sportsreference Documentation

Release 0.1.0

Author

Sep 04, 2018

Contents:

1 Examples **3**
- 1.1 Get instances of all NHL teams for the 2018 season . . . 3
- 1.2 Print every NBA team's name and abbreviation . . . 3
- 1.3 Get a specific NFL team's season information . . . 3
- 1.4 Print the date of every game for a NCAA Men's Basketball team . . . 4
- 1.5 Print the number of interceptions by the away team in a NCAA Football game . . . 4
- 1.6 Get a Pandas DataFrame of all stats for a MLB game . . . 4
  - 1.6.1 sportsreference package . . . 4
    - 1.6.1.1 Subpackages . . . 4

2 Indices and tables **71**

Python Module Index **73**

Sportsreference is a free python API that pulls the stats from www.sports-reference.com and allows them to be easily be used in python-based applications, especially ones involving data analytics and machine learning.

Sportsreference exposes a plethora of sports information from major sports leagues in North America, such as the MLB, NBA, College Football and Basketball, NFL, and NHL. Every sport has its own set of valid API queries ranging from the list of teams in a league, to the date and time of a game, to the total number of wins a team has secured during the season, and many, many more metrics that paint a more detailed picture of how a team has performed during a game or throughout a season.

CHAPTER 1

Examples

The following are a few examples showcasing how easy it can be to collect an abundance of metrics and information from all of the tracked leagues. The examples below are only a miniscule subset of the total number of statistics that can be pulled using sportsreference. Visit the documentation on Read The Docs for a complete list of all information exposed by the API.

1.1 Get instances of all NHL teams for the 2018 season

```
from sportsreference.nhl.teams import Teams

teams = Teams(2018)
```

1.2 Print every NBA team's name and abbreviation

```
from sportsreference.nba.teams import Teams

teams = Teams()
for team in teams:
    print(team.name, team.abbreviation)
```

1.3 Get a specific NFL team's season information

```
from sportsreference.nfl.teams import Teams

teams = Teams()
lions = teams('DET')
```

1.4 Print the date of every game for a NCAA Men's Basketball team

```
from sportsreference.ncaab.schedule import Schedule

purdue_schedule = Schedule('purdue')
for game in purdue_schedule:
    print(game.date)
```

1.5 Print the number of interceptions by the away team in a NCAA Football game

```
from sportsreference.ncaaf.boxscore import Boxscore

championship_game = Boxscore('2018-01-08-georgia')
print(championship_game.away_interceptions)
```

1.6 Get a Pandas DataFrame of all stats for a MLB game

```
from sportsreference.mlb.boxscore import Boxscore

game = Boxscore('BOS201806070')
df = game.dataframe
```

1.6.1 sportsreference package

1.6.1.1 Subpackages

sportsreference.mlb package

The MLB package offers multiple modules which can be used to retrieve information and statistics for Major League Baseball, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.mlb.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of runs scored to the number of sacrifice flies, to the slugging percentage and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.mlb.boxscore import Boxscore

game_data = Boxscore('BOS/BOS201808020')
print(game_data.home_runs)  # Prints 15
print(game_data.away_runs)  # Prints 7
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.mlb.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.mlb.boxscore.**Boxscore**(*uri*)

Bases: object

Detailed information about the final statistics for a game.

Stores all relevant information for a game such as the date, time, location, result, and more advanced metrics such as the number of strikes, a pitcher's influence on the game, the number of putouts and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as 'BOS/BOS201806070'.

attendance
Returns an `int` of the game's listed attendance.

away_assists
Returns an `int` of the number of assists the away team registered.

away_at_bats
Returns an `int` of the number of at bats the away team had.

away_average_leverage_index
Returns a `float` of the amount of pressure the away team's pitcher faced during the game. 1.0 denotes average pressure while numbers less than 0 denote lighter pressure.

away_base_out_runs_added
Returns a `float` of the number of base out runs added by the away team.

away_base_out_runs_saved
Returns a `float` of the number of runs saved by the away pitcher based on the number of players on bases. 0.0 denotes an average value.

away_bases_on_balls
Returns an `int` of the number of bases the away team registerd as a result of balls.

away_batting_average
Returns a `float` of the batting average for the away team.

away_earned_runs
Returns a `float` of the number of runs the away team earned.

away_fly_balls
Returns an `int` of the number of fly balls the away team allowed.

away_game_score
Returns an `int` of the starting away pitcher's score determine by many factors, such as number of runs scored against, number of strikes, etc.

away_grounded_balls
Returns an `int` of the number of grounded balls the away team allowed.

away_hits
Returns an `int` of the number of hits the away team had.

away_home_runs
Returns an `int` of the number of times the away team gave up a home run.

away_inherited_runners
Returns an `int` of the number of runners a pitcher inherited when he entered the game.

away_inherited_score
Returns an `int` of the number of scorers a pitcher inherited when he entered the game.

away_innings_pitched
Returns a `float` of the number of innings the away team pitched.

away_line_drives
Returns an `int` of the number of line drives the away team allowed.

away_on_base_percentage
Returns a `float` of the percentage of at bats that result in the batter getting on base.

away_on_base_plus
Returns a `float` of the on base percentage plus the slugging percentage. Percentage ranges from 0-1.

away_pitches
Returns an `int` of the number of pitches the away team faced.

away_plate_appearances
Returns an `int` of the number of plate appearances the away team made.

away_putouts
Returns an `int` of the number of putouts the away team registered.

away_rbi
Returns an `int` of the number of runs batted in the away team registered.

away_runs
Returns an `int` of the number of runs the away team scored.

away_slugging_percentage
Returns a `float` of the slugging percentage for the away team based on the number of bases gained per at-bat with bigger plays getting more weight.

away_strikeouts
Returns an `int` of the number of times the away team was struck out.

away_strikes
Returns an `int` of the number of times a strike was called against the away team.

away_strikes_by_contact
Returns an `int` of the number of times the away team struck out a batter who made contact with the pitch.

away_strikes_looking
Returns an `int` of the number of times the away team struck out a batter who was looking.

away_strikes_swinging
Returns an `int` of the number of times the away team struck out a batter who was swinging.

away_unknown_bat_type
Returns an `int` of the number of away at bats that were not properly tracked and therefore cannot be safely placed in another statistical category.

away_win_probability_added
Returns a `float` of the total positive influence the away team's offense had on the outcome of the game.

away_win_probability_by_pitcher
Returns a `float` of the amount of influence the away pitcher had on the game's result with 0.0 denoting zero influence and 1.0 denoting he was solely responsible for the team's win.

away_win_probability_for_offensive_player
Returns a `float` of the overall influence the away team's offense had on the outcome of the game where 0.0 denotes no influence and 1.0 denotes the offense was solely responsible for the outcome.

away_win_probability_subtracted
Returns a `float` of the total negative influence the away team's offense had on the outcome of the game.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as 'BOS201806070'.

date
Returns a `string` of the date the game took place.

duration
Returns a `string` of the game's duration in the format 'H – MM'.

home_assists
Returns an `int` of the number of assists the home team registered.

home_at_bats
Returns an `int` of the number of at bats the home team had.

home_average_leverage_index
Returns a `float` of the amount of pressure the home team's pitcher faced during the game. 1.0 denotes average pressure while numbers less than 0 denote lighter pressure.

home_base_out_runs_added
Returns a `float` of the number of base out runs added by the home team.

home_base_out_runs_saved
Returns a `float` of the number of runs saved by the home pitcher based on the number of players on bases. 0.0 denotes an average value.

home_bases_on_balls
Returns an `int` of the number of bases the home team registerd as a result of balls.

home_batting_average
Returns a `float` of the batting average for the home team.

home_earned_runs
Returns a `float` of the number of runs the home team earned.

home_fly_balls
Returns an `int` of the number of fly balls the home team allowed.

home_game_score
Returns an `int` of the starting home pitcher's score determine by many factors, such as number of runs scored against, number of strikes, etc.

home_grounded_balls
Returns an `int` of the number of grounded balls the home team allowed.

home_hits
Returns an `int` of the number of hits the home team had.

home_home_runs
Returns an `int` of the number of times the home team gave up a home run.

home_inherited_runners
Returns an `int` of the number of runners a pitcher inherited when he entered the game.

home_inherited_score
Returns an `int` of the number of scorers a pitcher inherited when he entered the game.

home_innings_pitched
Returns a `float` of the number of innings the home team pitched.

home_line_drives
Returns an `int` of the number of line drives the home team allowed.

home_on_base_percentage
Returns a `float` of the percentage of at bats that result in the batter getting on base.

home_on_base_plus
Returns a `float` of the on base percentage plus the slugging percentage. Percentage ranges from 0-1.

home_pitches
Returns an `int` of the number of pitches the home team faced.

home_plate_appearances
Returns an `int` of the number of plate appearances the home team made.

home_putouts
Returns an `int` of the number of putouts the home team registered.

home_rbi
Returns an `int` of the number of runs batted in the home team registered.

home_runs
Returns an `int` of the number of runs the home team scored.

home_slugging_percentage
Returns a `float` of the slugging percentage for the home team based on the number of bases gained per at-bat with bigger plays getting more weight.

home_strikeouts
Returns an `int` of the number of times the home team was struck out.

home_strikes
Returns an `int` of the number of times a strike was called against the home team.

home_strikes_by_contact
Returns an `int` of the number of times the home team struck out a batter who made contact with the pitch.

home_strikes_looking
Returns an `int` of the number of times the home team struck out a batter who was looking.

home_strikes_swinging
Returns an `int` of the number of times the home team struck out a batter who was swinging.

home_unknown_bat_type
Returns an `int` of the number of home at bats that were not properly tracked and therefore cannot be safely placed in another statistical category.

home_win_probability_added
Returns a `float` of the total positive influence the home team's offense had on the outcome of the game.

home_win_probability_by_pitcher
Returns a `float` of the amount of influence the home pitcher had on the game's result with 0.0 denoting zero influence and 1.0 denoting he was solely responsible for the team's win.

home_win_probability_for_offensive_player
Returns a `float` of the overall influence the home team's offense had on the outcome of the game where 0.0 denotes no influence and 1.0 denotes the offense was solely responsible for the outcome.

home_win_probability_subtracted
Returns a `float` of the total negative influence the home team's offense had on the outcome of the game.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'LAD' for the Los Angeles Dodgers.

losing_name
Returns a `string` of the losing team's name, such as 'Los Angeles Dodgers'.

time
Returns a `string` of the time the game started.

time_of_day
Returns a `string` constant indicated whether the game was played during the day or at night.

venue
Returns a `string` of the name of the ballpark where the game was played.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'HOU' for the Houston Astros.

winning_name
Returns a `string` of the winning team's name, such as 'Houston Astros'.

class sportsreference.mlb.boxscore.**Boxscores**(*date*)
Bases: `object`

Search for MLB games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters **date** (*datetime object*) – The date to search for any matches. The month, day, and year are required for the search, but time is not factored into the search.

games
Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{
    'boxscores': [
        'home_name': Name of the home team, such as 'New York
                     Yankees' (`str`),
        'home_abbr': Abbreviation for the home team, such as 'NYY'
                     (`str`),
        'away_name': Name of the away team, such as 'Houston
                     Astros' (`str`),
        'away_abbr': Abbreviation for the away team, such as 'HOU'
                     (`str`),
        'boxscore': String representing the boxscore URI, such as
                    'SLN/SLN201807280' (`str`)},
        { ... },
        ...
```

(continues on next page)

(continued from previous page)

```
    ]
}
```

If no games were played during the requested day, the list for ['boxscores'] will be empty.

sportsreference.mlb.schedule module

The Schedule module can be used to iterate over all games in a team's schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.mlb.schedule import Schedule

houston_schedule = Schedule('HOU')
for game in houston_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class sportsreference.mlb.schedule.**Game**(*game_data*, *year*)

Bases: object

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team's schedule including date, time, opponent, and result.

Parameters

- **game_data** (*string*) – The row containing the specified game information.
- **year** (*string*) – The year of the current season.

attendance

Returns an `int` of the total listed attendance for the game.

boxscore

Returns an instance of the Boxscore class containing more detailed stats on the game.

dataframe

Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended

Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date

Returns a `string` of the date the game was played on.

datetime

Returns a datetime object of the month, day, year, and time the game was played.

day_or_night

Returns a `string` constant to indicate whether the game was played during the day or night.

game
Returns an `int` of the game in the season, where 1 is the first game of the season.

game_duration
Returns a `string` of the game's total duration in the format 'H – MM'.

game_number_for_day
Returns an `int` denoting which game is played for the team during the given day. Default value is 1 where a team plays only one game during the day, but can be higher for double headers, etc. For example, if a team has a double header one day, the first game of the day will return 1 while the second game will return 2.

games_behind
Returns a `float` of the number of games behind the leader the team is. 0.0 indicates the team is tied for first. Negative numbers indicate the number of games a team is ahead of the second place team.

innings
Returns an `int` of the total number of innings that were played.

location
Returns a `string` constant to indicate whether the game was played at home or away.

loser
Returns a `string` of the name of the losing pitcher.

opponent_abbr
Returns a `string` of the opponent's 3-letter abbreviation, such as 'NYY' for the New York Yankees.

rank
Returns an `int` of the team's rank in the league with 1 being the best team.

record
Returns a `string` of the team's record in the format 'W-L'.

result
Returns a `string` constant to indicate whether the team won or lost.

runs_allowed
Returns an `int` of the total number of runs that the team allowed.

runs_scored
Returns an `int` of the total number of runs that were scored by the team.

save
Returns a `string` of the name of the pitcher credited with the save if applicable. If no saves, returns None.

streak
Returns a `string` of the team's winning/losing streak at the conclusion of the requested game. A winning streak is denoted by a number of '+' signs for the number of consecutive wins and a losing streak is denoted by a '-' sign.

winner
Returns a `string` of the name of the winning pitcher.

class sportsreference.mlb.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'HOU' for the Houston Astros.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.mlb.teams module

The Teams module exposes information for all MLB teams including the team name and abbreviation, the number of games they won during the season, the total number of bases they've stolen, and much more.

```
from sportsreference.mlb.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)  # Prints the team's name
    print(team.batting_average)  # Prints the team's season batting average
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.mlb.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.mlb.teams.**Team**(*team_data*, *rank*, *year=None*)
Bases: `object`

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as rank, name, and abbreviation, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **rank** (*int*) – A team's position in the league based on the number of points they obtained during the season.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a `string` of the team's abbreviation, such as 'HOU' for the Houston Astros.

at_bats
Returns an `int` of the total number of at bats for the team.

average_batter_age
Returns a `float` of the average batter age weighted by their number of at bats plus the number of games participated in.

average_pitcher_age
Returns a `float` of the average pitcher age weighted by the number of games started, followed by the number of games played and saves.

away_losses
Returns an `int` of the number of away losses during the season.

away_record
Returns a `string` of the team's away record. Record is in the format 'W-L'.

away_wins
Returns an `int` of the number of away wins during the season.

balks
Returns an `int` of the total number of times a pitcher has balked.

bases_on_balls
Returns an `int` of the number of bases on walks.

bases_on_walks_given
Returns an `int` of the total number of bases from walks given up by a team during the season.

bases_on_walks_given_per_nine_innings
Returns a `float` of the average number of walks conceded per nine innings.

batters_faced
Returns an `int` of the total number of batters all pitchers have faced during a season.

batting_average
Returns a `float` of the batting average for the team. Percentage ranges from 0-1.

complete_game_shutouts
Returns an `int` of the total number of complete games where the opponent scored zero runs.

complete_games
Returns an `int` of the total number of complete games a team has accumulated during the season.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'HOU'.

doubles
Returns an `int` of the total number of doubles hit by the team.

earned_runs_against
Returns a `float` of the average number of earned runs against for a team.

earned_runs_against_plus
Returns an `int` of the team's average earned runs against, adjusted for the home ballpark.

extra_inning_losses
Returns an `int` of the number of losses the team has when the game has gone to extra innings.

extra_inning_record
Returns a `string` of the team's record when the game has gone to extra innings. Record is in the format 'W-L'.

extra_inning_wins
Returns an `int` of the number of wins the team has when the game has gone to extra innings.

fielding_independent_pitching
Returns a `float` of the team's effectiveness at preventing home runs, walks, batters being hit by pitches, and strikeouts.

games
Returns an `int` of the number of games the team has played during the season.

games_finished
Returns an `int` of the number of games finished which is equivalent to the number of games played minus the number of complete games during the season.

grounded_into_double_plays
Returns an `int` of the total number double plays grounded into by the team.

hit_pitcher
Returns an `int` of the total number of times a pitcher has hit an opposing batter.

hits
Returns an `int` of the total number of hits during the season.

hits_allowed
Returns an `int` of the total number of hits allowed during the season.

hits_per_nine_innings
Returns a `float` of the average number of hits per nine innings by the opponent.

home_losses
Returns an `int` of the number of losses at home during the season.

home_record
Returns a `string` of the team's home record. Record is in the format 'W-L'.

home_runs
Returns an `int` of the total number of home runs hit by the team.

home_runs_against
Returns an `int` of the total number of home runs given up during the season.

home_runs_per_nine_innings
Returns a `float` of the average number of home runs per nine innings by the opponent.

home_wins
Returns an `int` of the number of wins at home during the season.

innings_pitched
Returns a `float` of the total number of innings pitched by a team during the season.

intentional_bases_on_balls
Returns an `int` of the total number of times a player took a base from an intentional walk.

interleague_record
Returns a `string` of the team's interleague record. Record is in the format 'W-L'.

last_ten_games_record
Returns a `string` of the team's record over the last ten games. Record is in the format 'W-L'.

last_thirty_games_record
Returns a `string` of the team's record over the last thirty games. Record is in the format 'W-L'.

last_twenty_games_record
Returns a `string` of the team's record over the last twenty games. Record is in the format 'W-L'.

league
Returns a `string` of the two letter abbreviation of the league, such as 'AL' for the American League.

losses
Returns an `int` of the total number of games the team lost during the season.

losses_last_ten_games
Returns an `int` of the number of losses in the last 10 games.

losses_last_thirty_games
Returns an `int` of the number of losses in the last 30 games.

losses_last_twenty_games
Returns an `int` of the number of losses in the last 20 games.

losses_vs_left_handed_pitchers
Returns an `int` of number of losses against left-handed pitchers.

losses_vs_right_handed_pitchers
Returns an `int` of the number of losses against right-handed pitchers.

losses_vs_teams_over_500
Returns an `int` of the number of losses against teams over 500.

losses_vs_teams_under_500
Returns an `int` of the number of losses against teams under 500.

luck
Returns an `int` of the difference between the current wins and losses compared to the pythagorean wins and losses.

name
Returns a `string` of the team's full name, such as 'Houston Astros'.

number_of_pitchers
Returns an `int` of the total number of pitchers used during a season.

number_players_used
Returns an `int` of the number of different players used during the season.

on_base_percentage
Returns a `float` of the percentage of at bats that result in a player taking a base. Percentage ranges from 0-1.

on_base_plus_slugging_percentage
Returns a `float` of the sum of the on base percentage plus the slugging percentage.

on_base_plus_slugging_percentage_plus
Returns an `int` of the on base percentage plus the slugging percentage, adjusted to the team's home ballpark.

opposing_runners_left_on_base
Returns an `int` of the total number of opponents a team has left on bases at the end of an inning.

plate_appearances
Returns an `int` of the total number of plate appearances for the team.

