SportMonks_API_ETL

September 5, 2024

1 Trevor Maxwell

1.1 Extracting Data from SportsMonk API

1.1.1 2024-08-01

[1]:
```
# import necessary libraries

import pandas as pd
import numpy as np
import json
import requests
import time
from pprint import pprint
import time
```

[2]:
```
# assign api key

api_key = 'your_API_key_here'
```

[3]:
```
# Assign url to obtain teams
url = f"https://api.sportmonks.com/v3/football/teams?
 ↪api_token={api_key}&per_page=50&timezone=America/New_York"

# create player dictionary that will become dataframe from API
teams = {'id':[], 'sport_id':[], 'country_id':[], 'venue_id':[], 'gender':[],␣
 ↪'name':[],
               'short_code':[], 'image_path':[], 'founded':[], 'type':[],␣
 ↪'placeholder':[], 'last_played_at':[]}

# loop through teams API endpoint to obtain all teams from every page
next_page = True
while next_page == True:
#     print(url)
#     time.sleep(1)
    try:
        # Make GET request
        team_dict = requests.get(url)
```

```
    except:
        # print if error occurred
        print(f'Error Occurred.')
        break

    else:
        # turn response to JSON text
        team_dict = team_dict.text
        team_dict_text = json.loads(team_dict)

        # "data" key contains team information
        for key, value in team_dict_text.items():
            if key == 'data':
                num_items = len(value)
    #             print(num_items)
                for team in range(num_items-1):
                    team_dict = value[team]
    #                 print(team_dict)
                    teams['id'].append(team_dict['id'])
                    teams['sport_id'].append(team_dict['sport_id'])
                    teams['country_id'].append(team_dict['country_id'])
                    teams['venue_id'].append(team_dict['venue_id'])
                    teams['gender'].append(team_dict['gender'])
                    teams['name'].append(team_dict['name'])
                    teams['short_code'].append(team_dict['short_code'])
                    teams['image_path'].append(team_dict['image_path'])
                    teams['founded'].append(team_dict['founded'])
                    teams['type'].append(team_dict['type'])
                    teams['placeholder'].append(team_dict['placeholder'])
                    teams['last_played_at'].append(team_dict['last_played_at'])

            # check pagination to see if there are additional pages
            if key == 'pagination':
                if value['has_more'] == True:
                    new_url_start = value["next_page"][:45]
                    new_url_end = value["next_page"][-46:]
                    api_token = f"api_token={api_key}&"
                    url=new_url_start + api_token + new_url_end
#                     print(url)

                # close loop if there are no longer more pages
                else:
                    next_page = False
```

[4]:
```
# create team dataframe from dictionary created from the team API endpoint
```

```
team_df = pd.DataFrame(teams)
team_df.sort_values(by='last_played_at', ascending=False).head()
```

[4]:
```
       id  sport_id  country_id  venue_id gender                    name  \
7   18597         1       479.0   13533.0   male                   Japan
1   15265         1      1739.0   10990.0   male  CS Uruguay de Coronado
0   15251         1       158.0   10900.0   male                 Uruguay
13  18720         1       353.0    2035.0   male                Colombia
4   18572         1      1004.0   11542.0   male                  Canada

   short_code                                          image_path  founded  \
7         JPN  https://cdn.sportmonks.com/images/soccer/teams…   1921.0
1        None  https://cdn.sportmonks.com/images/soccer/teams…   1936.0
0         URU  https://cdn.sportmonks.com/images/soccer/teams…   1900.0
13        COL  https://cdn.sportmonks.com/images/soccer/teams…   1924.0
4         CAN  https://cdn.sportmonks.com/images/soccer/teams…   1912.0

         type  placeholder       last_played_at
7   national        False  2024-09-05 10:35:00
1   national        False  2024-09-04 20:00:00
0   national        False  2024-09-01 23:00:00
13  national        False  2024-07-15 01:15:00
4   national        False  2024-07-14 00:00:00
```

[5]:
```
# See amount of teams and number of columns
team_df.shape
```

[5]:
```
(248, 12)
```

[6]:
```
# Assign url to obtain stages

url = f"https://api.sportmonks.com/v3/football/stages?
 ↪api_token={api_key}&per_page=25&timezone=America/New_York"

# create player dictionary that will become dataframe from API
stages = {'id':[], 'sport_id':[], 'season_id':[], 'type_id':[], 'name':[],␣
 ↪'sort_order':[],
        'finished':[], 'is_current':[], 'starting_at':[], 'ending_at':[],␣
 ↪'games_in_current_week':[],
        'tie_breaker_rule_id':[]}

# loop through teams API endpoint to obtain all stages from every page
next_page = True
while next_page == True:
#     print(url)
#     time.sleep(1)
    try:
```

```
        # Make GET request
        api_dict = requests.get(url)

    except :
        # print if error occurred
        print(f'Error Occurred.')

    else:
        # turn response to JSON text
        stages_dict = api_dict.text
        stages_dict_text = json.loads(stages_dict)

        # "data" key contains team information
        for key, value in stages_dict_text.items():
            if key == 'data':
                num_items = len(value)
    #             print(num_items)
                for team in range(num_items-1):
                    stages_dict = value[team]
    #                 print(team_dict)
                    stages['id'].append(stages_dict['id'])
                    stages['sport_id'].append(stages_dict['sport_id'])
                    stages['season_id'].append(stages_dict['season_id'])
                    stages['type_id'].append(stages_dict['type_id'])
                    stages['name'].append(stages_dict['name'])
                    stages['sort_order'].append(stages_dict['sort_order'])
                    stages['finished'].append(stages_dict['finished'])
                    stages['is_current'].append(stages_dict['is_current'])
                    stages['starting_at'].append(stages_dict['starting_at'])
                    stages['ending_at'].append(stages_dict['ending_at'])
                    stages['games_in_current_week'].
 ↪append(stages_dict['games_in_current_week'])
                    stages['tie_breaker_rule_id'].
 ↪append(stages_dict['tie_breaker_rule_id'])

            if key == 'pagination':
                if value['has_more'] == True:
                    new_url_start = value["next_page"][:46]
                    new_url_end = value["next_page"][-46:]
                    api_token = f"api_token={api_key}&"
                    url=new_url_start + api_token + new_url_end
#                     print(url)
                else:
                    next_page = False
```

[7]:
```
# create stage dataframe from dictionary created from the stage API endpoint
```

```
stages_df = pd.DataFrame(stages)
stages_df.sort_values(by='ending_at', ascending=False).head()
```

[7]:

	id	sport_id	season_id	type_id	name	sort_order	finished	is_current	starting_at	ending_at	games_in_current_week	tie_breaker_rule_id
29	77468453	1	22871	224	Final	5	True	False	2024-07-15	2024-07-15	False	None
30	77468454	1	22871	224	3rd Place Final	4	True	False	2024-07-14	2024-07-14	False	None
31	77468455	1	22871	224	Semi-finals	3	True	False	2024-07-10	2024-07-11	False	None
32	77468456	1	22871	224	Quarter-finals	2	True	False	2024-07-05	2024-07-07	False	None
33	77468457	1	22871	223	Group Stage	1	True	False	2024-06-21	2024-07-03	False	None

