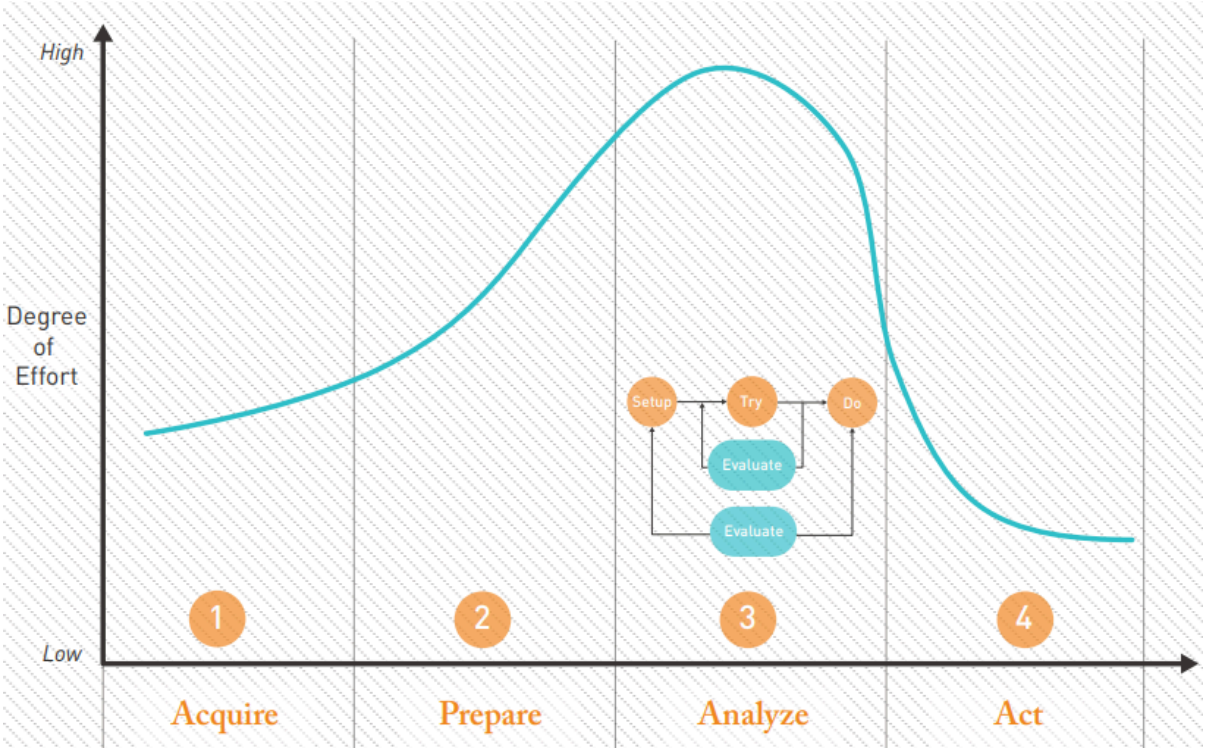
IDS a Wrap: Culminating Workshop for IDS

Instructors: Jhun Brian Andam | Timothy Jonah Borromeo

Course: Introduction to Data Science

Objectives:

- To reinforce and apply the stages of the data science pipeline—from data acquisition to analysis—using a real-time and semi-structured dataset.
- To gain hands-on experience in collecting data from APIs using Python's requests library, including authentication, handling JSON responses, and managing real-world API constraints.



This image illustrates a familiar reality we've already discussed: the practical workflow of data science. It highlights four broad stages—**Acquire**, **Prepare**, **Analyze**, and **Act**—and shows how the **degree of effort** typically peaks during the analysis phase, where we encounter the most trial-and-error through cycles of setting up, trying, and evaluating different approaches.

> The pipeline provides the methodological structure to manage and extract value from data that is `large`, `fast`, `diverse`, `uncertain`, and `complex`.