pythagorean_win_loss
Returns a `string` of the team's expected win-loss record based on the runs scored and allowed. Record is in the format 'W-L'.

rank
Returns an `int` of the team's rank based on their win percentage.

record_vs_left_handed_pitchers
Returns a `string` of the team's record against left-handed pitchers. Record is in the format 'W-L'.

record_vs_right_handed_pitchers
Returns a `string` of the team's record against right-handed pitchers. Record is in the format 'W-L'.

record_vs_teams_over_500
Returns a `string` of the team's record against teams with a win percentage over 500. Record is in the format 'W-L'.

record_vs_teams_under_500
Returns a `string` of the team's record against teams with a win percentage under 500. Record is in the format 'W-L'.

run_difference
Returns a `float` of the difference between the number of runs scored and the number of runs given up per game. Positive numbers indicate the team scores more per game than they are scored on.

runners_left_on_base
Returns an `int` of the total number of runners left on base at the end of an inning.

runs
Returns a `float` of the average number of runs scored per game by the team.

runs_against
Returns a `float` of the average number of runs scored per game by the opponent.

runs_allowed_per_game
Returns a `float` of the average number of runs a team has allowed per game.

runs_batted_in
Returns an `int` of the total number of runs batted in by the team.

sacrifice_flies
Returns an `int` of the total number of sacrifice flies the team made during the season.

sacrifice_hits
Returns an `int` of the total number of sacrifice hits the team made during the season.

saves
Returns an `int` of the total number of saves a team has accumulated during the season.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

shutouts
Returns an `int` of the total number of shutouts a team has accumulated during the season.

simple_rating_system
Returns a `float` of the average number of runs per game a team scores compared to average.

single_run_losses
Returns an `int` of the number of losses the team has when only one run is scored.

single_run_record
Returns a `string` of the team's record when only one run is scored. Record is in the format 'W-L'.

single_run_wins
Returns an `int` of the number of wins the team has when only one run is scored.

slugging_percentage
Returns a `float` of the ratio of total bases gained per at bat.

stolen_bases
Returns an `int` of the total number of bases stolen by the team.

streak
Returns a `string` of the team's current winning or losing streak, such as 'W 3' for a team on a 3-game winning streak.

strength_of_schedule
Returns a `float` denoting a team's strength of schedule, based on runs scores and conceded. Higher values result in more challenging schedules while 0.0 is an average schedule.

strikeouts
Returns an `int` of the total number of times a team has struck out an opponent.

strikeouts_per_base_on_balls
Returns a `float` of the average number of strikeouts per walk thrown by a team.

strikeouts_per_nine_innings
Returns a `float` of the average number of strikeouts a team throws per nine innings.

times_caught_stealing
Returns an `int` of the number of times a player was caught stealing.

times_hit_by_pitch
Returns an `int` of the total number of times a batter was hit by an opponent's pitch.

times_struck_out
Returns an `int` of the total number of times the team struck out.

total_bases
Returns an `int` of the total number of bases a team has gained during the season.

total_runs
Returns an `int` of the total number of runs scored during the season.

triples
Returns an `int` of the total number of tripes hit by the team.

whip
Returns a `float` of the average number of walks plus hits by the opponent per inning.

wild_pitches
Returns an `int` of the total number of wild pitches thrown by a team during a season.

win_percentage
Returns a `float` of the number of wins divided by the number of games played during the season. Percentage ranges from 0-1.

wins
Returns an `int` of the total number of games the team won during the season.

wins_last_ten_games
Returns an `int` of the number of wins in the last 10 games.

wins_last_thirty_games
Returns an `int` of the number of wins in the last 30 games.

wins_last_twenty_games
Returns an `int` of the number of wins in the last 20 games.

wins_vs_left_handed_pitchers
Returns an `int` of number of wins against left-handed pitchers.

wins_vs_right_handed_pitchers
Returns an `int` of the number of wins against right-handed pitchers.

wins_vs_teams_over_500
Returns an `int` of the number of wins against teams over 500.

wins_vs_teams_under_500
Returns an `int` of the number of wins against teams under 500.

class sportsreference.mlb.teams.**Teams**(*year=None*)
Bases: `object`

A list of all MLB teams and their stats in a given year.

Finds and retrieves a list of all MLB teams from www.baseball-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

sportsreference.mlb.teams.**mlb_int_property_decorator**(*func*)

sportsreference.nba package

The NBA package offers multiple modules which can be use to retrieve information and statistics for the National Basketball Association, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.nba.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of points scored to the number of free throws made, to the assist rate and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.nba.boxscore import Boxscore

game_data = Boxscore('201806080CLE')
print(game_data.away_points)  # Prints 108
print(game_data.home_points)  # Prints 85
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.nba.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.nba.boxscore.**Boxscore**(*uri*)
Bases: `object`

Detailed information about the final statistics for a game.

Stores all relevant metrics for a game such as the date, time, location, result, and more advanced metrics such as the effective field goal rate, the true shooting percentage, the game's pace, and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as '201710310LAL'.

away_assist_percentage
Returns a `float` of the percentage of the away team's field goals that were assisted. Percentage ranges from 0-100.

away_assists
Returns an `int` of the total number of assists by the away team.

away_block_percentage
Returns a `float` of the percentage of 2-point field goals that were blocked by the away team. Percentage ranges from 0-100.

away_blocks
Returns an `int` of the total number of blocks by the away team.

away_defensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

away_defensive_rebound_percentage
Returns a `float` of the percentage of available defensive rebounds the away team grabbed. Percentage ranges from 0-100.

away_defensive_rebounds
Returns an `int` of the total number of defensive rebounds by the away team.

away_effective_field_goal_percentage
Returns a `float` of the away team's field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

away_field_goal_attempts
Returns an `int` of the total number of field goal attempts by the away team.

away_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the away team. Percentage ranges from 0-1.

away_field_goals
Returns an `int` of the total number of field goals made by the away team.

away_free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt by the away team.

away_free_throw_attempts
Returns an `int` of the total number of free throw attempts by the away team.

away_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the away team.

away_free_throws
Returns an `int` of the total number of free throws made by the away team.

away_losses
Returns an `int` of the number of games the team has lost after the conclusion of the game.

away_minutes_played
Returns an `int` of the total number of minutes the team played during the game.

away_offensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

away_offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds the away team grabbed. Percentage ranges from 0-100.

away_offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the away team.

away_personal_fouls
Returns an `int` of the total number of personal fouls by the away team.

away_points
Returns an `int` of the number of points the away team scored.

away_steal_percentage
Returns a `float` of the percentage of possessions that ended in a steal by the away team. Percentage ranges from 0-100.

away_steals
Returns an `int` of the total number of steals by the away team.

away_three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range by the away team. Percentage ranges from 0-1.

away_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the away team.

away_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the away team. Percentage ranges from 0-1.

away_three_point_field_goals
Returns an `int` of the total number of three point field goals made by the away team.

away_total_rebound_percentage
Returns a `float` of the percentage of available rebounds the away team grabbed. Percentage ranges from 0-100.

away_total_rebounds
Returns an `int` of the total number of rebounds by the away team.

away_true_shooting_percentage
Returns a `float` of the away team's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

away_turnover_percentage
Returns a `float` of the number of times the away team turned the ball over per 100 possessions.

away_turnovers
Returns an `int` of the total number of turnovers by the away team.

away_two_point_field_goal_attempts
Returns an `int` of the total number of two point field goal attempts by the away team.

away_two_point_field_goal_percentage
Returns a `float` of the number of two point field goals made divided by the number of two point field goal attempts by the away team. Percentage ranges from 0-1.

away_two_point_field_goals
Returns an `int` of the total number of two point field goals made by the away team.

away_wins
Returns an `int` of the number of games the team has won after the conclusion of the game.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as '201710310LAL'.

date
Returns a `string` of the date the game took place.

home_assist_percentage
Returns a `float` of the percentage of the home team's field goals that were assisted. Percentage ranges from 0-100.

home_assists
Returns an `int` of the total number of assists by the home team.

home_block_percentage
Returns a `float` of the percentage of 2-point field goals that were blocked by the home team. Percentage ranges from 0-100.

home_blocks
Returns an `int` of the total number of blocks by the home team.

home_defensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

home_defensive_rebound_percentage
Returns a `float` of the percentage of available defensive rebounds the home team grabbed. Percentage ranges from 0-100.

home_defensive_rebounds
Returns an `int` of the total number of defensive rebounds by the home team.

home_effective_field_goal_percentage
Returns a `float` of the home team's field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

home_field_goal_attempts
Returns an `int` of the total number of field goal attempts by the home team.

home_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the home team. Percentage ranges from 0-1.

home_field_goals
Returns an `int` of the total number of field goals made by the home team.

home_free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt by the home team.

home_free_throw_attempts
Returns an `int` of the total number of free throw attempts by the home team.

home_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the home team.

home_free_throws
Returns an `int` of the total number of free throws made by the home team.

home_losses
Returns an `int` of the number of games the home team lost after the conclusion of the game.

home_minutes_played
Returns an `int` of the total number of minutes the team played during the game.

home_offensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the home team.

home_offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds the home team grabbed. Percentage ranges from 0-100.

home_offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the home team.

home_personal_fouls
Returns an `int` of the total number of personal fouls by the home team.

home_points
Returns an `int` of the number of points the home team scored.

home_steal_percentage
Returns a `float` of the percentage of possessions that ended in a steal by the home team. Percentage ranges from 0-100.

home_steals
Returns an `int` of the total number of steals by the home team.

home_three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range by the home team. Percentage ranges from 0-1.

home_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the home team.

home_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the home team. Percentage ranges from 0-1.

home_three_point_field_goals
Returns an `int` of the total number of three point field goals made by the home team.

home_total_rebound_percentage
Returns a `float` of the percentage of available rebounds the home team grabbed. Percentage ranges from 0-100.

home_total_rebounds
Returns an `int` of the total number of rebounds by the home team.

home_true_shooting_percentage
Returns a `float` of the home team's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

home_turnover_percentage
Returns a `float` of the number of times the home team turned the ball over per 100 possessions.

home_turnovers
Returns an `int` of the total number of turnovers by the home team.

home_two_point_field_goal_attempts
Returns an `int` of the total number of two point field goal attempts by the home team.

home_two_point_field_goal_percentage
Returns a `float` of the number of two point field goals made divided by the number of two point field goal attempts by the home team. Percentage ranges from 0-1.

home_two_point_field_goals
Returns an `int` of the total number of two point field goals made by the home team.

home_wins
Returns an `int` of the number of games the home team won after the conclusion of the game.

location
Returns a `string` of the name of the venue where the game was played.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'PHO' for the Phoenix Suns.

losing_name
Returns a `string` of the losing team's name, such as 'Phoenix Suns'.

pace
Returns a `float` of the game's overall pace, measured by the number of possessions per 40 minutes.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'DET' for the Detroit Pistons.

winning_name
Returns a `string` of the winning team's name, such as 'Detroit Pistons'.

class sportsreference.nba.boxscore.**Boxscores**(*date*)
Bases: `object`

Search for NBA games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters **date** (*datetime object*) – The date to search for any matches. The month, day, and year are required for the search, but time is not factored into the search.

games
Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{'boxscores' : [
    {'home_name': Name of the home team, such as 'Phoenix Suns'
                  (`str`),
     'home_abbr': Abbreviation for the home team, such as
                  'PHO' (`str`),
     'away_name': Name of the away team, such as 'Houston
                  Rockets' (`str`),
     'away_abbr': Abbreviation for the away team, such as
                  'HOU' (`str`),
     'boxscore': String representing the boxscore URI, such as
                 '201702040PHO' (`str`)},
    { ... },
    ...
    ]
}
```

If no games were played during the requested day, the list for ['boxscores'] will be empty.

sportsreference.nba.schedule module

The Schedule module can be used to iterate over all games in a team's schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.nba.schedule import Schedule

houston_schedule = Schedule('HOU')
for game in houston_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class sportsreference.nba.schedule.**Game**(*game_data*)

Bases: object

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team's schedule including date, time, opponent, and result.

Parameters **game_data** (*string*) – The row containing the specified game information.

assists
Returns an `int` of the total number of assists by the team.

blocks
Returns an `int` of the total number of blocks by the team.

boxscore
Returns an instance of the Boxscore class containing more detailed stats on the game.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended
Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date
Returns a `string` of the date the game took place at, such as 'Wed, Oct 18, 2017'.

datetime
Returns a datetime object to indicate the month, day, and year the game took place.

field_goal_attempts
Returns an `int` of the total number of field goal attempts by the team.

field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the team. Percentage ranges from 0-1.

field_goals
Returns an `int` of the total number of field goals made by the team.

free_throw_attempts
Returns an `int` of the total number of free throw attempts by the team.

free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the team.

free_throws
Returns an `int` of the total number of free throws made by the team.

game
Returns an `int` to indicate which game in the season was requested. The first game of the season returns 1.

location
Returns a `string` constant to indicate whether the game was played in the team's home arena or on the road.

offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the team.

opp_assists
Returns an `int` of the total number of assists by the opponent.

opp_blocks
Returns an `int` of the total number of blocks by the opponent.

opp_field_goal_attempts
Returns an `int` of the total number of field goal attempts by the opponent.

opp_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the opponent. Percentage ranges from 0-1.

opp_field_goals
Returns an `int` of the total number of field goals made by the opponent.

opp_free_throw_attempts
Returns an `int` of the total number of free throw attempts by the opponent.

opp_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the opponent.

opp_free_throws
Returns an `int` of the total number of free throws made by the opponent.

opp_offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the opponent.

opp_personal_fouls
Returns an `int` of the total number of personal fouls by the opponent.

opp_steals
Returns an `int` of the total number of steals by the opponent.

opp_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the opponent.

opp_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the opponent. Percentage ranges from 0-1.

opp_three_point_field_goals
Returns an `int` of the total number of three point field goals made by the opponent.

opp_total_rebounds
Returns an `int` of the total number of rebounds by the opponent.

opp_turnovers
Returns an `int` of the total number of turnovers by the opponent.

opponent_abbr
Returns a `string` of the opponent's 3-letter abbreviation, such as 'CHI' for the Chicago Bulls.

personal_fouls
Returns an `int` of the total number of personal fouls by the team.

points_allowed
Returns an `int` of the number of points the team allowed during the game.

points_scored
Returns an `int` of the number of points the team scored during the game.

result
Returns a `string` constant to indicate whether the team won or lost the game.

steals
Returns an `int` of the total number of steals by the team.

three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the team.

three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the team. Percentage ranges from 0-1.

three_point_field_goals
Returns an `int` of the total number of three point field goals made by the team.

total_rebounds
Returns an `int` of the total number of rebounds by the team.

turnovers
Returns an `int` of the total number of turnovers by the team.

class sportsreference.nba.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'PHO' for the Phoenix Suns.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.nba.teams module

The Teams module exposes information for all NBA teams including the team name and abbreviation, the number of games they won during the season, the total number of shots they've blocked, and much more.

```
from sportsreference.nba.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)   # Prints the team's name
    print(team.blocks)   # Prints the team's total blocked shots
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.nba.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.nba.teams.**Team**(*team_data*, *rank*, *year=None*)

Bases: `object`

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as rank, name, and abbreviation, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **rank** (*int*) – A team's position in the league based on the number of points they obtained during the season.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a `string` of the team's abbreviation, such as 'DET' for the Detroit Pistons.

assists
Returns an `int` of the total number of field goals that were assisted.

blocks
Returns an `int` of the total number of times the team blocked an opponent's shot.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'DET'.

defensive_rebounds
Returns an `int` of the total number of defensive rebounds the team has grabbed.

field_goal_attempts
Returns an `int` of the total number of field goals the team has attempted during the season.

field_goal_percentage
Returns a `float` of the percentage of field goals made divided by the number of attempts. Percentage ranges from 0-1.

field_goals
Returns an `int` of the total number of field goals the team has made during the season.

free_throw_attempts
Returns an `int` of the total number of free throw attempts during the season.

free_throw_percentage
Returns a `float` of the percentage of free throws made divided by the attempts. Percentage ranges from 0-1.

free_throws
Returns an `int` of the total number of free throws made during the season.

games_played
Returns an `int` of the total number of games the team has played during the season.

minutes_played
Returns an `int` of the total number of minutes played by all players on the team during the season.

name
Returns a `string` of the team's full name, such as 'Detroit Pistons'.

offensive_rebounds
Returns an `int` of the total number of offensive rebounds the team has grabbed.

opp_assists
Returns an `int` of the total number of field goals that were assisted by the opponent.

opp_blocks
Returns an `int` of the total number of times the opponent blocked the team's shot.

opp_defensive_rebounds
Returns an `int` of the total number of defensive rebounds the opponent grabbed.

opp_field_goal_attempts
Returns an `int` of the total number of field goals the opponents attempted during the season.

opp_field_goal_percentage
Returns a `float` of the percentage of field goals made divided by the number of attempts by the opponent. Percentage ranges from 0-1.

opp_field_goals
Returns an `int` of the total number of field goals the opponents made during the season.

opp_free_throw_attempts
Returns an `int` of the total number of free throw attempts during the season by the opponent.

opp_free_throw_percentage
Returns a `float` of the percentage of free throws made divided by the attempts by the opponent. Percentage ranges from 0-1.

opp_free_throws
Returns an `int` of the total number of free throws made during the season by the opponent.

opp_offensive_rebounds
Returns an `int` of the total number of offensive rebounds the opponent grabbed.

opp_personal_fouls
Returns an `int` of the total number of times the opponent fouled the team.

opp_points
Returns an `int` of the total number of points the team has been scored on during the season.

opp_steals
Returns an `int` of the total number of times the opponent stole the ball from the team.

opp_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goals the opponent attempted during the season.

opp_three_point_field_goal_percentage
Returns a `float` of the percentage of three point field goals made divided by the number of attempts by the opponent. Percentage ranges from 0-1.

opp_three_point_field_goals
Returns an `int` of the total number of three point field goals the opponent made during the season.

opp_total_rebounds
Returns an `int` of the total number of rebounds the opponent grabbed.

opp_turnovers
Returns an `int` of the total number of times the opponent turned the ball over.

opp_two_point_field_goal_attempts
Returns an `int` of the total number of two point field goals the opponent attempted during the season.

opp_two_point_field_goal_percentage
Returns a `float` of the percentage of two point field goals made divided by the number of attempts by the opponent. Percentage ranges from 0-1.

opp_two_point_field_goals
Returns an `int` of the total number of two point field goals the opponent made during the season.

personal_fouls
Returns an `int` of the total number of times the team has fouled an opponent.

points
Returns an `int` of the total number of points the team has scored during the season.

rank
Returns an `int` of the team's rank based on the number of points they score per game.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

steals
Returns an `int` of the total number of times the team stole the ball from the opponent.

three_point_field_goal_attempts
Returns an `int` of the total number of three point field goals the team has attempted during the season.

three_point_field_goal_percentage
Returns a `float` of the percentage of three point field goals made divided by the number of attempts. Percentage ranges from 0-1.

three_point_field_goals
Returns an `int` of the total number of three point field goals the team has made during the season.

total_rebounds
Returns an `int` of the total number of rebounds the team has grabbed.

turnovers
Returns an `int` of the total number of times the team has turned the ball over.

two_point_field_goal_attempts
Returns an `int` of the total number of two point field goals the team has attempted during the season.

two_point_field_goal_percentage
Returns a `float` of the percentage of two point field goals made divided by the number of attempts. Percentage ranges from 0-1.

two_point_field_goals
Returns an `int` of the total number of two point field goals the team has made during the season.

class sportsreference.nba.teams.**Teams**(*year=None*)
Bases: `object`

A list of all NBA teams and their stats in a given year.