[8]:
```
# See amount of stages and number of columns

stages_df.shape
```

[8]: (38, 12)

[9]:
```
# Assign url to obtain fixures

url = f"https://api.sportmonks.com/v3/football/fixtures?
 ↪api_token={api_key}&per_page=50&timezone=America/New_York"

# create player dictionary that will become dataframe from API
fixures = {'id':[], 'sport_id':[], 'league_id':[], 'season_id':[], 'stage_id':
 ↪[], 'group_id':[], 'aggregate_id':[],
        'round_id':[], 'state_id':[], 'venue_id':[], 'name':[], 'starting_at':
 ↪[], 'result_info':[],
        'leg':[], 'details':[], 'length':[], 'placeholder':[], 'has_odds':[],␣
 ↪'has_premium_odds':[],
        'starting_at_timestamp':[]}

# loop through teams API endpoint to obtain all stages from every page
next_page = True
while next_page == True:
```

```
#     print(url)
#     time.sleep(2)
    try:
        # Make GET request
        api_dict = requests.get(url)

    except :
        # print if error occurred
        print(f'Error Occurred.')
        break

    else:
        # turn response to JSON text
        fixtures_dict = api_dict.text
        fixtures_dict_text = json.loads(fixtures_dict)

        # "data" key contains team information
        for key, value in fixtures_dict_text.items():
            if key == 'data':
                num_items = len(value)
    #             print(num_items)
                for fixure in range(num_items-1):
                    fixtures_dict = value[fixure]
    #                 print(team_dict)
                    fixures['id'].append(fixtures_dict['id'])
                    fixures['sport_id'].append(fixtures_dict['sport_id'])
                    fixures['league_id'].append(fixtures_dict['league_id'])
                    fixures['season_id'].append(fixtures_dict['season_id'])
                    fixures['stage_id'].append(fixtures_dict['stage_id'])
                    fixures['group_id'].append(fixtures_dict['group_id'])
                    fixures['aggregate_id'].
 ↪append(fixtures_dict['aggregate_id'])
                    fixures['round_id'].append(fixtures_dict['round_id'])
                    fixures['state_id'].append(fixtures_dict['state_id'])
                    fixures['venue_id'].append(fixtures_dict['venue_id'])
                    fixures['name'].append(fixtures_dict['name'])
                    fixures['starting_at'].append(fixtures_dict['starting_at'])
                    fixures['result_info'].append(fixtures_dict['result_info'])
                    fixures['leg'].append(fixtures_dict['leg'])
                    fixures['details'].append(fixtures_dict['details'])
                    fixures['length'].append(fixtures_dict['length'])
                    fixures['placeholder'].append(fixtures_dict['placeholder'])
                    fixures['has_odds'].append(fixtures_dict['has_odds'])
                    fixures['has_premium_odds'].
 ↪append(fixtures_dict['has_premium_odds'])
                    fixures['starting_at_timestamp'].
 ↪append(fixtures_dict['starting_at_timestamp'])
```

```
            # look for additional pages to obtain data
            if key == 'pagination':
                if value['has_more'] == True:
                    new_url_start = value["next_page"][:48]
                    new_url_end = value["next_page"][-46:]
                    api_token = f"api_token={api_key}&"
                    url=new_url_start + api_token + new_url_end
#                     print(url)
                else:
                    next_page = False
```

[ ]:

[10]:
```
# create fixure dataframe from dictionary created from the fixure API endpoint

fixures_df = pd.DataFrame(fixures)
fixures_df.sort_values(by='starting_at', ascending=False).head()
```

[10]:

	id	sport_id	league_id	season_id	stage_id	group_id	aggregate_id	round_id	state_id	venue_id	name	starting_at	result_info	leg	details	length	placeholder	has_odds	has_premium_odds	starting_at_timestamp
161	19038185	1	1114	22871	77468453	NaN	None	NaN	7	78531.0	Argentina vs Colombia	2024-07-14 21:15:00	Argentina won after extra-time.	1/1	None	90	False	True	True	1721006100
162	19038186	1	1114	22871	77468454	NaN	None	NaN	8	14651.0	Canada vs Uruguay	2024-07-13 20:00:00	Canada won after penalties.	1/1	None	90	False	True	True	1720915200
164	19038188	1	1114	22871	77468455	NaN	None	NaN	5	14651.0	Uruguay vs Colombia	2024-07-10 20:00:00	Colombia won after full-time.	1/1	None	90	False	True	True	1720656000
163	19038187	1	1114	22871	77468455	NaN	None	NaN	5	21826.0	Argentina vs Canada	2024-07-09 20:00:00	Argentina won after full-time.	1/1	None	90	False	True	True	1720569600
167	19038191	1	1114	22871	77468456	NaN	None	NaN	8	343384.0	Uruguay vs Brazil	2024-07-06 21:00:00	Uruguay won after penalties.	1/1	None	90	False	True	True	1720314000

[11]:
```
# create data frame of only finals stage
final_stages = stages_df[stages_df.name == 'Final']
final_stages.head()
```

[11]:
```
          id  sport_id  season_id  type_id   name  sort_order  finished  \
3       2058         1       1037      224  Final           5      True
7      13328         1       5875      224  Final           5      True
15     13353         1       5876      224  Final           5      True
16     13354         1       5877      224  Final           5      True
22  77442284         1      15688      224  Final           5      True

    is_current starting_at   ending_at  games_in_current_week  \
3        False  2016-06-27  2016-06-27                  False
7        False  2007-07-15  2007-07-15                  False
15       False  2011-07-24  2011-07-24                  False
16       False  2015-07-04  2015-07-04                  False
22       False  2019-07-07  2019-07-07                  False

   tie_breaker_rule_id
3                 None
7                 None
15                None
16                None
22                None
```

[12]:
```
# create data frame of last two Copa America Tournaments

last_two_year_stages = stages_df[stages_df.starting_at > '2021-01-01']
last_two_year_stages.head()
```

[12]:
```
          id  sport_id  season_id  type_id            name  sort_order  \
24  77446324         1      16761      224           Final           5
25  77446325         1      16761      224  3rd Place Final           4
26  77446326         1      16761      224      Semi-finals           3
27  77446327         1      16761      224   Quarter-finals           2
28  77446328         1      16761      223      Group Stage           1

    finished  is_current starting_at   ending_at  games_in_current_week  \
24      True       False  2021-07-11  2021-07-11                  False
25      True       False  2021-07-10  2021-07-10                  False
26      True       False  2021-07-05  2021-07-07                  False
27      True       False  2021-07-02  2021-07-04                  False
28      True       False  2021-06-13  2021-06-29                  False

   tie_breaker_rule_id
24                None
25                None
```

```
26                None
27                None
28                None
```