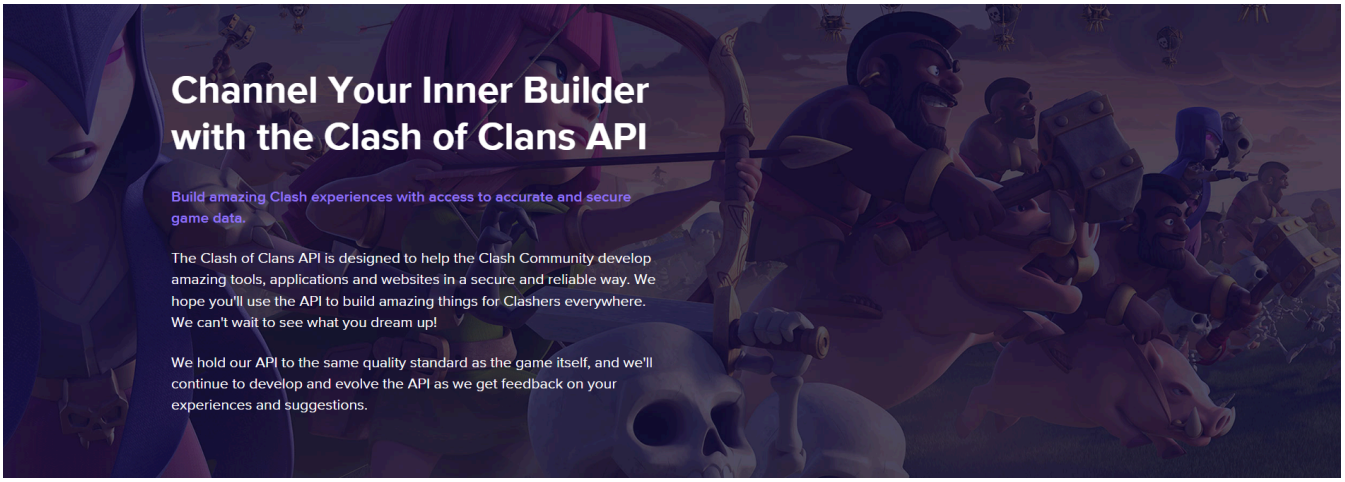
However, as we've also acknowledged in our discussions, the data science pipeline is **not as simple or linear** as these four steps suggest. In practice, each stage contains a range of **more granular and specific methods**—from data cleaning, transformation, feature engineering, and model tuning, to interpreting results and communicating insights. The process is iterative, often messy, and highly dependent on the context and constraints of each project. This diagram serves as a helpful high-level summary, but the real work lies in the complexities within and between these stages.

Workshop Overview

In this culminating workshop, we'll put that understanding into action. The objective is to **gather real-world data from a live data source**, particularly using an **API**, and walk it through the full data science pipeline. What makes this workshop especially relevant to today's tools and trends is our integration of **AI-powered assistants like ChatGPT** to support decision-making, troubleshoot challenges, and streamline various tasks throughout the process.

From forming the right API queries to shaping the data for analysis and deriving meaningful insights, this activity highlights not just technical execution—but how **AI can enhance human judgment** and improve the overall flow of a data science project.

For this workshop, we'll be featuring the **Clash of Clans API** as our primary data source. This API offers an excellent opportunity to experience real-world data work for several key reasons:



1. **It delivers real-time data**, making it ideal for understanding concepts related to data velocity and live system integration.
2. The data is **not perfectly clean or structured**, reflecting the **messiness** of real-world datasets, where inconsistencies, missing values, and nested structures are common.
3. It presents a **semi-structured format (JSON)**, challenging participants to parse and transform data before it can be used for analysis—mirroring typical preparation and wrangling tasks.
4. The data is **dynamic and contextual**, requiring logical reasoning to extract insights (e.g., understanding player stats, clan activity, and performance metrics).
5. Accessing the API involves **authentication and rate limits**, simulating real-world constraints when working with third-party data services.

Pre-workshop questions to ponder:

1. How will we collect the data?
2. What challenges might we face when cleaning or organizing this data?
3. How can we turn raw game data into valuable insights?