Finds and retrieves a list of all NBA teams from www.basketball-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

sportsreference.ncaab package

The NCAAB package offers multiple modules which can be used to retrieve information and statistics for Men's Division I College Basketball, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.ncaab.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of points scored to the number of blocked shots, to the assist percentage and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.ncaab.boxscore import Boxscore

game_data = Boxscore('2018-04-02-21-villanova')
print(game_data.home_points)  # Prints 79
print(game_data.away_points)  # Prints 62
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.ncaab.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.ncaab.boxscore.**Boxscore**(*uri*)
Bases: `object`

Detailed information about the final statistics for a game.

Stores all relevant metrics for a game such as the date, time, location, result, and more advanced metrics such as the effective field goal rate, the true shooting percentage, the game's pace, and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as '2017-11-10-21-kansas'.

away_assist_percentage
Returns a `float` of the percentage of the away team's field goals that were assisted. Percentage ranges from 0-100.

away_assists
Returns an `int` of the total number of assists by the away team.

away_block_percentage
Returns a `float` of the percentage of 2-point field goals that were blocked by the away team. Percentage ranges from 0-100.

away_blocks
Returns an `int` of the total number of blocks by the away team.

away_defensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

away_defensive_rebound_percentage
Returns a `float` of the percentage of available defensive rebounds the away team grabbed. Percentage ranges from 0-100.

away_defensive_rebounds
Returns an `int` of the total number of defensive rebounds by the away team.

away_effective_field_goal_percentage
Returns a `float` of the away team's field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

away_field_goal_attempts
Returns an `int` of the total number of field goal attempts by the away team.

away_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the away team. Percentage ranges from 0-1.

away_field_goals
Returns an `int` of the total number of field goals made by the away team.

away_free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt by the away team.

away_free_throw_attempts
Returns an `int` of the total number of free throw attempts by the away team.

away_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the away team.

away_free_throws
Returns an `int` of the total number of free throws made by the away team.

away_losses
Returns an `int` of the number of games the team has lost after the conclusion of the game.

away_minutes_played
Returns an `int` of the total number of minutes the team played during the game.

away_offensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

away_offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds the away team grabbed. Percentage ranges from 0-100.

away_offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the away team.

away_personal_fouls
Returns an `int` of the total number of personal fouls by the away team.

away_points
Returns an `int` of the number of points the away team scored.

away_steal_percentage
Returns a `float` of the percentage of possessions that ended in a steal by the away team. Percentage ranges from 0-100.

away_steals
Returns an `int` of the total number of steals by the away team.

away_three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range by the away team. Percentage ranges from 0-1.

away_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the away team.

away_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the away team. Percentage ranges from 0-1.

away_three_point_field_goals
Returns an `int` of the total number of three point field goals made by the away team.

away_total_rebound_percentage
Returns a `float` of the percentage of available rebounds the away team grabbed. Percentage ranges from 0-100.

away_total_rebounds
Returns an `int` of the total number of rebounds by the away team.

away_true_shooting_percentage
Returns a `float` of the away team's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

away_turnover_percentage
Returns a `float` of the number of times the away team turned the ball over per 100 possessions.

away_turnovers
Returns an `int` of the total number of turnovers by the away team.

away_two_point_field_goal_attempts
Returns an `int` of the total number of two point field goal attempts by the away team.

away_two_point_field_goal_percentage
Returns a `float` of the number of two point field goals made divided by the number of two point field goal attempts by the away team. Percentage ranges from 0-1.

away_two_point_field_goals
Returns an `int` of the total number of two point field goals made by the away team.

away_wins
Returns an `int` of the number of games the team has won after the conclusion of the game.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as '2017-11-10-21-kansas'.

date
Returns a `string` of the date the game took place.

home_assist_percentage
Returns a `float` of the percentage of the home team's field goals that were assisted. Percentage ranges from 0-100.

home_assists
Returns an `int` of the total number of assists by the home team.

home_block_percentage
Returns a `float` of the percentage of 2-point field goals that were blocked by the home team. Percentage ranges from 0-100.

home_blocks
Returns an `int` of the total number of blocks by the home team.

home_defensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the away team.

home_defensive_rebound_percentage
Returns a `float` of the percentage of available defensive rebounds the home team grabbed. Percentage ranges from 0-100.

home_defensive_rebounds
Returns an `int` of the total number of defensive rebounds by the home team.

home_effective_field_goal_percentage
Returns a `float` of the home team's field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

home_field_goal_attempts
Returns an `int` of the total number of field goal attempts by the home team.

home_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by the home team. Percentage ranges from 0-1.

home_field_goals
Returns an `int` of the total number of field goals made by the home team.

home_free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt by the home team.

home_free_throw_attempts
Returns an `int` of the total number of free throw attempts by the home team.

home_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts by the home team.

home_free_throws
Returns an `int` of the total number of free throws made by the home team.

home_losses
Returns an `int` of the number of games the home team lost after the conclusion of the game.

home_minutes_played
Returns an `int` of the total number of minutes the team played during the game.

home_offensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the home team.

home_offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds the home team grabbed. Percentage ranges from 0-100.

home_offensive_rebounds
Returns an `int` of the total number of offensive rebounds by the home team.

home_personal_fouls
Returns an `int` of the total number of personal fouls by the home team.

home_points
Returns an `int` of the number of points the home team scored.

home_steal_percentage
Returns a `float` of the percentage of possessions that ended in a steal by the home team. Percentage ranges from 0-100.

home_steals
Returns an `int` of the total number of steals by the home team.

home_three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range by the home team. Percentage ranges from 0-1.

home_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts by the home team.

home_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by the home team. Percentage ranges from 0-1.

home_three_point_field_goals
Returns an `int` of the total number of three point field goals made by the home team.

home_total_rebound_percentage
Returns a `float` of the percentage of available rebounds the home team grabbed. Percentage ranges from 0-100.

home_total_rebounds
Returns an `int` of the total number of rebounds by the home team.

home_true_shooting_percentage
Returns a `float` of the home team's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

home_turnover_percentage
Returns a `float` of the number of times the home team turned the ball over per 100 possessions.

home_turnovers
Returns an `int` of the total number of turnovers by the home team.

home_two_point_field_goal_attempts
Returns an `int` of the total number of two point field goal attempts by the home team.

home_two_point_field_goal_percentage
Returns a `float` of the number of two point field goals made divided by the number of two point field goal attempts by the home team. Percentage ranges from 0-1.

home_two_point_field_goals
Returns an `int` of the total number of two point field goals made by the home team.

home_wins
Returns an `int` of the number of games the home team won after the conclusion of the game.

location
Returns a `string` of the name of the venue where the game was played.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'INDIANA' for the Indiana Hoosiers.

losing_name
Returns a `string` of the losing team's name, such as 'Indiana' Hoosiers'.

pace
Returns a `float` of the game's overall pace, measured by the number of possessions per 40 minutes.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'PURDUE' for the Purdue Boilermakers.

winning_name
Returns a `string` of the winning team's name, such as 'Purdue Boilermakers'.

class sportsreference.ncaab.boxscore.**Boxscores**(*date*)
Bases: `object`

Search for NCAAB games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, a boolean value which indicates if the game is between two Division-I teams or not, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters **date** (*datetime object*) – The date to search for any matches. The month, day, and year are required for the search, but time is not factored into the search.

games
Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{'boxscores' : [
    {'home_name': Name of the home team, such as 'Purdue
                  Boilermakers' (`str`),
     'home_abbr': Abbreviation for the home team, such as
                  'PURDUE' (`str`),
     'away_name': Name of the away team, such as 'Indiana
                  Hoosiers' (`str`),
     'away_abbr': Abbreviation for the away team, such as
                  'INDIANA' (`str`),
     'boxscore': String representing the boxscore URI, such as
                 '2018-01-28-15-indiana' (`str`),
     'non_di': Boolean value which evaluates to True when at least
               one of the teams does not compete in NCAA
```

(continues on next page)

(continued from previous page)

```
                Division-I basketball (`str`)},
    { ... },
    ...
    ]
}
```

If no games were played during the requested day, the list for ['boxscores'] will be empty.

sportsreference.ncaab.schedule module

The Schedule module can be used to iterate over all games in a team's schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.ncaab.schedule import Schedule

purdue_schedule = Schedule('PURDUE')
for game in purdue_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class sportsreference.ncaab.schedule.**Game**(*game_data*)

Bases: object

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team's schedule including date, time, opponent, and result.

Parameters **game_data** (*string*) – The row containing the specified game information.

arena

Returns a `string` of the name of the arena the game was played at.

boxscore

Returns an instance of the Boxscore class containing more detailed stats on the game.

dataframe

Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended

Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date

Returns a `string` of the game's date, such as 'Fri, Nov 10, 2017'.

datetime

Returns a datetime object to indicate the month, day, year, and time the requested game took place.

game

Returns an `int` of the game's position in the season. The first game of the season returns 1.

location
Returns a `string` constant to indicate whether the game was played at the team's home venue, the opponent's venue, or at a neutral site.

opponent_abbr
Returns a `string` of the opponent's abbreviation, such as 'PURDUE' for the Purdue Boilermakers.

opponent_conference
Returns a `string` of the opponent's conference, such as 'Big Ten' for a team participating in the Big Ten Conference. If the team is not a Division-I school, a string constant for non-majors is returned.

opponent_name
Returns a `string` of the opponent's name, such as the 'Purdue Boilermakers'.

opponent_rank
Returns a `string` of the opponent's rank when the game was played and None if the team was unranked.

overtimes
Returns an `int` of the number of overtimes that were played during the game and 0 if the game finished at the end of regulation time.

points_against
Returns an `int` of the number of points the team allowed during the game.

points_for
Returns an `int` of the number of points the team scored during the game.

result
Returns a `string` constant to indicate whether the team won or lost the game.

season_losses
Returns an `int` of the number of games the team has lost after the conclusion of the requested game.

season_wins
Returns an `int` of the number of games the team has won after the conclusion of the requested game.

streak
Returns a `string` of the team's win streak at the conclusion of the requested game. Streak is in the format '[W|L] #' (ie. 'W 3' indicates a 3-game winning streak while 'L 2' indicates a 2-game losing streak.

time
Returns a `string` to indicate the time the game started, such as '9:00 pm/est'.

type
Returns a `string` constant to indicate whether the game was played during the regular season or in the post season.

class sportsreference.ncaab.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'PURDUE' for the Purdue Boilermakers.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.ncaab.teams module

The Teams module exposes information for all NCAAB teams including the team name and abbreviation, the number of games they won during the season, the total number of shots they've blocked, and much more.

```
from sportsreference.ncaab.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)   # Prints the team's name
    print(team.blocks)   # Prints the number of shots the team blocked
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.ncaab.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.ncaab.teams.**Team**(*team_data*, *year=None*)
Bases: `object`

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as full and short names, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a `string` of the team's short name, such as 'PURDUE' for the Purdue Boilermakers.

assist_percentage
Returns a `float` of the percentage of field goals that were assisted. Percentage ranges from 0-100.

assists
Returns an `int` of the total number of assists during the season.

away_losses
Returns an `int` of the total number of away games the team lost during the season.

away_wins
Returns an `int` of the total number of away games the team won during the season.

block_percentage
Returns a `float` of the percentage of 2-point field goals by the opponent that were blocked. Percentage ranges from 0-100.

blocks
Returns an `int` of the total number of blocks during the season.

conference_losses
Returns an `int` of the total number of conference games the team lost during the season.

conference_wins
Returns an `int` of the total number of conference games the team won during the season.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'PURDUE'.

effective_field_goal_percentage
Returns a `float` of the field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

field_goal_attempts
Returns an `int` of the total number of field goal attempts during the season.

field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts. Percentage ranges from 0-1.

field_goals
Returns an `int` of the total number of field goals made during the season.

free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt.

free_throw_attempts
Returns an `int` of the total number of free throw attempts during the season.

free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts during the season.

free_throws
Returns an `int` of the total number of free throws made during the season.

free_throws_per_field_goal_attempt
Returns a `float` of the number of free throws per field goal attempt.

games_played
Returns an `int` of the total number of games the team has played during the season.

home_losses
Returns an `int` of the total number of home games the team lost during the season.

home_wins
Returns an `int` of the total number of home games the team won during the season.

losses
Returns an `int` of the total number of games the team lost during the season.

minutes_played
Returns an `int` of the total number of minutes played by the team during the season.

name
Returns a `string` of the team's full name, such as 'Purdue Boilermakers'.

offensive_rating
Returns a `float` of the average number of points scored per 100 possessions.

offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds a team grabbed. Percentage ranges from 0-100.

offensive_rebounds
Returns an `int` of the total number of offensive rebounds during the season.

opp_assist_percentage
Returns a `float` of the percentage of the opponent's field goals that were assisted. Percentage ranges from 0-100.

opp_assists
Returns an `int` of the total number of assists during the season by opponents.

opp_block_percentage
Returns a `float` of the percentage of 2-point field goals that were blocked by the opponent. Percentage ranges from 0-100.

opp_blocks
Returns an `int` of the total number of blocks during the season by opponents.

opp_effective_field_goal_percentage
Returns a `float` of the opponent's field goal percentage while giving extra weight to 3-point field goals. Percentage ranges from 0-1.

opp_field_goal_attempts
Returns an `int` of the total number of field goal attempts during the season by opponents.

opp_field_goal_percentage
Returns a `float` of the number of field goals made divided by the total number of field goal attempts by opponents. Percentage ranges from 0-1.

opp_field_goals
Returns an `int` of the total number of field goals made during the season by opponents.

opp_free_throw_attempt_rate
Returns a `float` of the average number of free throw attempts per field goal attempt by the opponent.

opp_free_throw_attempts
Returns an `int` of the total number of free throw attempts during the season by opponents.

opp_free_throw_percentage
Returns a `float` of the number of free throws made divided by the number of free throw attempts during the season by opponents.

opp_free_throws
Returns an `int` of the total number of free throws made during the season by opponents.

opp_free_throws_per_field_goal_attempt
Returns a `float` of the number of free throws per field goal attempt by the opponent.

opp_offensive_rating
Returns a `float` of the average number of points scored per 100 possessions by the opponent.

opp_offensive_rebound_percentage
Returns a `float` of the percentage of available offensive rebounds the opponent grabbed. Percentage ranges from 0-100.

opp_offensive_rebounds
Returns an `int` of the total number of offensive rebounds during the season by opponents.

opp_personal_fouls
Returns an `int` of the total number of personal fouls during the season by opponents.

opp_points
Returns an `int` of the total number of points opponents have scored during the season.

opp_steal_percentage
Returns a `float` of the percentage of possessions that ended in a steal by the opponent. Percentage ranges from 0-100.

opp_steals
Returns an `int` of the total number of steals during the season by opponents.

opp_three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range by the opponent. Percentage ranges from 0-1.

opp_three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts during the season by opponents.

opp_three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts by opponents. Percentage ranges from 0-1.

opp_three_point_field_goals
Returns an `int` of the total number of three point field goals made during the season by opponents.

opp_total_rebound_percentage
Returns a `float` of the percentage of available rebounds the opponent grabbed. Percentage ranges from 0-100.

opp_total_rebounds
Returns an `int` of the total number of rebounds during the season by opponents.

opp_true_shooting_percentage
Returns a `float` of the opponent's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

opp_turnover_percentage
Returns a `float` of the number of times the opponent turned the ball over per 100 possessions.

opp_turnovers
Returns an `int` of the total number of turnovers during the season by opponents.

pace
Returns a `float` of the average number of possessions per 40 minutes.

personal_fouls
Returns an `int` of the total number of personal fouls during the season.

points
Returns an `int` of the total number of points the team scored during the season.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

simple_rating_system
Returns a `float` of the team's average point differential compared to the strength of schedule. Higher values indicate stronger teams. An average team is denoted with 0.0. Negative numbers are comparatively worse than average.

steal_percentage
Returns a `float` of the percentage of opponent possessions that ended in a steal. Percentage ranges from 0-100.

steals
Returns an `int` of the total number of steals during the season.

strength_of_schedule
Returns a `float` of the team's strength of schedule based on the points above and below average. An average strength of schedule is denoted with 0.0. Negative numbers are comparatively easier than average.

three_point_attempt_rate
Returns a `float` of the percentage of field goal attempts from 3-point range. Percentage ranges from 0-1.

three_point_field_goal_attempts
Returns an `int` of the total number of three point field goal attempts during the season.

three_point_field_goal_percentage
Returns a `float` of the number of three point field goals made divided by the number of three point field goal attempts. Percentage ranges from 0-1.

three_point_field_goals
Returns an `int` of the total number of three point field goals made during the season.

total_rebound_percentage
Returns a `float` of the percentage of available rebounds a team grabbed. Percentage ranges from 0-100.

total_rebounds
Returns an `int` of the total number of rebounds during the season.

true_shooting_percentage
Returns a `float` of the team's true shooting percentage which considers free throws, 2-point field goals, and 3-point field goals. Percentage ranges from 0-1.

turnover_percentage
Returns a `float` of the number of times the team turned the ball over per 100 possessions.

turnovers
Returns an `int` of the total number of turnovers during the season.

win_percentage
Returns a `float` of the number of wins divided by the number of games played during the season. Percentage ranges from 0-1.

wins
Returns an `int` of the total number of games the team won during the season.

class sportsreference.ncaab.teams.**Teams**(*year=None*)
Bases: `object`

A list of all NCAA Men's Basketball teams and their stats in a given year.

Finds and retrieves a list of all NCAA Men's Basketball teams from www.sports-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

sportsreference.ncaaf package

The NCAAF package offers multiple modules which can be used to retrieve information and statistics for Division-I College Football, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.ncaaf.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of points scored to the number of pass yards, to the yards from penalties and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.ncaaf.boxscore import Boxscore

game_data = Boxscore('2018-01-08-georgia')
print(game_data.home_points)  # Prints 23
print(game_data.away_points)  # Prints 26
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.ncaaf.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.ncaaf.boxscore.**Boxscore**(*uri*)

Bases: `object`

Detailed information about the final statistics for a game.