[13]:
```
# join finals to fixures to create list for Argentina's last two championships

fixure_data = fixures_df[['stage_id', 'id', 'result_info']]

last_two_year_fixures = pd.merge(last_two_year_stages, fixure_data,␣
 ↪left_on='id', right_on='stage_id', how='inner')
# final_fixures.head()
all_fixures = last_two_year_fixures['id_y']

all_fixure_ids = list(all_fixures)
# all_fixure_ids
```

[14]:
```
last_two_year_fixures.sort_values(by='starting_at', ascending=False).head()
```

[14]:
```
        id_x  sport_id  season_id  type_id             name  sort_order  \
27  77468453         1      22871      224            Final           5
28  77468454         1      22871      224  3rd Place Final           4
29  77468455         1      22871      224      Semi-finals           3
30  77468455         1      22871      224      Semi-finals           3
32  77468456         1      22871      224   Quarter-finals           2

    finished  is_current starting_at   ending_at  games_in_current_week  \
27      True       False  2024-07-15  2024-07-15                  False
28      True       False  2024-07-14  2024-07-14                  False
29      True       False  2024-07-10  2024-07-11                  False
30      True       False  2024-07-10  2024-07-11                  False
32      True       False  2024-07-05  2024-07-07                  False

   tie_breaker_rule_id  stage_id      id_y                       result_info
27                None  77468453  19038185  Argentina won after extra-time.
28                None  77468454  19038186      Canada won after penalties.
29                None  77468455  19038187   Argentina won after full-time.
30                None  77468455  19038188    Colombia won after full-time.
32                None  77468456  19038190   Venezuela won after penalties.
```

[34]:
```
# Assign url to obtain statistics per fixure ID

# stat IDs used to obtain stats from API
stat_id_list = [45, 42, 58, 34, 52, 57,  80, 56, 81, 55, 99, 84, 47, 86, 41,␣
 ↪83, 51]

# create player dictionary that will become dataframe from API
```

```
stats = {'fixure_id':[], 'season_id':[], 'stage_id':[],␣
 ↪'home_ball_possession_pct':[], 'away_ball_possession_pct':[],
        'home_total_shots':[], 'away_total_shots':[], 'home_shots_blocked':[],␣
 ↪'away_shots_blocked':[], 'home_saves':[],
        'away_saves':[], 'home_goals':[], 'away_goals':[], 'home_team_id':[],␣
 ↪'away_team_id':[],
        'home_fouls':[], 'away_fouls':[],'home_successful_passes':[],␣
 ↪'away_successful_passes':[],
        'home_free_kicks':[], 'away_free_kicks':[], 'home_accurate_crosses':
 ↪[], 'away_accurate_crosses':[],
        'home_yellowcards':[], 'away_yellowcards':[], 'home_corners':[],␣
 ↪'away_corners':[],
        'home_penalty_kicks':[], 'away_penalty_kicks':
 ↪[],'home_shots_on_target':[],
        'away_shots_on_target':[], 'home_shots_off_target':[],␣
 ↪'away_shots_off_target':[], 'home_redcards':[],
        'away_redcards':[], 'home_offsides':[], 'away_offsides':[]}

# Counter for the number of fixure that is being looped through
fixure_num = 0

# loop through each fixure ID
for fixure_id in all_fixure_ids:

    # keep tally of the number of fixures looped through
    fixure_num = fixure_num + 1

#     time.sleep(1)
#     print(fixure_id)
    url = f"https://api.sportmonks.com/v3/football/fixtures/{fixure_id}?
 ↪api_token={api_key}&include=statistics.type"
    print(url)
    stats['fixure_id'].append(fixure_id)

    try:
        # Make GET request
        api_dict = requests.get(url)

    except :
        # print if error occurred
        print(f'Error Occurred.')
    else:
        # turn response to JSON text
        stats_dict = api_dict.text
        stats_dict_text = json.loads(stats_dict)
```

```
        # "data" key contains team information
        for key, value in stats_dict_text.items():
            if key == 'data':

                # "statistics" key contains game stats
                for key2, value2 in value.items():
                    if key2 == 'statistics':

                        # count how many stats to loop through
                        stat_length = len(value2)

                        for stat in range(stat_length):
                            new_stat_dict = value2[stat]

                            # "type" and "data" keys contain the statistical␣
↪features
                            statistic = new_stat_dict['type']
                            stat_value_dict = new_stat_dict['data']
                            stat_value = stat_value_dict['value']

                            if statistic['id'] in stat_id_list:

                                # append home and away ball possession pct and␣
↪team IDs
                                if statistic['id'] == 45:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_team_id'].
↪append(new_stat_dict['participant_id'])
                                        stats['home_ball_possession_pct'].
↪append(stat_value)
                                    else:
                                        stats['away_ball_possession_pct'].
↪append(stat_value)
                                        stats['away_team_id'].
↪append(new_stat_dict['participant_id'])

                                # append home and away total shots
                                elif statistic['id'] == 42:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_total_shots'].
↪append(stat_value)
                                    else:
                                        stats['away_total_shots'].
↪append(stat_value)

                                # append home and away shots blocked
```

```
                                elif statistic['id'] == 58:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_shots_blocked'].
↪append(stat_value)
                                    else:
                                        stats['away_shots_blocked'].
↪append(stat_value)

                                # append home and away saves
                                elif statistic['id'] == 57:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_saves'].append(stat_value)
                                    else:
                                        stats['away_saves'].append(stat_value)

                                # append home and away goals
                                elif statistic['id'] == 52:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_goals'].append(stat_value)
                                    else:
                                        stats['away_goals'].append(stat_value)

                                # append home and away corner kicks
                                elif statistic['id'] == 34:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_corners'].append(stat_value)
                                    else:
                                        stats['away_corners'].append(stat_value)

                                # append home and away fouls
                                elif statistic['id'] == 56:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_fouls'].append(stat_value)
                                    else:
                                        stats['away_fouls'].append(stat_value)

                                # append home and away penalty kicks
                                elif statistic['id'] == 47:
                                    if new_stat_dict['location'] == 'home':
                                        stats['home_penalty_kicks'].
↪append(stat_value)
                                    else:
                                        stats['away_penalty_kicks'].
↪append(stat_value)

                                # append home and away shots on target
                                elif statistic['id'] == 86:
```

```
                                if new_stat_dict['location'] == 'home':
                                    stats['home_shots_on_target'].
↪append(stat_value)
                                else:
                                    stats['away_shots_on_target'].
↪append(stat_value)

                            # append home and away shots on target
                            elif statistic['id'] == 41:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_shots_off_target'].
↪append(stat_value)
                                else:
                                    stats['away_shots_off_target'].
↪append(stat_value)

                            # append home and away yellowcards
                            elif statistic['id'] == 84:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_yellowcards'].
↪append(stat_value)
                                else:
                                    stats['away_yellowcards'].
↪append(stat_value)

                            # append home and away interceptions
                            elif statistic['id'] == 99:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_accurate_crosses'].
↪append(stat_value)
                                else:
                                    stats['away_accurate_crosses'].
↪append(stat_value)

                            # append home and away free kicks
                            elif statistic['id'] == 55:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_free_kicks'].
↪append(stat_value)
                                else:
                                    stats['away_free_kicks'].
↪append(stat_value)

                            # append home and away successful passes
                            elif statistic['id'] == 81:
                                if new_stat_dict['location'] == 'home':
```

```
                                    stats['home_successful_passes'].
↪append(stat_value)
                                else:
                                    stats['away_successful_passes'].
↪append(stat_value)

                            # append home and away redcards
                            elif statistic['id'] == 83:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_redcards'].
↪append(stat_value)
                                else:
                                    stats['away_redcards'].
↪append(stat_value)

                            # append home and away offsides
                            elif statistic['id'] == 51:
                                if new_stat_dict['location'] == 'home':
                                    stats['home_offsides'].
↪append(stat_value)
                                else:
                                    stats['away_offsides'].
↪append(stat_value)

                    # append stage and season ID
                    stats['stage_id'].append(value['stage_id'])
                    stats['season_id'].append(value['season_id'])

                    # append nan value if stat ID was not found for the␣
↪current fixure ID
                    if len(stats['home_ball_possession_pct']) < fixure_num:
                        stats['home_ball_possession_pct'].append(np.nan)

                    if len(stats['away_ball_possession_pct']) < fixure_num:
                        stats['away_ball_possession_pct'].append(np.nan)

                    if len(stats['home_total_shots']) < fixure_num:
                        stats['home_total_shots'].append(np.nan)

                    if len(stats['away_total_shots']) < fixure_num:
                        stats['away_total_shots'].append(np.nan)

                    if len(stats['home_shots_blocked']) < fixure_num:
                        stats['home_shots_blocked'].append(np.nan)

                    if len(stats['away_shots_blocked']) < fixure_num:
```

```
                stats['away_shots_blocked'].append(np.nan)

            if len(stats['home_saves']) < fixure_num:
                stats['home_saves'].append(np.nan)

            if len(stats['away_saves']) < fixure_num:
                stats['away_saves'].append(np.nan)

            if len(stats['home_goals']) < fixure_num:
                stats['home_goals'].append(np.nan)

            if len(stats['away_goals']) < fixure_num:
                stats['away_goals'].append(np.nan)

            if len(stats['home_fouls']) < fixure_num:
                stats['home_fouls'].append(np.nan)

            if len(stats['away_fouls']) < fixure_num:
                stats['away_fouls'].append(np.nan)

            if len(stats['home_successful_passes']) < fixure_num:
                stats['home_successful_passes'].append(np.nan)

            if len(stats['away_successful_passes']) < fixure_num:
                stats['away_successful_passes'].append(np.nan)

            if len(stats['home_free_kicks']) < fixure_num:
                stats['home_free_kicks'].append(np.nan)

            if len(stats['away_free_kicks']) < fixure_num:
                stats['away_free_kicks'].append(np.nan)

            if len(stats['home_accurate_crosses']) < fixure_num:
                stats['home_accurate_crosses'].append(np.nan)

            if len(stats['away_accurate_crosses']) < fixure_num:
                stats['away_accurate_crosses'].append(np.nan)

            if len(stats['home_yellowcards']) < fixure_num:
                stats['home_yellowcards'].append(np.nan)

            if len(stats['away_yellowcards']) < fixure_num:
                stats['away_yellowcards'].append(np.nan)

            if len(stats['home_corners']) < fixure_num:
                stats['home_corners'].append(np.nan)
```

```
                        if len(stats['away_corners']) < fixure_num:
                            stats['away_corners'].append(np.nan)

                        if len(stats['home_penalty_kicks']) < fixure_num:
                            stats['home_penalty_kicks'].append(np.nan)

                        if len(stats['away_penalty_kicks']) < fixure_num:
                            stats['away_penalty_kicks'].append(np.nan)

                        if len(stats['home_shots_on_target']) < fixure_num:
                            stats['home_shots_on_target'].append(np.nan)

                        if len(stats['away_shots_on_target']) < fixure_num:
                            stats['away_shots_on_target'].append(np.nan)

                        if len(stats['home_shots_off_target']) < fixure_num:
                            stats['home_shots_off_target'].append(np.nan)

                        if len(stats['away_shots_off_target']) < fixure_num:
                            stats['away_shots_off_target'].append(np.nan)

                        if len(stats['home_redcards']) < fixure_num:
                            stats['home_redcards'].append(np.nan)

                        if len(stats['away_redcards']) < fixure_num:
                            stats['away_redcards'].append(np.nan)

                        if len(stats['home_offsides']) < fixure_num:
                            stats['home_offsides'].append(np.nan)

                        if len(stats['away_offsides']) < fixure_num:
                            stats['away_offsides'].append(np.nan)
```

