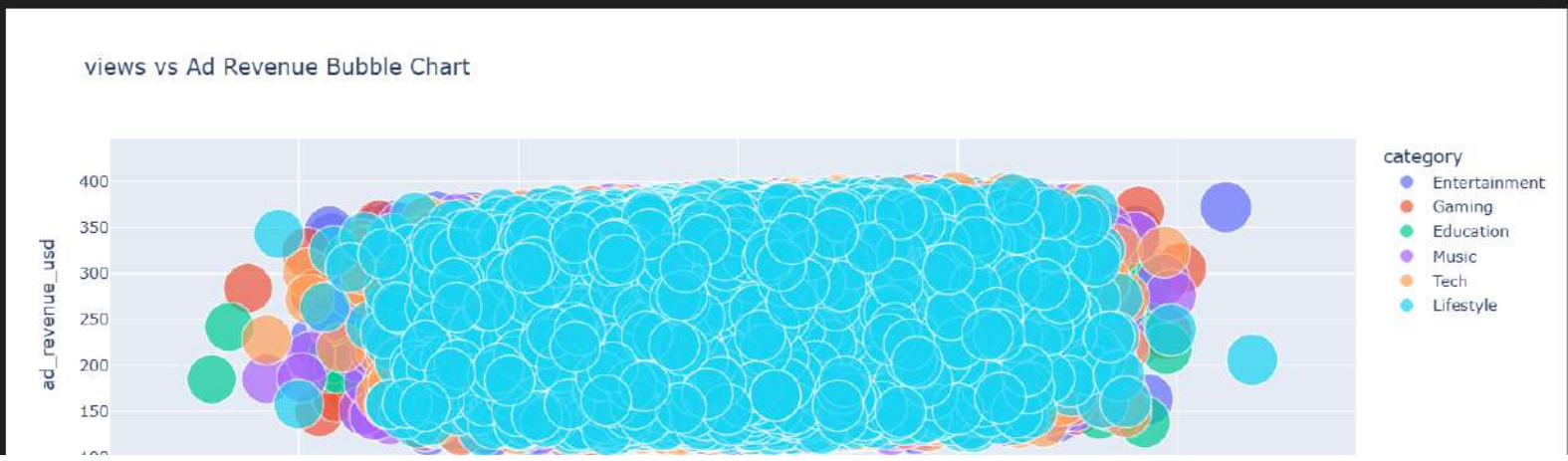
```python
import plotly.express as px

for col in numerical_columns:
    fig = px.histogram(
        utube,
        x=col,
        nbins=50,
        marginal="box",   # box plot above histogram
        title=f"Distribution of {col}",
        opacity=0.75
    )
    fig.show()
```

Python


Distribution of views
10k
8k
6k
4k
2k
0
count
9.6k
9.8k
10k
10.2k
10.4k
views

```
for col in numerical_columns[:-1]:  # exclude target
    fig = px.scatter(
        utube,
        x=col,
        y='ad_revenue_usd',
        size='views',          # bubble size based on views
        color='category',      # optional: color by category
        hover_data=['likes', 'comments'],  # extra info on hover
        title=f"{col} vs Ad Revenue Bubble Chart",
        size_max=40
    )
    fig.show()
```

Pytho



```python
import plotly.express as px

for col in numerical_columns:
    fig = px.box(
        utube,
        x=col,
        points="all",          # show all data points, outliers included
        title=f"Outlier Detection - {col}"
    )
    fig.show()
```

Pytho