Stores all relevant information for a game such as the date, time, location, result, and more advanced metrics such as the number of fumbles from sacks, a team's passing completion, rushing touchdowns and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as '2018-01-08-georgia'.

away_first_downs
Returns an `int` of the number of first downs the away team gained.

away_fumbles
Returns an `int` of the number of times the away team fumbled the ball.

away_fumbles_lost
Returns an `int` of the number of times the away team turned the ball over as the result of a fumble.

away_interceptions
Returns an `int` of the number of interceptions the away team threw.

away_pass_attempts
Returns an `int` of the number of passes that were thrown by the away team.

away_pass_completions
Returns an `int` of the number of completed passes the away team made.

away_pass_touchdowns
Returns an `int` of the number of passing touchdowns the away team scored.

away_pass_yards
Returns an `int` of the number of passing yards the away team gained.

away_penalties
Returns an `int` of the number of penalties called on the away team.

away_points
Returns an `int` of the number of points the away team scored.

away_rush_attempts
Returns an `int` of the number of rushing plays the away team made.

away_rush_touchdowns
Returns an `int` of the number of rushing touchdowns the away team scored.

away_rush_yards
Returns an `int` of the number of rushing yards the away team gained.

away_total_yards
Returns an `int` of the total number of yards the away team gained.

away_turnovers
Returns an `int` of the number of times the away team turned the ball over.

away_yards_from_penalties
Returns an `int` of the number of yards gifted as a result of penalties called on the away team.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as '2018-01-08-georgia'.

date
Returns a `string` of the date the game took place.

home_first_downs
Returns an `int` of the number of first downs the home team gained.

home_fumbles
Returns an `int` of the number of times the home team fumbled the ball.

home_fumbles_lost
Returns an `int` of the number of times the home team turned the ball over as the result of a fumble.

home_interceptions
Returns an `int` of the number of interceptions the home team threw.

home_pass_attempts
Returns an `int` of the number of passes that were thrown by the home team.

home_pass_completions
Returns an `int` of the number of completed passes the home team made.

home_pass_touchdowns
Returns an `int` of the number of passing touchdowns the home team scored.

home_pass_yards
Returns an `int` of the number of passing yards the home team gained.

home_penalties
Returns an `int` of the number of penalties called on the home team.

home_points
Returns an `int` of the number of points the home team scored.

home_rush_attempts
Returns an `int` of the number of rushing plays the home team made.

home_rush_touchdowns
Returns an `int` of the number of rushing touchdowns the home team scored.

home_rush_yards
Returns an `int` of the number of rushing yards the home team gained.

home_total_yards
Returns an `int` of the total number of yards the home team gained.

home_turnovers
Returns an `int` of the number of times the home team turned the ball over.

home_yards_from_penalties
Returns an `int` of the number of yards gifted as a result of penalties called on the home team.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'GEORGIA' for the Georgia Bulldogs.

losing_name
Returns a `string` of the losing team's name, such as 'Georgia'.

stadium
Returns a `string` of the name of the stadium where the game was played.

time
Returns a `string` of the time the game started.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'ALABAMA' for the Alabama Crimson Tide.

winning_name
Returns a `string` of the winning team's name, such as 'Alabama'.

class sportsreference.ncaaf.boxscore.**Boxscores**(*date*)
Bases: `object`

Search for NCAAF games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, a boolean value which indicates if the game is between two Division-I teams or not, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters **date** (*datetime object*) – The date to search for any matches. The month, day, and year are required for the search, but time is not factored into the search.

games
Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{'boxscores' : [
    {'home_name': Name of the home team, such as 'Purdue
                  Boilermakers' (`str`),
```

(continues on next page)

(continued from previous page)

```
    'home_abbr': Abbreviation for the home team, such as
                 'PURDUE' (`str`),
    'away_name': Name of the away team, such as 'Indiana
                 Hoosiers' (`str`),
    'away_abbr': Abbreviation for the away team, such as
                 'INDIANA' (`str`),
    'boxscore': String representing the boxscore URI, such as
                '2017-09-09-michigan' (`str`),
    'non_di': Boolean value which evaluates to True when at least
              one of the teams does not compete in NCAA
              Division-I football (`str`)},
   { ... },
   ...
   ]
}
```

If no games were played during the requested day, the list for ['boxscores'] will be empty.

sportsreference.ncaaf.boxscore.**ncaaf_int_property_sub_index**(*func*)

sportsreference.ncaaf.schedule module

The Schedule module can be used to iterate over all games in a team's schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.ncaaf.schedule import Schedule

purdue_schedule = Schedule('PURDUE')
for game in purdue_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class sportsreference.ncaaf.schedule.**Game**(*game_data*)

Bases: `object`

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team's schedule including date, time, opponent, and result.

Parameters **game_data** (*string*) – The row containing the specified game information.

boxscore

Returns an instance of the Boxscore class containing more detailed stats on the game.

dataframe

Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended

Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date
Returns a `string` of the date the game was played, such as 'Sep 2, 2017'.

datetime
Returns a datetime object of the month, day, year, and time the game was played. If the game doesn't include a time, the default value of '00:00' will be used.

day_of_week
Returns a `string` of the 3-letter abbreviation of the day of the week the game was played on, such as 'Sat' for Saturday.

game
Returns an `int` to indicate which game in the season was requested. The first game of the season returns 1.

location
Returns a `string` constant to indicate whether the game was played at home, away, or in a neutral location.

losses
Returns an `int` of the number of games the team has lost so far in the season at the conclusion of the requested game.

opponent_abbr
Returns a `string` of the opponent's abbreviation, such as 'PURDUE' for the Purdue Boilermakers.

opponent_conference
Returns a `string` of the conference the team participates in, such as 'Big Ten' for the Big Ten Conference. If a team does not compete in Division-I, a string constant for the non-major school will be returned.

opponent_name
Returns a `string` of the opponent's name, such as 'Purdue Boilermakers' for the Purdue Boilermakers.

opponent_rank
Returns an `int` of the opponent's rank at the time the game was played.

points_against
Returns an `int` of the number of points the team allowed during the game.

points_for
Returns an `int` of the number of points the team scored during the game.

rank
Returns an `int` of the team's rank at the time the game was played.

result
Returns a `string` constant to indicate whether the team won or lost the game.

streak
Returns a `string` of the team's winning streak at the conclusion of the requested game. Streaks are listed in the format '[W|L] #' (ie. 'W 3' for a 3-game winning streak and 'L 2' for a 2-game losing streak).

time
Returns a `string` of the time the game started, such as '12 – 00 PM'.

wins
Returns an `int` of the number of games the team has won so far in the season at the conclusion of the requested game.

class sportsreference.ncaaf.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'MICHIGAN' for the Michigan Wolverines.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.ncaaf.teams module

The Teams module exposes information for all NCAAF teams including the team name and abbreviation, the number of games they won during the season, the total number of pass yards, and much more.

```
from sportsreference.ncaaf.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)  # Prints the team's name
    print(team.pass_yards)  # Prints the team's total passing yards
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.ncaaf.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.ncaaf.teams.**Team**(*team_data*, *year=None*)
Bases: `object`

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as full and short names, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a `string` of the team's short name, such as 'PURDUE' for the Purdue Boilermakers.

conference_losses
Returns an `int` of the total number of conference games the team lost during the season.

conference_win_percentage
Returns a `float` of the percentage of conference wins divided by the number of conference games played during the season. Percentage ranges from 0-1.

conference_wins
Returns an `int` of the total number of conference games the team won during the season.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'PURDUE'.

first_downs
Returns a `float` of the total number of first downs achieved per game.

first_downs_from_penalties
Returns a `float` of the average number of first downs from an opponent's penalties per game.

fumbles_lost
Returns a `float` of the average number of fumbles per game.

games
Returns an `int` of the total number of games the team has played during the season.

interceptions
Returns a `float` of the average number of interceptions thrown per game.

losses
Returns an `int` of the total number of games the team lost during the season.

name
Returns a `string` of the team's full name, such as 'Purdue Boilermakers'.

pass_attempts
Returns a `float` of the average number of passes that are attempted per game.

pass_completion_percentage
Returns a `float` of the percentage of completed passes per game. Percentage ranges from 0-100.

pass_completions
Returns a `float` of the average number of completed passes per game.

pass_first_downs
Returns a `float` of the average number of first downs from passing plays per game.

pass_touchdowns
Returns a `float` of the average number of passing touchdowns scored per game.

pass_yards
Returns a `float` of the average number of yards gained from passing per game.

penalties
Returns a `float` the average number of penalties conceded per game.

plays
Returns a `float` of the average number of offensive plays per game.

points_against_per_game
Returns a `float` of the average number of points conceded per game.

points_per_game
Returns a `float` of the average number of points scored by the team per game.

rush_attempts
Returns a `float` of the average number of rushing plays per game.

rush_first_downs
Returns a `float` of the average number of first downs from rushing plays per game.

rush_touchdowns
Returns a `float` of the average number of rushing touchdowns scored per game.

rush_yards
Returns a `float` of the average number of yards gained from rushing per game.

rush_yards_per_attempt
Returns a `float` of the average number of yards gained per rushing attempt per game.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

simple_rating_system
Returns a `float` of the team's relative strength based on the average margin of victory and the strength of schedule. An average team is denoted with 0.0 while a negative score indicates a comparatively weak team.

strength_of_schedule
Returns a `float` of the team's strength of schedule based on the number of points above or below average. An average difficulty schedule is denoted with 0.0 while a negative score indicates a comparatively easy schedule.

turnovers
Returns a `float` of the average number of turnovers per game.

win_percentage
Returns a `float` of the percentage of wins divided by the number of games played during the season. Percentage ranges from 0-1.

wins
Returns an `int` of the total number of games the team won during the season.

yards
Returns a `float` of the average number of yards gained per game.

yards_from_penalties
Returns a `float` of the average number of yards gained from an opponent's penalties per game.

yards_per_play
Returns a `float` of the average number of yards gained per play.

class sportsreference.ncaaf.teams.**Teams**(*year=None*)
Bases: `object`

A list of all NCAA Men's Football teams and their stats in a given year.

Finds and retrieves a list of all NCAA Men's Football teams from www.sports-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

sportsreference.nfl package

The NFL package offers multiple modules which can be used to retrieve information and statistics for the National Football League, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.nfl.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of points scored to the number of passing yards, to the number of yards lost from sacks and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.nfl.boxscore import Boxscore

game_data = Boxscore('201802040nwe')
print(game_data.home_points)  # Prints 33
print(game_data.away_points)  # Prints 41
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.nfl.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.nfl.boxscore.**Boxscore**(*uri*)

Bases: `object`

Detailed information about the final statistics for a game.

Stores all relevant information for a game such as the date, time, location, result, and more advanced metrics such as the number of yards from sacks, a team's passing completion, rushing touchdowns and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as '201802040nwe'.

attendance
Returns an `int` of the game's listed attendance.

away_first_downs
Returns an `int` of the number of first downs the away team gained.

away_fourth_down_attempts
Returns an `int` of the number of fourth down plays the away team attempted to convert.

away_fourth_down_conversions
Returns an `int` of the number of fourth down plays the away team successfully converted.

away_fumbles
Returns an `int` of the number of times the away team fumbled the ball.

away_fumbles_lost
Returns an `int` of the number of times the away team turned the ball over as the result of a fumble.

away_interceptions
Returns an `int` of the number of interceptions the away team threw.

away_net_pass_yards
Returns an `int` of the net pass yards gained by the away team.

away_pass_attempts
Returns an `int` of the number of passes that were thrown by the away team.

away_pass_completions
Returns an `int` of the number of completed passes the away team made.

away_pass_touchdowns
Returns an `int` of the number of passing touchdowns the away team scored.

away_pass_yards
Returns an `int` of the number of passing yards the away team gained.

away_penalties
Returns an `int` of the number of penalties called on the away team.

away_points
Returns an `int` of the number of points the away team scored.

away_rush_attempts
Returns an `int` of the number of rushing plays the away team made.

away_rush_touchdowns
Returns an `int` of the number of rushing touchdowns the away team scored.

away_rush_yards
Returns an `int` of the number of rushing yards the away team gained.

away_third_down_attempts
Returns an `int` of the number of third down plays the away team attempted to convert.

away_third_down_conversions
Returns an `int` of the number of third down plays the away team successfully converted.

away_time_of_possession
Returns a `string` of the amount of time the home team had possession of the football in the format 'MM:SS'.

away_times_sacked
Returns an `int` of the number of times the away team was sacked.

away_total_yards
Returns an `int` of the total number of yards the away team gained.

away_turnovers
Returns an `int` of the number of times the away team turned the ball over.

away_yards_from_penalties
Returns an `int` of the number of yards gifted as a result of penalties called on the away team.

away_yards_lost_from_sacks
Returns an `int` of the number of yards the away team lost as the result of a sack.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as '201802040nwe'.

date
Returns a `string` of the date the game took place.

duration
Returns a `string` of the game's duration in the format 'H – MM'.

home_first_downs
Returns an `int` of the number of first downs the home team gained.

home_fourth_down_attempts
Returns an `int` of the number of fourth down plays the home team attempted to convert.

home_fourth_down_conversions
Returns an `int` of the number of fourth down plays the home team successfully converted.

home_fumbles
Returns an `int` of the number of times the home team fumbled the ball.

home_fumbles_lost
Returns an `int` of the number of times the home team turned the ball over as the result of a fumble.

home_interceptions
Returns an `int` of the number of interceptions the home team threw.

home_net_pass_yards
Returns an `int` of the net pass yards gained by the home team.

home_pass_attempts
Returns an `int` of the number of passes that were thrown by the home team.

home_pass_completions
Returns an `int` of the number of completed passes the home team made.

home_pass_touchdowns
Returns an `int` of the number of passing touchdowns the home team scored.

home_pass_yards
Returns an `int` of the number of passing yards the home team gained.

home_penalties
Returns an `int` of the number of penalties called on the home team.

home_points
Returns an `int` of the number of points the home team scored.

home_rush_attempts
Returns an `int` of the number of rushing plays the home team made.

home_rush_touchdowns
Returns an `int` of the number of rushing touchdowns the home team scored.

home_rush_yards
Returns an `int` of the number of rushing yards the home team gained.

home_third_down_attempts
Returns an `int` of the number of third down plays the home team attempted to convert.

home_third_down_conversions
Returns an `int` of the number of third down plays the home team successfully converted.

home_time_of_possession
Returns a `string` of the amount of time the home team had possession of the football in the format 'MM:SS'.

home_times_sacked
Returns an `int` of the number of times the home team was sacked.

home_total_yards
Returns an `int` of the total number of yards the home team gained.

home_turnovers
Returns an `int` of the number of times the home team turned the ball over.

home_yards_from_penalties
Returns an `int` of the number of yards gifted as a result of penalties called on the home team.

home_yards_lost_from_sacks
Returns an `int` of the number of yards the home team lost as the result of a sack.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'KAN' for the Kansas City Chiefs.

losing_name
Returns a `string` of the losing team's name, such as 'Kansas City Chiefs'.

stadium
Returns a `string` of the name of the stadium where the game was played.

time
Returns a `string` of the time the game started.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'NWE' for the New England Patriots.

winning_name
Returns a `string` of the winning team's name, such as 'New England Patriots'.

class sportsreference.nfl.boxscore.**Boxscores**(*week*, *year*)
Bases: `object`

Search for NFL games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters

- **week** (*int*) – The week number to pull games from.
- **year** (*int*) – The 4-digit year to pull games from.

games
Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{'boxscores' : [
    {'home_name': Name of the home team, such as 'Kansas City
                  Chiefs' (`str`),
     'home_abbr': Abbreviation for the home team, such as
                  'KAN' (`str`),
     'away_name': Name of the away team, such as 'Houston
                  Texans' (`str`),
     'away_abbr': Abbreviation for the away team, such as
                  'HOU' (`str`),
     'boxscore': String representing the boxscore URI, such as
                 'SLN/SLN201807280' (`str`)},
```

(continues on next page)

(continued from previous page)

```
    { ... },
    ...
    ]
}
```

If no games were played during the requested day, the list for ['boxscores'] will be empty.

sportsreference.nfl.boxscore.**nfl_int_property_sub_index**(*func*)

sportsreference.nfl.schedule module

The Schedule module can be used to iterate over all games in a team's schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.nfl.schedule import Schedule

houston_schedule = Schedule('HOU')
for game in houston_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class sportsreference.nfl.schedule.**Game**(*game_data*, *game_type*, *year*)

Bases: `object`

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team's schedule including date, time, opponent, and result.

Parameters

- **game_data** (*string*) – The row containing the specified game's information.
- **game_type** (*string*) – A constant to denote whether a game took place in the regular season or in the playoffs.
- **year** (*string*) – The year as a 4-digit string. Note that this is the year that the bulk of the season took place. For example the Super Bowl for the 2017 season took place in early Feburary 2018, but 2017 should be passed as that was the year the bulk of the season was played in.

boxscore

Returns an instance of the Boxscore class containing more detailed stats on the game.

dataframe

Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended

Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date

Returns a `string` of the month and day the game was played, such as 'September 7'.

datetime
Returns a datetime object representing the date the game was played.

day
Returns a `string` of the day of the week the game was played as a 3-letter abbreviation, such as 'Sun' for Sunday.

extra_points_attempted
Returns an `int` of the number of times the team attempted to convert an extra point after scoring a touchdown.

extra_points_made
Returns an `int` of the number of extra points the team successfully converted after scoring a touchdown.

field_goals_attempted
Returns an `int` of the total number of times the team attempted a field goal.

field_goals_made
Returns an `int` of the total number of field goals the team scored.

fourth_down_attempts
Returns an `int` of the total number of fourth downs the team attempted to convert.

fourth_down_conversions
Returns an `int` of the number of fourth downs the team successfully converted.

interceptions
Returns an `int` of the number of interceptions the team threw.

location
Returns a `string` constant indicating whether the game was played at home, away, or a neutral site, such as the Super Bowl.

opponent_abbr
Returns a `string` of the opponent's 3-letter abbreviation, such as 'NWE' for the New England Patriots.

opponent_name
Returns a `string` of the opponent's full name, such as the 'New England Patriots'.

overtime
Returns a `boolean` value that evaluates to True if the game when to overtime and False if it ended in regulation.

pass_attempts
Returns an `int` of the number of passes the team attempted during the game.

pass_completion_rate
Returns a `float` of the percentage of passes that were completed by the team. Percentage ranges from 0-100.

pass_completions
Returns an `int` of the number of completed passed by the team.

pass_touchdowns
Returns an `int` of the number of touchdowns the team scored as a result of passing plays.

pass_yards
Returns an `int` of the number of yards the team gained as a result of passing plays.

pass_yards_per_attempt
Returns a `float` of the average number of yards gained per passing play.

points_allowed
Returns an `int` of the number of points allowed by the team.

points_scored
Returns an `int` of the number of points scored by the team.

punt_yards
Returns an `int` of the total number of yards the team punted the ball.

punts
Returns an `int` of the number of times the team punted the ball.

quarterback_rating
Returns a `float` of the quarterback's rating for the game.

result
Returns a `string` constant indicating whether the team won or lost the game.

rush_attempts
Returns an `int` of the total number of times the team attempted a rushing play.

rush_touchdowns
Returns an `int` of the number of touchdowns the team scored as a result of rushing plays.

rush_yards
Returns an `int` of the total number of yards the team gain as a result of rushing plays.

rush_yards_per_attempt
Returns a `float` of the average number of yards gained per rushing play.

third_down_attempts
Returns an `int` of the total number of third downs the team attempted to convert.

third_down_conversions
Returns an `int` of the number of third downs the team successfully converted.

time_of_possession
Returns a `string` of the total time the team spent with the ball. Time is in the format 'MM:SS'.

times_sacked
Returns an `int` of the number of times the quarterback was sacked by the opponent.

type
Returns a `string` constant indicating whether the game is a regular season or playoff matchup.

week
Returns an `int` of the week number in the season, such as 1 for the first week of the regular season.

yards_lost_from_sacks
Returns an `int` of the total number of yards lost as a result of a sack.

class sportsreference.nfl.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'NWE' for the New England Patriots.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.nfl.teams module

The Teams module exposes information for all MLB teams including the team name and abbreviation, the number of games they won during the season, the average margin of victory, and much more.