Basic Concepts

To get the most out of this workshop, it's important to have a basic grasp of the following concepts and tools, especially since we'll be working directly with a real-world API:

1. **API (Application Programming Interface)**
2. **Endpoint**
3. **Base URL**
4. **Request**
5. **Request Methods**
6. **Headers**
7. **Authentication**
8. **API Token**
9. **Status Code**
10. **JSON**
11. **Query Parameters**

Groupings

Form 6 groups, each composed of at most 5 members.

To ensure a smoother experience during the workshop, try to include at least one member familiar with Clash of Clans in each group.

Implementation

In [1]:
```
# import standard libraries

import requests                # for communicating with the API framework
import json
import pandas as pd            # structuring responses
from dotenv import load_dotenv # handling sensitive keys (best practice)
import os                      # extract local keys
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Before you can start collecting data, you'll need to register and gain access to the Clash of Clans API. This involves creating a developer account and generating an API token that will authenticate your requests. Visit the following link and register an account.

- https://developer.clashofclans.com/#/

Generating Your API Key

Once you've successfully registered for a Clash of Clans developer account:

1. **Navigate to** `My Account` > **Create New Key**.
2. You'll be asked to provide an **Allowed IP Address**. This is required to ensure that only your machine can use the key. To find your current IP address for local development, visit: https://whatismyipaddress.com/
3. Look for your **IPv4 Address**, copy it, and paste it into the **Allowed IP field** on the key creation page.
4. After submitting, your **API key will be generated**—copy and store it securely. You'll use it in your request headers to access the API.

Note: If you change networks (e.g., from home to school Wi-Fi), you'll need to update your allowed IP address accordingly.

To protect your API key and avoid accidentally sharing it (e.g., when uploading code to GitHub), it's good practice to **store sensitive information in a `.env` file**.

1. **Install the `python-dotenv` library** (if not already installed):

   ```
   pip install python-dotenv
   ```
2. **Create a `.env` file** in the **same directory** as your Jupyter notebook or Python script.
3. Inside the `.env` file, store your API key like this:

   ```
   API_TOKEN=your_actual_api_key_here
   ```
4. Then, in your notebook, load it securely with:

   ```
   from dotenv import load_dotenv
   import os

   load_dotenv()
   api_key = os.getenv("API_TOKEN")
   ```

In [2]:
```
load_dotenv()

