In [3]:
```
import pandas as pd
youtube=pd.read_csv(r'C:\Users\sowmi\OneDrive\Desktop\python\youtube\youtube_a
youtube.head()
```

Out[3]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [4]:
```
youtube
```

Out[4]:

	video_id	date	views	likes	comments	watch_time_minute
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.21418
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.74744
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.65849
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.51777
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.66643
...	...	...	...	...	...	.
122395	vid_2902	2024-12-14 10:50:40.993199	9853	1673.0	147.0	42075.70488
122396	vid_3890	2024-07-13 10:50:40.993199	10128	1709.0	63.0	57563.70304
122397	vid_3934	2024-06-10 10:50:40.993199	10267	700.0	NaN	27549.71465
122398	vid_4260	2024-12-22 10:50:40.993199	10240	1616.0	106.0	56967.38438
122399	vid_1056	2024-06-25 10:50:40.993199	9931	770.0	NaN	38466.83713

122400 rows × 12 columns

In [5]:
```
youtube.shape
```

Out[5]:
```
(122400, 12)
```

In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [8]:
```
youtube.describe()
```

Out[8]:

	views	likes	comments	watch_time_minutes	video_
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	

In [9]:
```
youtube.info()
```

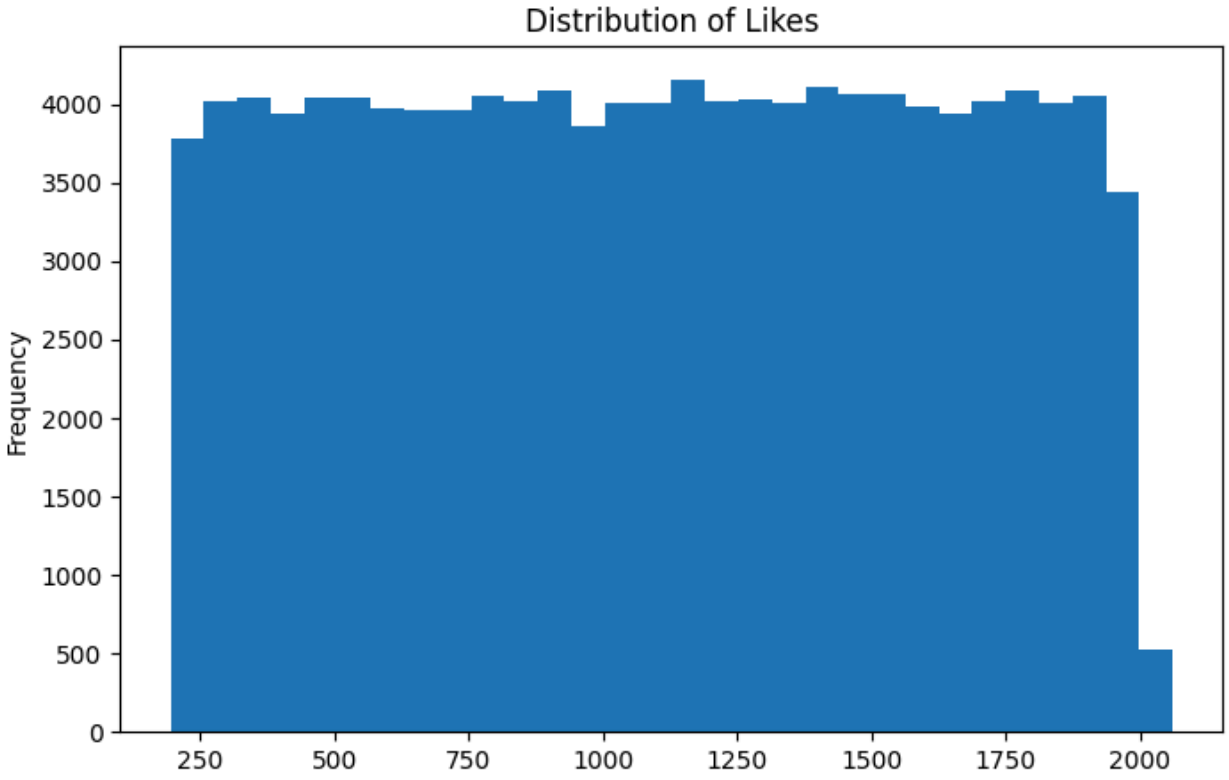
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122400 entries, 0 to 122399
Data columns (total 12 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   video_id              122400 non-null  object 
 1   date                  122400 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
```

In [10]:
```
youtube.isnull().sum()
```

Out[10]:
```
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
```

In [11]:
```
youtube['likes'].dropna().plot(kind='hist', bins=30, figsize=(8,5), title="Dis
```

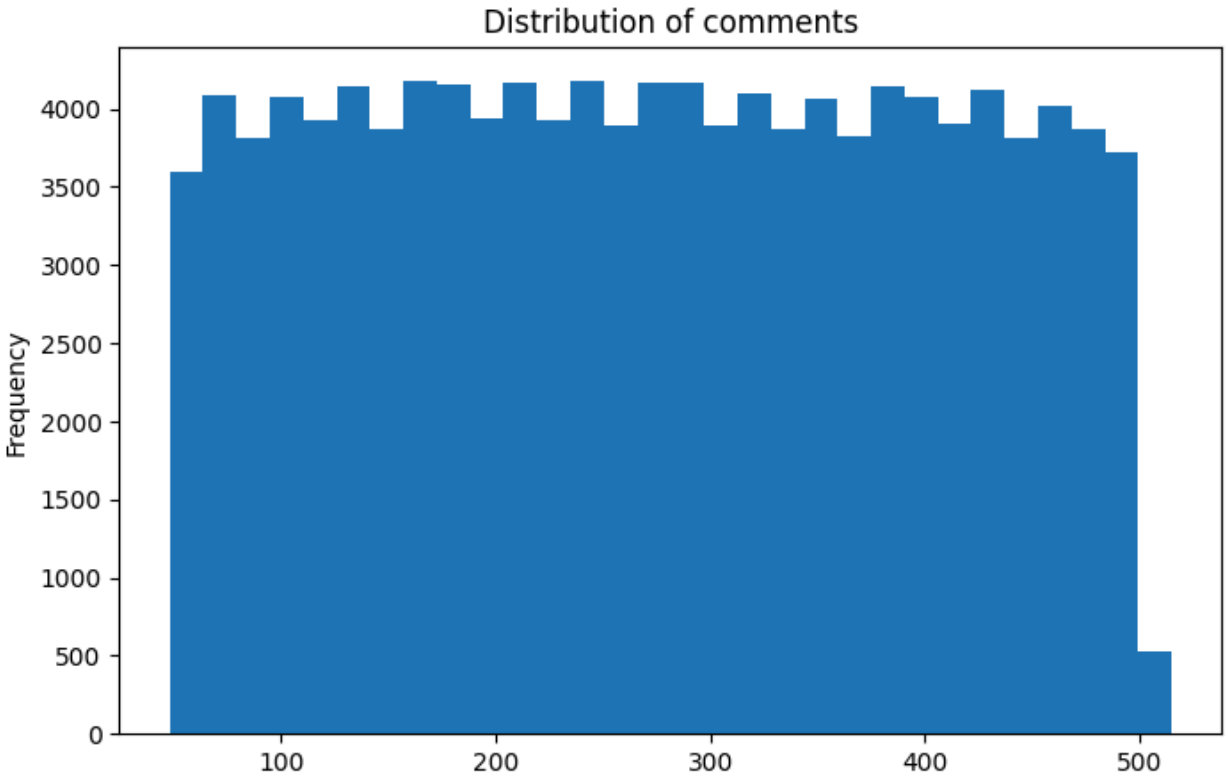
Out[11]:
```
<Axes: title={'center': 'Distribution of Likes'}, ylabel='Frequency'>
```



In [12]:
```
youtube['likes'] = youtube['likes'].fillna(youtube['likes'].median())
```

In [13]:
```
youtube['comments'].dropna().plot(kind='hist', bins=30, figsize=(8,5), title="
```

Out[13]:
```
<Axes: title={'center': 'Distribution of comments'}, ylabel='Frequency'>
```



In [14]:
```
youtube['comments'] = youtube['comments'].fillna(youtube['comments'].median())
```

In [15]:
```
youtube['watch_time_minutes'] = youtube['watch_time_minutes'].fillna(youtube['
```

In [16]:
```
youtube.isnull().sum()
```

Out[16]:
```
video_id                0
date                    0
views                   0
likes                   0
comments                0
watch_time_minutes      0
video_length_minutes    0
subscribers             0
category                0
device                  0
country                 0
ad_revenue_usd          0
dtype: int64
```

In [17]:
```
youtube.info()
```

```
<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                 122400 non-null  float64
 4   comments              122400 non-null  float64
 5   watch_time_minutes    122400 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
```

In [18]:
```
youtube.head()
```

Out[18]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [19]:
```
youtube['date'] = pd.to_datetime(youtube['date'])
```

In [20]:
```
youtube.info()
```