```
from sportsreference.nfl.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)  # Prints the team's name
    # Prints the team's average margin of victory
    print(team.margin_of_victory)
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.nfl.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.nfl.teams.**Team**(*team_data*, *rank*, *year=None*)
Bases: object

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as rank, name, and abbreviation, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **rank** (*int*) – A team's position in the league based on the number of points they obtained during the season.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a string of team's abbreviation, such as 'KAN' for the Kansas City Chiefs.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'KAN'.

defensive_simple_rating_system
Returns a `float` of the team's defensive strength according to the simple rating system. An average team is denoted with 0.0 and a negative score is a comparatively weaker team.

first_downs
Returns an `int` of the total number of first downs the team achieved during the season.

first_downs_from_penalties
Returns an `int` of the total number of first downs conceded as a result of penalties called on the team.

fumbles
Returns an `int` of the total number of times the team fumbled the ball during the season.

games_played
Returns an `int` of the number of games played during the season.

interceptions
Returns an `int` of the total number of interceptions the team has thrown.

losses
Returns an `int` of the number of games the team lost during the season.

margin_of_victory
Returns a `float` of the average margin of victory per game.

name
Returns a `string` of the team's full name, such as 'Kansas City Chiefs'.

offensive_simple_rating_system
Returns a `float` of the team's offensive strength according to the simple rating system. An average team is denoted with 0.0 and a negative score is a comparatively weaker team.

pass_attempts
Returns an `int` of the total number of passes that were attempted.

pass_completions
Returns an `int` of the total number of passes that were completed.

pass_first_downs
Returns an `int` of the number of first downs the team gained from passing plays.

pass_net_yards_per_attempt
Returns a `float` of the net yards gained per passing play including sacks.

pass_touchdowns
Returns an `int` of the total number of touchdowns the team has scored from passing.

pass_yards
Returns an `int` of the total number of yards the team gained from passing.

penalties
Returns an `int` of the total number of penalties called on the team during the season.

percent_drives_with_points
Returns a `float` of the percentage of drives that result in points for the offense. Percentage ranges from 0-100.

percent_drives_with_turnovers
Returns a `float` of the percentage of drives that result in an offensive turnover. Percentage ranges from 0-100.

plays
Returns an `int` of the total number of offensive plays the team has made during the season.

points_against
Returns an `int` of the total number of points allowed during the season.

points_contributed_by_offense
Returns a `float` of the number of expected points contributed by the offense.

points_difference
Returns an `int` of the difference between the number of points scored and allowed during the season.

points_for
Returns an `int` of the total number of points scored during the season.

rank
Returns an `int` of the team's rank based on the number of points they scored during the season.

rush_attempts
Returns an `int` of the total number of rushing plays that were attempted.

rush_first_downs
Returns an `int` of the total number of first downs gained from rushing plays.

rush_touchdowns
Returns an `int` of the total number of touchdowns from rushing plays.

rush_yards
Returns an `int` of the total number of yards that were gained from rushing plays.

rush_yards_per_attempt
Returns a `float` of the average number of yards gained per rushing play.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

simple_rating_system
Returns a `float` of the team's relative strength based on average margin of victory plus strength of schedule. An average team is denoted with 0.0 and a negative score is a comparatively weaker team.

strength_of_schedule
Returns a `float` of the team's strength of schedule. An average difficulty schedule is denoted with a 0.0 and a negative number is comparatively easier than average.

turnovers
Returns an `int` of the total number of turnovers the team committed during the season.

win_percentage
Returns a `float` of the number of wins divided by the number of games played. Percentage ranges from 0-1.

wins
Returns an `int` of the number of games the team won during the season.

yards
Returns an `int` of the total number of yards the team has gained during the season.

yards_from_penalties
Returns an `int` of the total number of yards surrendered as a result of penalties called on the team.

yards_per_play
Returns a `float` of the average number of yards gained per play during the season.

class sportsreference.nfl.teams.**Teams**(*year=None*)
Bases: `object`

A list of all NFL teams and their stats in a given year.

Finds and retrieves a list of all NFL teams from www.pro-football-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

sportsreference.nhl package

The NHL package offers multiple modules which can be used to retrieve information and statistics for the National Hockey League, such as team names, season stats, game schedules, and boxscore metrics.

sportsreference.nhl.boxscore module

The Boxscore module can be used to grab information from a specific game. Metrics range from number of goals scored to the number of penalty minutes, to the save percentage and much more. The Boxscore can be easily queried by passing a boxscore's URI on sports-reference.com which can be retrieved from the `Schedule` class (see `Schedule` module below for more information on retrieving game-specific information).

```
from sportsreference.nhl.boxscore import Boxscore

game_data = Boxscore('201806070VEG')
print(game_data.home_goals)  # Prints 3
print(game_data.away_goals)  # Prints 4
df = game_data.dataframe  # Returns a Pandas DataFrame of game metrics
```

The Boxscore module also contains a `Boxscores` class which searches for all games played on a particular day and returns a dictionary of matchups between all teams on the requested day. The dictionary includes the names and abbreviations for each matchup as well as the boxscore link if applicable.

```
from datetime import datetime
from sportsreference.nhl.boxscore import Boxscores

games_today = Boxscores(datetime.today())
print(games_today.games)  # Prints a dictionary of all matchups for today
```

class sportsreference.nhl.boxscore.**Boxscore**(*uri*)
Bases: `object`

Detailed information about the final statistics for a game.

Stores all relevant information for a game such as the date, time, location, result, and more advanced metrics such as the number of goals scored, the number of points for a player, the amount of power play assists and much more.

Parameters **uri** (*string*) – The relative link to the boxscore HTML page, such as '201806070VEG'.

arena
Returns a `string` of the name of the ballpark where the game was played.

attendance
Returns an `int` of the game's listed attendance.

away_assists
Returns an `int` of the number of assists the away team registered.

away_even_strength_assists
Returns an `int` of the number of assists the away team registered while at even strength.

away_even_strength_goals
Returns an `int` of the number of goals the away team scored at even strength.

away_game_winning_goals
Returns an `int` of the number of game winning goals the away team scored.

away_goals
Returns an `int` of the number of goals the away team scored.

away_penalties_in_minutes
Returns an `int` of the length of time the away team spent in the penalty box.

away_points
Returns an `int` of the number of points the away team registered.

away_power_play_assists
Returns an `int` of the number of assists the away team registered while on a power play.

away_power_play_goals
Returns an `int` of the number of goals the away team scored while on a power play.

away_save_percentage
Returns a `float` of the percentage of shots the away team saved. Percentage ranges from 0-1.

away_saves
Returns an `int` of the number of saves the away team made.

away_shooting_percentage
Returns a `float` of the away team's shooting percentage. Percentage ranges from 0-100.

away_short_handed_assists
Returns an `int` of the number of assists the away team registered while short handed.

away_short_handed_goals
Returns an `int` of the number of goals the away team scored while short handed.

away_shots_on_goal
Returns an `int` of the number of shots on goal the away team registered.

away_shutout
Returns an `int` denoting whether or not the away team shutout the home team.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string URI that is used to instantiate the class, such as '201806070VEG'.

date
Returns a `string` of the date the game took place.

duration
Returns a `string` of the game's duration in the format 'H – MM'.

home_assists
Returns an `int` of the number of assists the home team registered.

home_even_strength_assists
Returns an `int` of the number of assists the home team registered while at even strength.

home_even_strength_goals
Returns an `int` of the number of goals the home team scored at even strength.

home_game_winning_goals
Returns an `int` of the number of game winning goals the home team scored.

home_goals
Returns an `int` of the number of goals the home team scored.

home_penalties_in_minutes
Returns an `int` of the length of time the home team spent in the penalty box.

home_points
Returns an `int` of the number of points the home team registered.

home_power_play_assists
Returns an `int` of the number of assists the home team registered while on a power play.

home_power_play_goals
Returns an `int` of the number of goals the home team scored while on a power play.

home_save_percentage
Returns a `float` of the percentage of shots the home team saved. Percentage ranges from 0-1.

home_saves
Returns an `int` of the number of saves the home team made.

home_shooting_percentage
Returns a `float` of the home team's shooting percentage. Percentage ranges from 0-100.

home_short_handed_assists
Returns an `int` of the number of assists the home team registered while short handed.

home_short_handed_goals
Returns an `int` of the number of goals the home team scored while short handed.

home_shots_on_goal
Returns an `int` of the number of shots on goal the home team registered.

home_shutout
Returns an `int` denoting whether or not the home team shutout the home team.

losing_abbr
Returns a `string` of the losing team's abbreviation, such as 'WSH' for the Washington Capitals.

losing_name
Returns a `string` of the losing team's name, such as 'Washington Capitals'.

time
Returns a `string` of the time the game started.

winner
Returns a `string` constant indicating whether the home or away team won.

winning_abbr
Returns a `string` of the winning team's abbreviation, such as 'VEG' for the Vegas Golden Knights.

winning_name
Returns a `string` of the winning team's name, such as 'Vegas Golden Knights'.

class sportsreference.nhl.boxscore.**Boxscores**(*date*)
Bases: `object`

Search for NHL games taking place on a particular day.

Retrieve a dictionary which contains a list of all games being played on a particular day. Output includes a link to the boxscore, and the names and abbreviations for both the home teams. If no games are played on a particular day, the list will be empty.

Parameters **date** (*datetime object*) – The date to search for any matches. The month, day, and year are required for the search, but time is not factored into the search.

games

Returns a `dictionary` object representing all of the games played on the requested day. Dictionary is in the following format:

```
{'boxscores' : [
    {'home_name': Name of the home team, such as 'New York
                  Rangers' (`str`),
     'home_abbr': Abbreviation for the home team, such as
                  'NYR' (`str`),
     'away_name': Name of the away team, such as 'Boston
                  Bruins' (`str`),
     'away_abbr': Abbreviation for the away team, such as
                  'BOS' (`str`),
     'boxscore': String representing the boxscore URI, such as
                 '201702040VAN' (`str`)},
    { ... },
    ...
    ]
}
```

If no games were played during the requested day, the list for [‘boxscores’] will be empty.

`sportsreference.nhl.boxscore.`**nhl_int_property_decorator**(*func*)

sportsreference.nhl.schedule module

The Schedule module can be used to iterate over all games in a team’s schedule to get game information such as the date, score, result, and more. Each game also has a link to the `Boxscore` class which has much more detailed information on the game metrics.

```
from sportsreference.nhl.schedule import Schedule

detroit_schedule = Schedule('DET')
for game in detroit_schedule:
    print(game.date)  # Prints the date the game was played
    print(game.result)  # Prints whether the team won or lost
    # Creates an instance of the Boxscore class for the game.
    boxscore = game.boxscore
```

class `sportsreference.nhl.schedule.`**Game**(*game_data*, *year*)

Bases: `object`

A representation of a matchup between two teams.

Stores all relevant high-level match information for a game in a team’s schedule including date, time, opponent, and result.

Parameters

- **game_data** (*string*) – The row containing the specified game information.
- **year** (*string*) – The year of the current season.

boxscore
Returns an instance of the Boxscore class containing more detailed stats on the game.

corsi_against
Returns an `int` of the Corsi Against at Even Strength metric which equals the number of shots + blocks + misses by the opponent.

corsi_for
Returns an `int` of the Corsi For at Even Strength metric which equals the number of shots + blocks + misses.

corsi_for_percentage
Returns a `float` of the percentage of control a team had of the puck which is calculated by the corsi_for value divided by the sum of corsi_for and corsi_against. Values greater than 50.0 indicate the team had more control of the puck than their opponent. Percentage ranges from 0-100.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the boxscore string.

dataframe_extended
Returns a pandas DataFrame representing the Boxscore class for the game. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property. The index for the DataFrame is the boxscore string.

date
Returns a `string` of the date the game was played, such as '2017-10-05'.

datetime
Returns a datetime object to indicate the month, day, and year the game was played at.

faceoff_losses
Returns an `int` of the number of faceoffs the team lost at even strength.

faceoff_win_percentage
Returns a `float` of percentage of faceoffs the team won while at even strength. Percentage ranges from 0-100.

faceoff_wins
Returns an `int` of the number of faceoffs the team won at even strength.

fenwick_against
Returns an `int` of the Fenwick Against at Even Strength metric which equals the number of shots + misses by the opponent.

fenwick_for
Returns an `int` of the Fenwick For at Even Strength metric which equals the number of shots + misses.

fenwick_for_percentage
Returns a `float` of the percentage of control a team had of the puck which is calculated by the fenwick_for value divided by the sum of fenwick_for and fenwick_against. Values greater than 50.0 indicate the team had more control of the puck than their opponent. Percentage ranges from 0-100.

game
Returns an `int` to indicate which game in the season was requested. The first game of the season returns 1.

goals_allowed
Returns an `int` of the number of goals the team allowed during the game.

goals_scored
Returns an `int` of the number of goals the team scored during the game.

location
Returns a `string` constant to indicate whether the game was played at home or away.

offensive_zone_start_percentage
Returns a `float` of the percentage of stats that took place in the offensive half. Value is calculated by the number of offensive zone starts divided by the sum of offensive zone starts and defensive zone starts. Percentage ranges from 0-100.

opp_penalties_in_minutes
Returns an `int` of the total number of minutes the opponent served for penalties.

opp_power_play_goals
Returns an `int` of the number of power play goals the opponent scored.

opp_power_play_opportunities
Returns an `int` of the number of power play opportunities the opponent had.

opp_short_handed_goals
Returns an `int` of the number of shorthanded goals the opponent scored.

opp_shots_on_goal
Returns an `int` of the total number of shots on goal the opponent registered.

opponent_abbr
Returns a `string` of the opponent's 3-letter abbreviation, such as 'NYR' for the New York Rangers.

opponent_name
Returns a `string` of the opponent's name, such as 'New York Rangers'.

overtime
Returns an `int` of the number of overtimes that were played during the game, or an int constant if the game went to a shootout.

pdo
Returns a `float` of the team's PDO at Even Strength metric which is calculated by the sum of the shooting percentage and save percentage. Percentage ranges from 0-100.

penalties_in_minutes
Returns an `int` of the total number of minutes the team served for penalties.

power_play_goals
Returns an `int` of the number of power play goals the team scored.

power_play_opportunities
Returns an `int` of the number of power play opportunities the team had.

result
Returns a `string` constant to indicate whether the team lost in regulation, lost in overtime, or won.

short_handed_goals
Returns an `int` of the number of shorthanded goals the team scored.

shots_on_goal
Returns an `int` of the total number of shots on goal the team registered.

class sportsreference.nhl.schedule.**Schedule**(*abbreviation*, *year=None*)
Bases: `object`

An object of the given team's schedule.

Generates a team's schedule for the season including wins, losses, and scores if applicable.

Parameters

- **abbreviation** (*string*) – A team's short name, such as 'NYR' for the New York Rangers.
- **year** (*string (optional)*) – The requested year to pull stats from.

dataframe
Returns a pandas DataFrame where each row is a representation of the Game class. Rows are indexed by the boxscore string.

dataframe_extended
Returns a pandas DataFrame where each row is a representation of the Boxscore class for every game in the schedule. Rows are indexed by the boxscore string. This property provides much richer context for the selected game, but takes longer to process compared to the lighter 'dataframe' property.

sportsreference.nhl.teams module

The Teams module exposes information for all NHL teams including the team name and abbreviation, the number of games they won during the season, the total number of shots on goal, and much more.

```
from sportsreference.nhl.teams import Teams

teams = Teams()
for team in teams:
    print(team.name)  # Prints the team's name
    print(team.shots_on_goal)  # Prints the team's total shots on goal
```

Each Team instance contains a link to the `Schedule` class which enables easy iteration over all games for a particular team. A Pandas DataFrame can also be queried to easily grab all stats for all games.

```
from sportsreference.nhl.teams import Teams

teams = Teams()
for team in teams:
    schedule = team.schedule  # Returns a Schedule instance for each team
    # Returns a Pandas DataFrame of all metrics for all game Boxscores for
    # a season.
    df = team.schedule.dataframe_extended
```

class sportsreference.nhl.teams.**Team**(*team_data*, *rank*, *year=None*)
Bases: `object`

An object containing all of a team's season information.

Finds and parses all team stat information and identifiers, such as rank, name, and abbreviation, and sets them as properties which can be directly read from for easy reference.

Parameters

- **team_data** (*string*) – A string containing all of the rows of stats for a given team. If multiple tables are being referenced, this will be comprised of multiple rows in a single string.
- **rank** (*int*) – A team's position in the league based on the number of points they obtained during the season.
- **year** (*string (optional)*) – The requested year to pull stats from.

abbreviation
Returns a `string` of the team's abbreviation, such as 'DET' for the Detroit Red Wings.

average_age
Returns a `float` of the average age of all players on the team, weighted by their time on ice.

dataframe
Returns a pandas DataFrame containing all other class properties and values. The index for the DataFrame is the string abbreviation of the team, such as 'DET'.

games_played
Returns an `int` of the total number of games the team has played in the season.

goals_against
Returns an `int` of the total number of goals opponents scored against the team during the season.

goals_for
Returns an `int` of the total number of goals a team scored during the season.

losses
Returns an `int` of the total number of losses the team had in the season.

name
Returns a `string` of the team's full name, such as 'Detroit Red Wings'.

overtime_losses
Returns an `int` of the total number of overtime losses the team had in the season.

pdo_at_even_strength
Returns a `float` of the PDO at even strength which equates to the shooting percentage plus the save percentage.

penalty_killing_percentage
Returns a `float` denoting the percentage of power plays that have been successfully defended without a goal being conceded. Percentage ranges from 0-100.

points
Returns an `int` of the total number of points the team gained in the season.

points_percentage
Returns a `float` denoting the percentage of points gained divided by the maximum possible points available during the season. Percentage ranges from 0-1.

power_play_goals
Returns an `int` of the total number of power play goals scored.

power_play_goals_against
Returns an `int` of the total number of power play goals conceded.

power_play_opportunities
Returns an `int` of the total number of power play opportunities for a team during the season.

power_play_opportunities_against
Returns an `int` of the total number of power play opportunities for the opponents during the season.

power_play_percentage
Returns a `float` denoting the percentage of power play opportunities where the team has scored. Percentage ranges from 0-100.

rank
Returns an `int` of the team's rank based on the number of points they obtained in the season.

save_percentage
Returns a `float` denoting the percentage of shots the team has saved during the season. Percentage ranges from 0-1.

schedule
Returns an instance of the Schedule class containing the team's complete schedule for the season.

shooting_percentage
Returns a `float` denoting the percentage of shots to goals during the season. Percentage ranges from 0-100.

short_handed_goals
Returns an `int` of the number of short handed goals the team has scored during the season.

short_handed_goals_against
Returns an `int` of the number of short handed goals the team has conceded during the season.

shots_against
Returns an `int` of the total number of shots on goal the team's opponents made during the season.

shots_on_goal
Returns an `int` of the total number of shots on goal the team made during the season.

simple_rating_system
Returns a `float` which takes into account the average goal differential vs a team's strength of schedule. The league average evaluates to 0.0. Teams which have a positive score are comparatively stronger than average while teams with a negative score are weaker.

strength_of_schedule
Returns a `float` denoting a team's strength of schedule, based on goals scores and conceded. Higher values result in more challenging schedules while 0.0 is an average schedule.

total_goals_per_game
Returns a `float` for the average number of goals scored per game.

wins
Returns an `int` of the total number of wins the team had in the season.

class sportsreference.nhl.teams.**Teams**(*year=None*)
Bases: `object`

A list of all NHL teams and their stats in a given year.

Finds and retrieves a list of all NHL teams from www.hockey-reference.com and creates a Team instance for every team that participated in the league in a given year. The Team class comprises a list of all major stats and a few identifiers for the requested season.