https://api.sportmonks.com/v3/football/fixtures/18101472?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101471?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101465?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101466?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101454?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101458?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101459?api_token=Fqe0lU7XfChKje0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/18101463?api_token=Fqe0lU7XfChKj

e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985346?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985347?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985349?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985350?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985352?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985353?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985355?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985356?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985359?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985360?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985361?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985363?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985364?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985367?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985368?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985370?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985371?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985373?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/11985374?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038185?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038186?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038187?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038188?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038189?api_token=Fqe0lU7XfChKj

e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038190?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038191?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038192?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038193?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038194?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038195?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038196?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038197?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038198?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038199?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038200?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038221?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038222?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038223?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038224?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038225?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038226?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038227?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19038228?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040236?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040237?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040238?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040239?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040240?api_token=Fqe0lU7XfChKj

```
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040241?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040242?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
https://api.sportmonks.com/v3/football/fixtures/19040243?api_token=Fqe0lU7XfChKj
e0TeFSFGig4BjZd77FGoQnil4YYwhr69EtoBLoPq2AJ8UNx&include=statistics.type
```

[ ]:

[ ]:

[16]:
```python
# confirm length for each value are the same to translate to DataFrame

for key, value in stats.items():
    value_length = len(value)
    print(f"{key} length: {value_length}")
```

```
fixure_id length: 59
season_id length: 59
stage_id length: 59
home_ball_possession_pct length: 59
away_ball_possession_pct length: 59
home_total_shots length: 59
away_total_shots length: 59
home_shots_blocked length: 59
away_shots_blocked length: 59
home_saves length: 59
away_saves length: 59
home_goals length: 59
away_goals length: 59
home_team_id length: 59
away_team_id length: 59
home_fouls length: 59
away_fouls length: 59
home_successful_passes length: 59
away_successful_passes length: 59
home_free_kicks length: 59
away_free_kicks length: 59
home_accurate_crosses length: 59
away_accurate_crosses length: 59
home_yellowcards length: 59
away_yellowcards length: 59
home_corners length: 59
away_corners length: 59
home_penalty_kicks length: 59
away_penalty_kicks length: 59
home_shots_on_target length: 59
```

```
away_shots_on_target length: 59
home_shots_off_target length: 59
away_shots_off_target length: 59
home_redcards length: 59
away_redcards length: 59
home_offsides length: 59
away_offsides length: 59
```

