Web Scraping - Soccer tables

Web Scraping project: gather soccer tables

Project's objectives:

1. To practice mining data from online website using `rvest`
2. To clean up the mined data into ready-to-use data.frame
3. To do exploratory data analysis
4. To study possible connection between points earned and clubs' performances and to build predictive models

---

Libraries

```
library('rvest', lib='C:/temp')
library('ggplot2', lib='C:/temp')
library('ggrepel', lib='C:/temp')
library('ggthemes', lib='C:/temp')
library('tidyverse', lib='C:/temp')
library('kableExtra', lib='C:/temp')
```

Part 1: Functions to get tables from multiple pages

In mining the data needed, I used the package `rvest`, link, together with web browser `SelectorGadget`, link.

The data is from ESPN, an example of which is link. Once I extracted into `R`, the tables from ESPN contained several pieces of data that were not related, such as:

```
urlsample <- 'http://www.espn.com/soccer/standings/_/league/ita.1/season/2017'
# Raw web info
web <- urlsample %>%
  read_html() %>%
  html_nodes('.stat-cell , .pr3 , .subHeader__item--content , #fittPageContainer a') %>%
  html_text()
matrix(web[c(1:4,79:106)], ncol = 4, byrow =T)
```

```
##      [,1]        [,2]        [,3]  [,4]
## [1,] "2017/2018" "2017/2018" "1"   ""
## [2,] "20"        ""          "BEN" "Benevento"
## [3,] "GP"        "GP"        "GP"  "W"
## [4,] "W"         "W"         "D"   "D"
## [5,] "D"         "L"         "L"   "L"
## [6,] "F"         "F"         "F"   "A"
## [7,] "A"         "A"         "GD"  "GD"
## [8,] "GD"        "P"         "P"   "P"
```

```
unrelatedrows <- c()
for (i in 0:3) {
  temp <- substr(web[length(web)-i],1,20)
  unrelatedrows <- c(unrelatedrows, temp)
}
unrelatedrows
```

```
## [1] "All Serie A News"     "Bonetti: Juve will g" "10 man Milan hang on"
## [4] "Will Man United relo"
```

The first 2 rows were the current table's season, and this number was fixed across all leagues and seasons. The last 4 rows were news related to the current league, and this number varied across different leagues, for example the Italian league had 4, and the Portugese league had 11 of them. The middle `GP`, `D`, ... were the table columns' names, they stood for:

1. `GP`: Games played
2. `W`: Wins
3. `D`: Draws
4. `L`: Losses
5. `F`: Goals For
6. `A`: Goals Against
7. `GD`: Goals Difference
8. `P`: Points

Each of these names were repeated 3 times across all leagues and seasons, and were at fixed positions: after information about the clubs in the league.

In dealing with the varying unrelated rows at the end, I wrote a function `check_link(url)` to check for the number of such rows.

1.1. `check_link(url)`: function to check # unrelated rows at end of raw table

Function will return the #(rows) to be removed at the end of table

```
check_link <- function(url) {
  url <- url
  # Raw web info
  web <- url %>%
    read_html() %>%
    html_nodes('.stat-cell , .pr3 , .subHeader__item--content , #fittPageContainer a') %>%
    html_text()
  # Flag: F if cell is string, T if cell is number: which is wanted
  flag <- FALSE
  # counter
  n <- 0
  while (flag == FALSE) {
    # Check for if current cell is a string
    if ( is.na(as.integer(web[length(web)-n]))) {
      # Yes: cell is string: increase count
      n <- n+1
    } else {
      # No: cell is number: Good
      flag <- TRUE
      n
    }
  }
  return(n)
}
```

After knowing that number, the desired rows were what remained. The function `get_table(url, leaguename, k)` helped automate the mining across different url addresses for different leagues and seasons.