```
<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  datetime64[ns]
 2   views                 122400 non-null  int64
 3   likes                 122400 non-null  float64
 4   comments              122400 non-null  float64
 5   watch_time_minutes    122400 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: datetime64[ns](1), float64(5), int64(2), object(4)
memory usage: 11.2+ MB
```

In [21]:
```
youtube.duplicated().sum()
```

Out[21]:
```
np.int64(2400)
```

In [22]:
```
youtube.drop_duplicates(inplace=True)
```

In [23]:
```
youtube.duplicated().sum()
```

Out[23]:
```
np.int64(0)
```

In [112…
```
youtube['engagement_rate'] = (youtube['likes'] + youtube['comments']) / youtub
youtube.head()
```

Out[112…

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

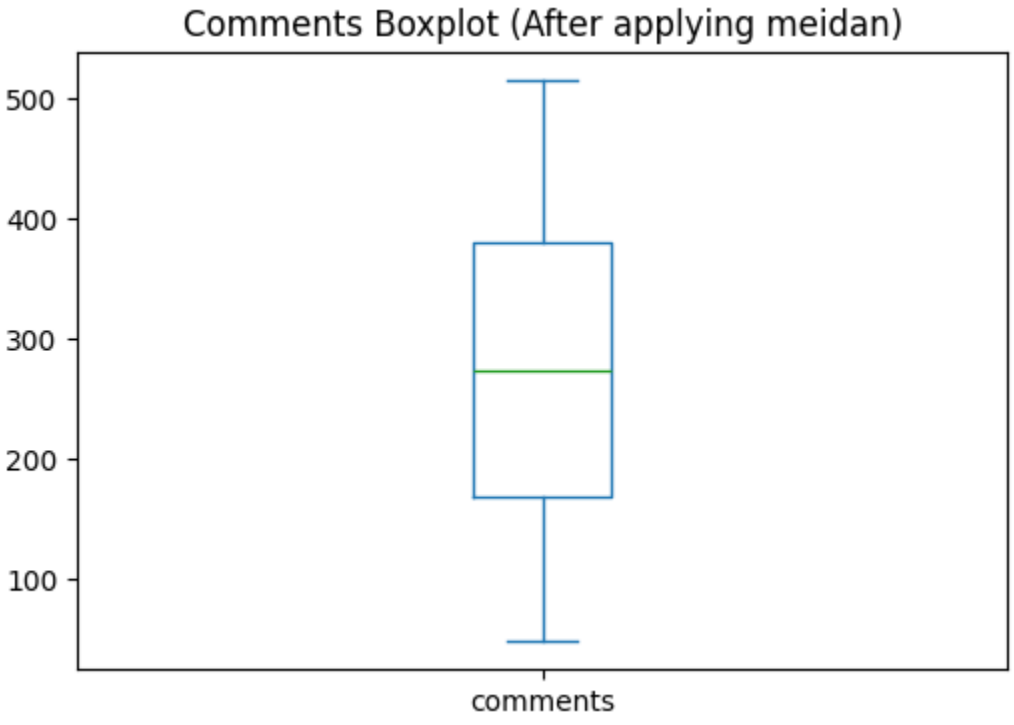
In [24]:
```
youtube.describe()
```

Out[24]:

	date	views	likes	comments	watch_t
count	120000	120000.000000	120000.000000	120000.000000	1
mean	2024-12-08 03:24:11.233198848	9999.832333	1099.755792	274.332350	
min	2024-06-09 10:50:40.993199	9521.000000	195.000000	48.000000	
25%	2024-09-07 10:50:40.993199104	9933.000000	673.000000	168.000000	
50%	2024-12-08 10:50:40.993199104	10000.000000	1103.000000	274.000000	
75%	2025-03-09 10:50:40.993199104	10067.000000	1524.000000	381.000000	
max	2025-06-08 10:50:40.993199	10468.000000	2061.000000	515.000000	
std	NaN	99.918405	506.372458	126.461529	

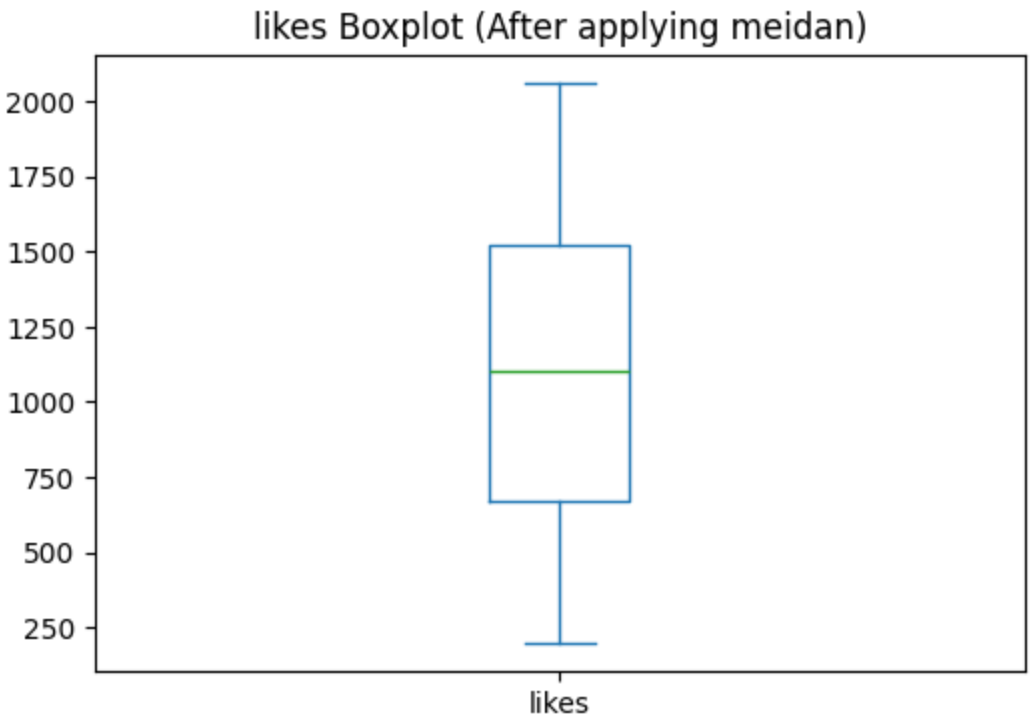
In [25]:
```
youtube['comments'].plot(kind='box', figsize=(6,4), title="Comments Boxplot (A
```

Out[25]:
```
<Axes: title={'center': 'Comments Boxplot (After applying meidan)'}>
```



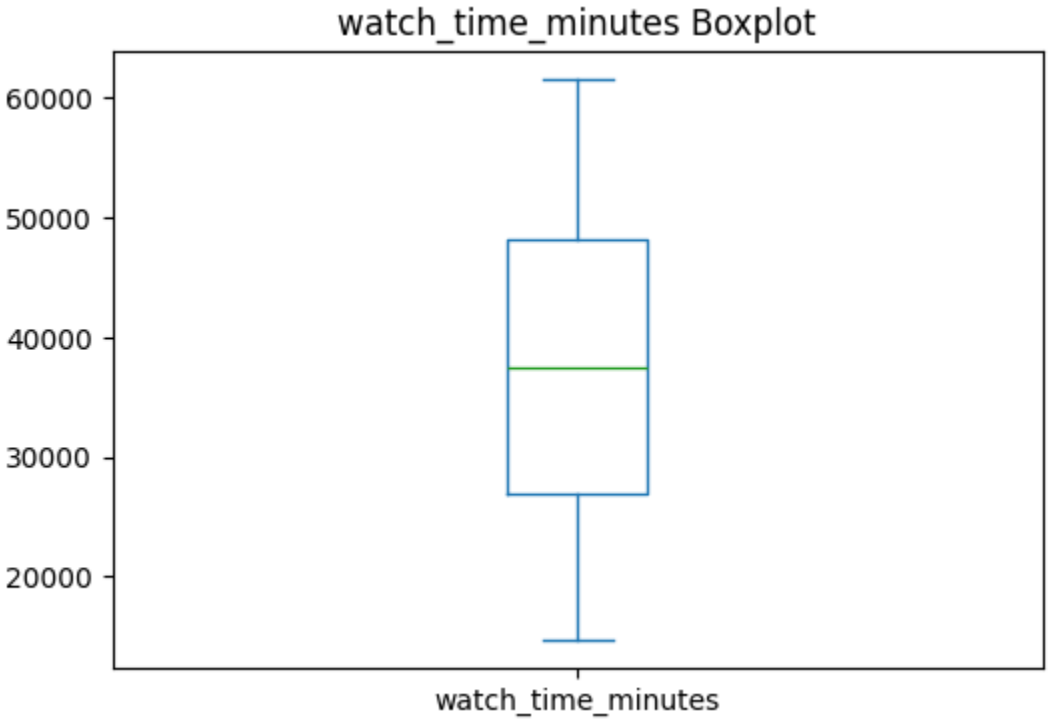
In [26]:
```
youtube['likes'].plot(kind='box', figsize=(6,4), title="likes Boxplot (After a
```

Out[26]:
```
<Axes: title={'center': 'likes Boxplot (After applying meidan)'}>
```



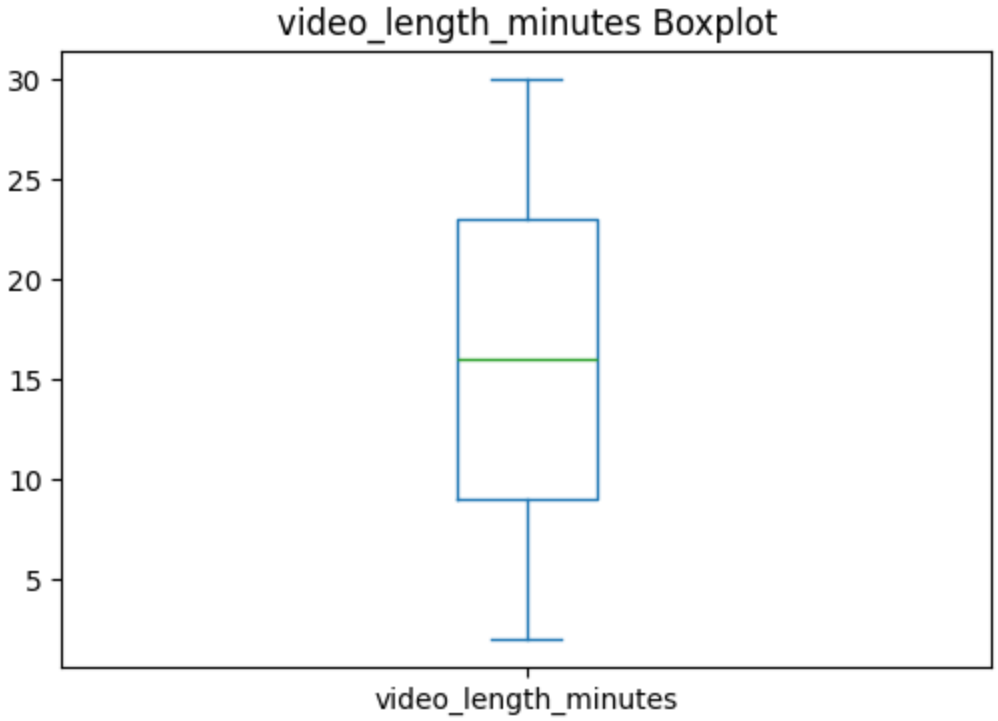
In [27]:
```
youtube['watch_time_minutes'].plot(kind='box', figsize=(6,4), title="watch_tim
```

Out[27]:
```
<Axes: title={'center': 'watch_time_minutes Boxplot'}>
```

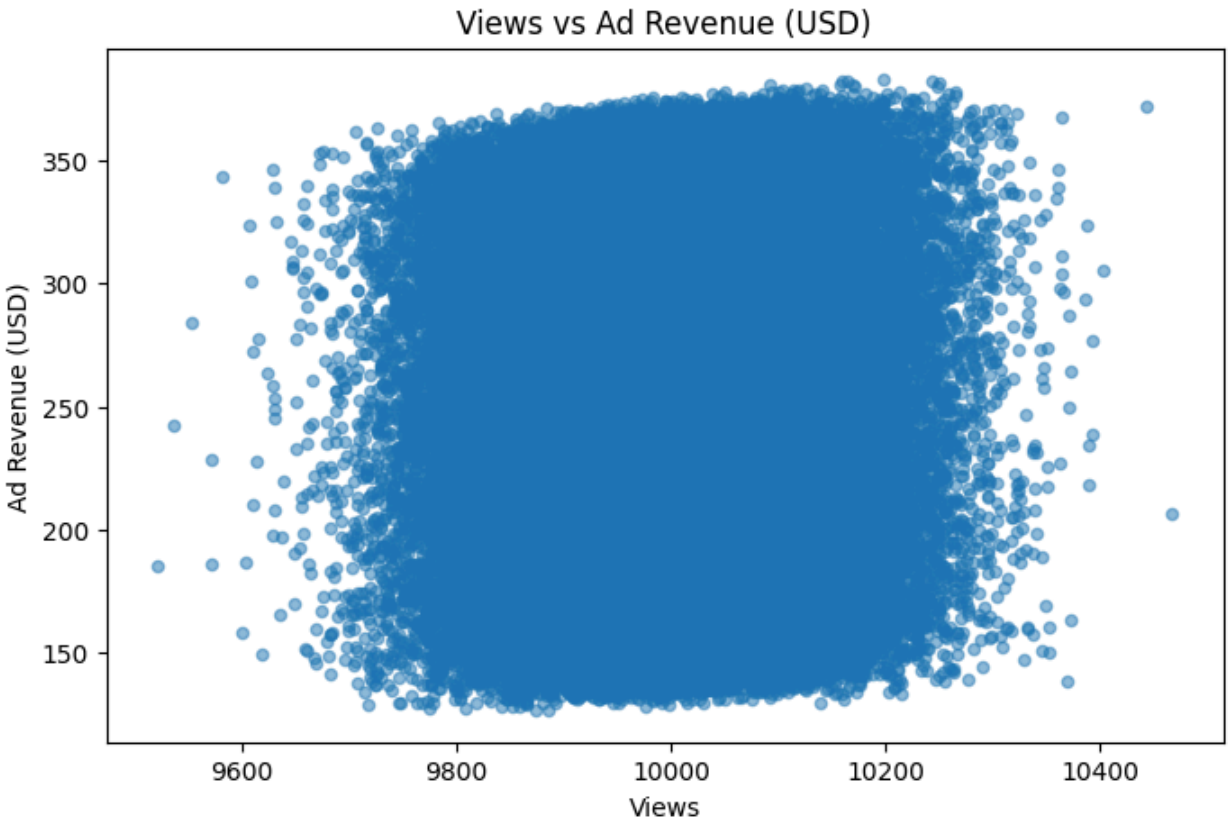


In [28]:
```
youtube['video_length_minutes'].plot(kind='box', figsize=(6,4), title="video_l
```

Out[28]:
```
<Axes: title={'center': 'video_length_minutes Boxplot'}>
```

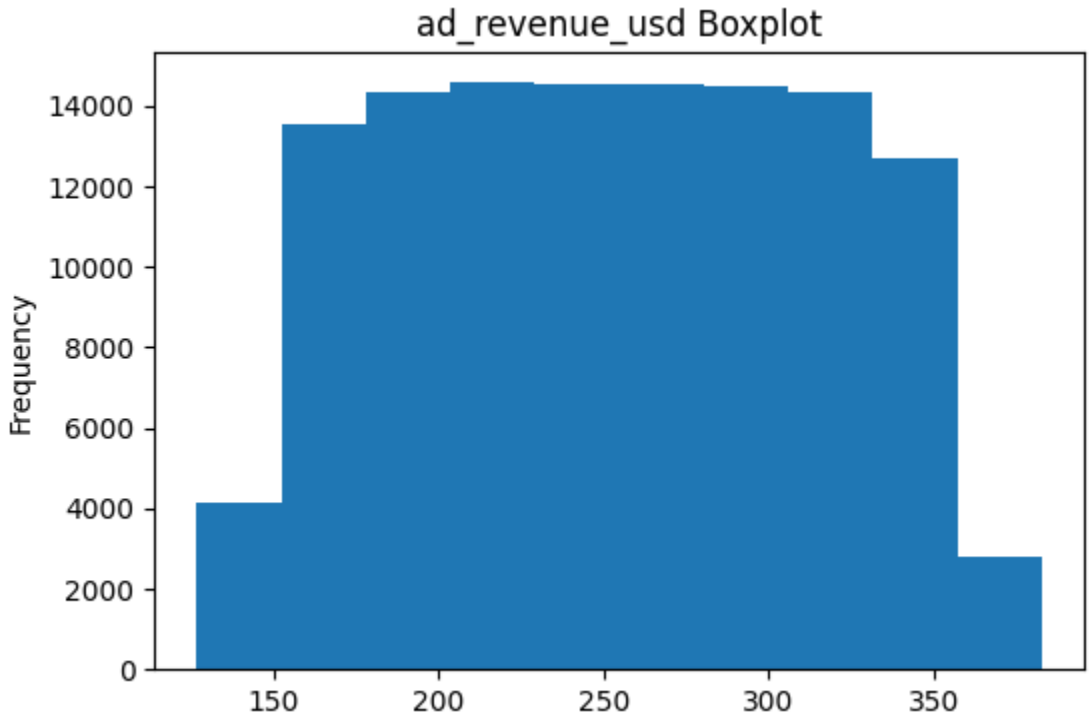


In [29]:
```
import matplotlib.pyplot as plt
youtube.plot(kind='scatter', x='views', y='ad_revenue_usd', figsize=(8,5), alp
plt.title("Views vs Ad Revenue (USD)")
plt.xlabel("Views")
plt.ylabel("Ad Revenue (USD)")
plt.show()
```



In [30]:
```
youtube['ad_revenue_usd'].plot(kind='hist', figsize=(6,4), title="ad_revenue_u
```

Out[30]:
```
<Axes: title={'center': 'ad_revenue_usd Boxplot'}, ylabel='Frequency'>
```



In [34]:
```
# Identify categorical columns
```

