TikTok_project_hypothesisTesting

April 18, 2025

1 TikTok project: Data exploration and hypothesis testing

The purpose of this project is to demostrate knowledge of how to prepare, create, and analyze hypothesis tests.

The goal is to apply descriptive and inferential statistics, probability distributions, and hypothesis testing in Python.

This activity has three parts:

Part 1: Imports and data loading * What data packages will be necessary for hypothesis testing?

Part 2: Conduct hypothesis testing * How will descriptive statistics help you analyze your data?

- How will you formulate your null hypothesis and alternative hypothesis?

Part 3: Communicate insights with stakeholders

- What key business insight(s) emerge from your hypothesis test?
- What business recommendations do you propose based on your results?

2 Data exploration and hypothesis testing

Consider the questions in your planning and those below to craft your response.

1. What is your research question for this data project? Later on, you will need to formulate the null and alternative hypotheses as the first step of your hypothesis test. Consider your research question now, at the start of this task.

Response:

There are a few possible ways to frame the research question. For example:

1) Do videos from verified accounts and videos unverified accounts have different average view counts?
2) Is there a relationship between the account being verified and the associated videos' view counts?

2.0.1 Task 1. Imports and Data Loading

Import packages and libraries needed to compute descriptive statistics and conduct a hypothesis test.

[1]:

```
# Import packages for data manipulation
import numpy as np
import pandas as pd

# Import packages for data visualization
import seaborn as sns
import matplotlib.pyplot as plt

# Import packages for statistical analysis/hypothesis testing
from scipy import stats
```

Load the dataset.

[3]:

```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

Consider the questions in your analyze stage and those below to craft your response: 1. Data professionals use descriptive statistics for Exploratory Data Analysis. How can computing descriptive statistics help you learn more about your data in this stage of your analysis?

Response: In general, descriptive statistics are useful because they let you quickly explore and understand large amounts of data. In this case, computing descriptive statistics helps you quickly compute the mean values of video_view_count for each group of verified_status in the sample data.

2.0.2 Task 2. Data exploration

Use descriptive statistics to conduct Exploratory Data Analysis (EDA).

Inspect the first five rows of the dataframe.

[5]:

```
# Display first few rows
data.head()
```

[5]:

```
   # claim_status    video_id  video_duration_sec  \
0  1        claim  7017666017                  59
1  2        claim  4014381136                  32
2  3        claim  9859838091                  31
3  4        claim  1866847991                  25
4  5        claim  7105231098                  19

                            video_transcription_text verified_status  \
0  someone shared with me that drone deliveries a…    not verified
1  someone shared with me that there are more mic…    not verified
2  someone shared with me that american industria…    not verified
3  someone shared with me that the metro of st. p…    not verified
4  someone shared with me that the number of busi…    not verified

  author_ban_status  video_view_count  video_like_count  video_share_count  \
0      under review          343296.0           19425.0              241.0
```

```
1            active          140877.0          77355.0            19034.0
2            active          902185.0          97690.0             2858.0
3            active          437506.0         239954.0            34812.0
4            active           56167.0          34987.0             4110.0

   video_download_count  video_comment_count
0                   1.0                  0.0
1                1161.0                684.0
2                 833.0                329.0
3                1234.0                584.0
4                 547.0                152.0
```

```
[7]: # Generate a table of descriptive statistics about the data
     data.describe()
```

```
[7]:                   #      video_id  video_duration_sec  video_view_count  \
     count  19382.000000  1.938200e+04        19382.000000      19084.000000
     mean    9691.500000  5.627454e+09           32.421732     254708.558688
     std     5595.245794  2.536440e+09           16.229967     322893.280814
     min        1.000000  1.234959e+09            5.000000         20.000000
     25%     4846.250000  3.430417e+09           18.000000       4942.500000
     50%     9691.500000  5.618664e+09           32.000000       9954.500000
     75%    14536.750000  7.843960e+09           47.000000     504327.000000
     max    19382.000000  9.999873e+09           60.000000     999817.000000

            video_like_count  video_share_count  video_download_count  \
     count      19084.000000       19084.000000          19084.000000
     mean       84304.636030       16735.248323           1049.429627
     std       133420.546814       32036.174350           2004.299894
     min            0.000000           0.000000              0.000000
     25%          810.750000         115.000000              7.000000
     50%         3403.500000         717.000000             46.000000
     75%       125020.000000       18222.000000           1156.250000
     max       657830.000000      256130.000000          14994.000000

            video_comment_count
     count         19084.000000
     mean            349.312146
     std             799.638865
     min               0.000000
     25%               1.000000
     50%               9.000000
     75%             292.000000
     max            9599.000000
```

Check for and handle missing values.

[9]:
```
# Check for missing values
data.isna().sum()
```

[9]:
```
#                             0
claim_status                298
video_id                      0
video_duration_sec            0
video_transcription_text    298
verified_status               0
author_ban_status             0
video_view_count            298
video_like_count            298
video_share_count           298
video_download_count        298
video_comment_count         298
dtype: int64
```

[11]:
```
# Drop rows with missing values
data = data.dropna(axis=0)
```

[13]:
```
# Display first few rows after handling missing values
data.head()
```

[13]:
```
   # claim_status    video_id  video_duration_sec  \
0  1        claim  7017666017                  59
1  2        claim  4014381136                  32
2  3        claim  9859838091                  31
3  4        claim  1866847991                  25
4  5        claim  7105231098                  19

