Project 1: NHL API Access and Analysis

Lee Pixton

6/20/2021

Contents

ST558 - NHL API Vignette

This is the first project for the course ST558 (Data Science for Statisticians) at NC State University. In this vignette, we will discuss reading data from the National Hockey League (NHL) API, and then providing summaries of the data pulled.

Required Project Packages

In order to run the code for the project, the required packages, as well as code to install them, are below.

- knitr
- httr
- RSQlite
- jsonlite
- tidyverse
- rmarkdown

Installing required packages

To install the packages required for this project, use the code chunk below, removing the beginning `#`

```
#install.packages("knitr", "httr", "RSQlite", "jsonlite", "rmarkdown", "tidyverse")
```

Function to Contact Franchise API

First thing to do is create a function that can access the franchise API. This will allow us to pull any number of information about each franchise. The `extension` option in the function takes in one of the following:

- `franchise` - Returns id, firstSeasonId and lastSeasonId and name of every team in the history of the nhl
- `franchise-team-totals` - Returns total stats for every franchise (ex roadTies, roadWins, etc)
- `franchise-season-records` - Returns drill-down into season records for a specific franchise
- `franchise-goalie-records` - Returns goalie records for the specified franchise
- `franchise-skater-records` - Returns skater records, same interaction as goalie endpoint
- `franchise-detail` - Returns captainHistory, coachingHistory, generalManagerHistory and a summary of retired numbers

And the `ID` option is available for specifying a team.

```
# This function is used to take input from the user and return appropriate data from the NHL franchise
franchiseAPI <- function(extension, ID = NULL, ...){
  base <- "https://records.nhl.com/site/api/"

  if(is.null(extension)){
    return("Must include a valid franchise call")
  } else{
    if(!is.null(ID)){
      if(extension == "franchise" | extension == "franchise-team-totals"){
        warning("ID is not allowed for these calls, will return ", extension, "without ID")
        URL <- paste0(base, extension)
      } else if(extension == "franchise-detail"){
        URL <- paste0(base, extension, "?cayenneExp=mostRecentTeamId=", ID)
      } else{
        URL <- paste0(base, extension, "?cayenneExp=franchiseId=", ID)
      }
    } else{
      URL <- paste0(base, extension)
    }
  }

  get_nhl <- GET(URL)
  nhl_text <- content(get_nhl, "text")
  nhl_json <- fromJSON(nhl_text, flatten=T)
  return(tbl_df(nhl_json$data))
}
```

Once we have the function, the next step is to put together a mapping from team name to ID number. This allows the user to input the team name without having to know the specific ID. We will use our API call to retrieve this information into a dataframe, that we can then use to complete a mapping in the wrapper function. The user will be able to use the full name, the abbreviation, or just the mascot to pull the ID.

```
team_mapping <- franchiseAPI("franchise")
team_mapping <- team_mapping %>% select(!(ends_with("SeasonId") | ends_with("TeamId") | ends_with("Plac
kable(team_mapping)
```

id	fullName	teamAbbrev	teamCommonName
1	Montréal Canadiens	MTL	Canadiens
2	Montreal Wanderers	MWN	Wanderers
3	St. Louis Eagles	SLE	Eagles
4	Hamilton Tigers	HAM	Tigers
5	Toronto Maple Leafs	TOR	Maple Leafs
6	Boston Bruins	BOS	Bruins
7	Montreal Maroons	MMR	Maroons
8	Brooklyn Americans	BRK	Americans
9	Philadelphia Quakers	QUA	Quakers
10	New York Rangers	NYR	Rangers
11	Chicago Blackhawks	CHI	Blackhawks
12	Detroit Red Wings	DET	Red Wings
13	Cleveland Barons	CLE	Barons
14	Los Angeles Kings	LAK	Kings
15	Dallas Stars	DAL	Stars
16	Philadelphia Flyers	PHI	Flyers
17	Pittsburgh Penguins	PIT	Penguins
18	St. Louis Blues	STL	Blues
19	Buffalo Sabres	BUF	Sabres
20	Vancouver Canucks	VAN	Canucks
21	Calgary Flames	CGY	Flames
22	New York Islanders	NYI	Islanders
23	New Jersey Devils	NJD	Devils
24	Washington Capitals	WSH	Capitals
25	Edmonton Oilers	EDM	Oilers
26	Carolina Hurricanes	CAR	Hurricanes
27	Colorado Avalanche	COL	Avalanche
28	Arizona Coyotes	ARI	Coyotes
29	San Jose Sharks	SJS	Sharks
30	Ottawa Senators	OTT	Senators
31	Tampa Bay Lightning	TBL	Lightning
32	Anaheim Ducks	ANA	Ducks
33	Florida Panthers	FLA	Panthers
34	Nashville Predators	NSH	Predators
35	Winnipeg Jets	WPG	Jets
36	Columbus Blue Jackets	CBJ	Blue Jackets
37	Minnesota Wild	MIN	Wild
38	Vegas Golden Knights	VGK	Golden Knights
39	Seattle Kraken	SEA	Kraken

Function to Contact Stats API

Next we need a function to access the stats API. Though there are many modifiers available to access multiple data sets, for our purposes we will only provide access to the `?expand=team.stats` modifier. By providing an `ID`, it will pull specific team data, but the function alone will pull data on all the teams.

```
# This function is used to take user input and ouput a tibble with data from the NHL Stats API
statsAPI <- function(ID=NULL,...){
  base <- "https://statsapi.web.nhl.com/api/v1/teams"
  end <- "?expand=team.stats"
  if (!is.null(ID)){
```

```
    URL <- paste0(base,"/ID",end)
  } else{
    URL <- paste0(base,end)
  }

  get_stats <- GET(URL)
  stats_text <- content(get_stats, "text")
  stats_json <- fromJSON(stats_text, flatten=T)
  return(tbl_df(stats_json))
}
```