```
categorical_cols = youtube.select_dtypes(include=['object', 'category']).colum
print(categorical_cols)
```

```
Index(['video_id', 'category', 'device', 'country'], dtype='object')
```

In [32]:
```
youtube['category'].value_counts()
```

Out[32]:
```
category
Education        20123
Music            20065
Tech             20028
Entertainment    20025
Gaming           19974
Lifestyle        19785
Name: count, dtype: int64
```

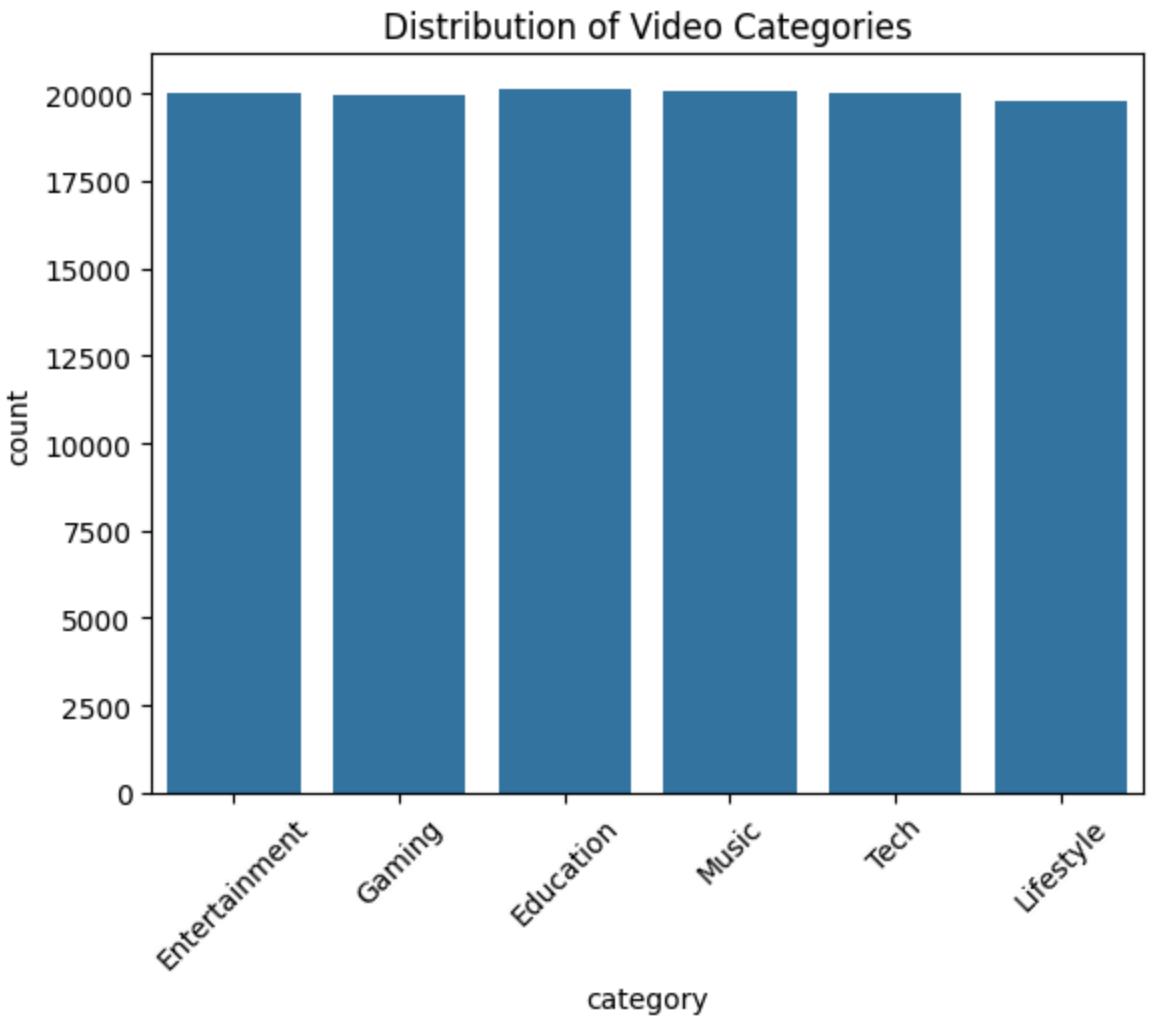
In [45]:
```
youtube['device'].value_counts()
```

Out[45]:
```
device
TV         30086
Mobile     29989
Desktop    29984
Tablet     29941
Name: count, dtype: int64
```

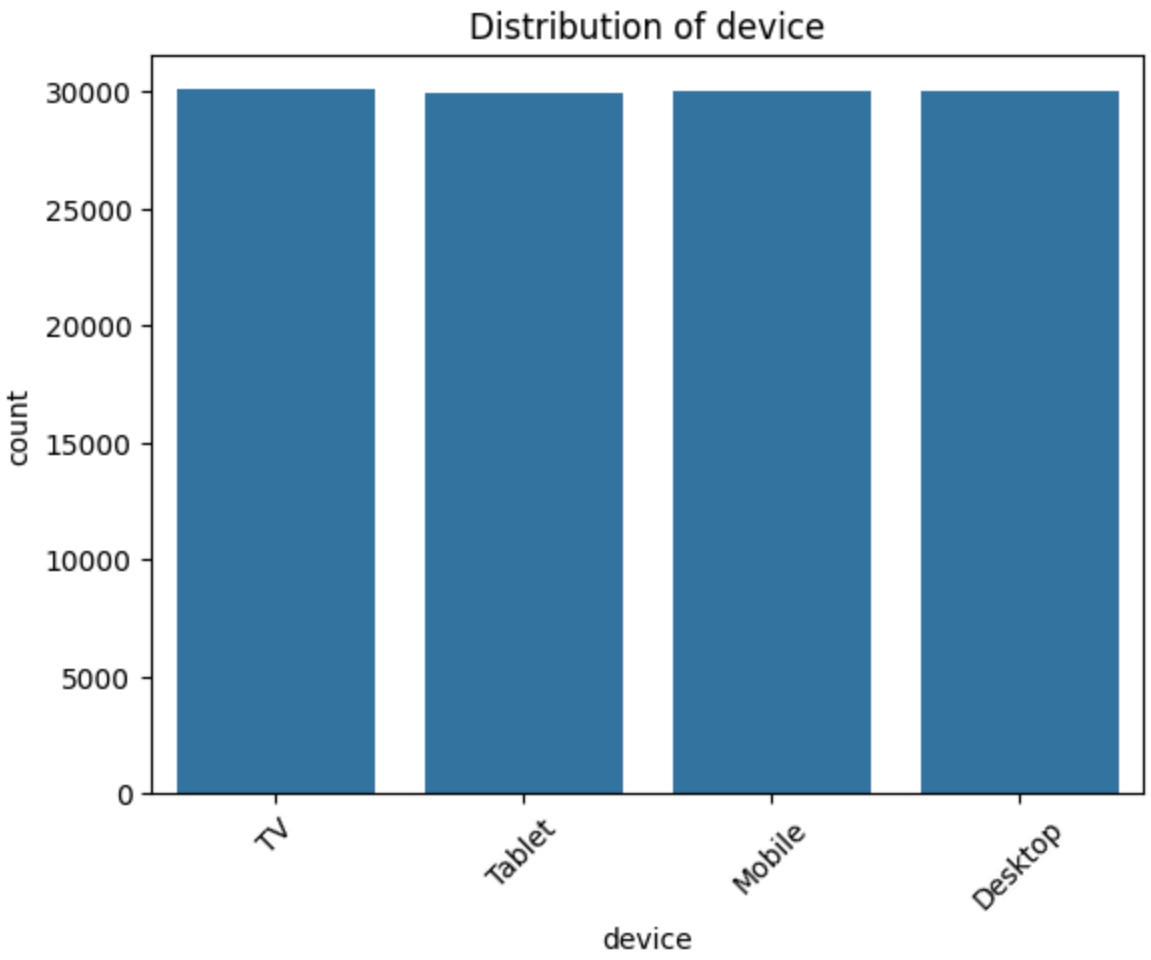
In [46]:
```
youtube['country'].value_counts()
```

Out[46]:
```
country
CA    20198
DE    20160
IN    20156
AU    19911
UK    19893
US    19682
Name: count, dtype: int64
```

In [47]:
```
sns.countplot(x='category', data=youtube)
plt.xticks(rotation=45)
plt.title("Distribution of Video Categories")
plt.show()
```



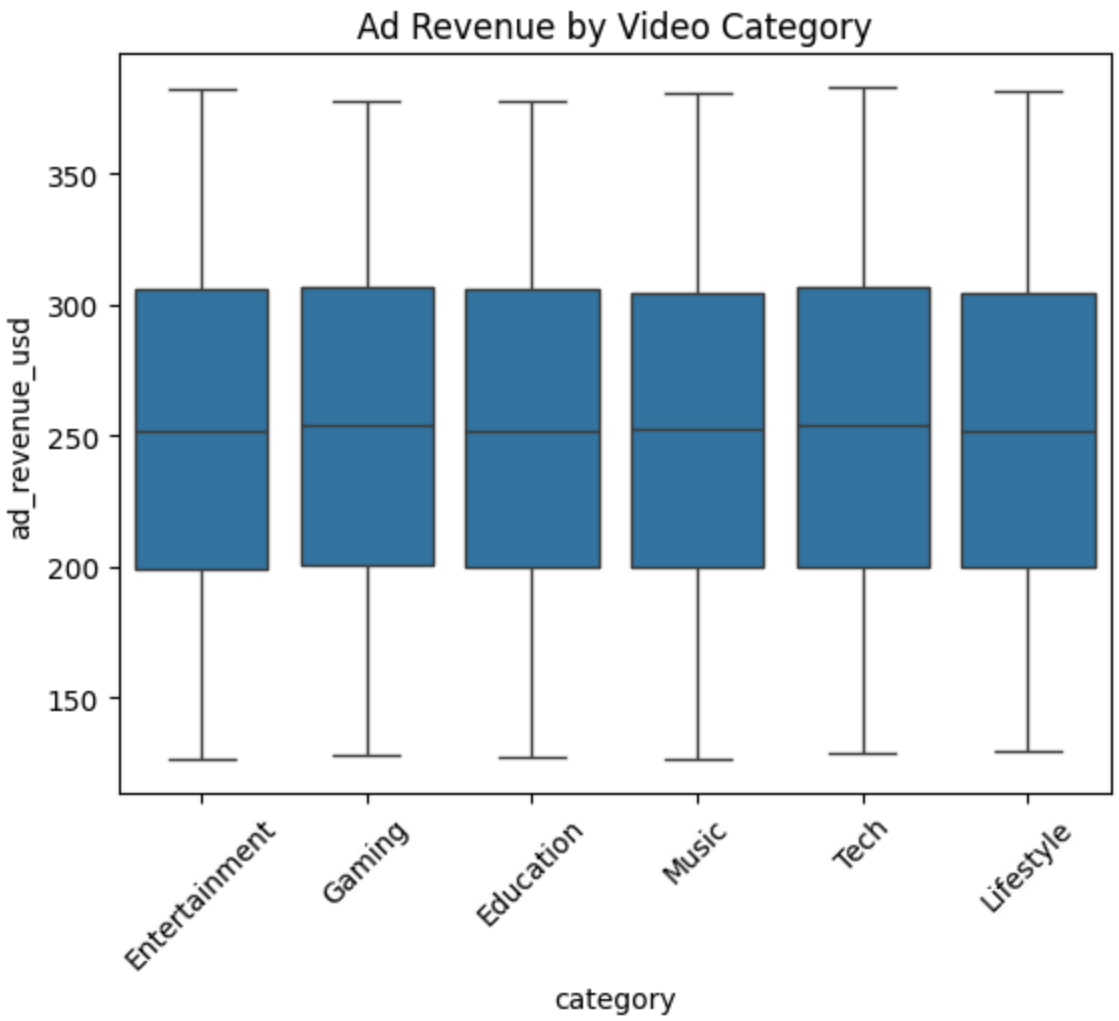
In [48]:
```
sns.countplot(x='device', data=youtube)
plt.xticks(rotation=45)
plt.title("Distribution of device")
plt.show()
```



In [49]:
```
youtube.groupby('category')['views'].mean().sort_values()
```

Out[49]:
```
category
Tech              9999.009986
Gaming            9999.773706
Lifestyle         9999.808137
Education         9999.914078
Entertainment    10000.106367
Music            10000.379915
Name: views, dtype: float64
```

In [50]:
```
sns.boxplot(x='category', y='ad_revenue_usd', data=youtube)
plt.xticks(rotation=45)
plt.title("Ad Revenue by Video Category")
plt.show()
```



In [52]:
```
corr_value = youtube['views'].corr(youtube['ad_revenue_usd'])
print(f"Correlation between views and ad revenue: {corr_value:.2f}")
# Correlation is positive and moderate, indicating that as views increase, ad
```