[17]:
```
# create match statistics dataframe from API

match_stats_df = pd.DataFrame(stats)
match_stats_df.head()
```

[17]:

	fixure_id	season_id	stage_id	home_ball_possession_pct	away_ball_possession_pct	home_total_shots	away_total_shots	home_shots_blocked	away_shots_blocked	home_saves	…	home_penalty_kicks	away_penalty_kicks	home_shots_on_target	away_shots_on_target	home_shots_off_target	away_shots_off_target
0	18101472	16761	77446324	41	59	6	13	2.0	3.0	4.0	…	0	0	2	2	2	5
1	18101471	16761	77446325	45	55	12	10	1.0	2.0	1.0	…	0	0	6	3	6	7
2	18101465	16761	77446326	56	44	15	7	NaN	NaN	2.0	…	0	0	8	2	7	5
3	18101466	16761	77446326	50	50	13	14	NaN	NaN	3.0	…	0	0	4	4	4	3
4	18101454	16761	77446327	41	59	10	11	NaN	NaN	NaN	…	0	0	5	5	5	6

```
   home_redcards  away_redcards  home_offsides  away_offsides
0              0              0            0.0            3.0
1              0              0            1.0            1.0
2              0              0            0.0            3.0
3              0              0            0.0            1.0
4              1              0            NaN            NaN

[5 rows x 37 columns]
```

[ ]:

[18]:
```
# append home and away team information to match stats dataframe

home_team_info = team_df[['id', 'name', 'image_path']].add_prefix('home_')
away_team_info = team_df[['id', 'name', 'image_path']].add_prefix('away_')

stats_x_home_df = pd.merge(match_stats_df, home_team_info,␣
 ↪left_on='home_team_id', right_on='home_id', how='left')
stats_x_team_df = pd.merge(stats_x_home_df, away_team_info,␣
 ↪left_on='away_team_id', right_on='away_id', how='left')
stats_x_team_df.head()
```

[18]:
```
   fixure_id  season_id  stage_id  home_ball_possession_pct  \
0   18101472      16761  77446324                        41
1   18101471      16761  77446325                        45
2   18101465      16761  77446326                        56
3   18101466      16761  77446326                        50
4   18101454      16761  77446327                        41

   away_ball_possession_pct  home_total_shots  away_total_shots  \
0                        59                 6                13
1                        55                12                10
2                        44                15                 7
3                        50                13                14
4                        59                10                11

   home_shots_blocked  away_shots_blocked  home_saves  …  home_redcards  \
0                 2.0                 3.0         4.0  …              0
1                 1.0                 2.0         1.0  …              0
2                 NaN                 NaN         2.0  …              0
3                 NaN                 NaN         3.0  …              0
4                 NaN                 NaN         NaN  …              1

   away_redcards  home_offsides  away_offsides  home_id  home_name  \
0              0            0.0            3.0    18644  Argentina
1              0            1.0            1.0    18720   Colombia
```

```
2              0            0.0            3.0    18704     Brazil
3              0            0.0            1.0    18644  Argentina
4              0            NaN            NaN    18704     Brazil

                                    home_image_path  away_id  away_name  \
0  https://cdn.sportmonks.com/images/soccer/teams…    18704     Brazil
1  https://cdn.sportmonks.com/images/soccer/teams…    18775       Peru
2  https://cdn.sportmonks.com/images/soccer/teams…    18775       Peru
3  https://cdn.sportmonks.com/images/soccer/teams…    18720   Colombia
4  https://cdn.sportmonks.com/images/soccer/teams…    18774      Chile

                                    away_image_path
0  https://cdn.sportmonks.com/images/soccer/teams…
1  https://cdn.sportmonks.com/images/soccer/teams…
2  https://cdn.sportmonks.com/images/soccer/teams…
3  https://cdn.sportmonks.com/images/soccer/teams…
4  https://cdn.sportmonks.com/images/soccer/teams…

[5 rows x 43 columns]
```

[ ]:

[19]:
```
# append fixture info to stats and team info

fixures_info = fixures_df[['id', 'name', 'starting_at', 'result_info']].
 ↪add_prefix('fixure_')
fixures_info

stats_x_team_x_fixure_df = pd.merge(stats_x_team_df, fixures_info,␣
 ↪on='fixure_id', how='left')
stats_x_team_x_fixure_df.head()
```

[19]:
```
   fixure_id  season_id  stage_id  home_ball_possession_pct  \
0   18101472      16761  77446324                        41
1   18101471      16761  77446325                        45
2   18101465      16761  77446326                        56
3   18101466      16761  77446326                        50
4   18101454      16761  77446327                        41

   away_ball_possession_pct  home_total_shots  away_total_shots  \
0                        59                 6                13
1                        55                12                10
2                        44                15                 7
3                        50                13                14
4                        59                10                11

   home_shots_blocked  away_shots_blocked  home_saves  …  away_offsides  \
```

```
0                2.0                3.0         4.0  …            3.0
1                1.0                2.0         1.0  …            1.0
2                NaN                NaN         2.0  …            3.0
3                NaN                NaN         3.0  …            1.0
4                NaN                NaN         NaN  …            NaN

   home_id  home_name                                    home_image_path  \
0    18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
1    18720   Colombia  https://cdn.sportmonks.com/images/soccer/teams…
2    18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…
3    18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
4    18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…

   away_id  away_name                                    away_image_path  \
0    18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…
1    18775       Peru  https://cdn.sportmonks.com/images/soccer/teams…
2    18775       Peru  https://cdn.sportmonks.com/images/soccer/teams…
3    18720   Colombia  https://cdn.sportmonks.com/images/soccer/teams…
4    18774      Chile  https://cdn.sportmonks.com/images/soccer/teams…

             fixure_name   fixure_starting_at             fixure_result_info
0    Argentina vs Brazil  2021-07-10 20:00:00  Argentina won after full-time.
1       Colombia vs Peru  2021-07-09 20:00:00   Colombia won after full-time.
2         Brazil vs Peru  2021-07-05 19:00:00     Brazil won after full-time.
3  Argentina vs Colombia  2021-07-06 21:00:00  Argentina won after penalties.
4        Brazil vs Chile  2021-07-02 20:00:00     Brazil won after full-time.

[5 rows x 46 columns]
```

[ ]:

[20]:
```
# append stages info to stats, team and fixure info

stages_info = stages_df[['id', 'name']].add_prefix('stage_')

stats_x_team_x_fixure_x_stage_df = pd.merge(stats_x_team_x_fixure_df,␣
 ↪stages_info, on='stage_id', how='left')
stats_x_team_x_fixure_x_stage_df.head()
```