1.2. `get_table(url, leaguename, k)`: function to get table

```
get_table <- function(url, leaguename, k) {
  web <- url %>%
    read_html() %>%
    html_nodes('.stat-cell , .pr3 , .subHeader__item--content , #fittPageContainer a') %>%
    html_text()

  # Get number of clubs for current season, by counting the number of unrelated rows
  # Top: 2, Middle: 24, for table headings: 8 unique values, each repeated 3x
  # Bottom: k, to be checked link-by-link
  n <- (length(web) - 2 - k - 24)/12

  # Get the season's year & League
  year <- rep(web[1], n)
  league <- rep(leaguename, n)
  # Filter out the unneeded rows
  web <- web[3:(length(web) - k)]

  # Dataframe of clubs
  df <- web %>%
    head(4*n) %>%
    matrix(nrow = n, byrow = T) %>%
    data.frame()
  # Dummy column: pos, to be referenced when mergin data.frames
  df$pos <- as.integer(as.character(df[,1]))

  # 1st 4*n rows are Club names, already used above
  web <- web %>% tail(length(web) - 4*n)

  # Temp data.frame of Clubs' attributes
  temp <- web %>%
    # Remove first 24 cells
    tail(length(web) - 24) %>%
    as.numeric() %>%
    # Convert to matrix, with n rows for n clubs
    matrix(nrow = n, byrow = T) %>%
    # Dummy var: position, to be referenced when merging 2 data.frames
    cbind(1:n) %>%
    data.frame()

  # Merge 2 data.frames
  df <- merge(df, temp, by.x = 'pos', by.y = 'X9')

  # Clean up the data.frame
  df <- df %>%
    select(-c(2,3)) %>%
    mutate(year = year,
           league = league)

  # Rename df
  colnames(df) <- c('Standing', 'Club_S', 'Club', 'Games', 'Wins', 'Draws', 'Losses',
                    'G_For', 'G_Against', 'G_Diff', 'Points', 'Season', 'League')
  return(df)
}
```

Part 2: Gather tables from leagues

In gathering the tables, my idea was to supply the link for each leagues, with the available years, the return of which would be a data.frame, to be appended to an empty data.frame created below.

National league tables

In checking the ESPN websites, I decided to gather information from the following leagues and the season years:

```
# Default data.frame, to be added in
df <- data.frame()

leagues <- c('Italy', 'England', 'Spain', 'Germany', 'France', 'Netherland',
             'Portugal', 'Russia', 'Turkey', 'Greece', 'Brazil', 'Argentina')
years <- c('01/02 - 17/18', '01/02 - 17/18', '01/02 - 17/18', '01/02 - 17/18',
           '02/03 - 17/18', '01/02 - 17/18', '06/07 - 17/18', '06/07 - 16/17',
           '06/07 - 17/18', '06/07 - 17/18', '06 - 18', '03/04 - 17/18')
urls <- c('http://www.espn.com/soccer/standings/_/league/ita.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/eng.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/esp.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/ger.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/fra.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/ned.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/por.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/rus.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/tur.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/GRE.1/seasontype/1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/bra.1/season/2017',
         'http://www.espn.com/soccer/standings/_/league/arg.1/season/2017')
data.frame(leagues, years, urls)
```

```
##       leagues         years
## 1       Italy 01/02 - 17/18
## 2     England 01/02 - 17/18
## 3       Spain 01/02 - 17/18
## 4     Germany 01/02 - 17/18
## 5      France 02/03 - 17/18
## 6  Netherland 01/02 - 17/18
## 7    Portugal 06/07 - 17/18
## 8      Russia 06/07 - 16/17
## 9      Turkey 06/07 - 17/18
## 10     Greece 06/07 - 17/18
## 11     Brazil       06 - 18
## 12  Argentina 03/04 - 17/18
##                                                                        urls
## 1               http://www.espn.com/soccer/standings/_/league/ita.1/season/2017
## 2               http://www.espn.com/soccer/standings/_/league/eng.1/season/2017
## 3               http://www.espn.com/soccer/standings/_/league/esp.1/season/2017
## 4               http://www.espn.com/soccer/standings/_/league/ger.1/season/2017
## 5               http://www.espn.com/soccer/standings/_/league/fra.1/season/2017
## 6               http://www.espn.com/soccer/standings/_/league/ned.1/season/2017
## 7               http://www.espn.com/soccer/standings/_/league/por.1/season/2017
## 8               http://www.espn.com/soccer/standings/_/league/rus.1/season/2017
## 9               http://www.espn.com/soccer/standings/_/league/tur.1/season/2017
## 10 http://www.espn.com/soccer/standings/_/league/GRE.1/seasontype/1/season/2017
## 11              http://www.espn.com/soccer/standings/_/league/bra.1/season/2017
## 12              http://www.espn.com/soccer/standings/_/league/arg.1/season/2017
```

Prepare and Set-up needed data.frames

1. Prepare data.frame of information on what to mine from the ESPN website:

    - Season year
    - League name
    - URL

```
yrita <- c(2017:2001); yreng <- c(2017:2001); yresp <- c(2017:2001)
yrger <- c(2017:2001); yrfra <- c(2017:2002); yrned <- c(2017:2001)
yrpor <- c(2017:2010, 2008:2006); yrrus <- c(2016:2006)
yrtur <- c(2017:2006); yrgre <- c(2017:2016, 2014:2006)
yrbra <- c(2017:2006); yrarg <- c(2017, 2015:2003)
yrs <- c(yrita, yreng, yresp, yrger, yrfra, yrned, yrpor, yrrus, yrtur, yrgre, yrbra, yrarg)
leagues <-
  c(rep(c('Serie A', 'http://www.espn.com/soccer/standings/_/league/ita.1/season/'), length(yrita)),
    rep(c('EPL', 'http://www.espn.com/soccer/standings/_/league/eng.1/season/'), length(yreng)),
    rep(c('La Liga', 'http://www.espn.com/soccer/standings/_/league/esp.1/season/'), length(yresp)),
    rep(c('Bundesliga', 'http://www.espn.com/soccer/standings/_/league/ger.1/season/'), length(yrger)),
    rep(c('Ligue 1', 'http://www.espn.com/soccer/standings/_/league/fra.1/season/'), length(yrfra)),
    rep(c('Netherland', 'http://www.espn.com/soccer/standings/_/league/ned.1/season/'), length(yrned)),
    rep(c('Portugal', 'http://www.espn.com/soccer/standings/_/league/por.1/season/'), length(yrpor)),
    rep(c('Russia', 'http://www.espn.com/soccer/standings/_/league/rus.1/season/'), length(yrrus)),
    rep(c('Turkey', 'http://www.espn.com/soccer/standings/_/league/tur.1/season/'), length(yrtur)),
    rep(c('Greece', 'http://www.espn.com/soccer/standings/_/league/GRE.1/seasontype/1/season/'), length(yrgre
    rep(c('Brazil', 'http://www.espn.com/soccer/standings/_/league/bra.1/season/'), length(yrbra)),
    rep(c('Argentina', 'http://www.espn.com/soccer/standings/_/league/arg.1/season/'), length(yrarg))
    )

df_url <- data.frame(yrs, leagues[c(TRUE, FALSE)], leagues[c(FALSE, TRUE)])
colnames(df_url) <- c('Season', 'League', 'URL')
print(paste0('Dimension of the Information data.frame: ', dim(df_url)[1], ' x ', dim(df_url)[2]))
```

```
## [1] "Dimension of the Information data.frame: 172 x 3"
```

2. Get index of first instance of new league: to check for number of unrelated rows at end of table

This was the problem that the function `check_link(url)` from 1.1. was written to solve.

```
# Step 1: Create intervals: only need to check 1st instance of new league, and not all of them
intervals <- c(length(yrita), length(yreng), length(yresp), length(yrger),
               length(yrfra), length(yrned), length(yrpor), length(yrrus),
               length(yrtur), length(yrgre), length(yrbra), length(yrarg))
# Step 2: Create an index of what to test
idx <- c(1 + cumsum(c(0, intervals)), end)
# to access idx, use idx[[i]]
print(paste0('Examples of such indices: ', idx[[1]], ', ', idx[[2]]))
```

```
## [1] "Examples of such indices: 1, 18"
```

3. Gather the tables into the main data.frame

```
for (i in 1:length(yrs)) {
  # Current season year
  season <- df_url$Season[i]
  # Current league name
```

```
  league <- df_url$League[i]
  # Current url
  url <- paste0(df_url$URL[i], season)
  # Check for if the current i is in index, if yes, check for #(unrelated rows)
  if (i %in% idx) {
    k <- check_link(url)
  }
  temptable <- get_table(url, league, k)
  # bind_rows to attach new data.frame below df
  df <- bind_rows(df, temptable)
}
```

Samples of clubs per leagues

After mining the website and cleaning up the data, the sample data.frame is as follows:

```
# Print head of each league
df %>%
  group_by(League) %>%
  filter(row_number() == c(1:3)) %>%
  kable(align = 'c') %>%
  kable_styling(bootstrap_options = "striped", font_size = 6)
```

Standing	Club_S	Club	Games	Wins	Draws	Losses	G_For	G_Against	G_Diff	Points	Season	League
1	JUV	Juventus	38	30	5	3	86	24	62	95	2017/2018	Serie A
2	NAP	Napoli	38	28	7	3	77	29	48	91	2017/2018	Serie A
3	ROMA	AS Roma	38	23	8	7	61	28	33	77	2017/2018	Serie A
1	MNC	Manchester City	38	32	4	2	106	27	79	100	2017-2018	EPL
2	MAN	Manchester United	38	25	6	7	68	28	40	81	2017-2018	EPL
3	TOT	Tottenham Hotspur	38	23	8	7	74	36	38	77	2017-2018	EPL
1	BAR	Barcelona	38	28	9	1	99	29	70	93	2017/2018	La Liga
2	ATM	Atletico Madrid	38	23	10	5	58	22	36	79	2017/2018	La Liga
3	MAD	Real Madrid	38	22	10	6	94	44	50	76	2017/2018	La Liga
1	BMU	Bayern Munich	34	27	3	4	92	28	64	84	2017/2018	Bundesliga
2	SCH	Schalke 04	34	18	9	7	53	37	16	63	2017/2018	Bundesliga
3	HOF	TSG Hoffenheim	34	15	10	9	66	48	18	55	2017/2018	Bundesliga
1	PSG	Paris Saint-Germain	38	29	6	3	108	29	79	93	2017/2018	Ligue 1
2	MON	AS Monaco	38	24	8	6	85	45	40	80	2017/2018	Ligue 1
3	LYON	Lyon	38	23	9	6	87	43	44	78	2017/2018	Ligue 1
1	PSV	PSV Eindhoven	34	26	5	3	87	39	48	83	2017/2018	Netherland
2	AJAX	Ajax Amsterdam	34	25	4	5	89	33	56	79	2017/2018	Netherland
3	ALK	AZ Alkmaar	34	22	5	7	72	38	34	71	2017/2018	Netherland
1	POR	FC Porto	34	28	4	2	82	18	64	88	2017/2018	Portugal
2	BEN	Benfica	34	25	6	3	80	22	58	81	2017/2018	Portugal
3	SCP	Sporting CP	34	24	6	4	63	24	39	78	2017/2018	Portugal
1	SPM	Spartak Moscow	30	22	3	5	46	27	19	69	2016	Russia
2	CSKA	CSKA Moscow	30	18	8	4	47	15	32	62	2016	Russia
3	ZEN	Zenit St Petersburg	30	18	7	5	50	19	31	61	2016	Russia
1	GAL	Galatasaray	34	24	3	7	75	33	42	75	2017/2018	Turkey
2	FEN	Fenerbahce	34	21	9	4	78	36	42	72	2017/2018	Turkey
3	ISTB	Istanbul Basaksehir	34	22	6	6	62	34	28	72	2017/2018	Turkey
1	AEK	AEK Athens	30	21	7	2	50	12	38	70	2017/2018	Greece
2	PAOK	PAOK Salonika	30	21	4	5	59	19	40	64	2017/2018	Greece
3	OLY	Olympiakos	30	18	6	6	63	28	35	57	2017/2018	Greece
1	COR	Corinthians	38	21	9	8	50	30	20	72	Brasileirao 2017	Brazil
2	PAL	Palmeiras	38	19	6	13	61	45	16	63	Brasileirao 2017	Brazil
3	SAN	Santos	38	17	12	9	42	32	10	63	Brasileirao 2017	Brazil
1	CABJ	Boca Juniors	27	18	4	5	50	22	28	58	2017/18 Superliga	Argentina
2	GCM	Godoy Cruz Antonio Tomba	27	17	5	5	45	24	21	56	2017/18 Superliga	Argentina
3	SLOR	San Lorenzo	27	14	8	5	31	20	11	50	2017/18 Superliga	Argentina

Part 3: Exploratory Data Analysis

```
leagues <- df %>%
  select(League) %>%
  unique()
```

Plots

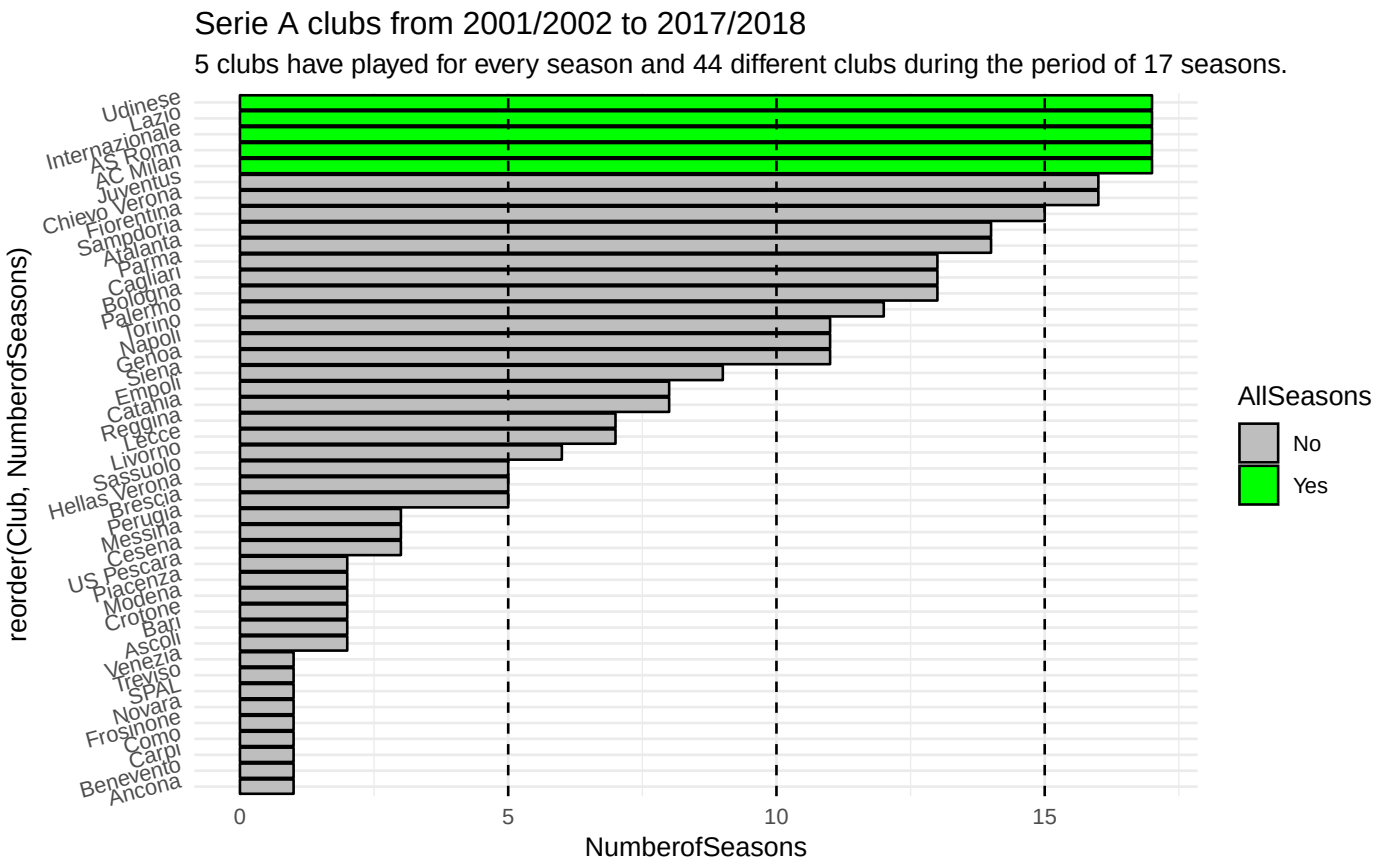
3.1. `plotseasons(league)`: function to draw plot of number of seasons for each club

```
plotseasons <- function(league) {
  # n: number of seasons in dataset/league
  s <- df %>%
    filter(League == league) %>%
    select(Season) %>%
    unique() %>%
    nrow()

  # Numbers about clubs/seasons
  temp <- df %>%
    filter(League == league) %>%
    group_by(Club) %>%
    summarise(NumberofSeasons = n()) %>%
    mutate(AllSeasons = ifelse(NumberofSeasons == s, 1, 0)) %>%
    summarise(k = sum(AllSeasons),
              n = n())
  # k: #(clubs appearing in all seasons)
  # n: #(clubs with >= 1 season)
  k <- temp[1,1]
  n <- temp[1,2]

  # Season years First and Last
  year <- df %>%
    filter(League == league) %>%
    select(Season) %>%
    filter(row_number() == 1 | row_number() == n())

  # Barplot
  df %>%
    filter(League == league) %>%
    group_by(Club) %>%
    summarise(NumberofSeasons = n()) %>%
    arrange(desc(NumberofSeasons)) %>%
    mutate(AllSeasons = ifelse(NumberofSeasons == s, 'Yes', 'No')) %>%
    ggplot(aes(x=reorder(Club, NumberofSeasons),
               y=NumberofSeasons,
               fill = AllSeasons)) +
    geom_col(color = 'black') +
    scale_fill_manual(values = c('grey', 'green')) +
    coord_flip() +
    geom_hline(yintercept = 15, linetype = 'dashed') +
    geom_hline(yintercept = 10, linetype = 'dashed') +
    geom_hline(yintercept = 5, linetype = 'dashed') +
    labs(title = paste0(league, ' clubs from ', year[2,1], ' to ', year[1,1]),
         subtitle = paste0(k, ' clubs have played for every season and ', n, ' different clubs during the per
    theme_minimal() +
```

```
    theme(axis.text.y = element_text(angle = 15, vjust = 0))
}
```

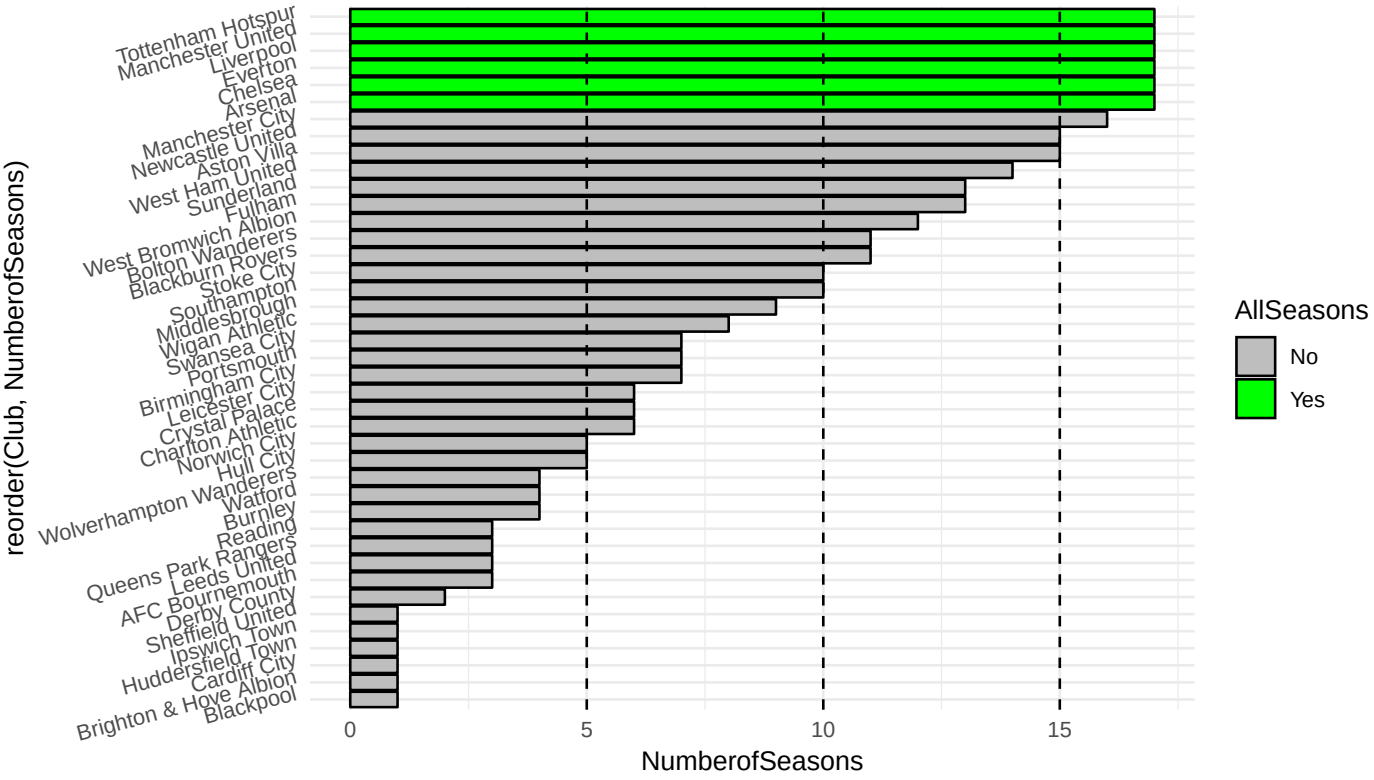
Use the function to plot some sample leagues.

```
for (i in 1:4) {
  league <- leagues[i,1]
  print(plotseasons(league))
}
```



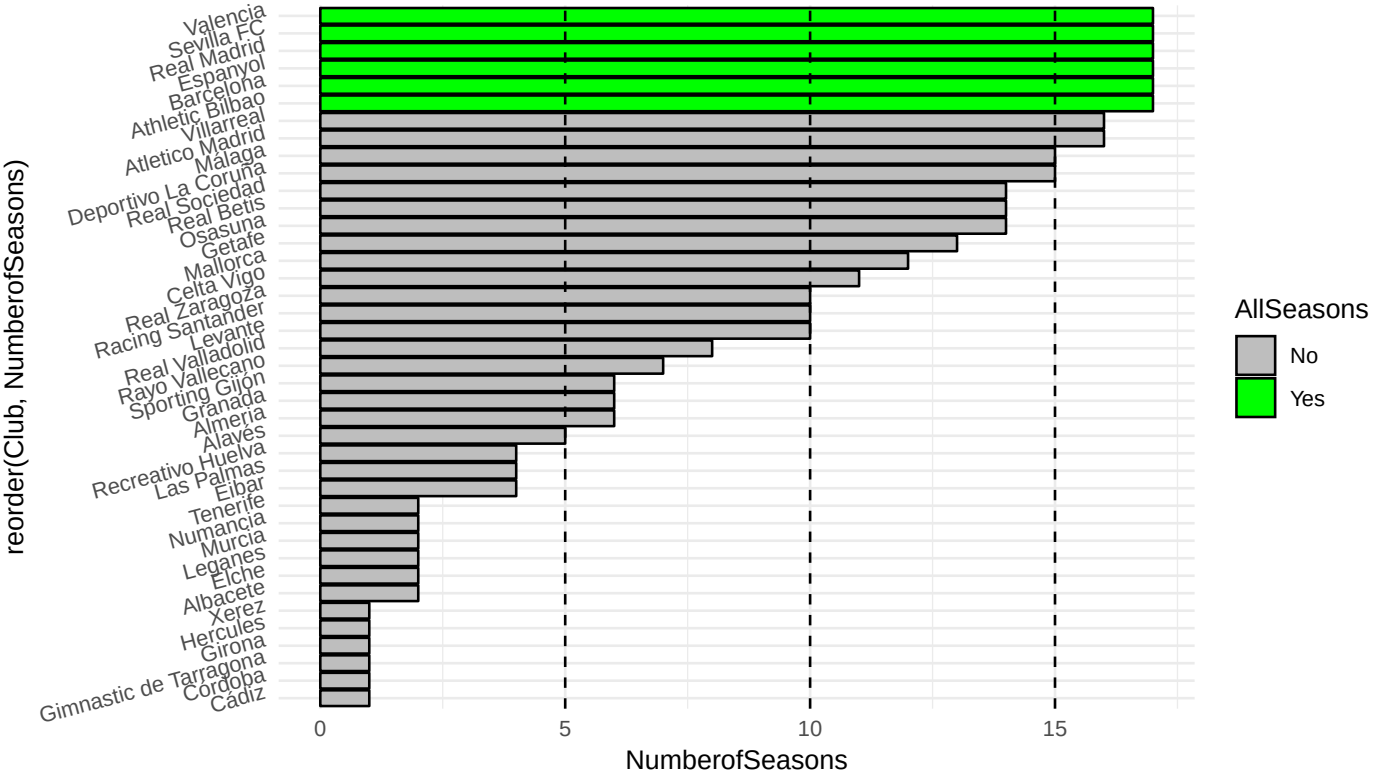
EPL clubs from 2001−2002 to 2017−2018

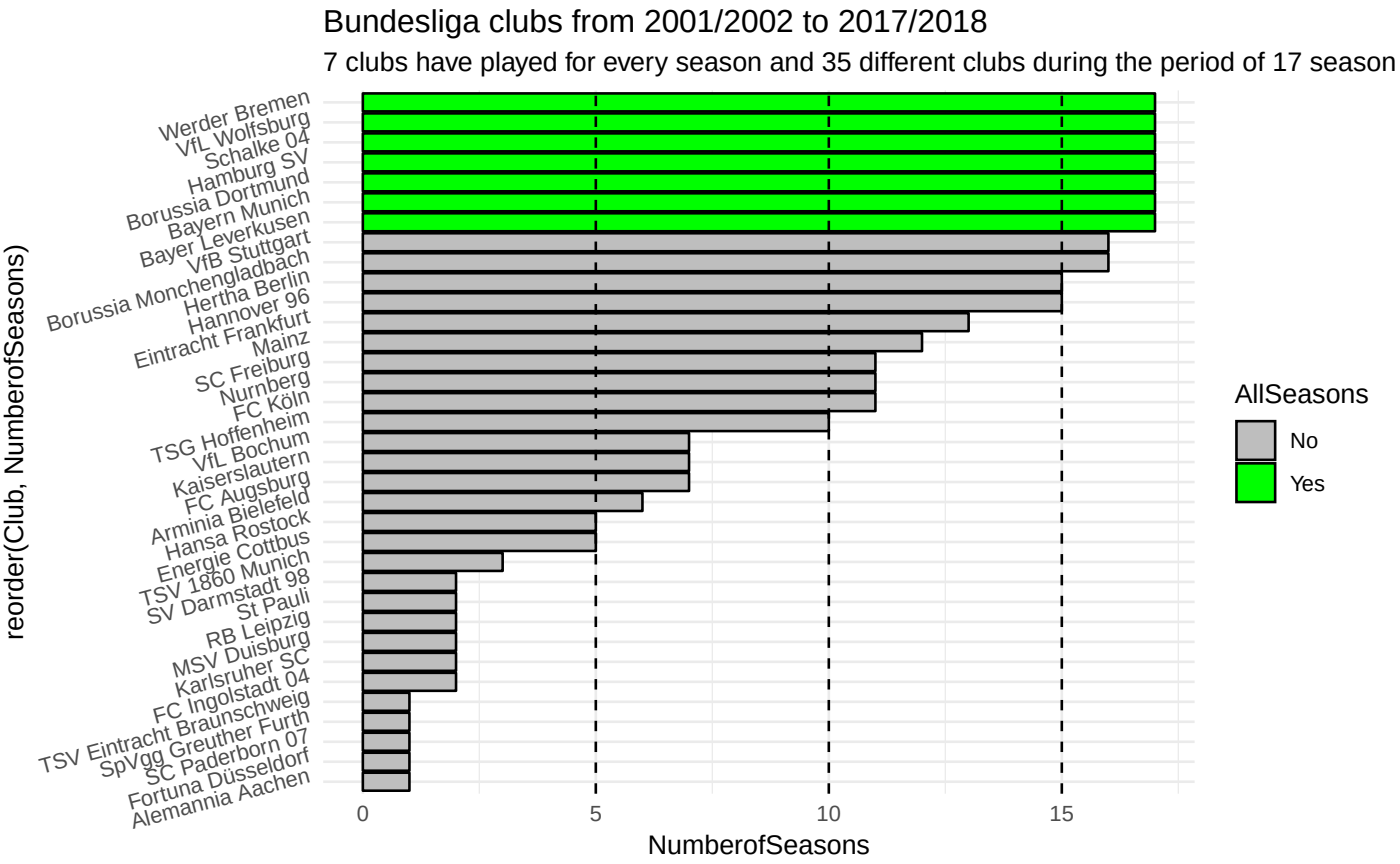
6 clubs have played for every season and 41 different clubs during the period of 17 seasons



La Liga clubs from 2001/2002 to 2017/2018

6 clubs have played for every season and 40 different clubs during the period of 17 seasons.




Bundesliga clubs from 2001/2002 to 2017/2018
7 clubs have played for every season and 35 different clubs during the period of 17 season
reorder(Club, NumberofSeasons)
Werder Bremen
VfL Wolfsburg
Schalke 04
Hamburg SV
Borussia Dortmund
Bayern Munich
Bayer Leverkusen
VfB Stuttgart
Borussia Monchengladbach
Hertha Berlin
Hannover 96
Eintracht Frankfurt
Mainz
SC Freiburg
Nurnberg
FC Köln
TSG Hoffenheim
VfL Bochum
Kaiserslautern
FC Augsburg
Arminia Bielefeld
Hansa Rostock
Energie Cottbus
TSV 1860 Munich
SV Darmstadt 98
St Pauli
RB Leipzig
MSV Duisburg
Karlsruher SC
FC Ingolstadt 04
TSV Eintracht Braunschweig
SpVgg Greuther Furth
SC Paderborn 07
Fortuna Düsseldorf
Alemannia Aachen
0
5
10
15
NumberofSeasons
AllSeasons
No
Yes