```
Correlation between views and ad revenue: 0.04
```

In [54]:
```
print(youtube.columns)
```

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

In [55]:
```
# Example formula: (likes + comments) / views
youtube["engagement_rate"] = (youtube["likes"] + youtube["comments"]) / youtub
```

In [56]:
```
corr_value = youtube["engagement_rate"].corr(youtube["ad_revenue_usd"])
print(f"Correlation between engagement rate and ad revenue: {corr_value:.2f}")
```

```
Correlation between engagement rate and ad revenue: 0.15
```

In [57]:
```
corr_value = youtube['video_length_minutes'].corr(youtube['ad_revenue_usd'])
print(f"Correlation between video length minutes and ad revenue: {corr_value:.
```

```
Correlation between video length minutes and ad revenue: 0.00
```

In [58]:
```
corr_value = youtube['watch_time_minutes'].corr(youtube['ad_revenue_usd'])
print(f"Correlation between watch time minutes and ad revenue: {corr_value:.2f
#very strong correlation (0.96) between watch time (minutes) and ad revenue.
```

```
Correlation between watch time minutes and ad revenue: 0.96
```

In [59]:
```
corr_value = youtube['subscribers'].corr(youtube['ad_revenue_usd'])
print(f"Correlation between subscribers and ad revenue: {corr_value:.2f}")
```

```
Correlation between subscribers and ad revenue: 0.01
```

In [50]:
```
pip install scipy
```

```
Requirement already satisfied: scipy in c:\users\sowmi\onedrive\desktop\python\
youtube\venv\lib\site-packages (1.16.2)
Requirement already satisfied: numpy<2.6,>=1.25.2 in c:\users\sowmi\onedrive\de
sktop\python\youtube\venv\lib\site-packages (from scipy) (2.3.3)
Note: you may need to restart the kernel to use updated packages.
```

In [60]:
```
import scipy.stats as stats
groups = [youtube[youtube['category'] == cat]['ad_revenue_usd'] for cat in you
f_stat, pval = stats.f_oneway(*groups)

print("ANOVA F-stat:", f_stat)
print("p-value:", pval)
#Statistically, there’s no strong evidence that ad revenue differs across vide
```

```
ANOVA F-stat: 1.9170641627638705
p-value: 0.08788330768704347
```

In [61]:
```
groups = [youtube[youtube['country'] == cat]['ad_revenue_usd'] for cat in yout
f_stat, pval = stats.f_oneway(*groups)

print("ANOVA F-stat:", f_stat)
print("p-value:", pval)
```

```
ANOVA F-stat: 0.38138224024052614
p-value: 0.8618687432381148
```

In [62]:
```
youtube.head()
```

Out[62]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [63]:
```
youtube['date'] = pd.to_datetime(youtube['date'])

youtube['year'] = youtube['date'].dt.year
youtube['month'] = youtube['date'].dt.month
youtube['day_of_weak'] = youtube['date'].dt.dayofweek  # 0=Mon, 6=Sun
youtube['hour'] = youtube['date'].dt.hour
youtube.head()
```

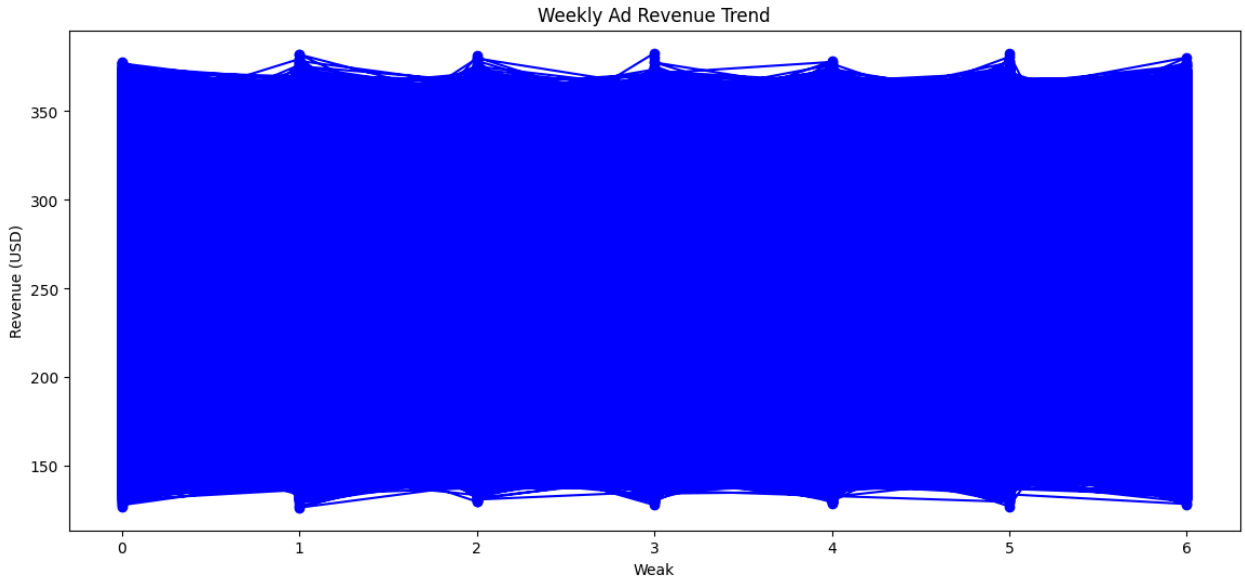
Out[63]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [64]:
```
correlation = youtube['day_of_weak'].corr(youtube['ad_revenue_usd'])
print("Correlation between Weak and Revenue:", correlation)
```