API_TOKEN = os.getenv('API_TOKEN')
BASE_URL = "https://api.clashofclans.com/v1/"
```

In [3]:
```
headers = {
    "Authorization": f"Bearer {API_TOKEN}"
}
```

Say, we want to query a list of clans.

In [4]:
```
loc_resp = requests.get(os.path.join(BASE_URL, 'locations'), headers=headers)
```

In [5]:
```
loc_resp.json().keys()
```

Out[5]:
```
dict_keys(['items', 'paging'])
```

In [6]:
```
loc_df = pd.DataFrame([i for i in loc_resp.json()['items'] if i['isCountry']])
loc_df.head()
```

Out[6]:

	id	name	isCountry	countryCode
0	32000008	Åland Islands	True	AX
1	32000009	Albania	True	AL
2	32000010	Algeria	True	DZ
3	32000011	American Samoa	True	AS
4	32000012	Andorra	True	AD

In [7]:
```
loc_df[loc_df['name'] == 'Philippines']
```

Out[7]:

	id	name	isCountry	countryCode
177	32000185	Philippines	True	PH

In [8]:
```
loc_params = {
    'locationId':32000185
}
```

In [9]:
```
clan_resp = requests.get(os.path.join(BASE_URL, 'clans'), headers=headers, params=loc_params)
clan_resp.reason
```

Out[9]:
```
'OK'
```

In [10]:
```
clan_resp.json().keys()
```

Out[10]:
```
dict_keys(['items', 'paging'])
```

In [11]:
```
clan_df = pd.DataFrame(clan_resp.json()['items'])
clan_df.head()
```

Out[11]:

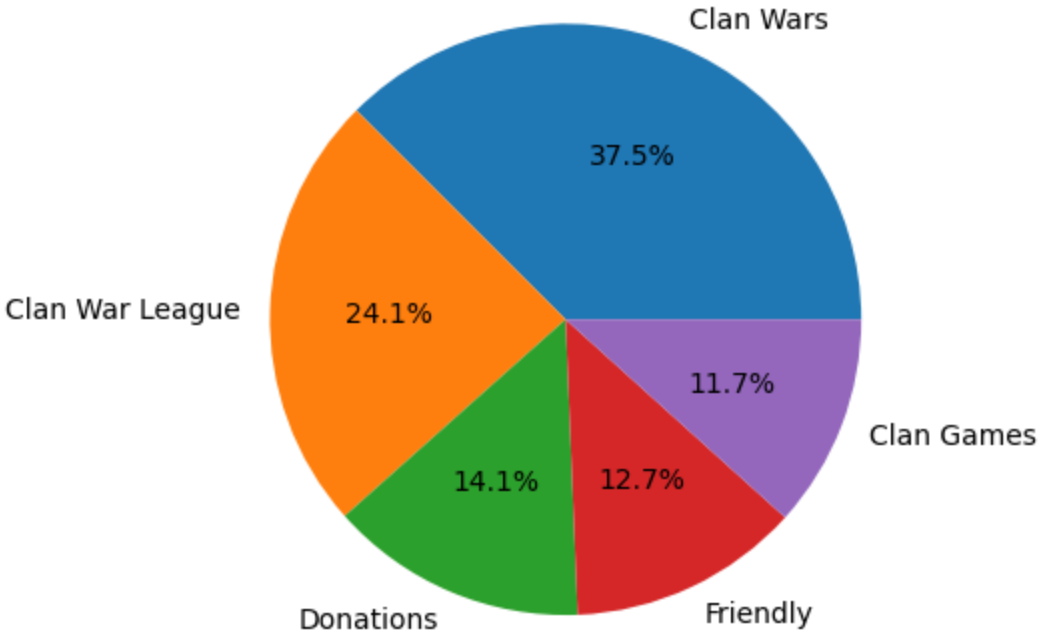
	tag	name	type	location	isFamilyFriendly	badgeUrls	clanLev
0	#2C8V89VC	"DoNT GiVE UP"]	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	
1	#2QP2LR020	"LEGENDARY"	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	
2	#2LCYY0GRY	"NΔTTONZ"	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	
3	#2GQY2U0RQ	"War Guru" OG's	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	
4	#2RGC0R0C8	#BLACKHEART07	open	{'id': 32000185, 'name': 'Philippines', 'isCou...	True	{'small': 'https://api-assets.clashofclans.com...	

5 rows × 24 columns

What are the labels of most Filipino clans?

In [12]:
```
label = []
for i in clan_df['labels']:
    if i:
        for j in i:
            label.append(j['name'])
```

In [13]:
```
label_vc = pd.Series(label).value_counts()
plt.pie(x=label_vc.values[:5], labels=label_vc.index[:5], autopct='%1.1f%%');
```



Is there a relationship between the clan labels and the clan points?

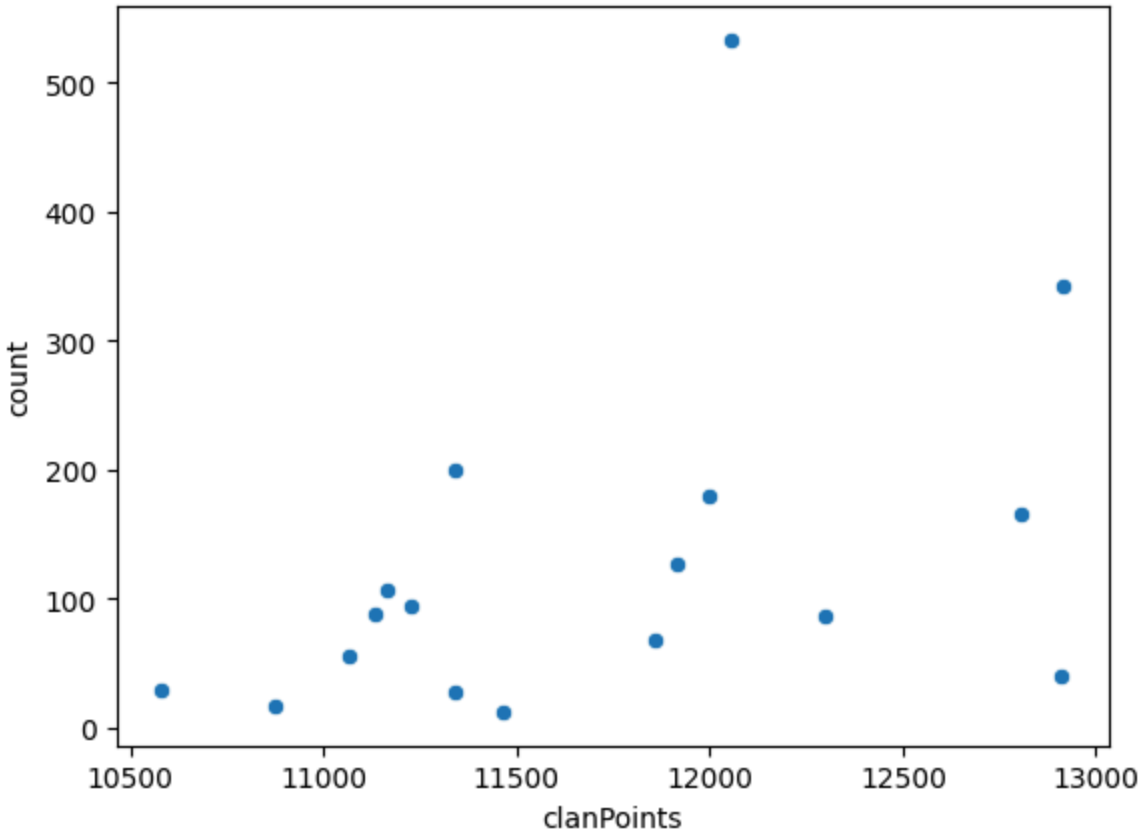
In [14]:
```
label_points = []