Parameters **year** (*string (optional)*) – The requested year to pull stats from.

dataframes
Returns a pandas DataFrame where each row is a representation of the Team class. Rows are indexed by the team abbreviation.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

sportsreference.mlb.boxscore, 5
sportsreference.mlb.schedule, 10
sportsreference.mlb.teams, 12
sportsreference.nba.boxscore, 18
sportsreference.nba.schedule, 24
sportsreference.nba.teams, 27
sportsreference.ncaab.boxscore, 30
sportsreference.ncaab.schedule, 36
sportsreference.ncaab.teams, 38
sportsreference.ncaaf.boxscore, 43
sportsreference.ncaaf.schedule, 46
sportsreference.ncaaf.teams, 48
sportsreference.nfl.boxscore, 51
sportsreference.nfl.schedule, 55
sportsreference.nfl.teams, 58
sportsreference.nhl.boxscore, 61
sportsreference.nhl.schedule, 64
sportsreference.nhl.teams, 67

Index

A

abbreviation (sportsreference.mlb.teams.Team attribute), 12
abbreviation (sportsreference.nba.teams.Team attribute), 27
abbreviation (sportsreference.ncaab.teams.Team attribute), 38
abbreviation (sportsreference.ncaaf.teams.Team attribute), 48
abbreviation (sportsreference.nfl.teams.Team attribute), 58
abbreviation (sportsreference.nhl.teams.Team attribute), 67
arena (sportsreference.ncaab.schedule.Game attribute), 36
arena (sportsreference.nhl.boxscore.Boxscore attribute), 61
assist_percentage (sportsreference.ncaab.teams.Team attribute), 38
assists (sportsreference.nba.schedule.Game attribute), 24
assists (sportsreference.nba.teams.Team attribute), 27
assists (sportsreference.ncaab.teams.Team attribute), 38
at_bats (sportsreference.mlb.teams.Team attribute), 12
attendance (sportsreference.mlb.boxscore.Boxscore attribute), 5
attendance (sportsreference.mlb.schedule.Game attribute), 10
attendance (sportsreference.nfl.boxscore.Boxscore attribute), 51
attendance (sportsreference.nhl.boxscore.Boxscore attribute), 61
average_age (sportsreference.nhl.teams.Team attribute), 67
average_batter_age (sportsreference.mlb.teams.Team attribute), 13
average_pitcher_age (sportsreference.mlb.teams.Team attribute), 13
away_assist_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_assist_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_assists (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_assists (sportsreference.nba.boxscore.Boxscore attribute), 19
away_assists (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_assists (sportsreference.nhl.boxscore.Boxscore attribute), 61
away_at_bats (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_average_leverage_index (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_base_out_runs_added (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_base_out_runs_saved (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_bases_on_balls (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_batting_average (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_block_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_block_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_blocks (sportsreference.nba.boxscore.Boxscore attribute), 19
away_blocks (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_defensive_rating (sportsreference.nba.boxscore.Boxscore attribute), 19
away_defensive_rating (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_defensive_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_defensive_rebound_percentage (sportsrefer-

ence.ncaab.boxscore.Boxscore attribute), 31
away_defensive_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 19
away_defensive_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_earned_runs (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_effective_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_effective_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_even_strength_assists (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_even_strength_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 19
away_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 19
away_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_first_downs (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_first_downs (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_fly_balls (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_fourth_down_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_fourth_down_conversions (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_free_throw_attempt_rate (sportsreference.nba.boxscore.Boxscore attribute), 19
away_free_throw_attempt_rate (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_free_throw_attempts (sportsreference.nba.boxscore.Boxscore attribute), 19
away_free_throw_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_free_throw_percentage (sportsreference.nba.boxscore.Boxscore attribute), 19
away_free_throw_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_free_throws (sportsreference.nba.boxscore.Boxscore attribute), 19
away_free_throws (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_fumbles (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_fumbles (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_fumbles_lost (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_fumbles_lost (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_game_score (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_game_winning_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_grounded_balls (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_hits (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_home_runs (sportsreference.mlb.boxscore.Boxscore attribute), 5
away_inherited_runners (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_inherited_score (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_innings_pitched (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_interceptions (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_interceptions (sportsreference.nfl.boxscore.Boxscore attribute), 51
away_line_drives (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_losses (sportsreference.mlb.teams.Team attribute), 13
away_losses (sportsreference.nba.boxscore.Boxscore attribute), 19
away_losses (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_losses (sportsreference.ncaab.teams.Team attribute), 38
away_minutes_played (sportsreference.nba.boxscore.Boxscore attribute), 19

away_minutes_played (sportsreference.ncaab.boxscore.Boxscore attribute), 31
away_net_pass_yards (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_offensive_rating (sportsreference.nba.boxscore.Boxscore attribute), 20
away_offensive_rating (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_offensive_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_offensive_rebound_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_offensive_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 20
away_offensive_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_on_base_percentage (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_on_base_plus (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_pass_attempts (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_pass_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_pass_completions (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_pass_completions (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_pass_touchdowns (sportsreference.ncaaf.boxscore.Boxscore attribute), 43
away_pass_touchdowns (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_pass_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_pass_yards (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_penalties (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_penalties (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_penalties_in_minutes (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_personal_fouls (sportsreference.nba.boxscore.Boxscore attribute), 20
away_personal_fouls (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_pitches (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_plate_appearances (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_points (sportsreference.nba.boxscore.Boxscore attribute), 20
away_points (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_points (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_points (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_points (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_power_play_assists (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_power_play_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_putouts (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_rbi (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_record (sportsreference.mlb.teams.Team attribute), 13
away_runs (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_rush_attempts (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_rush_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_rush_touchdowns (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_rush_touchdowns (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_rush_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_rush_yards (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_save_percentage (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_saves (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_shooting_percentage (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_short_handed_assists (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_short_handed_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_shots_on_goal (sportsreference.

ence.nhl.boxscore.Boxscore attribute), 62
away_shutout (sportsreference.nhl.boxscore.Boxscore attribute), 62
away_slugging_percentage (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_steal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_steal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_steals (sportsreference.nba.boxscore.Boxscore attribute), 20
away_steals (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_strikeouts (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_strikes (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_strikes_by_contact (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_strikes_looking (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_strikes_swinging (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_third_down_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_third_down_conversions (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_three_point_attempt_rate (sportsreference.nba.boxscore.Boxscore attribute), 20
away_three_point_attempt_rate (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_three_point_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 20
away_three_point_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_three_point_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_three_point_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_three_point_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 20
away_three_point_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_time_of_possession (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_times_sacked (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_total_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_total_rebound_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_total_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 20
away_total_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_total_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_total_yards (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_true_shooting_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_true_shooting_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_turnover_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_turnover_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_turnovers (sportsreference.nba.boxscore.Boxscore attribute), 20
away_turnovers (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_turnovers (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_turnovers (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_two_point_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 20
away_two_point_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_two_point_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 20
away_two_point_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_two_point_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 20
away_two_point_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 32
away_unknown_bat_type (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_win_probability_added (sportsreference.mlb.boxscore.Boxscore attribute), 6
away_win_probability_by_pitcher (sportsreference.mlb.boxscore.Boxscore attribute), 6

away_win_probability_for_offensive_player (sportsreference.mlb.boxscore.Boxscore attribute), 7
away_win_probability_subtracted (sportsreference.mlb.boxscore.Boxscore attribute), 7
away_wins (sportsreference.mlb.teams.Team attribute), 13
away_wins (sportsreference.nba.boxscore.Boxscore attribute), 21
away_wins (sportsreference.ncaab.boxscore.Boxscore attribute), 33
away_wins (sportsreference.ncaab.teams.Team attribute), 38
away_yards_from_penalties (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
away_yards_from_penalties (sportsreference.nfl.boxscore.Boxscore attribute), 52
away_yards_lost_from_sacks (sportsreference.nfl.boxscore.Boxscore attribute), 52

B

balks (sportsreference.mlb.teams.Team attribute), 13
bases_on_balls (sportsreference.mlb.teams.Team attribute), 13
bases_on_walks_given (sportsreference.mlb.teams.Team attribute), 13
bases_on_walks_given_per_nine_innings (sportsreference.mlb.teams.Team attribute), 13
batters_faced (sportsreference.mlb.teams.Team attribute), 13
batting_average (sportsreference.mlb.teams.Team attribute), 13
block_percentage (sportsreference.ncaab.teams.Team attribute), 38
blocks (sportsreference.nba.schedule.Game attribute), 24
blocks (sportsreference.nba.teams.Team attribute), 27
blocks (sportsreference.ncaab.teams.Team attribute), 39
Boxscore (class in sportsreference.mlb.boxscore), 5
Boxscore (class in sportsreference.nba.boxscore), 18
Boxscore (class in sportsreference.ncaab.boxscore), 30
Boxscore (class in sportsreference.ncaaf.boxscore), 43
Boxscore (class in sportsreference.nfl.boxscore), 51
Boxscore (class in sportsreference.nhl.boxscore), 61
boxscore (sportsreference.mlb.schedule.Game attribute), 10
boxscore (sportsreference.nba.schedule.Game attribute), 24
boxscore (sportsreference.ncaab.schedule.Game attribute), 36
boxscore (sportsreference.ncaaf.schedule.Game attribute), 46
boxscore (sportsreference.nfl.schedule.Game attribute), 55
boxscore (sportsreference.nhl.schedule.Game attribute), 64
Boxscores (class in sportsreference.mlb.boxscore), 9
Boxscores (class in sportsreference.nba.boxscore), 23
Boxscores (class in sportsreference.ncaab.boxscore), 35
Boxscores (class in sportsreference.ncaaf.boxscore), 45
Boxscores (class in sportsreference.nfl.boxscore), 54
Boxscores (class in sportsreference.nhl.boxscore), 63

C

complete_game_shutouts (sportsreference.mlb.teams.Team attribute), 13
complete_games (sportsreference.mlb.teams.Team attribute), 13
conference_losses (sportsreference.ncaab.teams.Team attribute), 39
conference_losses (sportsreference.ncaaf.teams.Team attribute), 48
conference_win_percentage (sportsreference.ncaaf.teams.Team attribute), 49
conference_wins (sportsreference.ncaab.teams.Team attribute), 39
conference_wins (sportsreference.ncaaf.teams.Team attribute), 49
corsi_against (sportsreference.nhl.schedule.Game attribute), 65
corsi_for (sportsreference.nhl.schedule.Game attribute), 65
corsi_for_percentage (sportsreference.nhl.schedule.Game attribute), 65

D

dataframe (sportsreference.mlb.boxscore.Boxscore attribute), 7
dataframe (sportsreference.mlb.schedule.Game attribute), 10
dataframe (sportsreference.mlb.schedule.Schedule attribute), 12
dataframe (sportsreference.mlb.teams.Team attribute), 13
dataframe (sportsreference.nba.boxscore.Boxscore attribute), 21
dataframe (sportsreference.nba.schedule.Game attribute), 24
dataframe (sportsreference.nba.schedule.Schedule attribute), 26
dataframe (sportsreference.nba.teams.Team attribute), 27
dataframe (sportsreference.ncaab.boxscore.Boxscore attribute), 33
dataframe (sportsreference.ncaab.schedule.Game attribute), 36
dataframe (sportsreference.ncaab.schedule.Schedule attribute), 37
dataframe (sportsreference.ncaab.teams.Team attribute), 39

dataframe (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
dataframe (sportsreference.ncaaf.schedule.Game attribute), 46
dataframe (sportsreference.ncaaf.schedule.Schedule attribute), 48
dataframe (sportsreference.ncaaf.teams.Team attribute), 49
dataframe (sportsreference.nfl.boxscore.Boxscore attribute), 52
dataframe (sportsreference.nfl.schedule.Game attribute), 55
dataframe (sportsreference.nfl.schedule.Schedule attribute), 57
dataframe (sportsreference.nfl.teams.Team attribute), 58
dataframe (sportsreference.nhl.boxscore.Boxscore attribute), 62
dataframe (sportsreference.nhl.schedule.Game attribute), 65
dataframe (sportsreference.nhl.schedule.Schedule attribute), 67
dataframe (sportsreference.nhl.teams.Team attribute), 68
dataframe_extended (sportsreference.mlb.schedule.Game attribute), 10
dataframe_extended (sportsreference.mlb.schedule.Schedule attribute), 12
dataframe_extended (sportsreference.nba.schedule.Game attribute), 24
dataframe_extended (sportsreference.nba.schedule.Schedule attribute), 26
dataframe_extended (sportsreference.ncaab.schedule.Game attribute), 36
dataframe_extended (sportsreference.ncaab.schedule.Schedule attribute), 37
dataframe_extended (sportsreference.ncaaf.schedule.Game attribute), 46
dataframe_extended (sportsreference.ncaaf.schedule.Schedule attribute), 48
dataframe_extended (sportsreference.nfl.schedule.Game attribute), 55
dataframe_extended (sportsreference.nfl.schedule.Schedule attribute), 58
dataframe_extended (sportsreference.nhl.schedule.Game attribute), 65
dataframe_extended (sportsreference.nhl.schedule.Schedule attribute), 67
dataframes (sportsreference.mlb.teams.Teams attribute), 18
dataframes (sportsreference.nba.teams.Teams attribute), 30
dataframes (sportsreference.ncaab.teams.Teams attribute), 42
dataframes (sportsreference.ncaaf.teams.Teams attribute), 50
dataframes (sportsreference.nfl.teams.Teams attribute), 61
dataframes (sportsreference.nhl.teams.Teams attribute), 69
date (sportsreference.mlb.boxscore.Boxscore attribute), 7
date (sportsreference.mlb.schedule.Game attribute), 10
date (sportsreference.nba.boxscore.Boxscore attribute), 21
date (sportsreference.nba.schedule.Game attribute), 24
date (sportsreference.ncaab.boxscore.Boxscore attribute), 33
date (sportsreference.ncaab.schedule.Game attribute), 36
date (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
date (sportsreference.ncaaf.schedule.Game attribute), 46
date (sportsreference.nfl.boxscore.Boxscore attribute), 52
date (sportsreference.nfl.schedule.Game attribute), 55
date (sportsreference.nhl.boxscore.Boxscore attribute), 62
date (sportsreference.nhl.schedule.Game attribute), 65
datetime (sportsreference.mlb.schedule.Game attribute), 10
datetime (sportsreference.nba.schedule.Game attribute), 24
datetime (sportsreference.ncaab.schedule.Game attribute), 36
datetime (sportsreference.ncaaf.schedule.Game attribute), 47
datetime (sportsreference.nfl.schedule.Game attribute), 55
datetime (sportsreference.nhl.schedule.Game attribute), 65
day (sportsreference.nfl.schedule.Game attribute), 56
day_of_week (sportsreference.ncaaf.schedule.Game attribute), 47
day_or_night (sportsreference.mlb.schedule.Game attribute), 10
defensive_rebounds (sportsreference.nba.teams.Team attribute), 27
defensive_simple_rating_system (sportsreference.nfl.teams.Team attribute), 58
doubles (sportsreference.mlb.teams.Team attribute), 13
duration (sportsreference.mlb.boxscore.Boxscore attribute), 7
duration (sportsreference.nfl.boxscore.Boxscore attribute), 52
duration (sportsreference.nhl.boxscore.Boxscore attribute), 62

E

earned_runs_against (sportsreference.mlb.teams.Team attribute), 13

earned_runs_against_plus (sportsreference.mlb.teams.Team attribute), 13
effective_field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 39
extra_inning_losses (sportsreference.mlb.teams.Team attribute), 13
extra_inning_record (sportsreference.mlb.teams.Team attribute), 13
extra_inning_wins (sportsreference.mlb.teams.Team attribute), 13
extra_points_attempted (sportsreference.nfl.schedule.Game attribute), 56
extra_points_made (sportsreference.nfl.schedule.Game attribute), 56

F

faceoff_losses (sportsreference.nhl.schedule.Game attribute), 65
faceoff_win_percentage (sportsreference.nhl.schedule.Game attribute), 65
faceoff_wins (sportsreference.nhl.schedule.Game attribute), 65
fenwick_against (sportsreference.nhl.schedule.Game attribute), 65
fenwick_for (sportsreference.nhl.schedule.Game attribute), 65
fenwick_for_percentage (sportsreference.nhl.schedule.Game attribute), 65
field_goal_attempts (sportsreference.nba.schedule.Game attribute), 24
field_goal_attempts (sportsreference.nba.teams.Team attribute), 27
field_goal_attempts (sportsreference.ncaab.teams.Team attribute), 39
field_goal_percentage (sportsreference.nba.schedule.Game attribute), 24
field_goal_percentage (sportsreference.nba.teams.Team attribute), 27
field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 39
field_goals (sportsreference.nba.schedule.Game attribute), 24
field_goals (sportsreference.nba.teams.Team attribute), 28
field_goals (sportsreference.ncaab.teams.Team attribute), 39
field_goals_attempted (sportsreference.nfl.schedule.Game attribute), 56
field_goals_made (sportsreference.nfl.schedule.Game attribute), 56
fielding_independent_pitching (sportsreference.mlb.teams.Team attribute), 14
first_downs (sportsreference.ncaaf.teams.Team attribute), 49
first_downs (sportsreference.nfl.teams.Team attribute), 59
first_downs_from_penalties (sportsreference.ncaaf.teams.Team attribute), 49
first_downs_from_penalties (sportsreference.nfl.teams.Team attribute), 59
fourth_down_attempts (sportsreference.nfl.schedule.Game attribute), 56
fourth_down_conversions (sportsreference.nfl.schedule.Game attribute), 56
free_throw_attempt_rate (sportsreference.ncaab.teams.Team attribute), 39
free_throw_attempts (sportsreference.nba.schedule.Game attribute), 24
free_throw_attempts (sportsreference.nba.teams.Team attribute), 28
free_throw_attempts (sportsreference.ncaab.teams.Team attribute), 39
free_throw_percentage (sportsreference.nba.schedule.Game attribute), 25
free_throw_percentage (sportsreference.nba.teams.Team attribute), 28
free_throw_percentage (sportsreference.ncaab.teams.Team attribute), 39
free_throws (sportsreference.nba.schedule.Game attribute), 25
free_throws (sportsreference.nba.teams.Team attribute), 28
free_throws (sportsreference.ncaab.teams.Team attribute), 39
free_throws_per_field_goal_attempt (sportsreference.ncaab.teams.Team attribute), 39
fumbles (sportsreference.nfl.teams.Team attribute), 59
fumbles_lost (sportsreference.ncaaf.teams.Team attribute), 49