```
Correlation between Weak and Revenue: -0.0010100738071614941
```

In [65]:
```
plt.figure(figsize=(14,6))
plt.plot(youtube['day_of_weak'], youtube['ad_revenue_usd'], marker='o', color=
plt.title("Weekly Ad Revenue Trend")
plt.xlabel("Weak")
plt.ylabel("Revenue (USD)")
plt.show()
```



In [66]:
```
youtube.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120000 entries, 0 to 122399
Data columns (total 17 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   video_id              120000 non-null  object
 1   date                  120000 non-null  datetime64[ns]
 2   views                 120000 non-null  int64
 3   likes                 120000 non-null  float64
 4   comments              120000 non-null  float64
 5   watch_time_minutes    120000 non-null  float64
 6   video_length_minutes  120000 non-null  float64
 7   subscribers           120000 non-null  int64
 8   category              120000 non-null  object
 9   device                120000 non-null  object
 10  country               120000 non-null  object
 11  ad_revenue_usd        120000 non-null  float64
 12  engagement_rate       120000 non-null  float64
 13  year                  120000 non-null  int32
 14  month                 120000 non-null  int32
 15  day_of_weak           120000 non-null  int32
 16  hour                  120000 non-null  int32
dtypes: datetime64[ns](1), float64(6), int32(4), int64(2), object(4)
memory usage: 14.6+ MB
```

In [67]:
```
x=youtube.drop('ad_revenue_usd', axis=1)
y=youtube['ad_revenue_usd']
```

In [68]:
```
x.head()
```

Out[68]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [69]:
```
# dropping subscribers, video_length_minutes
X = x.drop(['video_id','subscribers', 'video_length_minutes','date'], axis=1)
```

In [70]:
```
x.head()
```

Out[70]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [71]:
```
y.head()
```

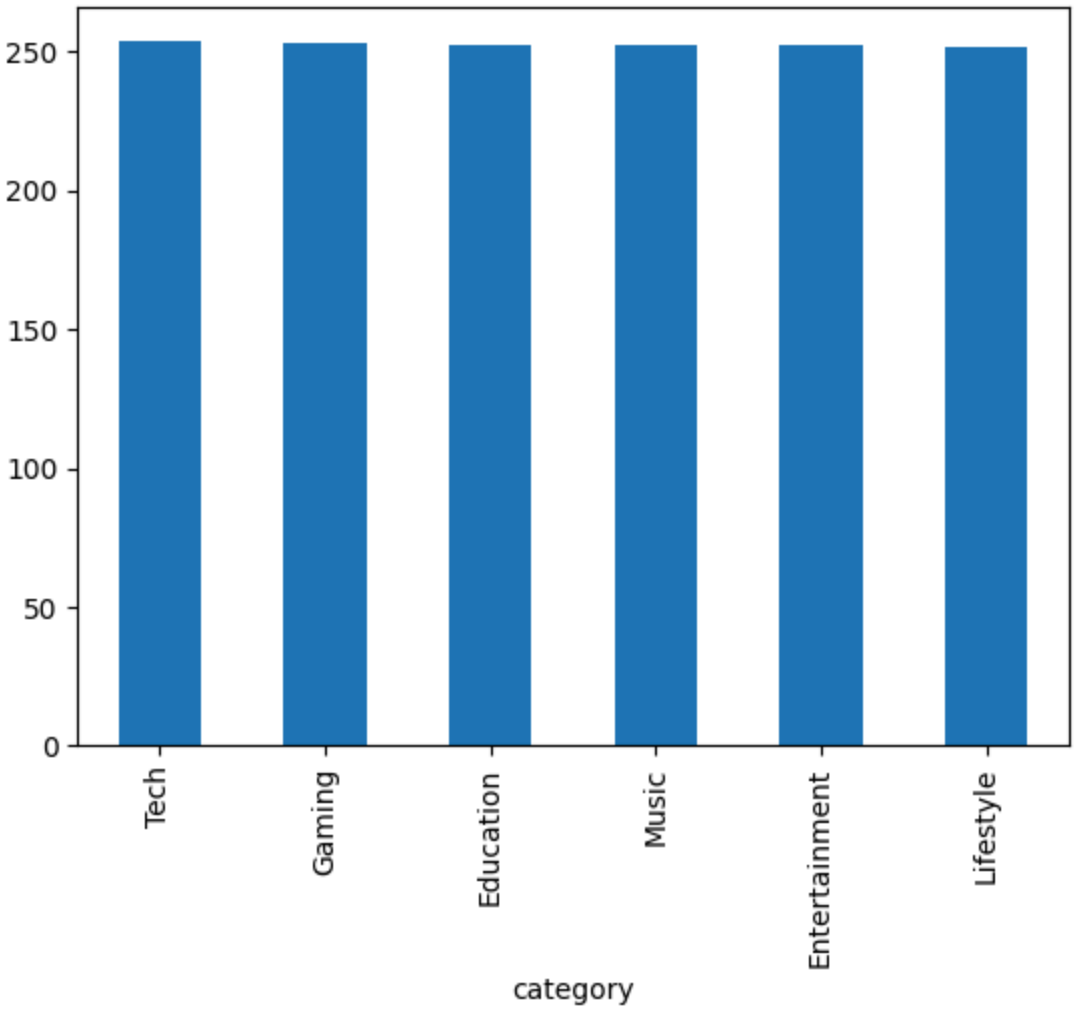
Out[71]:
```
0    203.178237
1    140.880508
2    360.134008
3    224.638261
4    165.514388
Name: ad_revenue_usd, dtype: float64
```

In [72]:
```
categorical_cols = X.select_dtypes(include=['object', 'category']).columns
print(categorical_cols)
```

```
Index(['category', 'device', 'country'], dtype='object')
```

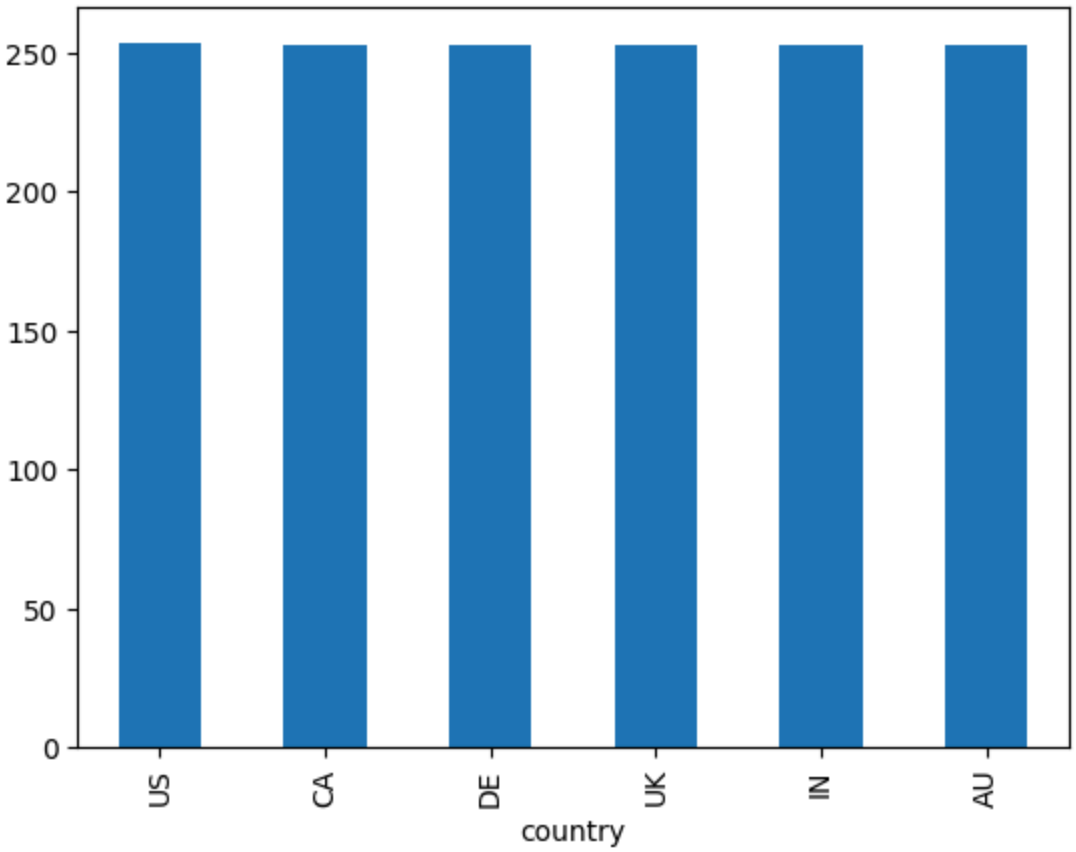
In [73]:
```
youtube.groupby(['category'])['ad_revenue_usd'].mean().sort_values(ascending=F
```

Out[73]:
```
<Axes: xlabel='category'>
```



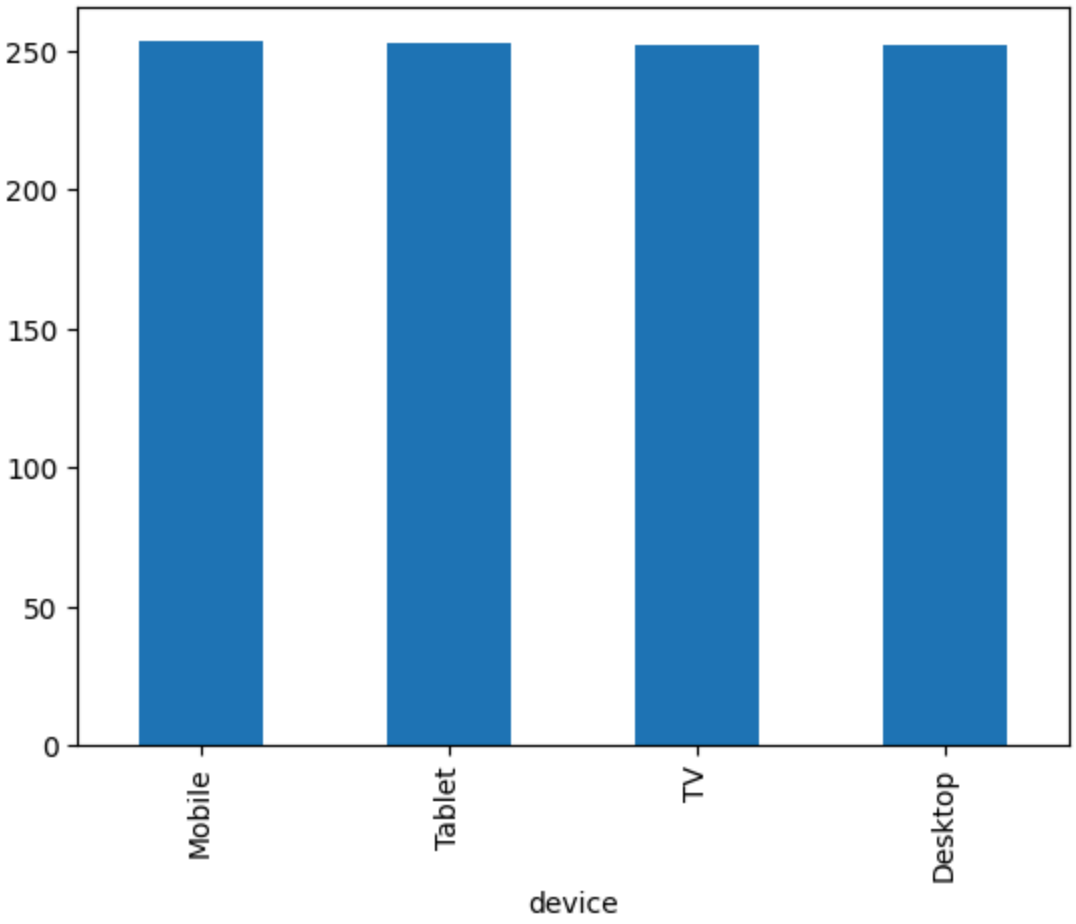
In [74]:
```
youtube.groupby(['country'])['ad_revenue_usd'].mean().sort_values(ascending=Fa
```

Out[74]: <Axes: xlabel='country'>



In [75]:
```
youtube.groupby(['device'])['ad_revenue_usd'].mean().sort_values(ascending=Fal
```

Out[75]: <Axes: xlabel='device'>



In [67]:
```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\sowmi\onedrive\desktop\
python\youtube\venv\lib\site-packages (1.7.2)
Requirement already satisfied: numpy>=1.22.0 in c:\users\sowmi\onedrive\deskto
p\python\youtube\venv\lib\site-packages (from scikit-learn) (2.3.3)
Requirement already satisfied: scipy>=1.8.0 in c:\users\sowmi\onedrive\desktop\
python\youtube\venv\lib\site-packages (from scikit-learn) (1.16.2)
Requirement already satisfied: joblib>=1.2.0 in c:\users\sowmi\onedrive\deskto
p\python\youtube\venv\lib\site-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\sowmi\onedrive\
desktop\python\youtube\venv\lib\site-packages (from scikit-learn) (3.6.0)
Note: you may need to restart the kernel to use updated packages.
```

In [76]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

In [77]:
```
x.head()
```

Out[77]:

	video_id	date	views	likes	comments	watch_time_minutes	vid
0	vid_3092	2024-09-24 10:50:40.993199	9936	1221.0	320.0	26497.214184	
1	vid_3459	2024-09-22 10:50:40.993199	10017	642.0	346.0	15209.747445	
2	vid_4784	2024-11-21 10:50:40.993199	10097	1979.0	187.0	57332.658498	
3	vid_4078	2025-01-28 10:50:40.993199	10034	1191.0	242.0	31334.517771	
4	vid_3522	2025-04-28 10:50:40.993199	9889	1858.0	477.0	15665.666434	

In [78]:
```
y.head()
```

Out[78]:
```
0    203.178237
1    140.880508
2    360.134008
3    224.638261
4    165.514388
Name: ad_revenue_usd, dtype: float64
```

In [79]:
```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 120000 entries, 0 to 122399
Data columns (total 16 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   video_id              120000 non-null  object
 1   date                  120000 non-null  datetime64[ns]
 2   views                 120000 non-null  int64
 3   likes                 120000 non-null  float64
 4   comments              120000 non-null  float64
 5   watch_time_minutes    120000 non-null  float64
 6   video_length_minutes  120000 non-null  float64
 7   subscribers           120000 non-null  int64
 8   category              120000 non-null  object
 9   device                120000 non-null  object
 10  country               120000 non-null  object
 11  engagement_rate       120000 non-null  float64
 12  year                  120000 non-null  int32
 13  month                 120000 non-null  int32
 14  day_of_weak           120000 non-null  int32
 15  hour                  120000 non-null  int32
dtypes: datetime64[ns](1), float64(5), int32(4), int64(2), object(4)
memory usage: 13.7+ MB
```

In [80]:
```
x_cat=X[['category', 'country', 'device']]
x_num=X[['engagement_rate', 'views', 'likes', 'comments', 'watch_time_minutes'
```

In [81]:
```
x_num
```

Out[81]:

	engagement_rate	views	likes	comments	watch_time_minutes
0	0.155093	9936	1221.0	320.0	26497.214184
1	0.098632	10017	642.0	346.0	15209.747445
2	0.214519	10097	1979.0	187.0	57332.658498
3	0.142814	10034	1191.0	242.0	31334.517771
4	0.236121	9889	1858.0	477.0	15665.666434
...	...	...	...	...	...
122395	0.184715	9853	1673.0	147.0	42075.704885
122396	0.174961	10128	1709.0	63.0	57563.703040
122397	0.094867	10267	700.0	274.0	27549.714659
122398	0.168164	10240	1616.0	106.0	56967.384382
122399	0.105125	9931	770.0	274.0	38466.837135

120000 rows × 5 columns

In [82]:
```
x_cat
```

Out[82]:

	category	country	device
0	Entertainment	IN	TV
1	Gaming	CA	Tablet
2	Education	CA	TV
3	Entertainment	UK	Mobile
4	Education	CA	Mobile
...	...	...	...
122395	Education	US	Tablet
122396	Music	UK	Desktop
122397	Tech	CA	Tablet
122398	Music	UK	Mobile
122399	Tech	CA	TV

120000 rows × 3 columns

In [83]:
```
for col in x_num + x_cat:
    if col not in X.columns:
        print("⚠ Not found:", col)
```