[20]:
```
   fixure_id  season_id  stage_id  home_ball_possession_pct  \
0   18101472      16761  77446324                        41
1   18101471      16761  77446325                        45
2   18101465      16761  77446326                        56
3   18101466      16761  77446326                        50
4   18101454      16761  77446327                        41

   away_ball_possession_pct  home_total_shots  away_total_shots  \
```

```
0                      59                6                13
1                      55               12                10
2                      44               15                 7
3                      50               13                14
4                      59               10                11

   home_shots_blocked  away_shots_blocked  home_saves  …  home_id  \
0                 2.0                 3.0         4.0  …    18644
1                 1.0                 2.0         1.0  …    18720
2                 NaN                 NaN         2.0  …    18704
3                 NaN                 NaN         3.0  …    18644
4                 NaN                 NaN         NaN  …    18704

   home_name                                  home_image_path  away_id  \
0  Argentina  https://cdn.sportmonks.com/images/soccer/teams…    18704
1   Colombia  https://cdn.sportmonks.com/images/soccer/teams…    18775
2     Brazil  https://cdn.sportmonks.com/images/soccer/teams…    18775
3  Argentina  https://cdn.sportmonks.com/images/soccer/teams…    18720
4     Brazil  https://cdn.sportmonks.com/images/soccer/teams…    18774

   away_name                                  away_image_path  \
0     Brazil  https://cdn.sportmonks.com/images/soccer/teams…
1       Peru  https://cdn.sportmonks.com/images/soccer/teams…
2       Peru  https://cdn.sportmonks.com/images/soccer/teams…
3   Colombia  https://cdn.sportmonks.com/images/soccer/teams…
4      Chile  https://cdn.sportmonks.com/images/soccer/teams…

             fixure_name   fixure_starting_at             fixure_result_info  \
0    Argentina vs Brazil  2021-07-10 20:00:00  Argentina won after full-time.
1       Colombia vs Peru  2021-07-09 20:00:00   Colombia won after full-time.
2         Brazil vs Peru  2021-07-05 19:00:00     Brazil won after full-time.
3  Argentina vs Colombia  2021-07-06 21:00:00  Argentina won after penalties.
4        Brazil vs Chile  2021-07-02 20:00:00     Brazil won after full-time.

        stage_name
0            Final
1  3rd Place Final
2      Semi-finals
3      Semi-finals
4   Quarter-finals

[5 rows x 47 columns]
```

[21]:
```
# keep and order columns wanted

ordered_cols = [
'fixure_id',
```

```
'season_id',
'stage_id',
'fixure_name',
'fixure_result_info',
'fixure_starting_at',
'stage_name',
'home_team_id',
'home_name',
'home_image_path',
'away_team_id',
'away_name',
'away_image_path',
'home_goals',
'away_goals',
'home_ball_possession_pct',
'away_ball_possession_pct',
'home_total_shots',
'away_total_shots',
'home_shots_blocked',
'away_shots_blocked',
'home_saves',
'away_saves',
'home_fouls',
'away_fouls',
'home_successful_passes',
'away_successful_passes',
'home_free_kicks',
'away_free_kicks',
'home_accurate_crosses',
'away_accurate_crosses',
'home_yellowcards',
'away_yellowcards',
'home_corners',
'away_corners',
'home_penalty_kicks',
'away_penalty_kicks',
'home_shots_on_target',
'away_shots_on_target',
'home_shots_off_target',
'away_shots_off_target',
'home_redcards',
'away_redcards',
'home_offsides',
'away_offsides'
]

final_stats_df = stats_x_team_x_fixure_x_stage_df[ordered_cols]
```

```
final_stats_df.sort_values(by='fixure_starting_at', ascending=False).head()
```

[21]:

```
    fixure_id  season_id  stage_id            fixure_name  \
27   19038185      22871  77468453  Argentina vs Colombia
28   19038186      22871  77468454      Canada vs Uruguay
30   19038188      22871  77468455    Uruguay vs Colombia
29   19038187      22871  77468455    Argentina vs Canada
33   19038191      22871  77468456      Uruguay vs Brazil

                fixure_result_info   fixure_starting_at       stage_name  \
27  Argentina won after extra-time.  2024-07-14 21:15:00            Final
28     Canada won after penalties.  2024-07-13 20:00:00  3rd Place Final
30    Colombia won after full-time.  2024-07-10 20:00:00      Semi-finals
29   Argentina won after full-time.  2024-07-09 20:00:00      Semi-finals
33     Uruguay won after penalties.  2024-07-06 21:00:00   Quarter-finals

    home_team_id  home_name  \
27         18644  Argentina
28         18572     Canada
30         15251    Uruguay
29         18644  Argentina
33         15251    Uruguay

                                    home_image_path  …  \
27  https://cdn.sportmonks.com/images/soccer/teams…  …
28  https://cdn.sportmonks.com/images/soccer/teams…  …
30  https://cdn.sportmonks.com/images/soccer/teams…  …
29  https://cdn.sportmonks.com/images/soccer/teams…  …
33  https://cdn.sportmonks.com/images/soccer/teams…  …

    home_penalty_kicks away_penalty_kicks home_shots_on_target  \
27                   0                  0                    6
28                   0                  0                    6
30                   0                  0                    2
29                   0                  0                    3
33                   0                  0                    1

    away_shots_on_target  home_shots_off_target  away_shots_off_target  \
27                     4                      4                      8
28                     5                      5                      3
30                     4                      8                      7
29                     2                      6                      5
33                     3                      7                      1

    home_redcards  away_redcards  home_offsides  away_offsides
27              0              0            2.0            1.0
28              0              0            2.0            2.0
```

```
30              0              1            4.0            2.0
29              0              0            2.0            0.0
33              1              0            2.0            0.0

[5 rows x 45 columns]
```

[22]:
```
# check datatypes

final_stats_df.dtypes
```

[22]:
```
fixure_id                   int64
season_id                   int64
stage_id                    int64
fixure_name                object
fixure_result_info         object
fixure_starting_at         object
stage_name                 object
home_team_id                int64
home_name                  object
home_image_path            object
away_team_id                int64
away_name                  object
away_image_path            object
home_goals                  int64
away_goals                  int64
home_ball_possession_pct    int64
away_ball_possession_pct    int64
home_total_shots            int64
away_total_shots            int64
home_shots_blocked        float64
away_shots_blocked        float64
home_saves                float64
away_saves                float64
home_fouls                  int64
away_fouls                  int64
home_successful_passes    float64
away_successful_passes    float64
home_free_kicks             int64
away_free_kicks             int64
home_accurate_crosses     float64
away_accurate_crosses     float64
home_yellowcards            int64
away_yellowcards            int64
home_corners                int64
away_corners                int64
home_penalty_kicks          int64
away_penalty_kicks          int64
```

```
home_shots_on_target          int64
away_shots_on_target          int64
home_shots_off_target         int64
away_shots_off_target         int64
home_redcards                 int64
away_redcards                 int64
home_offsides               float64
away_offsides               float64
dtype: object
```

