Math 660

Sep 2023

Name: Krisha Shah

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## New names:
## * `xG` -> `xG...6`
## * `xG` -> `xG...8`
```

1. Answer the following

(a) How many teams are there in the league?

```
num_of_teams <- length(unique(standings$Squad))
print(num_of_teams)
```

```
## [1] 20
```

Comment: The standings-1920 dataset consist of final ranking of all teams for the league. Thus we can find all the team names in this dataset. There are a total of 20 teams.

(b) How many players are there in the league?

```
num_of_players <- length(unique(players$Player))
print(num_of_players)
```

```
## [1] 515
```

Comment: The players-1920 dataset consist of all the players information. Thus we can find the total number of players in this dataset. I also cofirmed that goalkeepers are included in the players-1920 dataset. Some of the player names are repeated in two different squad guessing they would be transferred mid-season. Hence there are 515 players in the league.

(c) What is the total number of goals scored for the season?

```
num_of_goals1 <- sum(players$Gls)
print(num_of_goals1)
```

```
## [1] 1002
```

```
num_of_goals3 <- sum(teamstats$Gls)
print(num_of_goals3)
```

```
## [1] 1002
```

Comment: We can find the total number of goals scored in the season either by using the players data or teamstats data. The total number of goals scores for the season is 1002.

(d) What is the average attendance (spectators) at the games?

```
total_attendance <- mean(na.omit(scoresfixtures$Attendance))
print(total_attendance)
```

```
## [1] 39314.22
```

Comment: We can find the sum of total attendance in the standings dataset. Later to find the average attendace for games we divide by 380 (38 matches every week for 10 weeks). Average attendance at the games is 1568.216

2. Identify the primary key for each table. Verify your answers. Note: ignore the Rk variable, as it usually just represents the row number.

Answer:

(1) For goalkeepers-1920: (PrimaryKey - Player) - Player column has 37 unique names stating all of them are players.

(2) For players-1920: (PrimaryKey - Player,Nation) - The player names are not unique since there might be some players who are tranferred to another team inbetween the league. Thus, a combination of player and nation will give us unique values. Thus it can be considered as a primary key.

(3) For scoresfixtures-1920: (PrimaryKey - Date,Time,Home) - This dataset provides information about each match. There are 380 total matches that took place in the league. To identify them uniquely, we can say that each unique match is played on a specific date,time and home. No two matches can be played simultaneously at the same place and same time.

(4) For standings-1920: (PrimaryKey - Squad) - This dataset gives information about the final ranking of the teams. Thus 20 unique teams make the primary key for the dataset.

(5) For teamgoalkeeping-1920: (PrimaryKey - Squad) - This dataset gives information about the goalkeepers in each team. Thus 20 unique teams make the primary key for the dataset.

```
scoresfixtures %>% count(Date, Time, Home) %>% filter(n>1)
```

(6) For teamstats-1920: (PrimaryKey - Squad) - This dataset gives information about the players of the teams. Thus 20 unique teams make the primary key for the dataset.

```
## # A tibble: 0 x 4
## # i 4 variables: Date <date>, Time <chr>, Home <chr>, n <int>
```

3. Find a table with a foreign key(s), and write down the name of the table, the foreign key(s) and the associated table(s).

Answer:

(1) Table(goalkeepers) has foreign key(Player) associated with table(players)

(2) Table(players) has foreign key(Squad) associated with table(standings)

(3) Table(teamgoalkeeping) has foreign key(Squad) associated with table(standings)

(4) Table(teamstats) has foreign key(Squad) associated with table(standings)

4. For the players and goalkeepers data sets, the "Player" variable has two versions of the player name, and the "Nation" variable also has two abbreviations listed for the nationalities of the players. Create a new players tibble and a new goalkeepers tibble which contain only the first version of the player name (and not the second version), and the two abbreviations for "Nation" separated out into two columns, plus all the other columns unmodified. Assign the resulting tibbles to new names different from the original ones.

```
players_new <- tibble(players %>% mutate(Player = strsplit(Player, "\\\\") %>% sapply(function(x) x[1]))
print(players_new)
```

```
## # A tibble: 522 x 30
##       Rk Player Nation_Abbr_1 Nation_Abbr_2 Pos   Squad   Age  Born    MP Starts
##    <dbl> <chr>  <chr>         <chr>         <chr> <chr> <dbl> <dbl> <dbl>  <dbl>
##  1     1 Patri~ nl            NED           DF    Crys~    28  1990    29     29
##  2     2 Max A~ eng           ENG           DF    Norw~    19  2000    36     36
##  3     3 Tammy~ eng           ENG           FW    Chel~    21  1997    34     25
##  4     4 Che A~ eng           ENG           FW    Sout~    23  1996    30     12
##  5     5 Adrián es            ESP           GK    Live~    32  1987    11      9
```

```
##  6     6 Sergi~ ar            ARG           FW    Manc~    31  1988    24     18
##  7     7 Albia~ ch            SUI           MFFW  West~    22  1997     9      0
##  8     8 Natha~ nl            NED           DF    Bour~    24  1995    29     29
##  9     9 Marc ~ eng           ENG           MFDF  Leic~    29  1989    20      9
## 10    10 Toby ~ be            BEL           DF    Tott~    30  1989    33     33
## # i 512 more rows
## # i 20 more variables: Min <dbl>, Gls <dbl>, Ast <dbl>, PK <dbl>, PKatt <dbl>,
## #   CrdY <dbl>, CrdR <dbl>, `Gls Per 90 Minutes` <dbl>,
## #   `Ast Per 90 Minutes` <dbl>, `G+A Per 90 Minutes` <dbl>,
## #   `G-PK Per 90 Minutes` <dbl>, `G+A-PK Per 90 Minutes` <dbl>, xG <dbl>,
## #   npxG <dbl>, xA <dbl>, `xG Per 90 Minutes` <dbl>, `xA Per 90 Minutes` <dbl>,
## #   `xG+xA Per 90 Minutes` <dbl>, `npxG Per 90 Minutes` <dbl>, ...
```

```
goalkeepers_new <- tibble(goalkeepers %>% mutate(Player = strsplit(Player, "\\\\") %>% sapply(function(
print(goalkeepers_new)
```

```
## # A tibble: 37 x 25
##       Rk Player Nation_Abbr_1 Nation_Abbr_2 Pos   Squad   Age  Born    MP Starts
##    <dbl> <chr>  <chr>         <chr>         <chr> <chr> <dbl> <dbl> <dbl>  <dbl>
##  1     1 Adrián es            ESP           GK    Live~    32  1987    11      9
##  2     2 Aliss~ br            BRA           GK    Live~    26  1992    29     29
##  3     3 Kepa ~ es            ESP           GK    Chel~    24  1994    33     33
##  4     4 Claud~ cl            CHI           GK    Manc~    36  1983     4      3
##  5     5 Willy~ ar            ARG           GK    Chel~    37  1981     5      5
##  6     6 Marti~ sk            SVK           GK    Newc~    30  1989    38     38
##  7     7 Eders~ br            BRA           GK    Manc~    25  1993    35     35
##  8     8 Łukas~ pl            POL           GK    West~    34  1985    25     25
##  9     9 Ralf ~ de            GER           GK    Norw~    30  1988     1      1
## 10    10 Ben F~ eng           ENG           GK    Watf~    36  1983    38     38
## # i 27 more rows
## # i 15 more variables: Min <dbl>, GA <dbl>, GA90 <dbl>, SoTA <dbl>,
## #   Saves <dbl>, `Save%` <dbl>, W <dbl>, D <dbl>, L <dbl>, CS <dbl>,
## #   `CS%` <dbl>, PKatt <dbl>, PKA <dbl>, PKsv <dbl>, PKm <dbl>
```

5. Find, for each team, the mean player age, the age of the youngest player and the age of the oldest player. Use this info to find the name(s) of the youngest player(s) for each team. Your output for the latter, separate from the output of the first part, should only contain the team name, the minimum age, and the name of the player.

```
age <- players %>% group_by(Squad) %>% summarize(mean_age = mean(na.omit(Age)), youngest_player = min(Ag
print(age)
```

```
## # A tibble: 20 x 4
##    Squad           mean_age youngest_player oldest_player
##    <chr>              <dbl>           <dbl>         <dbl>
##  1 Arsenal             24.6              17            33
##  2 Aston Villa         25.8              18            36
##  3 Bournemouth         24.7              20            34
##  4 Brighton            25.6              18            35
##  5 Burnley             27.4              17            34
##  6 Chelsea             24.1              17            37
```

```
##  7 Crystal Palace      27.4             17           33
##  8 Everton             25.2             17           34
##  9 Leicester City      25.7             NA           NA
## 10 Liverpool           25.5             16           33
## 11 Manchester City     25.7             17           36
## 12 Manchester Utd      23.9             17           34
## 13 Newcastle Utd       25.8             19           30
## 14 Norwich City        25.5             NA           NA
## 15 Sheffield Utd       26.8             21           36
## 16 Southampton         24.2             19           32
## 17 Tottenham           24.6             17           32
## 18 Watford             27.2             17           36
## 19 West Ham            27.0             18           34
## 20 Wolves              24.4             19           32
```

Comment: We found that there were some null values which lead to wrong output for the Squad->Leicester City. Therefore, we omit the null values of the entire data. This is one of the possible ways of handling explicit null values. Later, group the dataset by Squad and find the mean, minimum and maximum age. Note that we do not have data for players age for Leicester City.

```
youngest_players_info <- players %>% inner_join(age, by ="Squad") %>% filter(Age == youngest_player) %>%

print(youngest_players_info)
```

```
## # A tibble: 24 x 3
##    Squad            Age Player
##    <chr>          <dbl> <chr>
##  1 Arsenal           17 "Bukayo Saka\\Bukayo-Saka"
##  2 Aston Villa       18 "Indiana Vassilev\\Indiana-Vassilev"
##  3 Bournemouth       20 "Lloyd Kelly\\Lloyd-Kelly"
##  4 Bournemouth       20 "Mark Travers\\Mark-Travers"
##  5 Brighton          18 "Tariq Lamptey\\Tariq-Lamptey"
##  6 Burnley           17 "Max Thompson\\Max-Thompson"
##  7 Chelsea           17 "Faustino Anjorin\\Faustino-Anjorin"
##  8 Chelsea           17 "Armando Broja\\Armando-Broja"
##  9 Crystal Palace    17 "Brandon Pierrick\\Brandon-Pierrick"
## 10 Everton           17 "Jarrad Branthwaite\\Jarrad-Branthwaite"
## # i 14 more rows
```

Comment: Since there are multiple players in the same team/Squad with same minimum age we try to print them all in the order of the Squad names.

6. Add the team performance numbers (specifically these ones: Possession, Assists, Penalty Kicks scored, Save percentage, and Clean sheet percentage) to the standings table. Add also the age statistics found in Q5 above to the standings table.

```
columns_to_include1 <- c("Squad","Poss", "Ast", "PK")
columns_to_include2 <- c("Squad","Save%", "CS%")
# Adding team performance number to the "standings" dataset.
```

```r
standings_with_team_perf <- left_join(standings, teamstats %>% select(all_of(columns_to_include1)), by =
# Adding team performance number to the "standings" dataset.
standings_with_team_perf <- left_join(standings_with_team_perf, teamgoalkeeping %>% select(all_of(colum
# Adding the age statistics found in Q5 to the "standings" dataset.
standings <- left_join(standings_with_team_perf, age, by = "Squad")

print(standings)
```

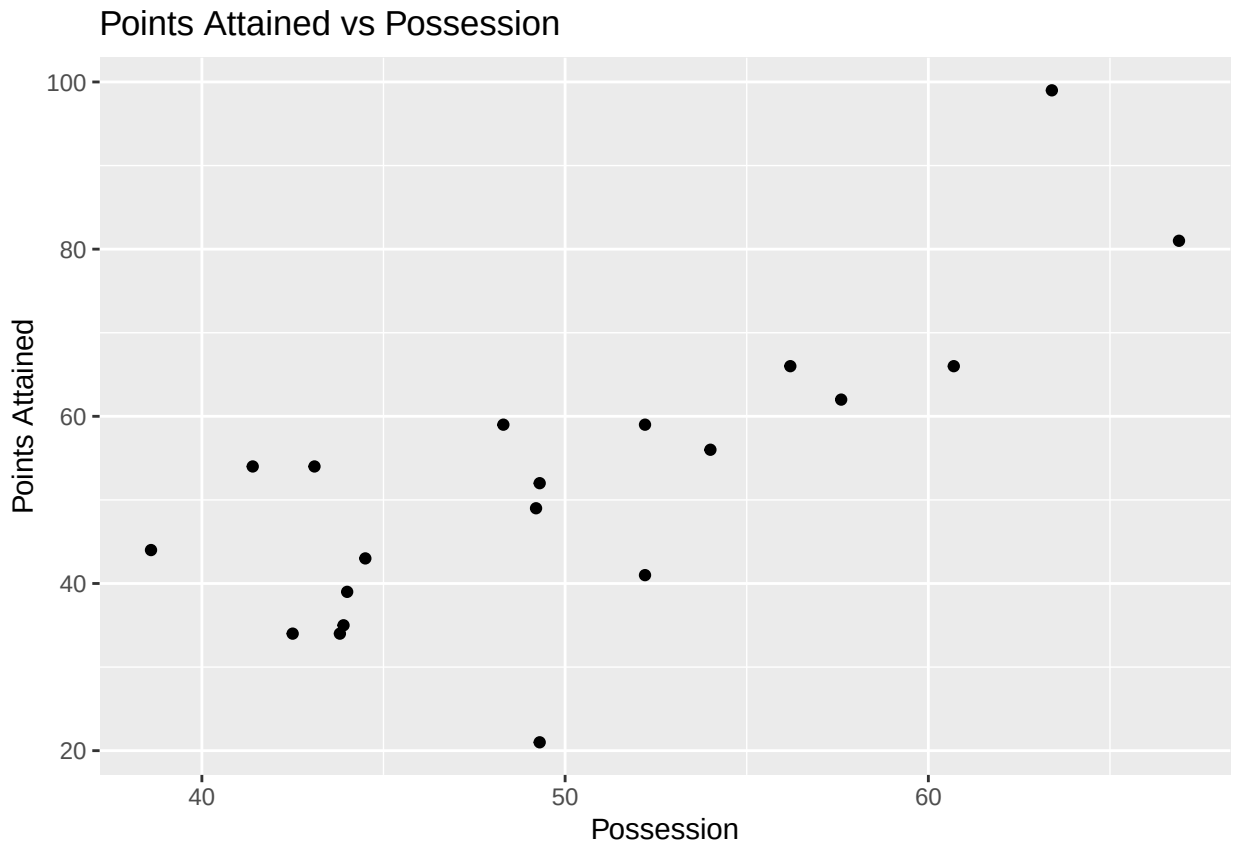
```
## # A tibble: 20 x 26
##       Rk Squad          MP     W     D     L    GF    GA GDiff   Pts    xG   xGA
##    <dbl> <chr>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1 Liverpool      38    32     3     3    85    33    52    99  71.5  40
##  2     2 Manchester~    38    26     3     9   102    35    67    81  93    34.7
##  3     3 Manchester~    38    18    12     8    66    36    30    66  59.4  37.4
##  4     4 Chelsea        38    20     6    12    69    54    15    66  66.6  37.9
##  5     5 Leicester ~    38    18     8    12    67    41    26    62  61.5  44.5
##  6     6 Tottenham      38    16    11    11    61    47    14    59  46.1  52
##  7     7 Wolves         38    15    14     9    51    40    11    59  47.1  34.8
##  8     8 Arsenal        38    14    14    10    56    48     8    56  49.2  56.6
##  9     9 Sheffield ~    38    14    12    12    39    39     0    54  41.5  47.9
## 10    10 Burnley        38    15     9    14    43    50    -7    54  43.9  48.3
## 11    11 Southampton    38    15     7    16    51    60    -9    52  53.9  53.1
## 12    12 Everton        38    13    10    15    44    56   -12    49  49.3  48.3
## 13    13 Newcastle ~    38    11    11    16    38    58   -20    44  33.1  58.3
## 14    14 Crystal Pa~    38    11    10    17    31    50   -19    43  34    51.3
## 15    15 Brighton       38     9    14    15    39    54   -15    41  41.2  54.5
## 16    16 West Ham       38    10     9    19    49    62   -13    39  46.2  61.1
## 17    17 Aston Villa    38     9     8    21    41    67   -26    35  40    65.9
## 18    18 Bournemouth    38     9     7    22    40    65   -25    34  42.7  57.5
## 19    19 Watford        38     8    10    20    36    64   -28    34  45.1  57.3
## 20    20 Norwich Ci~    38     5     6    27    26    75   -49    21  38    61.9
## # i 14 more variables: xGDiff <dbl>, `xGDiff/90` <dbl>, Attendance <dbl>,
## #   `Top Team Scorer` <chr>, Goalkeeper <chr>, Notes <chr>, Poss <dbl>,
## #   Ast <dbl>, PK <dbl>, `Save%` <dbl>, `CS%` <dbl>, mean_age <dbl>,
## #   youngest_player <dbl>, oldest_player <dbl>
```

Then make the following plots (A vs B means A is on the y axis):

(a) Points attained vs Possession

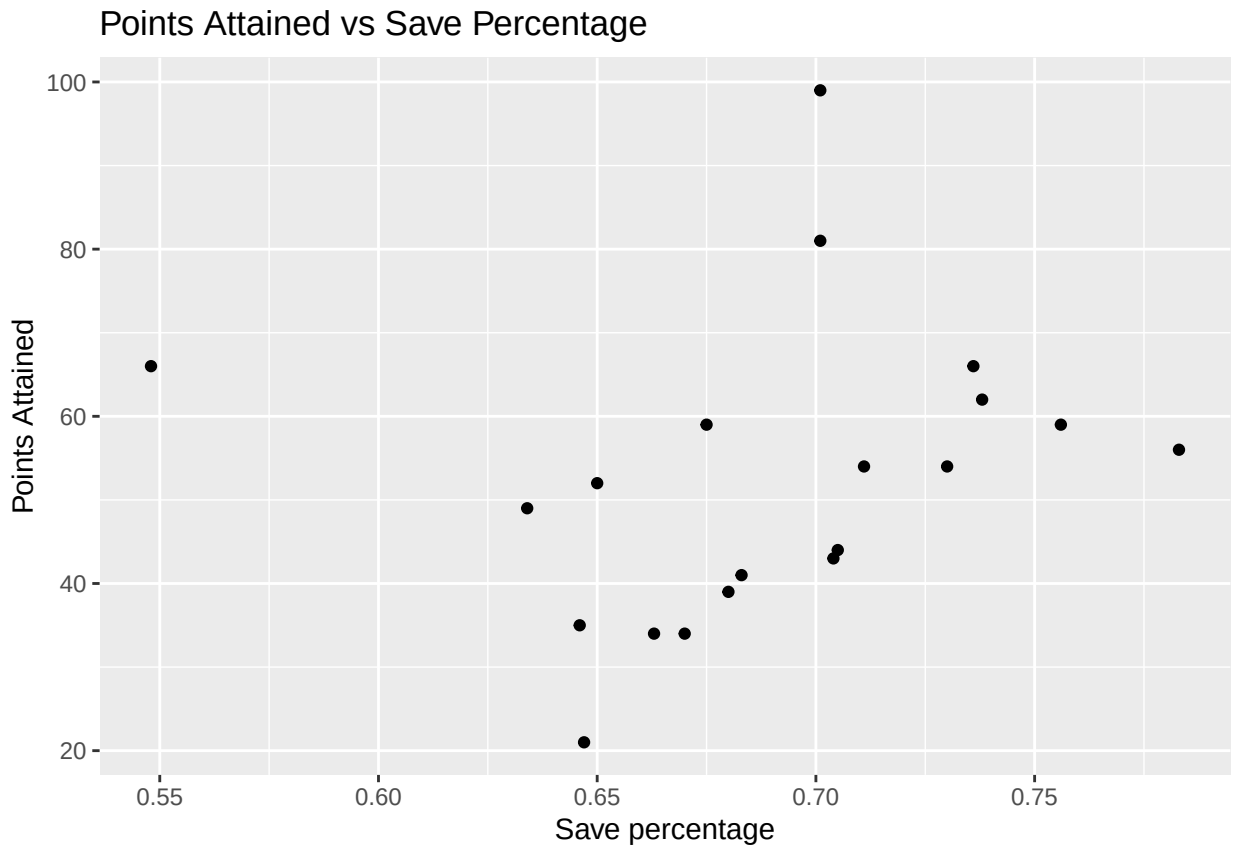
```r
qplot(x = Poss, y = Pts, data = standings, xlab = "Possession", ylab = "Points Attained", main = "Points
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



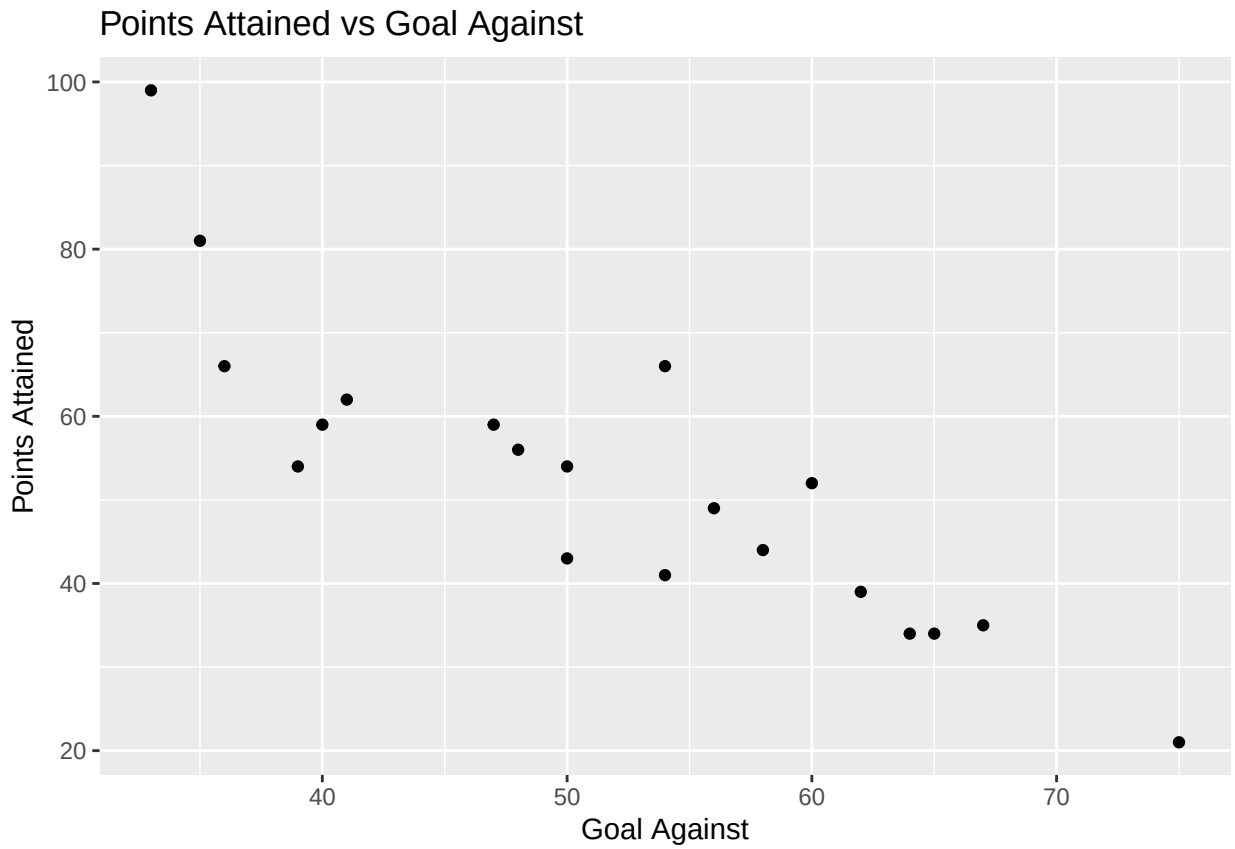
(b) Points attained vs Save percentage

```
qplot(x = `Save%`, y = Pts, data = standings, xlab = "Save percentage", ylab = "Points Attained", main =
```



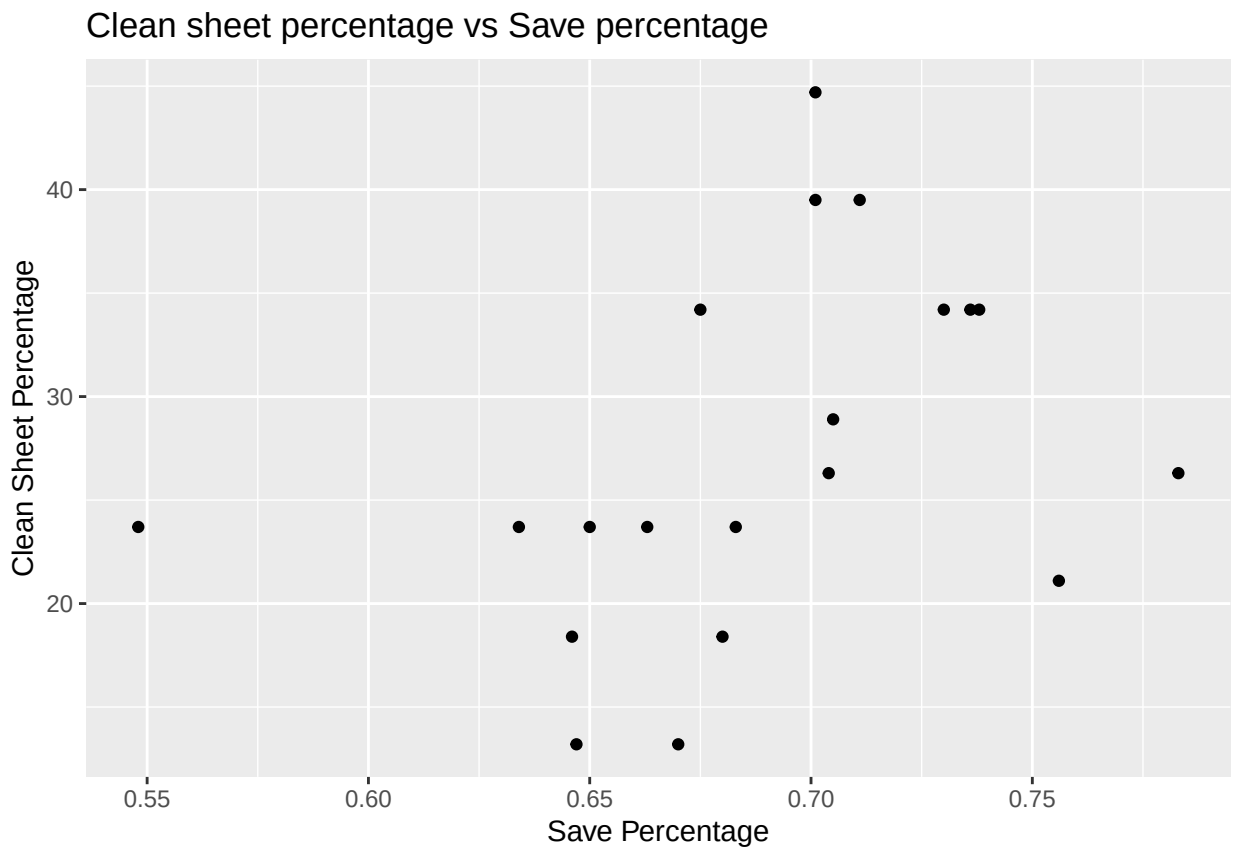
(c) Points attained vs Goal against

```
qplot(x = `GA`, y = Pts, data = standings, xlab = "Goal Against", ylab = "Points Attained", main = "Poi
```



(d) Clean sheet percentage vs Save percentage

```
qplot(x = `Save%`, y = `CS%`, data = standings, xlab = "Save Percentage", ylab = "Clean Sheet Percentag
```



7. Identify the names of the top 3 referees that refereed the most games. Your R output should only show three referees. Obtain all the games they refereed. Your output for this should only contain columns for the week and date the game was played, name of the home and away teams, and the name of the referee.

```
referee <- table(scoresfixtures$Referee)
sorted_referee <- sort(referee, decreasing = TRUE)
top_3_referees <- names(sorted_referee[1:3])
top_3_referee <- data.frame(Referee = top_3_referees)
print(top_3_referee)
```

```
##           Referee
## 1  Anthony Taylor
## 2 Martin Atkinson
## 3  Michael Oliver
```

```
columns_to_include1 <- c("Wk","Date", "Home", "Away", "Referee")
top_3_referee_table <- left_join(top_3_referee, scoresfixtures %>% select(all_of(columns_to_include1)),
print(top_3_referee_table)
```

```
##            Referee Wk       Date            Home            Away
## 1   Anthony Taylor  1 2019-08-11  Manchester Utd         Chelsea
## 2   Anthony Taylor  2 2019-08-17        Brighton        West Ham
```

```
## 3   Anthony Taylor  3 2019-08-24       Liverpool         Arsenal
## 4   Anthony Taylor  4 2019-09-01         Everton          Wolves
## 5   Anthony Taylor  5 2019-09-15         Watford         Arsenal
## 6   Anthony Taylor  6 2019-09-22        West Ham  Manchester Utd
## 7   Anthony Taylor  7 2019-09-28   Sheffield Utd       Liverpool
## 8   Anthony Taylor  9 2019-10-19  Crystal Palace Manchester City
## 9   Anthony Taylor 10 2019-10-27       Liverpool       Tottenham
## 10  Anthony Taylor 11 2019-11-02         Watford         Chelsea
## 11  Anthony Taylor 12 2019-11-10          Wolves     Aston Villa
## 12  Anthony Taylor 13 2019-11-23         Everton    Norwich City
## 13  Anthony Taylor 15 2019-12-03  Crystal Palace     Bournemouth
## 14  Anthony Taylor 16 2019-12-07 Manchester City  Manchester Utd
## 15  Anthony Taylor 18 2019-12-22       Tottenham         Chelsea
## 16  Anthony Taylor 19 2019-12-26         Everton         Burnley
## 17  Anthony Taylor 20 2019-12-29       Liverpool          Wolves
## 18  Anthony Taylor 22 2020-01-11  Manchester Utd    Norwich City
## 19  Anthony Taylor 23 2020-01-19         Burnley  Leicester City
## 20  Anthony Taylor 25 2020-02-01     Bournemouth     Aston Villa
## 21  Anthony Taylor 26 2020-02-17         Chelsea  Manchester Utd
## 22  Anthony Taylor 28 2020-02-29        West Ham     Southampton
## 23  Anthony Taylor 29 2020-03-07  Crystal Palace         Watford
## 24  Anthony Taylor 28 2020-06-17 Manchester City         Arsenal
## 25  Anthony Taylor 30 2020-06-20        West Ham          Wolves
## 26  Anthony Taylor 31 2020-06-24  Manchester Utd   Sheffield Utd
## 27  Anthony Taylor 32 2020-07-02 Manchester City       Liverpool
## 28  Anthony Taylor 34 2020-07-07         Watford    Norwich City
## 29  Anthony Taylor 35 2020-07-12          Wolves         Everton
## 30  Anthony Taylor 36 2020-07-16         Everton     Aston Villa
## 31  Anthony Taylor 37 2020-07-19       Tottenham  Leicester City
## 32  Anthony Taylor 38 2020-07-26   Newcastle Utd       Liverpool
## 33 Martin Atkinson  1 2019-08-11   Newcastle Utd         Arsenal
## 34 Martin Atkinson  2 2019-08-17     Aston Villa     Bournemouth
## 35 Martin Atkinson  3 2019-08-24    Norwich City         Chelsea
## 36 Martin Atkinson  4 2019-09-01         Arsenal       Tottenham
## 37 Martin Atkinson  5 2019-09-14  Manchester Utd  Leicester City
## 38 Martin Atkinson  6 2019-09-21   Newcastle Utd        Brighton
## 39 Martin Atkinson  8 2019-10-06         Arsenal     Bournemouth
## 40 Martin Atkinson  9 2019-10-20  Manchester Utd       Liverpool
## 41 Martin Atkinson 10 2019-10-27         Arsenal  Crystal Palace
## 42 Martin Atkinson 11 2019-11-03         Everton       Tottenham
## 43 Martin Atkinson 12 2019-11-09   Newcastle Utd     Bournemouth
## 44 Martin Atkinson 13 2019-11-23 Manchester City         Chelsea
## 45 Martin Atkinson 14 2019-11-30       Liverpool        Brighton
## 46 Martin Atkinson 16 2019-12-07         Watford  Crystal Palace
## 47 Martin Atkinson 17 2019-12-14     Southampton        West Ham
## 48 Martin Atkinson 18 2019-12-21     Bournemouth         Burnley
## 49 Martin Atkinson 19 2019-12-27          Wolves Manchester City
## 50 Martin Atkinson 21 2020-01-01   Newcastle Utd  Leicester City
## 51 Martin Atkinson 22 2020-01-11       Tottenham       Liverpool
## 52 Martin Atkinson 24 2020-01-21     Aston Villa         Watford
## 53 Martin Atkinson 25 2020-02-01   Newcastle Utd    Norwich City
## 54 Martin Atkinson 26 2020-02-16     Aston Villa       Tottenham
## 55 Martin Atkinson 27 2020-02-23  Manchester Utd         Watford
## 56 Martin Atkinson 28 2020-02-29        Brighton  Crystal Palace
```