```
    else:
        print("🔍 Found:", col)
```

```
🔍 Found: category
🔍 Found: comments
🔍 Found: country
🔍 Found: device
🔍 Found: engagement_rate
🔍 Found: likes
🔍 Found: views
🔍 Found: watch_time_minutes
```

In [84]:
```
print(type(X))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [85]:
```
print("Num features:", x_num, type(x_num))
print("Cat features:", x_cat, type(x_cat))

for col in x_num + x_cat:
    print(col, "->", col in X.columns)
```

```
Num features:         engagement_rate  views   likes  comments  watch_time_minu
tes
0               0.155093   9936  1221.0     320.0        26497.214184
1               0.098632  10017   642.0     346.0        15209.747445
2               0.214519  10097  1979.0     187.0        57332.658498
3               0.142814  10034  1191.0     242.0        31334.517771
4               0.236121   9889  1858.0     477.0        15665.666434
...                  ...    ...     ...       ...                 ...
122395          0.184715   9853  1673.0     147.0        42075.704885
122396          0.174961  10128  1709.0      63.0        57563.703040
122397          0.094867  10267   700.0     274.0        27549.714659
122398          0.168164  10240  1616.0     106.0        56967.384382
122399          0.105125   9931   770.0     274.0        38466.837135

[120000 rows x 5 columns] <class 'pandas.core.frame.DataFrame'>
Cat features:              category country   device
0       Entertainment      IN       TV
1              Gaming      CA   Tablet
2           Education      CA       TV
3       Entertainment      UK   Mobile
4           Education      CA   Mobile
...               ...     ...      ...
122395      Education      US   Tablet
122396          Music      UK  Desktop
122397           Tech      CA   Tablet
122398          Music      UK   Mobile
122399           Tech      CA       TV

[120000 rows x 3 columns] <class 'pandas.core.frame.DataFrame'>
category -> True
comments -> True
country -> True
device -> True
engagement_rate -> True
likes -> True
views -> True
watch_time_minutes -> True
```

In [86]:

```python
import pickle
num_features = ["engagement_rate", "views", "likes", "comments", "watch_time_m
cat_features = ["category", "country", "device"]

preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_features),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat_features)
    ]
)

# Pipeline
preprocess_pipeline = Pipeline(steps=[("preprocessor", preprocessor)])

# Fit + Transform
X_processed = preprocess_pipeline.fit_transform(X)
```

```
print("Shape before processing:", X.shape)
print("Shape after processing:", X_processed.shape)
```

```
Shape before processing: (120000, 12)
Shape after processing: (120000, 21)
```

In [92]:
```
from sklearn.model_selection import train_test_split

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

In [93]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
import pickle

# Features
num_features = ["views", "likes", "comments", "watch_time_minutes", "engagemer
cat_features = ["category", "device", "country"]

# Preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_features),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat_features)
    ]
)

# Final pipeline = Preprocessor + Linear Regression
lr_pipeline = Pipeline(steps=[
    ("preprocessor", preprocessor),
    ("model", LinearRegression())
])
# Train pipeline
lr_pipeline.fit(X_train, y_train)

# 📦 Save (Pickle) the pipeline
with open("linear_regression_pipeline.pkl", "wb") as f:
    pickle.dump(lr_pipeline, f)
```

In [94]:
```
X_train, X_test, y_train, y_test = train_test_split(X_processed, y, test_size=
```

In [95]:
```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
```

In [96]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

```
y_pred = lr.predict(X_test)
print("R²:", r2_score(y_test, y_pred))
```

```
R²: 0.9525715958524404
```

In [97]:
```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)
print("R²:", r2)
print("RMSE:", rmse)
print("MAE:", mae)
```

```
R²: 0.9525715958524404
RMSE: 13.480384120734533
MAE: 3.109322363017399
```

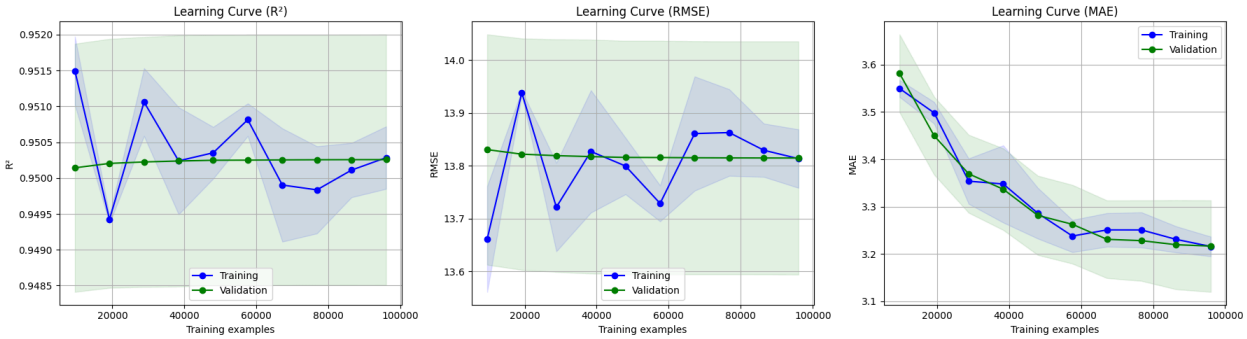
In [98]:
```
Train_y_pred=lr.predict(X_train)
Test_y_pred=lr.predict(X_test)
```

In [99]:
```
print('Train MAE:',mean_absolute_error(y_train,Train_y_pred))
print('Test MAE:',mean_absolute_error(y_test,Test_y_pred))
print('Train R2:',r2_score(y_train,Train_y_pred))
print('Test R2:',r2_score(y_test,Test_y_pred))
```

```
Train MAE: 3.261322124748466
Test MAE: 3.109322363017399
Train R2: 0.949708712825048
Test R2: 0.9525715958524404
```

In [100…
```
import matplotlib.pyplot as plt
from sklearn.model_selection import learning_curve
import numpy as np

def plot_learning_curves(model, X, y):
    # Define metrics
    scoring_metrics = {
        "R²": "r2",
        "RMSE": "neg_root_mean_squared_error",
        "MAE": "neg_mean_absolute_error"
    }

    fig, axes = plt.subplots(1, 3, figsize=(18, 5))

    for ax, (metric_name, scoring) in zip(axes, scoring_metrics.items()):
        train_sizes, train_scores, test_scores = learning_curve(
            model, X, y, cv=5, scoring=scoring, n_jobs=-1,
            train_sizes=np.linspace(0.1, 1.0, 10)
        )

        # Fix sklearn convention (negative errors for RMSE/MAE)
        if "neg" in scoring:
            train_scores = -train_scores
            test_scores = -test_scores
```

```
        train_mean = np.mean(train_scores, axis=1)
        test_mean = np.mean(test_scores, axis=1)
        train_std = np.std(train_scores, axis=1)
        test_std = np.std(test_scores, axis=1)

        # Plot curves
        ax.plot(train_sizes, train_mean, 'o-', color="blue", label="Training")
        ax.plot(train_sizes, test_mean, 'o-', color="green", label="Validation

        ax.fill_between(train_sizes, train_mean-train_std, train_mean+train_st
        ax.fill_between(train_sizes, test_mean-test_std, test_mean+test_std, a

        ax.set_title(f"Learning Curve ({metric_name})")
        ax.set_xlabel("Training examples")
        ax.set_ylabel(metric_name)
        ax.legend(loc="best")
        ax.grid()

    plt.tight_layout()
    plt.show()
   # One-hot encode categorical columns in X
X_numeric = pd.get_dummies(X, drop_first=True)

# Call the function with numeric features
plot_learning_curves(lr, X_numeric, y)
```



In [88]:
```
#ELASTIC NET MODEL
```

In [101…
```
from sklearn.linear_model import  Ridge, Lasso, ElasticNet
elastic_net = ElasticNet(alpha=0.1, l1_ratio=0.5, random_state=42)
elastic_net.fit(X_train, y_train)
y_pred_train = elastic_net.predict(X_train)
y_pred_test = elastic_net.predict(X_test)
train_r2 = r2_score(y_train, y_pred_train)
test_r2 = r2_score(y_test, y_pred_test)
mae = mean_absolute_error(y_test, y_pred_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred_test))

print("Train R²:", train_r2)
print("Test R²:", test_r2)
print("Test MAE:", mae)
print("Test RMSE:", rmse)
```

```
Train R²: 0.9475089221327974
Test R²: 0.9503472714113815
Test MAE: 5.248553970175287
Test RMSE: 13.792867724244145
```