[23]:
```
# translate fixure_starting_at from object datatype to datetime

final_stats_df['fixure_starting_at'] = pd.
 ↪to_datetime(final_stats_df['fixure_starting_at'], format='%Y-%m-%d %H:%M:%S')
final_stats_df['fixure_starting_at'].dtype
```

```
C:\Users\trevo\AppData\Local\Temp\ipykernel_19500\1342700550.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  final_stats_df['fixure_starting_at'] =
pd.to_datetime(final_stats_df['fixure_starting_at'], format='%Y-%m-%d %H:%M:%S')
```

[23]:
```
dtype('<M8[ns]')
```

[24]:
```
# check fixure_starting_at datatype after translation

final_stats_df.dtypes
```

[24]:
```
fixure_id                          int64
season_id                          int64
stage_id                           int64
fixure_name                       object
fixure_result_info                object
fixure_starting_at          datetime64[ns]
stage_name                        object
home_team_id                       int64
home_name                         object
home_image_path                   object
away_team_id                       int64
away_name                         object
away_image_path                   object
home_goals                         int64
away_goals                         int64
home_ball_possession_pct           int64
```

```
away_ball_possession_pct            int64
home_total_shots                    int64
away_total_shots                    int64
home_shots_blocked                float64
away_shots_blocked                float64
home_saves                        float64
away_saves                        float64
home_fouls                          int64
away_fouls                          int64
home_successful_passes            float64
away_successful_passes            float64
home_free_kicks                     int64
away_free_kicks                     int64
home_accurate_crosses             float64
away_accurate_crosses             float64
home_yellowcards                    int64
away_yellowcards                    int64
home_corners                        int64
away_corners                        int64
home_penalty_kicks                  int64
away_penalty_kicks                  int64
home_shots_on_target                int64
away_shots_on_target                int64
home_shots_off_target               int64
away_shots_off_target               int64
home_redcards                       int64
away_redcards                       int64
home_offsides                     float64
away_offsides                     float64
dtype: object
```

[25]:
```
# view US games

final_stats_df[(final_stats_df.home_team_id == 18571) | (final_stats_df.
 ↪away_team_id == 18571)]
```

[25]:
```
    fixure_id  season_id  stage_id              fixure_name  \
40   19038198      22871  77468457  United States vs Bolivia
47   19038225      22871  77468457   Panama vs United States
48   19038226      22871  77468457  United States vs Uruguay

                   fixure_result_info  fixure_starting_at   stage_name  \
40  United States won after full-time. 2024-06-23 18:00:00  Group Stage
47         Panama won after full-time. 2024-06-27 18:00:00  Group Stage
48        Uruguay won after full-time. 2024-07-01 21:00:00  Group Stage

    home_team_id      home_name  \
```

```
40         18571  United States
47         18717         Panama
48         18571  United States

                                    home_image_path  …  \
40  https://cdn.sportmonks.com/images/soccer/teams…  …
47  https://cdn.sportmonks.com/images/soccer/teams…  …
48  https://cdn.sportmonks.com/images/soccer/teams…  …

    home_penalty_kicks away_penalty_kicks home_shots_on_target  \
40                   0                  0                    8
47                   1                  0                    4
48                   0                  0                    3

    away_shots_on_target  home_shots_off_target  away_shots_off_target  \
40                     3                      6                      2
47                     3                      2                      3
48                     5                      2                      6

    home_redcards  away_redcards  home_offsides  away_offsides
40              0              0            1.0            1.0
47              1              1            2.0            3.0
48              0              0            1.0            0.0

[3 rows x 45 columns]
```

[26]:
```
# determine max length of each field for SQL table creation

for col in final_stats_df.columns:
    if final_stats_df[col].dtype == 'object':
        max_length = final_stats_df[col].apply(len).max()
        print(f"max length for {col}: {max_length}")
```

```
max length for fixure_name: 24
max length for fixure_result_info: 34
max length for stage_name: 15
max length for home_name: 13
max length for home_image_path: 59
max length for away_name: 13
max length for away_image_path: 59
```

[27]:
```
# load final_stats_df to SQL

from sqlalchemy import create_engine
import urllib

# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES
# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES
```

```
# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES

# params = urllib.parse.quote_plus(
#     "DRIVER={SQL Server};"
#     "SERVER=your_server_name;"
#     "DATABASE=Projects;"
#     "Trusted_Connection=yes;"
# )

# engine = create_engine(f"mssql+pyodbc:///?odbc_connect={params}")

# final_stats_df.to_sql('Copa_America_Matches', con=engine, if_exists='append',␣
 ↪index=False)

print("data imported successfully")
```

```
data imported successfully
```

[28]:
```
# melt data for tableau

melted_cols =␣
 ↪['home_goals','away_goals','home_ball_possession_pct','away_ball_possession_pct','home_tota
              ␣
 ↪'away_total_shots','home_shots_blocked','away_shots_blocked','home_saves','away_saves','hom
              ␣
 ↪'away_fouls','home_successful_passes','away_successful_passes','home_free_kicks','away_free
              ␣
 ↪'home_accurate_crosses','away_accurate_crosses','home_yellowcards','away_yellowcards','home
              ␣
 ↪'away_corners','home_penalty_kicks','away_penalty_kicks','home_shots_on_target','away_shots
              ␣
 ↪'home_shots_off_target','away_shots_off_target','home_redcards','away_redcards','home_offsi
               'away_offsides']

id_cols =␣
 ↪['fixure_id','season_id','stage_id','fixure_name','fixure_result_info','fixure_starting_at'
         ␣
 ↪'home_team_id','home_name','home_image_path','away_team_id','away_name','away_image_path']

final_stats_df_melted = final_stats_df.melt(id_vars=id_cols,
                                            value_vars=melted_cols,
                                            var_name = 'statistic',
                                            value_name = 'value')
```

[29]:
```
# Add row number to dataframe as a primary key

final_stats_df_melted['row_id']= range(1, len(final_stats_df_melted) + 1)
```

```
final_stats_df_melted.head()
```

[29]:
```
   fixure_id  season_id  stage_id            fixure_name  \
0   18101472      16761  77446324    Argentina vs Brazil
1   18101471      16761  77446325       Colombia vs Peru
2   18101465      16761  77446326         Brazil vs Peru
3   18101466      16761  77446326  Argentina vs Colombia
4   18101454      16761  77446327        Brazil vs Chile

               fixure_result_info  fixure_starting_at       stage_name  \
0  Argentina won after full-time. 2021-07-10 20:00:00            Final
1   Colombia won after full-time. 2021-07-09 20:00:00  3rd Place Final
2     Brazil won after full-time. 2021-07-05 19:00:00      Semi-finals
3  Argentina won after penalties. 2021-07-06 21:00:00      Semi-finals
4     Brazil won after full-time. 2021-07-02 20:00:00   Quarter-finals

   home_team_id  home_name                                    home_image_path  \
0         18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
1         18720   Colombia  https://cdn.sportmonks.com/images/soccer/teams…
2         18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…
3         18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
4         18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…

   away_team_id away_name                                    away_image_path  \
0         18704    Brazil  https://cdn.sportmonks.com/images/soccer/teams…
1         18775      Peru  https://cdn.sportmonks.com/images/soccer/teams…
2         18775      Peru  https://cdn.sportmonks.com/images/soccer/teams…
3         18720  Colombia  https://cdn.sportmonks.com/images/soccer/teams…
4         18774     Chile  https://cdn.sportmonks.com/images/soccer/teams…

    statistic  value  row_id
0  home_goals    1.0       1
1  home_goals    3.0       2
2  home_goals    1.0       3
3  home_goals    1.0       4
4  home_goals    1.0       5
```