                            video_transcription_text verified_status  \
0  someone shared with me that drone deliveries a…    not verified
1  someone shared with me that there are more mic…    not verified
2  someone shared with me that american industria…    not verified
3  someone shared with me that the metro of st. p…    not verified
4  someone shared with me that the number of busi…    not verified

  author_ban_status  video_view_count  video_like_count  video_share_count  \
0      under review          343296.0           19425.0              241.0
1            active          140877.0           77355.0            19034.0
2            active          902185.0           97690.0             2858.0
3            active          437506.0          239954.0            34812.0
4            active           56167.0           34987.0             4110.0

   video_download_count  video_comment_count
0                   1.0                  0.0
1                1161.0                684.0
```

```
2              833.0                 329.0
3             1234.0                 584.0
4              547.0                 152.0
```

You are interested in the relationship between `verified_status` and `video_view_count`. One approach is to examine the mean value of `video_view_count` for each group of `verified_status` in the sample data.

[15]:

```python
# Compute the mean `video_view_count` for each group in `verified_status`
data.groupby('verified_status')['video_view_count'].mean()
```

[15]:

```
verified_status
not verified    265663.785339
verified         91439.164167
Name: video_view_count, dtype: float64
```

2.0.3 Task 3. Hypothesis testing

Before you conduct your hypothesis test, consider the following questions where applicable to complete your code response:

1. Recall the difference between the null hypothesis and the alternative hypotheses. What are your hypotheses for this data project?

Response:

- **Null hypothesis**: There is no difference in number of views between TikTok videos posted by verified accounts and TikTok videos posted by unverified accounts (any observed difference in the sample data is due to chance or sampling variability).
- **Alternative hypothesis**: There is a difference in number of views between TikTok videos posted by verified accounts and TikTok videos posted by unverified accounts (any observed difference in the sample data is due to an actual difference in the corresponding population means).

Your goal in this step is to conduct a two-sample t-test. Recall the steps for conducting a hypothesis test:

1. State the null hypothesis and the alternative hypothesis
2. Choose a signficance level
3. Find the p-value
4. Reject or fail to reject the null hypothesis

H_0: There is no difference in number of views between TikTok videos posted by verified accounts and TikTok videos posted by unverified accounts (any observed difference in the sample data is due to chance or sampling variability).

H_A: There is a difference in number of views between TikTok videos posted by verified accounts and TikTok videos posted by unverified accounts (any observed difference in the sample data is due to an actual difference in the corresponding population means).

You choose 5% as the significance level and proceed with a two-sample t-test.

[17]:

```
# Conduct a two-sample t-test to compare means

# Save each sample in a variable
not_verified = data[data["verified_status"] == "not␣
 ↪verified"]["video_view_count"]
verified = data[data["verified_status"] == "verified"]["video_view_count"]

# Implement a t-test using the two samples
stats.ttest_ind(a=not_verified, b=verified, equal_var=False)
```

[17]:

```
TtestResult(statistic=25.499441780633777, pvalue=2.6088823687177823e-120,
df=1571.163074387424)
```

Question: Based on the p-value you got above, do you reject or fail to reject the null hypothesis?

Response: Since the p-value is extremely small (much smaller than the significance level of 5%), you reject the null hypothesis. You conclude that there **is** a statistically significant difference in the mean video view count between verified and unverified accounts on TikTok.

2.1 Step 4: Communicate insights with stakeholders

Ask yourself the following questions:

1. What business insight(s) can you draw from the result of your hypothesis test?

Response:

The analysis shows that there is a statistically significant difference in the average view counts between videos from verified accounts and videos from unverified accounts. This suggests there might be fundamental behavioral differences between these two groups of accounts.

It would be interesting to investigate the root cause of this behavioral difference. For example, do unverified accounts tend to post more clickbait-y videos? Or are unverified accounts associated with spam bots that help inflate view counts?

The next step will be to build a regression model on verified_status. A regression model is the natural next step because the end goal is to make predictions on claim status. A regression model for verified_status can help analyze user behavior in this group of verified users. Technical note to prepare regression model: because the data is skewed, and there is a significant difference in account types, it will be key to build a logistic regression model.

[ ]:

```

```