DSC350 - Term Project - Milestone 3

Connecting to an API/Pulling in the Data and Cleaning/Formatting

Perform at least 5 data transformation and/or cleansing steps to your API data.

=== ; Title: Term Project - Milestone 2 ; Author: API-Football ; Date: 10 July 2024 ; Modified By: Tyler Heflin ; Description: This program demonstrates the use of Python for ; modification and transformations to API data. ;=== */

We begin the assignment by importing the necessary libraries for completion.

In [1]:

```
import requests
import pandas as pd
import json
import warnings
warnings.filterwarnings('ignore')
```

Perform Request-Response Procedures on an API

In [2]:

```
# Define the headers for the API
headers = {
        "x-rapidapi-key": "f75a7c49dfmsh438cd6ef8f61b7fp1d69d3jsn72233cf9ffa6",
        "x-rapidapi-host": "api-football-v1.p.rapidapi.com"
}

# Define the endpoints for the API
url1 = "https://api-football-v1.p.rapidapi.com/v3/players"
url2 = "https://api-football-v1.p.rapidapi.com/v3/leagues"

# Define the parameters for the players endpoint
params_players = {
    "league": "253",
    "season": "2023"
}

# Define the parameters for the leagues endpoint
params_leagues = {
    "id": "253"
}
```

In [3]:

```
# Define a funtion to verify JSON format from API
def format_json(response):
    try:
        data = response.json()
    except ValueError as e:
        print("Response is not in JSON format:", e)
    return data
```

In [4]:

```
# Fetch data from 'players' endpoint
response_players = requests.get(url1, headers=headers, params=params_players, verif
players_data = format_json(response_players)

# Fetch data from 'leagues' endpoint
response_leagues = requests.get(url2, headers=headers, params=params_leagues, verif
leagues_data = format_json(response_leagues)
```

In [5]:

```
# Combine data into a DataFrame
if players_data and leagues_data:
    players_df = pd.json_normalize(players_data['response'])
    leagues_df = pd.json_normalize(leagues_data['response'])

    # Merge the DataFrames
    combined_df = pd.concat([players_df, leagues_df], axis=1)
    # Export newly created DataFrame
    combined_df.to_csv(r'C:\Users\thefli0\Downloads\combined_data.csv', index=False
    print("Data successfully exported to file.")

else:
    print("Failed to fetch data in correct format.")
```

```
Data successfully exported to file.
```

Load the CSV File

In [6]:

```
df = pd.read_csv(r'C:\Users\thefli0\Downloads\combined_data.csv')
```

Remove Columns

In [7]:

```
# Define columns to be removed for cleaner dataset
columns_to_drop = ['player.injured', 'player.photo', 'seasons', 'league.id', 'leagu
                   'country.name', 'country.code', 'country.flag']
df = df.drop(columns=[col for col in columns_to_drop if col in df.columns])
```

Rename Columns

In [8]:

```
# Define the columns to be changed and what the new headers will be
df = df.rename(columns={
    'statistics': 'player_stats',
    'player.id': 'player_id_num',
    'player.name': 'player_full_name',
    'player.firstname': 'player_first_name',
    'player.lastname': 'player_last_name',
    'player.age': 'age',
    'player.birth.date': 'player_birthdate',
    'player.birth.place': 'place_of_birth',
    'player.birth.country': 'birth_country',
    'player.nationality': 'nationality',
    'player.height': 'height',
    'player.weight': 'weight'
})
```

Change Date Format

In [9]:
```
# Change date format to standard for dates
if 'player_birthdate' in df.columns:
    df['player_birthdate'] = pd.to_datetime(df['player_birthdate'], errors='coerce'
```

Fill Missing Values

In [10]:
```
# Define function to fill missing values with 'NULL'
df = df.fillna('NULL')
```

Remove Duplicates

In [11]:
```
# Perform cleansing of duplicates for unique ID number
df = df.drop_duplicates(subset='player_id_num')
```

Write to New File

In [12]:
```
# Export the newly cleaned data
df.to_csv(r'C:\Users\thefli0\Downloads\cleaned_combined_data.csv', index=False)

print("Data cleaning and transformation complete. New file is saved to 'cleaned_com
```

```
Data cleaning and transformation complete. New file is saved to 'cleaned_combined_da
ta.csv'.
```