G

Game (class in sportsreference.mlb.schedule), 10
Game (class in sportsreference.nba.schedule), 24
Game (class in sportsreference.ncaab.schedule), 36
Game (class in sportsreference.ncaaf.schedule), 46
Game (class in sportsreference.nfl.schedule), 55
Game (class in sportsreference.nhl.schedule), 64
game (sportsreference.mlb.schedule.Game attribute), 10
game (sportsreference.nba.schedule.Game attribute), 25
game (sportsreference.ncaab.schedule.Game attribute), 36
game (sportsreference.ncaaf.schedule.Game attribute), 47
game (sportsreference.nhl.schedule.Game attribute), 65
game_duration (sportsreference.mlb.schedule.Game attribute), 11
game_number_for_day (sportsreference.mlb.schedule.Game attribute), 11
games (sportsreference.mlb.boxscore.Boxscores attribute), 9

games (sportsreference.mlb.teams.Team attribute), 14
games (sportsreference.nba.boxscore.Boxscores attribute), 23
games (sportsreference.ncaab.boxscore.Boxscores attribute), 35
games (sportsreference.ncaaf.boxscore.Boxscores attribute), 45
games (sportsreference.ncaaf.teams.Team attribute), 49
games (sportsreference.nfl.boxscore.Boxscores attribute), 54
games (sportsreference.nhl.boxscore.Boxscores attribute), 64
games_behind (sportsreference.mlb.schedule.Game attribute), 11
games_finished (sportsreference.mlb.teams.Team attribute), 14
games_played (sportsreference.nba.teams.Team attribute), 28
games_played (sportsreference.ncaab.teams.Team attribute), 39
games_played (sportsreference.nfl.teams.Team attribute), 59
games_played (sportsreference.nhl.teams.Team attribute), 68
goals_against (sportsreference.nhl.teams.Team attribute), 68
goals_allowed (sportsreference.nhl.schedule.Game attribute), 65
goals_for (sportsreference.nhl.teams.Team attribute), 68
goals_scored (sportsreference.nhl.schedule.Game attribute), 65
grounded_into_double_plays (sportsreference.mlb.teams.Team attribute), 14

H

hit_pitcher (sportsreference.mlb.teams.Team attribute), 14
hits (sportsreference.mlb.teams.Team attribute), 14
hits_allowed (sportsreference.mlb.teams.Team attribute), 14
hits_per_nine_innings (sportsreference.mlb.teams.Team attribute), 14
home_assist_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_assist_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_assists (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_assists (sportsreference.nba.boxscore.Boxscore attribute), 21
home_assists (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_assists (sportsreference.nhl.boxscore.Boxscore attribute), 62
home_at_bats (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_average_leverage_index (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_base_out_runs_added (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_base_out_runs_saved (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_bases_on_balls (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_batting_average (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_block_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_block_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_blocks (sportsreference.nba.boxscore.Boxscore attribute), 21
home_blocks (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_defensive_rating (sportsreference.nba.boxscore.Boxscore attribute), 21
home_defensive_rating (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_defensive_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_defensive_rebound_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_defensive_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 21
home_defensive_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_earned_runs (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_effective_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_effective_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_even_strength_assists (sportsreference.nhl.boxscore.Boxscore attribute), 62
home_even_strength_goals (sportsreference.nhl.boxscore.Boxscore attribute), 62
home_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 21
home_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 33

home_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 21
home_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_first_downs (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_first_downs (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_fly_balls (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_fourth_down_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_fourth_down_conversions (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_free_throw_attempt_rate (sportsreference.nba.boxscore.Boxscore attribute), 21
home_free_throw_attempt_rate (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_free_throw_attempts (sportsreference.nba.boxscore.Boxscore attribute), 21
home_free_throw_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_free_throw_percentage (sportsreference.nba.boxscore.Boxscore attribute), 21
home_free_throw_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_free_throws (sportsreference.nba.boxscore.Boxscore attribute), 21
home_free_throws (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_fumbles (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_fumbles (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_fumbles_lost (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_fumbles_lost (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_game_score (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_game_winning_goals (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_goals (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_grounded_balls (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_hits (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_home_runs (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_inherited_runners (sportsreference.mlb.boxscore.Boxscore attribute), 7
home_inherited_score (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_innings_pitched (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_interceptions (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_interceptions (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_line_drives (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_losses (sportsreference.mlb.teams.Team attribute), 14
home_losses (sportsreference.nba.boxscore.Boxscore attribute), 21
home_losses (sportsreference.ncaab.boxscore.Boxscore attribute), 33
home_losses (sportsreference.ncaab.teams.Team attribute), 39
home_minutes_played (sportsreference.nba.boxscore.Boxscore attribute), 22
home_minutes_played (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_net_pass_yards (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_offensive_rating (sportsreference.nba.boxscore.Boxscore attribute), 22
home_offensive_rating (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_offensive_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_offensive_rebound_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_offensive_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 22
home_offensive_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_on_base_percentage (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_on_base_plus (sportsrefer-

ence.mlb.boxscore.Boxscore attribute), 8
home_pass_attempts (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_pass_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_pass_completions (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_pass_completions (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_pass_touchdowns (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_pass_touchdowns (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_pass_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_pass_yards (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_penalties (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_penalties (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_penalties_in_minutes (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_personal_fouls (sportsreference.nba.boxscore.Boxscore attribute), 22
home_personal_fouls (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_pitches (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_plate_appearances (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_points (sportsreference.nba.boxscore.Boxscore attribute), 22
home_points (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_points (sportsreference.ncaaf.boxscore.Boxscore attribute), 44
home_points (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_points (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_power_play_assists (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_power_play_goals (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_putouts (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_rbi (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_record (sportsreference.mlb.teams.Team attribute), 14
home_runs (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_runs (sportsreference.mlb.teams.Team attribute), 14
home_runs_against (sportsreference.mlb.teams.Team attribute), 14
home_runs_per_nine_innings (sportsreference.mlb.teams.Team attribute), 14
home_rush_attempts (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_rush_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_rush_touchdowns (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_rush_touchdowns (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_rush_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_rush_yards (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_save_percentage (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_saves (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_shooting_percentage (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_short_handed_assists (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_short_handed_goals (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_shots_on_goal (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_shutout (sportsreference.nhl.boxscore.Boxscore attribute), 63
home_slugging_percentage (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_steal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_steal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_steals (sportsreference.nba.boxscore.Boxscore attribute), 22
home_steals (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_strikeouts (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_strikes (sportsreference.mlb.boxscore.Boxscore at-

tribute), 8
home_strikes_by_contact (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_strikes_looking (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_strikes_swinging (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_third_down_attempts (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_third_down_conversions (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_three_point_attempt_rate (sportsreference.nba.boxscore.Boxscore attribute), 22
home_three_point_attempt_rate (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_three_point_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 22
home_three_point_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_three_point_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_three_point_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_three_point_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 22
home_three_point_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_time_of_possession (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_times_sacked (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_total_rebound_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_total_rebound_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_total_rebounds (sportsreference.nba.boxscore.Boxscore attribute), 22
home_total_rebounds (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_total_yards (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_total_yards (sportsreference.nfl.boxscore.Boxscore attribute), 53
home_true_shooting_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_true_shooting_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_turnover_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_turnover_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_turnovers (sportsreference.nba.boxscore.Boxscore attribute), 22
home_turnovers (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_turnovers (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_turnovers (sportsreference.nfl.boxscore.Boxscore attribute), 54
home_two_point_field_goal_attempts (sportsreference.nba.boxscore.Boxscore attribute), 22
home_two_point_field_goal_attempts (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_two_point_field_goal_percentage (sportsreference.nba.boxscore.Boxscore attribute), 22
home_two_point_field_goal_percentage (sportsreference.ncaab.boxscore.Boxscore attribute), 34
home_two_point_field_goals (sportsreference.nba.boxscore.Boxscore attribute), 23
home_two_point_field_goals (sportsreference.ncaab.boxscore.Boxscore attribute), 35
home_unknown_bat_type (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_win_probability_added (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_win_probability_by_pitcher (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_win_probability_for_offensive_player (sportsreference.mlb.boxscore.Boxscore attribute), 8
home_win_probability_subtracted (sportsreference.mlb.boxscore.Boxscore attribute), 9
home_wins (sportsreference.mlb.teams.Team attribute), 14
home_wins (sportsreference.nba.boxscore.Boxscore attribute), 23
home_wins (sportsreference.ncaab.boxscore.Boxscore attribute), 35
home_wins (sportsreference.ncaab.teams.Team attribute), 39
home_yards_from_penalties (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
home_yards_from_penalties (sportsreference.nfl.boxscore.Boxscore attribute), 54

home_yards_lost_from_sacks (sportsreference.nfl.boxscore.Boxscore attribute), 54

I

innings (sportsreference.mlb.schedule.Game attribute), 11
innings_pitched (sportsreference.mlb.teams.Team attribute), 14
intentional_bases_on_balls (sportsreference.mlb.teams.Team attribute), 14
interceptions (sportsreference.ncaaf.teams.Team attribute), 49
interceptions (sportsreference.nfl.schedule.Game attribute), 56
interceptions (sportsreference.nfl.teams.Team attribute), 59
interleague_record (sportsreference.mlb.teams.Team attribute), 14

L

last_ten_games_record (sportsreference.mlb.teams.Team attribute), 14
last_thirty_games_record (sportsreference.mlb.teams.Team attribute), 14
last_twenty_games_record (sportsreference.mlb.teams.Team attribute), 14
league (sportsreference.mlb.teams.Team attribute), 14
location (sportsreference.mlb.schedule.Game attribute), 11
location (sportsreference.nba.boxscore.Boxscore attribute), 23
location (sportsreference.nba.schedule.Game attribute), 25
location (sportsreference.ncaab.boxscore.Boxscore attribute), 35
location (sportsreference.ncaab.schedule.Game attribute), 36
location (sportsreference.ncaaf.schedule.Game attribute), 47
location (sportsreference.nfl.schedule.Game attribute), 56
location (sportsreference.nhl.schedule.Game attribute), 65
loser (sportsreference.mlb.schedule.Game attribute), 11
losing_abbr (sportsreference.mlb.boxscore.Boxscore attribute), 9
losing_abbr (sportsreference.nba.boxscore.Boxscore attribute), 23
losing_abbr (sportsreference.ncaab.boxscore.Boxscore attribute), 35
losing_abbr (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
losing_abbr (sportsreference.nfl.boxscore.Boxscore attribute), 54
losing_abbr (sportsreference.nhl.boxscore.Boxscore attribute), 63
losing_name (sportsreference.mlb.boxscore.Boxscore attribute), 9
losing_name (sportsreference.nba.boxscore.Boxscore attribute), 23
losing_name (sportsreference.ncaab.boxscore.Boxscore attribute), 35
losing_name (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
losing_name (sportsreference.nfl.boxscore.Boxscore attribute), 54
losing_name (sportsreference.nhl.boxscore.Boxscore attribute), 63
losses (sportsreference.mlb.teams.Team attribute), 15
losses (sportsreference.ncaab.teams.Team attribute), 39
losses (sportsreference.ncaaf.schedule.Game attribute), 47
losses (sportsreference.ncaaf.teams.Team attribute), 49
losses (sportsreference.nfl.teams.Team attribute), 59
losses (sportsreference.nhl.teams.Team attribute), 68
losses_last_ten_games (sportsreference.mlb.teams.Team attribute), 15
losses_last_thirty_games (sportsreference.mlb.teams.Team attribute), 15
losses_last_twenty_games (sportsreference.mlb.teams.Team attribute), 15
losses_vs_left_handed_pitchers (sportsreference.mlb.teams.Team attribute), 15
losses_vs_right_handed_pitchers (sportsreference.mlb.teams.Team attribute), 15
losses_vs_teams_over_500 (sportsreference.mlb.teams.Team attribute), 15
losses_vs_teams_under_500 (sportsreference.mlb.teams.Team attribute), 15
luck (sportsreference.mlb.teams.Team attribute), 15

M

margin_of_victory (sportsreference.nfl.teams.Team attribute), 59
minutes_played (sportsreference.nba.teams.Team attribute), 28
minutes_played (sportsreference.ncaab.teams.Team attribute), 39
mlb_int_property_decorator() (in module sportsreference.mlb.teams), 18

N

name (sportsreference.mlb.teams.Team attribute), 15
name (sportsreference.nba.teams.Team attribute), 28
name (sportsreference.ncaab.teams.Team attribute), 39
name (sportsreference.ncaaf.teams.Team attribute), 49
name (sportsreference.nfl.teams.Team attribute), 59
name (sportsreference.nhl.teams.Team attribute), 68

ncaaf_int_property_sub_index() (in module sportsreference.ncaaf.boxscore), 46
nfl_int_property_sub_index() (in module sportsreference.nfl.boxscore), 55
nhl_int_property_decorator() (in module sportsreference.nhl.boxscore), 64
number_of_pitchers (sportsreference.mlb.teams.Team attribute), 15
number_players_used (sportsreference.mlb.teams.Team attribute), 15

O

offensive_rating (sportsreference.ncaab.teams.Team attribute), 40
offensive_rebound_percentage (sportsreference.ncaab.teams.Team attribute), 40
offensive_rebounds (sportsreference.nba.schedule.Game attribute), 25
offensive_rebounds (sportsreference.nba.teams.Team attribute), 28
offensive_rebounds (sportsreference.ncaab.teams.Team attribute), 40
offensive_simple_rating_system (sportsreference.nfl.teams.Team attribute), 59
offensive_zone_start_percentage (sportsreference.nhl.schedule.Game attribute), 66
on_base_percentage (sportsreference.mlb.teams.Team attribute), 15
on_base_plus_slugging_percentage (sportsreference.mlb.teams.Team attribute), 15
on_base_plus_slugging_percentage_plus (sportsreference.mlb.teams.Team attribute), 15
opp_assist_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_assists (sportsreference.nba.schedule.Game attribute), 25
opp_assists (sportsreference.nba.teams.Team attribute), 28
opp_assists (sportsreference.ncaab.teams.Team attribute), 40
opp_block_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_blocks (sportsreference.nba.schedule.Game attribute), 25
opp_blocks (sportsreference.nba.teams.Team attribute), 28
opp_blocks (sportsreference.ncaab.teams.Team attribute), 40
opp_defensive_rebounds (sportsreference.nba.teams.Team attribute), 28
opp_effective_field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_field_goal_attempts (sportsreference.nba.schedule.Game attribute), 25
opp_field_goal_attempts (sportsreference.nba.teams.Team attribute), 28
opp_field_goal_attempts (sportsreference.ncaab.teams.Team attribute), 40
opp_field_goal_percentage (sportsreference.nba.schedule.Game attribute), 25
opp_field_goal_percentage (sportsreference.nba.teams.Team attribute), 28
opp_field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_field_goals (sportsreference.nba.schedule.Game attribute), 25
opp_field_goals (sportsreference.nba.teams.Team attribute), 28
opp_field_goals (sportsreference.ncaab.teams.Team attribute), 40
opp_free_throw_attempt_rate (sportsreference.ncaab.teams.Team attribute), 40
opp_free_throw_attempts (sportsreference.nba.schedule.Game attribute), 25
opp_free_throw_attempts (sportsreference.nba.teams.Team attribute), 28
opp_free_throw_attempts (sportsreference.ncaab.teams.Team attribute), 40
opp_free_throw_percentage (sportsreference.nba.schedule.Game attribute), 25
opp_free_throw_percentage (sportsreference.nba.teams.Team attribute), 28
opp_free_throw_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_free_throws (sportsreference.nba.schedule.Game attribute), 25
opp_free_throws (sportsreference.nba.teams.Team attribute), 28
opp_free_throws (sportsreference.ncaab.teams.Team attribute), 40
opp_free_throws_per_field_goal_attempt (sportsreference.ncaab.teams.Team attribute), 40
opp_offensive_rating (sportsreference.ncaab.teams.Team attribute), 40
opp_offensive_rebound_percentage (sportsreference.ncaab.teams.Team attribute), 40
opp_offensive_rebounds (sportsreference.nba.schedule.Game attribute), 25
opp_offensive_rebounds (sportsreference.nba.teams.Team attribute), 28
opp_offensive_rebounds (sportsreference.ncaab.teams.Team attribute), 40
opp_penalties_in_minutes (sportsreference.nhl.schedule.Game attribute), 66
opp_personal_fouls (sportsreference.nba.schedule.Game attribute), 25
opp_personal_fouls (sportsreference.nba.teams.Team attribute), 28

opp_personal_fouls (sportsreference.ncaab.teams.Team attribute), 41
opp_points (sportsreference.nba.teams.Team attribute), 28
opp_points (sportsreference.ncaab.teams.Team attribute), 41
opp_power_play_goals (sportsreference.nhl.schedule.Game attribute), 66
opp_power_play_opportunities (sportsreference.nhl.schedule.Game attribute), 66
opp_short_handed_goals (sportsreference.nhl.schedule.Game attribute), 66
opp_shots_on_goal (sportsreference.nhl.schedule.Game attribute), 66
opp_steal_percentage (sportsreference.ncaab.teams.Team attribute), 41
opp_steals (sportsreference.nba.schedule.Game attribute), 25
opp_steals (sportsreference.nba.teams.Team attribute), 29
opp_steals (sportsreference.ncaab.teams.Team attribute), 41
opp_three_point_attempt_rate (sportsreference.ncaab.teams.Team attribute), 41
opp_three_point_field_goal_attempts (sportsreference.nba.schedule.Game attribute), 25
opp_three_point_field_goal_attempts (sportsreference.nba.teams.Team attribute), 29
opp_three_point_field_goal_attempts (sportsreference.ncaab.teams.Team attribute), 41
opp_three_point_field_goal_percentage (sportsreference.nba.schedule.Game attribute), 25
opp_three_point_field_goal_percentage (sportsreference.nba.teams.Team attribute), 29
opp_three_point_field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 41
opp_three_point_field_goals (sportsreference.nba.schedule.Game attribute), 25
opp_three_point_field_goals (sportsreference.nba.teams.Team attribute), 29
opp_three_point_field_goals (sportsreference.ncaab.teams.Team attribute), 41
opp_total_rebound_percentage (sportsreference.ncaab.teams.Team attribute), 41
opp_total_rebounds (sportsreference.nba.schedule.Game attribute), 26
opp_total_rebounds (sportsreference.nba.teams.Team attribute), 29
opp_total_rebounds (sportsreference.ncaab.teams.Team attribute), 41
opp_true_shooting_percentage (sportsreference.ncaab.teams.Team attribute), 41
opp_turnover_percentage (sportsreference.ncaab.teams.Team attribute), 41
opp_turnovers (sportsreference.nba.schedule.Game attribute), 26
opp_turnovers (sportsreference.nba.teams.Team attribute), 29
opp_turnovers (sportsreference.ncaab.teams.Team attribute), 41
opp_two_point_field_goal_attempts (sportsreference.nba.teams.Team attribute), 29
opp_two_point_field_goal_percentage (sportsreference.nba.teams.Team attribute), 29
opp_two_point_field_goals (sportsreference.nba.teams.Team attribute), 29
opponent_abbr (sportsreference.mlb.schedule.Game attribute), 11
opponent_abbr (sportsreference.nba.schedule.Game attribute), 26
opponent_abbr (sportsreference.ncaab.schedule.Game attribute), 37
opponent_abbr (sportsreference.ncaaf.schedule.Game attribute), 47
opponent_abbr (sportsreference.nfl.schedule.Game attribute), 56
opponent_abbr (sportsreference.nhl.schedule.Game attribute), 66
opponent_conference (sportsreference.ncaab.schedule.Game attribute), 37
opponent_conference (sportsreference.ncaaf.schedule.Game attribute), 47
opponent_name (sportsreference.ncaab.schedule.Game attribute), 37
opponent_name (sportsreference.ncaaf.schedule.Game attribute), 47
opponent_name (sportsreference.nfl.schedule.Game attribute), 56
opponent_name (sportsreference.nhl.schedule.Game attribute), 66
opponent_rank (sportsreference.ncaab.schedule.Game attribute), 37
opponent_rank (sportsreference.ncaaf.schedule.Game attribute), 47
opposing_runners_left_on_base (sportsreference.mlb.teams.Team attribute), 15
overtime (sportsreference.nfl.schedule.Game attribute), 56
overtime (sportsreference.nhl.schedule.Game attribute), 66
overtime_losses (sportsreference.nhl.teams.Team attribute), 68
overtimes (sportsreference.ncaab.schedule.Game attribute), 37