[30]:
```
# create column determining if the stat is for the home team or away team

location_indicator = []

for value in final_stats_df_melted['statistic']:
    if value[:4] == 'home':
        location_indicator.append('home')
    else:
        location_indicator.append('away')
```

```
final_stats_df_melted['team_indicator'] = location_indicator

# remove home_ and away_ from value in stat
final_stats_df_melted['statistic'] = final_stats_df_melted['statistic'].str.
 ↪slice(start=5)
final_stats_df_melted.tail()
```

[30]:
```
      fixure_id  season_id  stage_id             fixure_name  \
1883   19040239      22871  77468457    Jamaica vs Venezuela
1884   19040240      22871  77468457      Uruguay vs Bolivia
1885   19040241      22871  77468457       Bolivia vs Panama
1886   19040242      22871  77468457  Colombia vs Costa Rica
1887   19040243      22871  77468457  Costa Rica vs Paraguay

                  fixure_result_info  fixure_starting_at   stage_name  \
1883   Venezuela won after full-time. 2024-06-30 20:00:00  Group Stage
1884     Uruguay won after full-time. 2024-06-27 21:00:00  Group Stage
1885      Panama won after full-time. 2024-07-01 21:00:00  Group Stage
1886    Colombia won after full-time. 2024-06-28 18:00:00  Group Stage
1887  Costa Rica won after full-time. 2024-07-02 21:00:00  Group Stage

      home_team_id   home_name  \
1883         18771     Jamaica
1884         15251     Uruguay
1885         18825     Bolivia
1886         18720    Colombia
1887         18598  Costa Rica

                                      home_image_path  away_team_id  \
1883  https://cdn.sportmonks.com/images/soccer/teams…         18711
1884  https://cdn.sportmonks.com/images/soccer/teams…         18825
1885  https://cdn.sportmonks.com/images/soccer/teams…         18717
1886  https://cdn.sportmonks.com/images/soccer/teams…         18598
1887  https://cdn.sportmonks.com/images/soccer/teams…         18723

       away_name                                    away_image_path statistic  \
1883   Venezuela  https://cdn.sportmonks.com/images/soccer/teams…  offsides
1884     Bolivia  https://cdn.sportmonks.com/images/soccer/teams…  offsides
1885      Panama  https://cdn.sportmonks.com/images/soccer/teams…  offsides
1886  Costa Rica  https://cdn.sportmonks.com/images/soccer/teams…  offsides
1887    Paraguay  https://cdn.sportmonks.com/images/soccer/teams…  offsides

      value  row_id team_indicator
1883    1.0    1884           away
1884    0.0    1885           away
1885    0.0    1886           away
```

```
1886    1.0    1887           away
1887    1.0    1888           away
```

[31]:
```
# order columns in DataFrame

ordered_cols = [
                'row_id',
                'fixure_id',
                'season_id',
                'stage_id',
                'fixure_name',
                'fixure_result_info',
                'fixure_starting_at',
                'stage_name',
                'home_team_id',
                'home_name',
                'home_image_path',
                'away_team_id',
                'away_name',
                'away_image_path',
                'team_indicator',
                'statistic',
                'value'
                ]

final_stats_df_melted = final_stats_df_melted[ordered_cols]
final_stats_df_melted.head()
```

[31]:
```
   row_id  fixure_id  season_id  stage_id            fixure_name  \
0       1   18101472      16761  77446324    Argentina vs Brazil
1       2   18101471      16761  77446325       Colombia vs Peru
2       3   18101465      16761  77446326         Brazil vs Peru
3       4   18101466      16761  77446326  Argentina vs Colombia
4       5   18101454      16761  77446327        Brazil vs Chile

                fixure_result_info  fixure_starting_at       stage_name  \
0  Argentina won after full-time. 2021-07-10 20:00:00            Final
1   Colombia won after full-time. 2021-07-09 20:00:00  3rd Place Final
2     Brazil won after full-time. 2021-07-05 19:00:00      Semi-finals
3  Argentina won after penalties. 2021-07-06 21:00:00      Semi-finals
4     Brazil won after full-time. 2021-07-02 20:00:00   Quarter-finals

   home_team_id  home_name                                  home_image_path  \
0         18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
1         18720   Colombia  https://cdn.sportmonks.com/images/soccer/teams…
2         18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…
3         18644  Argentina  https://cdn.sportmonks.com/images/soccer/teams…
```

```
4         18704     Brazil  https://cdn.sportmonks.com/images/soccer/teams…

   away_team_id away_name                                    away_image_path  \
0         18704    Brazil  https://cdn.sportmonks.com/images/soccer/teams…
1         18775      Peru  https://cdn.sportmonks.com/images/soccer/teams…
2         18775      Peru  https://cdn.sportmonks.com/images/soccer/teams…
3         18720  Colombia  https://cdn.sportmonks.com/images/soccer/teams…
4         18774     Chile  https://cdn.sportmonks.com/images/soccer/teams…

  team_indicator statistic  value
0           home     goals    1.0
1           home     goals    3.0
2           home     goals    1.0
3           home     goals    1.0
4           home     goals    1.0
```

[32]:
```python
# determine max length of each field for SQL table creation

for col in final_stats_df_melted.columns:
    if final_stats_df_melted[col].dtype == 'object':
        max_length = final_stats_df_melted[col].apply(len).max()
        print(f"max length for {col}: {max_length}")
```

```
max length for fixure_name: 24
max length for fixure_result_info: 34
max length for stage_name: 15
max length for home_name: 13
max length for home_image_path: 59
max length for away_name: 13
max length for away_image_path: 59
max length for team_indicator: 4
max length for statistic: 19
```

[33]:
```python
# load final_stats_df to SQL

from sqlalchemy import create_engine
import urllib

# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES
# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES
# LOAD TO SQL COMMENTED OUT TO AVOID ADDING MATCHES MULTIPLE TIMES

# params = urllib.parse.quote_plus(
#     "DRIVER={SQL Server};"
#     "SERVER=your_server_name;"
#     "DATABASE=Projects;"
#     "Trusted_Connection=yes;"
# )
```

```
# engine = create_engine(f"mssql+pyodbc:///?odbc_connect={params}")

# final_stats_df_melted.to_sql('Copa_America_Statistics', con=engine,␣
  ↪if_exists='append', index=False)

print("data imported successfully")
```

```
data imported successfully
```

[ ]: