

## 2 tipsport XML Feed

Ing. Peter Tomko, M.A.

27/8/2020

```
library(rvest)
library(stringr)
library(dplyr)
library(data.table)
library(xml2)
library(tidyverse)
library(DBI)
library(lubridate)

# - initialize the connection
con <- DBI::dbConnect(odbc::odbc(), "betting_ds")
```

Note: MIGHT BE REFRESHED ANYTIME.

### Introduction

The document provides an automated way of refreshing the data necessary for the project. There are two types of data source - historical match data [www.football-data.co.uk/data.php](http://www.football-data.co.uk/data.php) and actual odds provided by Tipsport.

### Tisport Data - Actual Bookmakers Odds

The betting company Tipsport provides the XML feed, in which the current bookmakers odds are included. The data source helps to create optimal investment strategy for the upcoming matches.

### Download and write into t\_tipsport\_bookmaker

Most of the obtained columns are described in the table creation script.

```
time_download <- Sys.time()

xml2_data <- xml2::download_xml("http://ban.tipsport.cz/f/oddsFeed.xml",
                                file = "oddsFeed.xml")
tipsport_data <- xml2::read_html(xml2_data, encoding = "UTF-8")

df_supersport <- xml_find_all(tipsport_data, "//supersport") %>%
  map_dfr(~ {
    # extract the attributes from the parent tag as a data.frame
    parent <- xml_attrs(.x) %>% enframe() %>% spread(name, value)

    # make a data.frame out of the attributes of the kids
    kids <- xml_children(.x) %>% map_dfr(~ as.list(xml_attrs(.x)))
```

```

    # combine them (bind_cols does not repeat parent rows)
    cbind.data.frame(parent, kids) %>% set_tidy_names() %>% as_tibble() }) %>%
as.data.frame() %>%
rename(sport = `name..1`,
       sport_category = `name..2`)

df_sport <- xml_find_all(tipsport_data, "//sport") %>%
  map_dfr(~ {
    # extract the attributes from the parent tag as a data.frame
    parent <- xml_attrs(.x) %>% enframe() %>% spread(name, value)

    # make a data.frame out of the attributes of the kids
    kids <- xml_children(.x) %>% map_dfr(~ as.list(xml_attrs(.x)))

    # combine them (bind_cols does not repeat parent rows)
    cbind.data.frame(parent, kids) %>% set_tidy_names() %>% as_tibble() }) %>%
as.data.frame() %>%
rename(sport_category = `name..1`,
       sport_league = `name..3`,
       sport_id = `id`,
       sport_urlold = urlold,
       sport_url = url)

df_competition <-
  data.frame(xml_find_all(tipsport_data, "//competition") %>%
    map_dfr(~ {
      # extract the attributes from the parent tag as a data.frame
      parent <-
        xml_attrs(.x) %>%
        enframe() %>%
        spread(name, value)

      # make a data.frame out of the attributes of the kids
      kids <-
        xml_children(.x) %>%
        map_dfr(~ as.list(xml_attrs(.x)))

      # combine them (bind_cols does not repeat parent rows)
      cbind.data.frame(parent, kids) %>%
        set_tidy_names() %>%
        as_tibble() }))) %>%

as.data.frame() %>%
rowwise() %>%
mutate(urlold = na.omit(c(`urlold..11`, urlold..12)),
       url = na.omit(c(`url..12`, url..13)),
       sport_league = `name..3`,
       sport_url = `url..4`,
       sport_urlold = `urlold..5`,
       sport_id = `id..2`,
       competition_name = na.omit(c(`name..8`, `name..9`))) %>%
select(-`id..6`, -`urlold..11`, -`url..12`, -`url..13`, -`name..3`,
       -`url..4`, -`urlold..5`, -`id..2`, -`urlold..12`, -`name..8`,

```

```

    -`name..9`)

df_match <- xml_find_all(tipsport_data, "//match") %>%
  map_dfr(~ {
    # extract the attributes from the parent tag as a data.frame
    parent <- xml_attrs(.x) %>% enframe() %>% spread(name, value)

    # make a data.frame out of the attributes of the kids
    kids <- xml_children(.x) %>% map_dfr(~ as.list(xml_attrs(.x)))

    # combine them (bind_cols does not repeat parent rows)
    cbind.data.frame(parent, kids) %>% set_tidy_names() %>% as_tibble() }) %>%
  as.data.frame() %>%
  rowwise() %>%
  mutate(eventid = na.omit(c(`id..8`, `id..9`)),
         eventname = na.omit(c(`name..9`, `name..10`)),
         competition_name = na.omit(c(`name..4`, `name..5`))) %>%
  select(-`id..2`, -`id..8`, -`id..9`, -`name..9`, -`name..10`,
        -`name..4`, -`name..5`)

df_event <- xml_find_all(tipsport_data, "//event") %>%
  map_dfr(~ {
    # extract the attributes from the parent tag as a data.frame
    parent <- xml_attrs(.x) %>% enframe() %>% spread(name, value)

    # make a data.frame out of the attributes of the kids
    kids <- xml_children(.x) %>% map_dfr(~ as.list(xml_attrs(.x)))

    # combine them (bind_cols does not repeat parent rows)
    cbind.data.frame(parent, kids) %>% set_tidy_names() %>% as_tibble() }) %>%
  as.data.frame() %>%
  rename(eventid = `id..4`,
         eventname = name,
         opportunityid = `id..6`)

t_tipsport_bookmaker <-
  df_event %>%
  left_join(., df_match) %>%
  left_join(., df_competition) %>%
  left_join(., df_sport) %>%
  left_join(., df_supersport) %>%
  mutate(rate = as.numeric(rate),
         created_at = time_download,
         dateclosed = dmy_hm(dateclosed)) %>%
  as.data.frame()

dbWriteTable(con, "t_tipsport_bookmaker", t_tipsport_bookmaker,
             row.names = F, append = T)

rm(df_competition)
rm(df_event)
rm(df_match)
rm(df_sport)

```

```
rm(df_supersport)
rm(tipsport_data)
rm(time_download)
rm(xml2_data)
```