P

pace (sportsreference.nba.boxscore.Boxscore attribute), 23

pace (sportsreference.ncaab.boxscore.Boxscore attribute), 35
pace (sportsreference.ncaab.teams.Team attribute), 41
pass_attempts (sportsreference.ncaaf.teams.Team attribute), 49
pass_attempts (sportsreference.nfl.schedule.Game attribute), 56
pass_attempts (sportsreference.nfl.teams.Team attribute), 59
pass_completion_percentage (sportsreference.ncaaf.teams.Team attribute), 49
pass_completion_rate (sportsreference.nfl.schedule.Game attribute), 56
pass_completions (sportsreference.ncaaf.teams.Team attribute), 49
pass_completions (sportsreference.nfl.schedule.Game attribute), 56
pass_completions (sportsreference.nfl.teams.Team attribute), 59
pass_first_downs (sportsreference.ncaaf.teams.Team attribute), 49
pass_first_downs (sportsreference.nfl.teams.Team attribute), 59
pass_net_yards_per_attempt (sportsreference.nfl.teams.Team attribute), 59
pass_touchdowns (sportsreference.ncaaf.teams.Team attribute), 49
pass_touchdowns (sportsreference.nfl.schedule.Game attribute), 56
pass_touchdowns (sportsreference.nfl.teams.Team attribute), 59
pass_yards (sportsreference.ncaaf.teams.Team attribute), 49
pass_yards (sportsreference.nfl.schedule.Game attribute), 56
pass_yards (sportsreference.nfl.teams.Team attribute), 59
pass_yards_per_attempt (sportsreference.nfl.schedule.Game attribute), 56
pdo (sportsreference.nhl.schedule.Game attribute), 66
pdo_at_even_strength (sportsreference.nhl.teams.Team attribute), 68
penalties (sportsreference.ncaaf.teams.Team attribute), 49
penalties (sportsreference.nfl.teams.Team attribute), 59
penalties_in_minutes (sportsreference.nhl.schedule.Game attribute), 66
penalty_killing_percentage (sportsreference.nhl.teams.Team attribute), 68
percent_drives_with_points (sportsreference.nfl.teams.Team attribute), 59
percent_drives_with_turnovers (sportsreference.nfl.teams.Team attribute), 59
personal_fouls (sportsreference.nba.schedule.Game attribute), 26
personal_fouls (sportsreference.nba.teams.Team attribute), 29
personal_fouls (sportsreference.ncaab.teams.Team attribute), 41
plate_appearances (sportsreference.mlb.teams.Team attribute), 15
plays (sportsreference.ncaaf.teams.Team attribute), 49
plays (sportsreference.nfl.teams.Team attribute), 59
points (sportsreference.nba.teams.Team attribute), 29
points (sportsreference.ncaab.teams.Team attribute), 41
points (sportsreference.nhl.teams.Team attribute), 68
points_against (sportsreference.ncaab.schedule.Game attribute), 37
points_against (sportsreference.ncaaf.schedule.Game attribute), 47
points_against (sportsreference.nfl.teams.Team attribute), 59
points_against_per_game (sportsreference.ncaaf.teams.Team attribute), 49
points_allowed (sportsreference.nba.schedule.Game attribute), 26
points_allowed (sportsreference.nfl.schedule.Game attribute), 56
points_contributed_by_offense (sportsreference.nfl.teams.Team attribute), 60
points_difference (sportsreference.nfl.teams.Team attribute), 60
points_for (sportsreference.ncaab.schedule.Game attribute), 37
points_for (sportsreference.ncaaf.schedule.Game attribute), 47
points_for (sportsreference.nfl.teams.Team attribute), 60
points_per_game (sportsreference.ncaaf.teams.Team attribute), 49
points_percentage (sportsreference.nhl.teams.Team attribute), 68
points_scored (sportsreference.nba.schedule.Game attribute), 26
points_scored (sportsreference.nfl.schedule.Game attribute), 57
power_play_goals (sportsreference.nhl.schedule.Game attribute), 66
power_play_goals (sportsreference.nhl.teams.Team attribute), 68
power_play_goals_against (sportsreference.nhl.teams.Team attribute), 68
power_play_opportunities (sportsreference.nhl.schedule.Game attribute), 66
power_play_opportunities (sportsreference.nhl.teams.Team attribute), 68
power_play_opportunities_against (sportsreference.nhl.teams.Team attribute), 68
power_play_percentage (sportsreference.nhl.teams.Team attribute), 68
punt_yards (sportsreference.nfl.schedule.Game attribute),

57
punts (sportsreference.nfl.schedule.Game attribute), 57
pythagorean_win_loss (sportsreference.mlb.teams.Team attribute), 15

Q

quarterback_rating (sportsreference.nfl.schedule.Game attribute), 57

R

rank (sportsreference.mlb.schedule.Game attribute), 11
rank (sportsreference.mlb.teams.Team attribute), 15
rank (sportsreference.nba.teams.Team attribute), 29
rank (sportsreference.ncaaf.schedule.Game attribute), 47
rank (sportsreference.nfl.teams.Team attribute), 60
rank (sportsreference.nhl.teams.Team attribute), 68
record (sportsreference.mlb.schedule.Game attribute), 11
record_vs_left_handed_pitchers (sportsreference.mlb.teams.Team attribute), 15
record_vs_right_handed_pitchers (sportsreference.mlb.teams.Team attribute), 16
record_vs_teams_over_500 (sportsreference.mlb.teams.Team attribute), 16
record_vs_teams_under_500 (sportsreference.mlb.teams.Team attribute), 16
result (sportsreference.mlb.schedule.Game attribute), 11
result (sportsreference.nba.schedule.Game attribute), 26
result (sportsreference.ncaab.schedule.Game attribute), 37
result (sportsreference.ncaaf.schedule.Game attribute), 47
result (sportsreference.nfl.schedule.Game attribute), 57
result (sportsreference.nhl.schedule.Game attribute), 66
run_difference (sportsreference.mlb.teams.Team attribute), 16
runners_left_on_base (sportsreference.mlb.teams.Team attribute), 16
runs (sportsreference.mlb.teams.Team attribute), 16
runs_against (sportsreference.mlb.teams.Team attribute), 16
runs_allowed (sportsreference.mlb.schedule.Game attribute), 11
runs_allowed_per_game (sportsreference.mlb.teams.Team attribute), 16
runs_batted_in (sportsreference.mlb.teams.Team attribute), 16
runs_scored (sportsreference.mlb.schedule.Game attribute), 11
rush_attempts (sportsreference.ncaaf.teams.Team attribute), 49
rush_attempts (sportsreference.nfl.schedule.Game attribute), 57
rush_attempts (sportsreference.nfl.teams.Team attribute), 60
rush_first_downs (sportsreference.ncaaf.teams.Team attribute), 50
rush_first_downs (sportsreference.nfl.teams.Team attribute), 60
rush_touchdowns (sportsreference.ncaaf.teams.Team attribute), 50
rush_touchdowns (sportsreference.nfl.schedule.Game attribute), 57
rush_touchdowns (sportsreference.nfl.teams.Team attribute), 60
rush_yards (sportsreference.ncaaf.teams.Team attribute), 50
rush_yards (sportsreference.nfl.schedule.Game attribute), 57
rush_yards (sportsreference.nfl.teams.Team attribute), 60
rush_yards_per_attempt (sportsreference.ncaaf.teams.Team attribute), 50
rush_yards_per_attempt (sportsreference.nfl.schedule.Game attribute), 57
rush_yards_per_attempt (sportsreference.nfl.teams.Team attribute), 60

S

sacrifice_flies (sportsreference.mlb.teams.Team attribute), 16
sacrifice_hits (sportsreference.mlb.teams.Team attribute), 16
save (sportsreference.mlb.schedule.Game attribute), 11
save_percentage (sportsreference.nhl.teams.Team attribute), 68
saves (sportsreference.mlb.teams.Team attribute), 16
Schedule (class in sportsreference.mlb.schedule), 11
Schedule (class in sportsreference.nba.schedule), 26
Schedule (class in sportsreference.ncaab.schedule), 37
Schedule (class in sportsreference.ncaaf.schedule), 47
Schedule (class in sportsreference.nfl.schedule), 57
Schedule (class in sportsreference.nhl.schedule), 66
schedule (sportsreference.mlb.teams.Team attribute), 16
schedule (sportsreference.nba.teams.Team attribute), 29
schedule (sportsreference.ncaab.teams.Team attribute), 41
schedule (sportsreference.ncaaf.teams.Team attribute), 50
schedule (sportsreference.nfl.teams.Team attribute), 60
schedule (sportsreference.nhl.teams.Team attribute), 68
season_losses (sportsreference.ncaab.schedule.Game attribute), 37
season_wins (sportsreference.ncaab.schedule.Game attribute), 37
shooting_percentage (sportsreference.nhl.teams.Team attribute), 69
short_handed_goals (sportsreference.nhl.schedule.Game attribute), 66
short_handed_goals (sportsreference.nhl.teams.Team attribute), 69

short_handed_goals_against (sportsreference.nhl.teams.Team attribute), 69
shots_against (sportsreference.nhl.teams.Team attribute), 69
shots_on_goal (sportsreference.nhl.schedule.Game attribute), 66
shots_on_goal (sportsreference.nhl.teams.Team attribute), 69
shutouts (sportsreference.mlb.teams.Team attribute), 16
simple_rating_system (sportsreference.mlb.teams.Team attribute), 16
simple_rating_system (sportsreference.ncaab.teams.Team attribute), 41
simple_rating_system (sportsreference.ncaaf.teams.Team attribute), 50
simple_rating_system (sportsreference.nfl.teams.Team attribute), 60
simple_rating_system (sportsreference.nhl.teams.Team attribute), 69
single_run_losses (sportsreference.mlb.teams.Team attribute), 16
single_run_record (sportsreference.mlb.teams.Team attribute), 16
single_run_wins (sportsreference.mlb.teams.Team attribute), 16
slugging_percentage (sportsreference.mlb.teams.Team attribute), 16
sportsreference.mlb.boxscore (module), 5
sportsreference.mlb.schedule (module), 10
sportsreference.mlb.teams (module), 12
sportsreference.nba.boxscore (module), 18
sportsreference.nba.schedule (module), 24
sportsreference.nba.teams (module), 27
sportsreference.ncaab.boxscore (module), 30
sportsreference.ncaab.schedule (module), 36
sportsreference.ncaab.teams (module), 38
sportsreference.ncaaf.boxscore (module), 43
sportsreference.ncaaf.schedule (module), 46
sportsreference.ncaaf.teams (module), 48
sportsreference.nfl.boxscore (module), 51
sportsreference.nfl.schedule (module), 55
sportsreference.nfl.teams (module), 58
sportsreference.nhl.boxscore (module), 61
sportsreference.nhl.schedule (module), 64
sportsreference.nhl.teams (module), 67
stadium (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
stadium (sportsreference.nfl.boxscore.Boxscore attribute), 54
steal_percentage (sportsreference.ncaab.teams.Team attribute), 41
steals (sportsreference.nba.schedule.Game attribute), 26
steals (sportsreference.nba.teams.Team attribute), 29
steals (sportsreference.ncaab.teams.Team attribute), 42
stolen_bases (sportsreference.mlb.teams.Team attribute), 16
streak (sportsreference.mlb.schedule.Game attribute), 11
streak (sportsreference.mlb.teams.Team attribute), 17
streak (sportsreference.ncaab.schedule.Game attribute), 37
streak (sportsreference.ncaaf.schedule.Game attribute), 47
strength_of_schedule (sportsreference.mlb.teams.Team attribute), 17
strength_of_schedule (sportsreference.ncaab.teams.Team attribute), 42
strength_of_schedule (sportsreference.ncaaf.teams.Team attribute), 50
strength_of_schedule (sportsreference.nfl.teams.Team attribute), 60
strength_of_schedule (sportsreference.nhl.teams.Team attribute), 69
strikeouts (sportsreference.mlb.teams.Team attribute), 17
strikeouts_per_base_on_balls (sportsreference.mlb.teams.Team attribute), 17
strikeouts_per_nine_innings (sportsreference.mlb.teams.Team attribute), 17

T

Team (class in sportsreference.mlb.teams), 12
Team (class in sportsreference.nba.teams), 27
Team (class in sportsreference.ncaab.teams), 38
Team (class in sportsreference.ncaaf.teams), 48
Team (class in sportsreference.nfl.teams), 58
Team (class in sportsreference.nhl.teams), 67
Teams (class in sportsreference.mlb.teams), 18
Teams (class in sportsreference.nba.teams), 30
Teams (class in sportsreference.ncaab.teams), 42
Teams (class in sportsreference.ncaaf.teams), 50
Teams (class in sportsreference.nfl.teams), 60
Teams (class in sportsreference.nhl.teams), 69
third_down_attempts (sportsreference.nfl.schedule.Game attribute), 57
third_down_conversions (sportsreference.nfl.schedule.Game attribute), 57
three_point_attempt_rate (sportsreference.ncaab.teams.Team attribute), 42
three_point_field_goal_attempts (sportsreference.nba.schedule.Game attribute), 26
three_point_field_goal_attempts (sportsreference.nba.teams.Team attribute), 29
three_point_field_goal_attempts (sportsreference.ncaab.teams.Team attribute), 42
three_point_field_goal_percentage (sportsreference.nba.schedule.Game attribute), 26
three_point_field_goal_percentage (sportsreference.nba.teams.Team attribute), 29

three_point_field_goal_percentage (sportsreference.ncaab.teams.Team attribute), 42
three_point_field_goals (sportsreference.nba.schedule.Game attribute), 26
three_point_field_goals (sportsreference.nba.teams.Team attribute), 29
three_point_field_goals (sportsreference.ncaab.teams.Team attribute), 42
time (sportsreference.mlb.boxscore.Boxscore attribute), 9
time (sportsreference.ncaab.schedule.Game attribute), 37
time (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
time (sportsreference.ncaaf.schedule.Game attribute), 47
time (sportsreference.nfl.boxscore.Boxscore attribute), 54
time (sportsreference.nhl.boxscore.Boxscore attribute), 63
time_of_day (sportsreference.mlb.boxscore.Boxscore attribute), 9
time_of_possession (sportsreference.nfl.schedule.Game attribute), 57
times_caught_stealing (sportsreference.mlb.teams.Team attribute), 17
times_hit_by_pitch (sportsreference.mlb.teams.Team attribute), 17
times_sacked (sportsreference.nfl.schedule.Game attribute), 57
times_struck_out (sportsreference.mlb.teams.Team attribute), 17
total_bases (sportsreference.mlb.teams.Team attribute), 17
total_goals_per_game (sportsreference.nhl.teams.Team attribute), 69
total_rebound_percentage (sportsreference.ncaab.teams.Team attribute), 42
total_rebounds (sportsreference.nba.schedule.Game attribute), 26
total_rebounds (sportsreference.nba.teams.Team attribute), 29
total_rebounds (sportsreference.ncaab.teams.Team attribute), 42
total_runs (sportsreference.mlb.teams.Team attribute), 17
triples (sportsreference.mlb.teams.Team attribute), 17
true_shooting_percentage (sportsreference.ncaab.teams.Team attribute), 42
turnover_percentage (sportsreference.ncaab.teams.Team attribute), 42
turnovers (sportsreference.nba.schedule.Game attribute), 26
turnovers (sportsreference.nba.teams.Team attribute), 29
turnovers (sportsreference.ncaab.teams.Team attribute), 42
turnovers (sportsreference.ncaaf.teams.Team attribute), 50
turnovers (sportsreference.nfl.teams.Team attribute), 60
two_point_field_goal_attempts (sportsreference.nba.teams.Team attribute), 29
two_point_field_goal_percentage (sportsreference.nba.teams.Team attribute), 30
two_point_field_goals (sportsreference.nba.teams.Team attribute), 30
type (sportsreference.ncaab.schedule.Game attribute), 37
type (sportsreference.nfl.schedule.Game attribute), 57

V

venue (sportsreference.mlb.boxscore.Boxscore attribute), 9

W

week (sportsreference.nfl.schedule.Game attribute), 57
whip (sportsreference.mlb.teams.Team attribute), 17
wild_pitches (sportsreference.mlb.teams.Team attribute), 17
win_percentage (sportsreference.mlb.teams.Team attribute), 17
win_percentage (sportsreference.ncaab.teams.Team attribute), 42
win_percentage (sportsreference.ncaaf.teams.Team attribute), 50
win_percentage (sportsreference.nfl.teams.Team attribute), 60
winner (sportsreference.mlb.boxscore.Boxscore attribute), 9
winner (sportsreference.mlb.schedule.Game attribute), 11
winner (sportsreference.nba.boxscore.Boxscore attribute), 23
winner (sportsreference.ncaab.boxscore.Boxscore attribute), 35
winner (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
winner (sportsreference.nfl.boxscore.Boxscore attribute), 54
winner (sportsreference.nhl.boxscore.Boxscore attribute), 63
winning_abbr (sportsreference.mlb.boxscore.Boxscore attribute), 9
winning_abbr (sportsreference.nba.boxscore.Boxscore attribute), 23
winning_abbr (sportsreference.ncaab.boxscore.Boxscore attribute), 35
winning_abbr (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
winning_abbr (sportsreference.nfl.boxscore.Boxscore attribute), 54
winning_abbr (sportsreference.nhl.boxscore.Boxscore attribute), 63
winning_name (sportsreference.mlb.boxscore.Boxscore attribute), 9

winning_name (sportsreference.nba.boxscore.Boxscore attribute), 23
winning_name (sportsreference.ncaab.boxscore.Boxscore attribute), 35
winning_name (sportsreference.ncaaf.boxscore.Boxscore attribute), 45
winning_name (sportsreference.nfl.boxscore.Boxscore attribute), 54
winning_name (sportsreference.nhl.boxscore.Boxscore attribute), 63
wins (sportsreference.mlb.teams.Team attribute), 17
wins (sportsreference.ncaab.teams.Team attribute), 42
wins (sportsreference.ncaaf.schedule.Game attribute), 47
wins (sportsreference.ncaaf.teams.Team attribute), 50
wins (sportsreference.nfl.teams.Team attribute), 60
wins (sportsreference.nhl.teams.Team attribute), 69
wins_last_ten_games (sportsreference.mlb.teams.Team attribute), 17
wins_last_thirty_games (sportsreference.mlb.teams.Team attribute), 17
wins_last_twenty_games (sportsreference.mlb.teams.Team attribute), 17
wins_vs_left_handed_pitchers (sportsreference.mlb.teams.Team attribute), 17
wins_vs_right_handed_pitchers (sportsreference.mlb.teams.Team attribute), 17
wins_vs_teams_over_500 (sportsreference.mlb.teams.Team attribute), 18
wins_vs_teams_under_500 (sportsreference.mlb.teams.Team attribute), 18

Y

yards (sportsreference.ncaaf.teams.Team attribute), 50
yards (sportsreference.nfl.teams.Team attribute), 60
yards_from_penalties (sportsreference.ncaaf.teams.Team attribute), 50
yards_from_penalties (sportsreference.nfl.teams.Team attribute), 60
yards_lost_from_sacks (sportsreference.nfl.schedule.Game attribute), 57
yards_per_play (sportsreference.ncaaf.teams.Team attribute), 50
yards_per_play (sportsreference.nfl.teams.Team attribute), 60