In [102…

```python
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Initialize model (you can tune max_depth, min_samples_split etc.)
dt = DecisionTreeRegressor(random_state=42, max_depth=10)

# Fit the model
dt.fit(X_train, y_train)

# Predictions
y_train_pred = dt.predict(X_train)
y_test_pred = dt.predict(X_test)

# Metrics
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)

train_mae = mean_absolute_error(y_train, y_train_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)

train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))
print("🔷 Decision Tree Results 🔷")
print(f"Train R²: {train_r2:.4f}, Test R²: {test_r2:.4f}")
print(f"Train MAE: {train_mae:.4f}, Test MAE: {test_mae:.4f}")
print(f"Train RMSE: {train_rmse:.4f}, Test RMSE: {test_rmse:.4f}")
```

```
🔷 Decision Tree Results 🔷
Train R²: 0.9511, Test R²: 0.9503
Train MAE: 4.1525, Test MAE: 4.2328
Train RMSE: 13.6981, Test RMSE: 13.8010
```

In [103…

```python
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import numpy as np

# Initialize model
rf = RandomForestRegressor(
    n_estimators=300,   # number of trees
    random_state=42,
    max_depth=15,
    n_jobs=-1
)

# Fit model
rf.fit(X_train, y_train)
# Predictions
y_train_pred = rf.predict(X_train)
```

```
y_test_pred = rf.predict(X_test)

# Metrics
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)

train_mae = mean_absolute_error(y_train, y_train_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)

train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))

print("🔷 Random Forest Results 🔷")
print(f"Train R²: {train_r2:.4f}, Test R²: {test_r2:.4f}")
print(f"Train MAE: {train_mae:.4f}, Test MAE: {test_mae:.4f}")
print(f"Train RMSE: {train_rmse:.4f}, Test RMSE: {test_rmse:.4f}")
```

```
🔷 Random Forest Results 🔷
Train R²: 0.9636, Test R²: 0.9511
Train MAE: 2.7837, Test MAE: 3.4703
Train RMSE: 11.8216, Test RMSE: 13.6816
```

In [ ]:
```
#SVR model
```

In [104…
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.svm import SVR
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

# 🔷 Identify feature types (update these column names as per your dataset)
categorical_features = ["category", "country", "device"]   # text columns
numeric_features = ["views", "likes", "comments", "watch_time_minutes"]  # num

# 🔷 Features and target
X = youtube[categorical_features + numeric_features]
y = youtube["ad_revenue_usd"]

# 🔷 Preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(handle_unknown="ignore"), categorical_features),
        ("num", StandardScaler(), numeric_features)
    ]
)

# 🔷 SVR pipeline
svr_pipeline = Pipeline([
    ("preprocessor", preprocessor),
    ("svr", SVR(kernel="rbf", C=10, gamma="scale", epsilon=0.1))  # smaller C
])
```

```
# 🔹 Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# ⚡ Optional: sample data if it's too big (for faster training)
if len(X_train) > 5000:
    X_train = X_train.sample(5000, random_state=42)
    y_train = y_train.loc[X_train.index]

# 🔹 Train model
svr_pipeline.fit(X_train, y_train)

# 🔹 Predictions
y_train_pred = svr_pipeline.predict(X_train)
y_test_pred = svr_pipeline.predict(X_test)

# 🔹 Evaluation
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)
train_mae = mean_absolute_error(y_train, y_train_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))

print("🔹 SVR Results 🔹")
print(f"Train R²: {train_r2:.4f}, Test R²: {test_r2:.4f}")
print(f"Train MAE: {train_mae:.4f}, Test MAE: {test_mae:.4f}")
print(f"Train RMSE: {train_rmse:.4f}, Test RMSE: {test_rmse:.4f}")
```

```
🔹 SVR Results 🔹
Train R²: 0.9515, Test R²: 0.9508
Train MAE: 3.7255, Test MAE: 4.1246
Train RMSE: 13.7944, Test RMSE: 13.7306
```

In [ ]:
```
#FINAL MODEL WITH STREAMLIT LINEAR REGRESSION
```

In [107…
```
print(youtube.columns.tolist())
```

```
['video_id', 'date', 'views', 'likes', 'comments', 'watch_time_minutes', 'vide
o_length_minutes', 'subscribers', 'category', 'device', 'country', 'ad_revenu
e_usd', 'engagement_rate', 'year', 'month', 'day_of_weak', 'hour']
```

In [108…
```
num_features = ["views", "likes", "comments", "watch_time_minutes"]
```

In [111…
```
youtube["engagement_rate"] = (
    (youtube["likes"] + youtube["comments"]) / youtube["views"]
).fillna(0)
```

In [112…
```
import pickle

# Features
```

```
num_features = ["views", "likes", "comments", "watch_time_minutes"]  # ⚠ remo

cat_features = ["category", "device", "country"]

# Train-test split
X = youtube[num_features + cat_features]
y = youtube["ad_revenue_usd"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, randc

preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_features),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat_features)
    ]
)

lr_pipeline = Pipeline(steps=[
    ("preprocessor", preprocessor),
    ("model", LinearRegression())
])

# Fit model
lr_pipeline.fit(X_train, y_train)

# Predict
y_pred = lr_pipeline.predict(X_test)

# Save pipeline
import pickle
with open("linear_regression_pipeline.pkl", "wb") as f:
    pickle.dump(lr_pipeline, f)
```

In [113…

```
pip install streamlit
```

```
Collecting streamlit
  Downloading streamlit-1.49.1-py3-none-any.whl.metadata (9.5 kB)
Collecting altair!=5.4.0,!=5.4.1,<6,>=4.0 (from streamlit)
  Downloading altair-5.5.0-py3-none-any.whl.metadata (11 kB)
Collecting blinker<2,>=1.5.0 (from streamlit)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting cachetools<7,>=4.0 (from streamlit)
  Downloading cachetools-6.2.0-py3-none-any.whl.metadata (5.4 kB)
Collecting click<9,>=7.0 (from streamlit)
  Downloading click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: numpy<3,>=1.23 in c:\users\sowmi\onedrive\deskto
p\python\youtube\venv\lib\site-packages (from streamlit) (2.3.3)
Requirement already satisfied: packaging<26,>=20 in c:\users\sowmi\onedrive\des
ktop\python\youtube\venv\lib\site-packages (from streamlit) (25.0)
Requirement already satisfied: pandas<3,>=1.4.0 in c:\users\sowmi\onedrive\desk
top\python\youtube\venv\lib\site-packages (from streamlit) (2.3.2)
Requirement already satisfied: pillow<12,>=7.1.0 in c:\users\sowmi\onedrive\des
ktop\python\youtube\venv\lib\site-packages (from streamlit) (11.3.0)
Collecting protobuf<7,>=3.20 (from streamlit)
  Downloading protobuf-6.32.1-cp310-abi3-win_amd64.whl.metadata (593 bytes)
Collecting pyarrow>=7.0 (from streamlit)
  Downloading pyarrow-21.0.0-cp313-cp313-win_amd64.whl.metadata (3.4 kB)
Collecting requests<3,>=2.27 (from streamlit)
  Downloading requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting tenacity<10,>=8.1.0 (from streamlit)
  Downloading tenacity-9.1.2-py3-none-any.whl.metadata (1.2 kB)
Collecting toml<2,>=0.10.1 (from streamlit)
  Downloading toml-0.10.2-py2.py3-none-any.whl.metadata (7.1 kB)
Collecting typing-extensions<5,>=4.4.0 (from streamlit)
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting watchdog<7,>=2.1.5 (from streamlit)
  Downloading watchdog-6.0.0-py3-none-win_amd64.whl.metadata (44 kB)
Collecting gitpython!=3.1.19,<4,>=3.0.7 (from streamlit)
  Downloading gitpython-3.1.45-py3-none-any.whl.metadata (13 kB)
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.9.1-py2.py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: tornado!=6.5.0,<7,>=6.0.3 in c:\users\sowmi\oned
rive\desktop\python\youtube\venv\lib\site-packages (from streamlit) (6.5.2)
Collecting jinja2 (from altair!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting jsonschema>=3.0 (from altair!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading jsonschema-4.25.1-py3-none-any.whl.metadata (7.6 kB)
Collecting narwhals>=1.14.2 (from altair!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading narwhals-2.4.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: colorama in c:\users\sowmi\onedrive\desktop\pyth
on\youtube\venv\lib\site-packages (from click<9,>=7.0->streamlit) (0.4.6)
Collecting gitdb<5,>=4.0.1 (from gitpython!=3.1.19,<4,>=3.0.7->streamlit)
  Downloading gitdb-4.0.12-py3-none-any.whl.metadata (1.2 kB)
Collecting smmap<6,>=3.0.1 (from gitdb<5,>=4.0.1->gitpython!=3.1.1
9,<4,>=3.0.7->streamlit)
  Downloading smmap-5.0.2-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sowmi\onedriv
e\desktop\python\youtube\venv\lib\site-packages (from pandas<3,>=1.4.0->streaml
it) (2.9.0.post0)
```

```
Requirement already satisfied: pytz>=2020.1 in c:\users\sowmi\onedrive\desktop\
python\youtube\venv\lib\site-packages (from pandas<3,>=1.4.0->streamlit) (202
5.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\sowmi\onedrive\deskto
p\python\youtube\venv\lib\site-packages (from pandas<3,>=1.4.0->streamlit) (202
5.2)
Collecting charset_normalizer<4,>=2 (from requests<3,>=2.27->streamlit)
  Downloading charset_normalizer-3.4.3-cp313-cp313-win_amd64.whl.metadata (37 k
B)
Collecting idna<4,>=2.5 (from requests<3,>=2.27->streamlit)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests<3,>=2.27->streamlit)
  Downloading urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi>=2017.4.17 (from requests<3,>=2.27->streamlit)
  Downloading certifi-2025.8.3-py3-none-any.whl.metadata (2.4 kB)
Collecting MarkupSafe>=2.0 (from jinja2->altair!=5.4.0,!=5.4.1,<6,>=4.0->stream
lit)
  Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
Collecting attrs>=22.2.0 (from jsonschema>=3.0->altai
r!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading attrs-25.3.0-py3-none-any.whl.metadata (10 kB)
Collecting jsonschema-specifications>=2023.03.6 (from jsonschema>=3.0->altai
r!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading jsonschema_specifications-2025.9.1-py3-none-any.whl.metadata (2.9
kB)
Collecting referencing>=0.28.4 (from jsonschema>=3.0->altai
r!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading referencing-0.36.2-py3-none-any.whl.metadata (2.8 kB)
Collecting rpds-py>=0.7.1 (from jsonschema>=3.0->altai
r!=5.4.0,!=5.4.1,<6,>=4.0->streamlit)
  Downloading rpds_py-0.27.1-cp313-cp313-win_amd64.whl.metadata (4.3 kB)
Requirement already satisfied: six>=1.5 in c:\users\sowmi\onedrive\desktop\pyth
on\youtube\venv\lib\site-packages (from python-dateutil>=2.8.2->panda
s<3,>=1.4.0->streamlit) (1.17.0)
Downloading streamlit-1.49.1-py3-none-any.whl (10.0 MB)
   ---------------------------------------- 0.0/10.0 MB ? eta -:--:--
   ---- ----------------------------------- 1.0/10.0 MB 6.3 MB/s eta 0:00:02
   ---------- ----------------------------- 2.6/10.0 MB 6.3 MB/s eta 0:00:02
   -------------- ------------------------- 3.9/10.0 MB 6.3 MB/s eta 0:00:01
   ----------------- ---------------------- 4.5/10.0 MB 6.3 MB/s eta 0:00:01
   ------------------- -------------------- 5.2/10.0 MB 5.1 MB/s eta 0:00:01
   ------------------------- ------------- 6.6/10.0 MB 5.3 MB/s eta 0:00:01
   ------------------------------ -------- 7.9/10.0 MB 5.4 MB/s eta 0:00:01
   ---------------------------------- ----- 9.4/10.0 MB 5.6 MB/s eta 0:00:01
   ---------------------------------------- 10.0/10.0 MB 5.6 MB/s  0:00:01
Downloading altair-5.5.0-py3-none-any.whl (731 kB)
   ---------------------------------------- 0.0/731.2 kB ? eta -:--:--
   ---------------------------------------- 731.2/731.2 kB 6.5 MB/s  0:00:00
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading cachetools-6.2.0-py3-none-any.whl (11 kB)
Downloading click-8.2.1-py3-none-any.whl (102 kB)
Downloading gitpython-3.1.45-py3-none-any.whl (208 kB)
Downloading gitdb-4.0.12-py3-none-any.whl (62 kB)
Downloading protobuf-6.32.1-cp310-abi3-win_amd64.whl (435 kB)
```

