Task 2 Solution Proposal

Sander Eide Dahling

2023-06-18

Task 2 Solution Proposal

First we need to import all relevant packages to the memory of R. I will make use of the Tidyverse in this task, as it is relatively simple. We will specifically use tidyr and dplyr

```
# Loading relevant libraries into memory
library(tidyverse)
```

Before loading in the data, you need to make sure that the working directory of the program is the same as the location of the file. I make sure this is the case by running this code:

```
# Getting the current path of the source file
current_path = dirname(rstudioapi::getActiveDocumentContext()$path)

# Then you are able to set the working directory as this file path
#setwd(current_path)
```

In my example, the working directory will be set to **C:/dir/git/nhh-course-help/GUIDES/Data-Science-Academy/R-Training-Tasks/Task2-PremierLeague-23-24/[solution]**. This might not look the same to you, especially if you are on a Mac.

With the working directory set up correctly, we can start by loading in the data. The data used for this task is a data set gathered from Kaggle. If you are unable to download the data set from this link, you can find the data in the "data" folder in the Task2 folder.

```
# Loading in the data set

df <- read_csv("../data/FPL_Schedule2324.csv")

head(df)
```

```
## # A tibble: 6 x 39
##   Team   GW1   GW2   GW3   GW4   GW5   GW6   GW7   GW8   GW9   GW10  GW11  GW12
##   <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Arsen~ Nott~ Crys~ Fulh~ Man ~ Ever~ Spurs Bour~ Man ~ Chel~ Shef~ Newc~ Burn~
## 2 Aston~ Newc~ Ever~ Burn~ Live~ Crys~ Chel~ Brig~ Wolv~ West~ Luto~ Nott~ Fulh~
## 3 Bourn~ West~ Live~ Spurs Bren~ Chel~ Brig~ Arse~ Ever~ Wolv~ Burn~ Man ~ Newc~
## 4 Brent~ Spurs Fulh~ Crys~ Bour~ Newc~ Ever~ Nott~ Man ~ Burn~ Chel~ West~ Live~
## 5 Brigh~ Luto~ Wolv~ West~ Newc~ Man ~ Bour~ Asto~ Live~ Man ~ Fulh~ Ever~ Shef~
## 6 Burnl~ Man ~ Luto~ Asto~ Spurs Nott~ Man ~ Newc~ Chel~ Bren~ Bour~ Crys~ Arse~
## # ... with 26 more variables: GW13 <chr>, GW14 <chr>, GW15 <chr>, GW16 <chr>,
## #   GW17 <chr>, GW18 <chr>, GW19 <chr>, GW20 <chr>, GW21 <chr>, GW22 <chr>,
## #   GW23 <chr>, GW24 <chr>, GW25 <chr>, GW26 <chr>, GW27 <chr>, GW28 <chr>,
## #   GW29 <chr>, GW30 <chr>, GW31 <chr>, GW32 <chr>, GW33 <chr>, GW34 <chr>,
## #   GW35 <chr>, GW36 <chr>, GW37 <chr>, GW38 <chr>
```

Here we can see that the data is in the wide format. According to Objective 1, we must make the format "longer".

```
# Use the tidyr function "pivot_longer" and specify which columns to convert to long format
df_long <- df %>%
  pivot_longer(cols = "GW1":"GW38")

# Completed Objective 1
head(df_long)
```

```
## # A tibble: 6 x 3
##   Team    name  value
##   <chr>   <chr> <chr>
## 1 Arsenal GW1   Nott'm Forest
## 2 Arsenal GW2   Crystal Palace
## 3 Arsenal GW3   Fulham
## 4 Arsenal GW4   Man Utd
## 5 Arsenal GW5   Everton
## 6 Arsenal GW6   Spurs
```

```
# Changing the names of the columns to make them more readable

df_renamed <- df_long %>%
  rename(team = Team,
         game_week = name,
         opponent = value)

# Completed Objective 2
df_renamed
```

```
## # A tibble: 760 x 3
##    team    game_week opponent
##    <chr>   <chr>     <chr>
##  1 Arsenal GW1       Nott'm Forest
##  2 Arsenal GW2       Crystal Palace
##  3 Arsenal GW3       Fulham
##  4 Arsenal GW4       Man Utd
##  5 Arsenal GW5       Everton
##  6 Arsenal GW6       Spurs
##  7 Arsenal GW7       Bournemouth
##  8 Arsenal GW8       Man City
##  9 Arsenal GW9       Chelsea
## 10 Arsenal GW10      Sheff Utd
## # ... with 750 more rows
```

```
# Using the filter function to only choose a specific set of observations
df_last_day <- df_renamed %>%
  filter(game_week == "GW38",
         team %in% c("Arsenal", "Liverpool", "Man City"))

# Completed Objective 3
df_last_day
```

```
## # A tibble: 3 x 3
##   team      game_week opponent
##   <chr>     <chr>     <chr>
```

```
## 1 Arsenal   GW38      Everton
## 2 Liverpool GW38      Wolves
## 3 Man City  GW38      West Ham
```