for idx, row in clan_df.iterrows():
    if row['labels']:  # Check if labels exist
        for label in row['labels']:
            label_points.append({
                'clan_id': row['tag'],  # or any unique identifier
                'label': label['name'],
                'clanPoints': row['clanPoints']
            })

label_points_df = pd.DataFrame(label_points)
```

In [15]:
```
avg_points_by_label = label_points_df.groupby('label')['clanPoints'].mean().sort_values(ascendin
pts_x_freq = pd.concat([avg_points_by_label, label_vc], axis=1, join='inner')
sns.scatterplot(data=pts_x_freq, x='clanPoints', y='count')
```

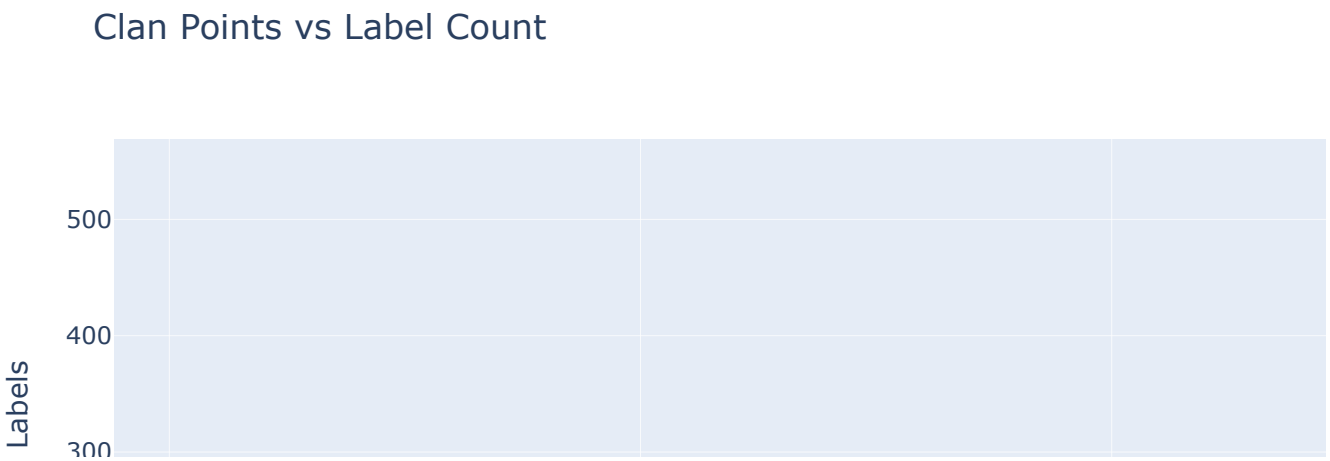
Out[15]:
```
<Axes: xlabel='clanPoints', ylabel='count'>
```



In [16]:

```
fig = px.scatter(
    pts_x_freq,
    x='clanPoints',
    y='count',
    title='Clan Points vs Label Count',
    labels={'clanPoints': 'Clan Points', 'count': 'Number of Labels'},
    hover_name=pts_x_freq.index.astype(str)  # Optional: show index on hover
)

fig.show()
```



What is the distibution of trophy for top n clans?

In [17]:
```
top_n_clan = clan_df.sort_values(by='clanPoints', ascending=False).head(5)
top_n_clan
```

Out[17]:

	tag	name	type	location	isFamilyFriendly	badgeUrls	clanLevel
60	#9Q82CRYJ	BARKADA	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	26
68	#2QQR8YJRR	Behave clan2022	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	15
603	#2QY8PU9C9	Saudi Braders	open	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	15
139	#L0LJG2G	clantari	inviteOnly	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	21
245	#PVGL0892	FURIOUS FIGHTER	open	{'id': 32000185, 'name': 'Philippines', 'isCou...	False	{'small': 'https://api-assets.clashofclans.com...	26

5 rows × 24 columns

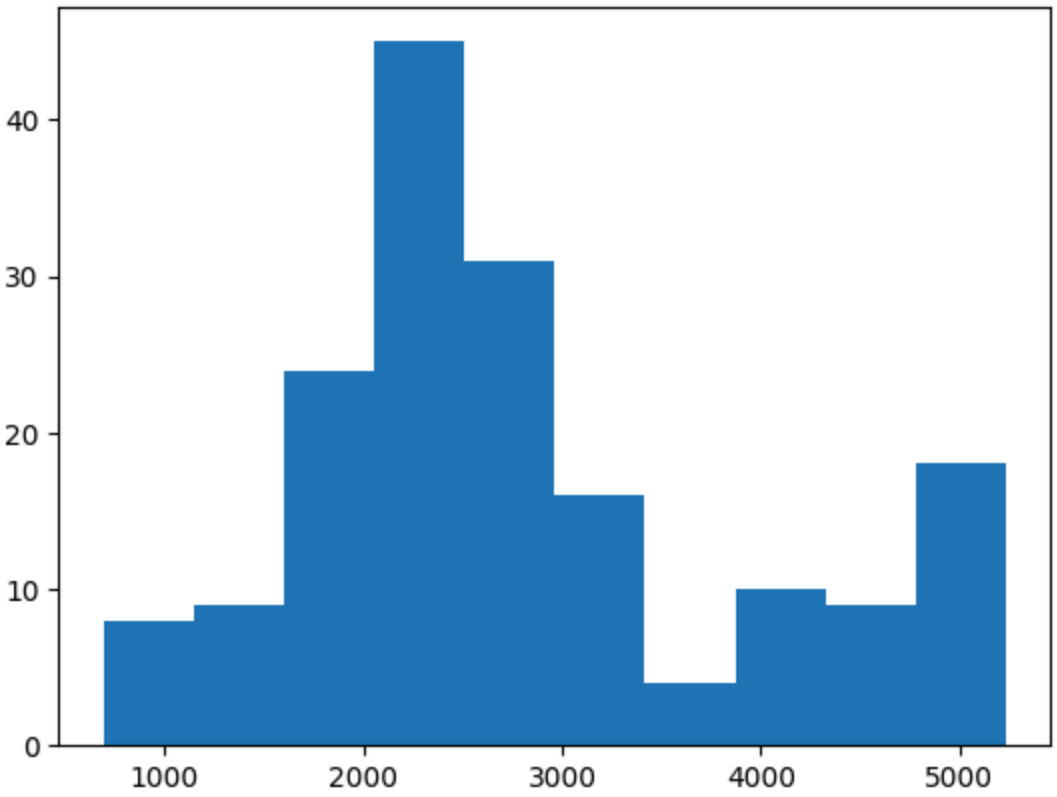
In [20]:
```
members = []
headers['Accept'] = 'application/json'