```
Downloading pydeck-0.9.1-py2.py3-none-any.whl (6.9 MB)
   ---------------------------------------- 0.0/6.9 MB ? eta -:--:--
   --- ------------------------------------ 0.5/6.9 MB 5.7 MB/s eta 0:00:02
   ---------- ----------------------------- 2.4/6.9 MB 5.6 MB/s eta 0:00:01
   --------------- ------------------------ 2.9/6.9 MB 4.8 MB/s eta 0:00:01
   -------------------- ------------------- 3.7/6.9 MB 4.5 MB/s eta 0:00:01
   ------------------------ --------------- 4.5/6.9 MB 4.4 MB/s eta 0:00:01
   ------------------------------- -------- 5.8/6.9 MB 4.5 MB/s eta 0:00:01
   ---------------------------------------  6.8/6.9 MB 4.6 MB/s eta 0:00:01
   ---------------------------------------- 6.9/6.9 MB 4.5 MB/s  0:00:01
Downloading requests-2.32.5-py3-none-any.whl (64 kB)
Downloading charset_normalizer-3.4.3-cp313-cp313-win_amd64.whl (107 kB)
Downloading idna-3.10-py3-none-any.whl (70 kB)
Downloading smmap-5.0.2-py3-none-any.whl (24 kB)
Downloading tenacity-9.1.2-py3-none-any.whl (28 kB)
Downloading toml-0.10.2-py2.py3-none-any.whl (16 kB)
Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Downloading urllib3-2.5.0-py3-none-any.whl (129 kB)
Downloading watchdog-6.0.0-py3-none-win_amd64.whl (79 kB)
Downloading certifi-2025.8.3-py3-none-any.whl (161 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading jsonschema-4.25.1-py3-none-any.whl (90 kB)
Downloading attrs-25.3.0-py3-none-any.whl (63 kB)
Downloading jsonschema_specifications-2025.9.1-py3-none-any.whl (18 kB)
Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl (15 kB)
Downloading narwhals-2.4.0-py3-none-any.whl (406 kB)
Downloading pyarrow-21.0.0-cp313-cp313-win_amd64.whl (26.1 MB)
   ---------------------------------------- 0.0/26.1 MB ? eta -:--:--
   - -------------------------------------- 1.0/26.1 MB 6.6 MB/s eta 0:00:04
   ---- ----------------------------------- 2.6/26.1 MB 6.6 MB/s eta 0:00:04
   ------ --------------------------------- 3.9/26.1 MB 6.6 MB/s eta 0:00:04
   -------- ------------------------------- 5.5/26.1 MB 6.6 MB/s eta 0:00:04
   ---------- ----------------------------- 6.6/26.1 MB 6.5 MB/s eta 0:00:03
   ------------ --------------------------- 8.1/26.1 MB 6.5 MB/s eta 0:00:03
   ------------- -------------------------- 8.7/26.1 MB 6.2 MB/s eta 0:00:03
   ------------- -------------------------- 8.9/26.1 MB 5.9 MB/s eta 0:00:03
   -------------- ------------------------- 9.2/26.1 MB 5.1 MB/s eta 0:00:04
   -------------- ------------------------- 9.2/26.1 MB 5.1 MB/s eta 0:00:04
   -------------- ------------------------- 9.4/26.1 MB 4.4 MB/s eta 0:00:04
   -------------- ------------------------- 9.7/26.1 MB 4.1 MB/s eta 0:00:05
   --------------- ------------------------ 10.2/26.1 MB 3.7 MB/s eta 0:00:05
   ---------------- ----------------------- 10.5/26.1 MB 3.6 MB/s eta 0:00:05
   ---------------- ----------------------- 11.0/26.1 MB 3.5 MB/s eta 0:00:05
   ----------------- ---------------------- 11.3/26.1 MB 3.4 MB/s eta 0:00:05
   ------------------ --------------------- 11.8/26.1 MB 3.3 MB/s eta 0:00:05
   ------------------ --------------------- 12.3/26.1 MB 3.3 MB/s eta 0:00:05
   ------------------- -------------------- 12.8/26.1 MB 3.2 MB/s eta 0:00:05
   -------------------- ------------------- 13.6/26.1 MB 3.2 MB/s eta 0:00:04
   --------------------- ------------------ 14.2/26.1 MB 3.2 MB/s eta 0:00:04
   --------------------- ------------------ 14.9/26.1 MB 3.2 MB/s eta 0:00:04
   ---------------------- ----------------- 15.7/26.1 MB 3.2 MB/s eta 0:00:04
   ----------------------- ---------------- 16.5/26.1 MB 3.3 MB/s eta 0:00:03
   ------------------------ --------------- 17.3/26.1 MB 3.3 MB/s eta 0:00:03
   ------------------------- -------------- 18.4/26.1 MB 3.3 MB/s eta 0:00:03
```

```
   ----------------------------- ---------- 19.1/26.1 MB 3.4 MB/s eta 0:00:03
   ------------------------------ --------- 20.2/26.1 MB 3.4 MB/s eta 0:00:02
   -------------------------------- ------- 21.2/26.1 MB 3.4 MB/s eta 0:00:02
   ---------------------------------- ----- 22.3/26.1 MB 3.5 MB/s eta 0:00:02
   ----------------------------------- ---- 23.3/26.1 MB 3.5 MB/s eta 0:00:01
   ------------------------------------- -- 24.4/26.1 MB 3.6 MB/s eta 0:00:01
   -------------------------------------- - 25.2/26.1 MB 3.6 MB/s eta 0:00:01
   ---------------------------------------  26.0/26.1 MB 3.6 MB/s eta 0:00:01
   ---------------------------------------- 26.1/26.1 MB 3.6 MB/s  0:00:07
Downloading referencing-0.36.2-py3-none-any.whl (26 kB)
Downloading rpds_py-0.27.1-cp313-cp313-win_amd64.whl (232 kB)
Installing collected packages: watchdog, urllib3, typing-extensions, toml, tena
city, smmap, rpds-py, pyarrow, protobuf, narwhals, MarkupSafe, idna, click, cha
rset_normalizer, certifi, cachetools, blinker, attrs, requests, referencing, ji
nja2, gitdb, pydeck, jsonschema-specifications, gitpython, jsonschema, altair,
streamlit

   ----------------------------------------  0/28 [watchdog]
   ----------------------------------------  0/28 [watchdog]
   ----------------------------------------  0/28 [watchdog]
   ----------------------------------------  0/28 [watchdog]
   ----------------------------------------  0/28 [watchdog]
   ----------------------------------------  0/28 [watchdog]
   - --------------------------------------  1/28 [urllib3]
   - --------------------------------------  1/28 [urllib3]
   - --------------------------------------  1/28 [urllib3]
   - --------------------------------------  1/28 [urllib3]
   - --------------------------------------  1/28 [urllib3]
   -- -------------------------------------  2/28 [typing-extensions]
   ---- -----------------------------------  3/28 [toml]
   ----- ----------------------------------  4/28 [tenacity]
   ------- --------------------------------  5/28 [smmap]
   ------- --------------------------------  5/28 [smmap]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
   ---------- -----------------------------  7/28 [pyarrow]
```

```
---------- -----------------------------  7/28 [pyarrow]
---------- -----------------------------  7/28 [pyarrow]
---------- -----------------------------  7/28 [pyarrow]
---------- -----------------------------  7/28 [pyarrow]
---------- -----------------------------  7/28 [pyarrow]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
----------- ----------------------------  8/28 [protobuf]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
------------ ---------------------------  9/28 [narwhals]
--------------- ------------------------ 11/28 [idna]
--------------- ------------------------ 11/28 [idna]
----------------- ---------------------- 12/28 [click]
----------------- ---------------------- 12/28 [click]
------------------ --------------------- 13/28 [charset_normalizer]
------------------ --------------------- 13/28 [charset_normalizer]
------------------ --------------------- 13/28 [charset_normalizer]
-------------------- ------------------- 14/28 [certifi]
---------------------- ----------------- 16/28 [blinker]
------------------------ --------------- 17/28 [attrs]
------------------------ --------------- 17/28 [attrs]
------------------------ --------------- 17/28 [attrs]
------------------------- -------------- 18/28 [requests]
------------------------- -------------- 18/28 [requests]
--------------------------- ------------ 19/28 [referencing]
--------------------------- ------------ 19/28 [referencing]
---------------------------- ----------- 20/28 [jinja2]
```

```
--------------------------- ----------- 20/28 [jinja2]
--------------------------- ----------- 20/28 [jinja2]
--------------------------- ----------- 20/28 [jinja2]
----------------------------- --------- 21/28 [gitdb]
----------------------------- --------- 21/28 [gitdb]
----------------------------- --------- 21/28 [gitdb]
------------------------------ -------- 22/28 [pydeck]
------------------------------ -------- 22/28 [pydeck]
------------------------------ -------- 22/28 [pydeck]
------------------------------ -------- 22/28 [pydeck]
------------------------------ -------- 22/28 [pydeck]
------------------------------ -------- 22/28 [pydeck]
-------------------------------- ----- 24/28 [gitpython]
-------------------------------- ----- 24/28 [gitpython]
-------------------------------- ----- 24/28 [gitpython]
-------------------------------- ----- 24/28 [gitpython]
-------------------------------- ----- 24/28 [gitpython]
-------------------------------- ----- 24/28 [gitpython]
---------------------------------- ---- 25/28 [jsonschema]
---------------------------------- ---- 25/28 [jsonschema]
---------------------------------- ---- 25/28 [jsonschema]
---------------------------------- ---- 25/28 [jsonschema]
---------------------------------- ---- 25/28 [jsonschema]
---------------------------------- ---- 25/28 [jsonschema]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
----------------------------------- -- 26/28 [altair]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
------------------------------------ - 27/28 [streamlit]
```

```
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   -------------------------------------- - 27/28 [streamlit]
   ---------------------------------------- 28/28 [streamlit]

Successfully installed MarkupSafe-3.0.2 altair-5.5.0 attrs-25.3.0 blinker-1.9.0
cachetools-6.2.0 certifi-2025.8.3 charset_normalizer-3.4.3 click-8.2.1 gitd
b-4.0.12 gitpython-3.1.45 idna-3.10 jinja2-3.1.6 jsonschema-4.25.1 jsonschema-s
pecifications-2025.9.1 narwhals-2.4.0 protobuf-6.32.1 pyarrow-21.0.0 pydec
k-0.9.1 referencing-0.36.2 requests-2.32.5 rpds-py-0.27.1 smmap-5.0.2 streamli
t-1.49.1 tenacity-9.1.2 toml-0.10.2 typing-extensions-4.15.0 urllib3-2.5.0 watc
hdog-6.0.0
Note: you may need to restart the kernel to use updated packages.
```