for t in top_n_clan['tag']:
    try:
        # Ensure the tag is URL-safe (replace '#' with '%23')
        safe_tag = t.replace('#', '%23')
        endpoint = f"{BASE_URL}clans/{safe_tag}/members"
        member_resp = requests.get(endpoint, headers=headers)
        member_resp.raise_for_status()  # Optional: raises an exception for bad status codes
        members.append(member_resp.json())
    except Exception as e:
        print(f"Error fetching members for clan {t}: {e}")
```

In [22]:
```
trophies = []
for clan in members:
    clan_members = clan['items']
```

```
    for member in clan_members:
        trophies.append(member['trophies'])
```

In [23]:
```
plt.hist(trophies);
```



Now, I'd like you to critically examine the dataset and identify potential analytical questions that could be asked. Consider what meaningful insights or patterns could be uncovered through analysis of this type of data.

Activity

Instruction: Analytical Exploration of Clash of Clans Data

You are tasked with analyzing the Clash of Clans data (using the available API endpoints) for a specific country. Your goal is to uncover meaningful insights that can help understand trends, behavior, and performance within the game environment.

Tasks:

1. **Formulate Analytical Questions**
    - Develop 3–4 analytical questions related to the available data from the Clash of Clans API (including clans, players, war logs, trophies, and more).
    - Think critically about what insights can be drawn from different parts of the data—whether that's trends in player performance, clan activity, or comparisons across various metrics.
    - Your questions should guide your analysis and shape the conclusions you aim to draw.
    - Feel free to use **AI tools** (like ChatGPT, Bard, or Copilot) to help brainstorm questions, structure your analysis, or clarify how different parts of the data relate to one another.
2. **Analyze the Data**
    - Use Python with libraries like **pandas**, **seaborn**, and **matplotlib** to analyze the data.
    - Consider questions such as:
        - How do **trophies** correlate with **clan level** or **war activity**?
        - What are the patterns in **player behavior** (e.g., win rates, participation in wars, etc.)?
        - Are there any notable **country-specific trends** (e.g., number of clans, trophies, or member distribution)?
        - How does **clan size** affect performance or participation in events?
3. **Visualize Your Findings**
    - Present your findings through **visualizations** that clearly communicate your insights. These might include:
        - **Bar charts** for distributions (e.g., clan levels, trophies, war participation)
        - **Boxplots** for variations (e.g., trophies by clan level or war frequency)
        - **Scatter plots** to visualize relationships (e.g., trophies vs. clan level, member count vs. activity)
    - Ensure the visuals are clear, informative, and help to support the analysis you conduct.
4. **Present Your Results**
    - Prepare a **short presentation (3–5 minutes)** summarizing:
        - Your **analytical questions**
        - The **methods** used for analysis
        - The **key findings** from your exploration
        - How these insights might inform **strategy, performance, or behavior in Clash of Clans**.
    - Explain why you selected those questions and how your analysis provides a deeper understanding of the game's mechanics.

Prepare to share your findings with the class and discuss how your analysis could be applied to improve player or clan strategies in Clash of Clans!