```
## 57 Martin Atkinson 29 2020-03-07         Arsenal        West Ham
## 58 Martin Atkinson 30 2020-06-20        Brighton         Arsenal
## 59 Martin Atkinson 31 2020-06-24       Liverpool  Crystal Palace
## 60 Martin Atkinson 32 2020-07-01        West Ham         Chelsea
## 61 Martin Atkinson 34 2020-07-08   Sheffield Utd          Wolves
## 62 Martin Atkinson 35 2020-07-12     Aston Villa  Crystal Palace
## 63 Martin Atkinson 36 2020-07-17        West Ham         Watford
## 64 Martin Atkinson 38 2020-07-26  Leicester City  Manchester Utd
## 65  Michael Oliver  1 2019-08-09       Liverpool    Norwich City
## 66  Michael Oliver  2 2019-08-17 Manchester City       Tottenham
## 67  Michael Oliver  3 2019-08-23     Aston Villa         Everton
## 68  Michael Oliver  5 2019-09-14        Brighton         Burnley
## 69  Michael Oliver  6 2019-09-22         Chelsea       Liverpool
## 70  Michael Oliver  7 2019-09-28         Everton Manchester City
## 71  Michael Oliver  8 2019-10-05        West Ham  Crystal Palace
## 72  Michael Oliver 10 2019-10-26         Burnley         Chelsea
## 73  Michael Oliver 11 2019-11-02         Arsenal          Wolves
## 74  Michael Oliver 12 2019-11-10       Liverpool Manchester City
## 75  Michael Oliver 13 2019-11-23        West Ham       Tottenham
## 76  Michael Oliver 14 2019-11-30     Southampton         Watford
## 77  Michael Oliver 16 2019-12-08     Aston Villa  Leicester City
## 78  Michael Oliver 17 2019-12-15  Manchester Utd         Everton
## 79  Michael Oliver 19 2019-12-26  Leicester City       Liverpool
## 80  Michael Oliver 21 2020-01-01         Burnley     Aston Villa
## 81  Michael Oliver 22 2020-01-10   Sheffield Utd        West Ham
## 82  Michael Oliver 23 2020-01-18         Watford       Tottenham
## 83  Michael Oliver 24 2020-01-23          Wolves       Liverpool
## 84  Michael Oliver 25 2020-02-01        West Ham        Brighton
## 85  Michael Oliver 27 2020-02-22         Chelsea       Tottenham
## 86  Michael Oliver 28 2020-02-29         Watford       Liverpool
## 87  Michael Oliver 29 2020-03-09  Leicester City     Aston Villa
## 88  Michael Oliver 28 2020-06-17     Aston Villa   Sheffield Utd
## 89  Michael Oliver 31 2020-06-24          Wolves     Bournemouth
## 90  Michael Oliver 32 2020-06-28         Watford     Southampton
## 91  Michael Oliver 33 2020-07-04          Wolves         Arsenal
## 92  Michael Oliver 34 2020-07-08        West Ham         Burnley
## 93  Michael Oliver 35 2020-07-12       Tottenham         Arsenal
## 94  Michael Oliver 36 2020-07-16  Leicester City   Sheffield Utd
## 95  Michael Oliver 37 2020-07-21         Watford Manchester City
## 96  Michael Oliver 38 2020-07-26        West Ham     Aston Villa
```

8. Obtain a list of the games played by the top 3 scoring teams (i.e. the 3 teams that scored the most goals), showing only week, date, home and away teams and the result (final score).

```
top_3_scoring_teams <- teamstats %>% arrange(desc(Gls)) %>% slice_head(n = 3) %>% pull(Squad)
print(top_3_scoring_teams)
```

```
## [1] "Manchester City" "Liverpool"       "Chelsea"
```

```
columns_to_include1 <- c("Wk","Date", "Home", "Away", "Score")
result <- scoresfixtures %>% select(all_of(columns_to_include1)) %>% filter(Home %in% top_3_scoring_tea
print(result)
```

```
## # A tibble: 108 x 5
##       Wk Date       Home            Away            Score
##    <dbl> <date>     <chr>           <chr>           <chr>
##  1     1 2019-08-09 Liverpool       Norwich City    4-1
##  2     1 2019-08-10 West Ham        Manchester City 0-5
##  3     1 2019-08-11 Manchester Utd  Chelsea         4-0
##  4     2 2019-08-17 Southampton     Liverpool       1-2
##  5     2 2019-08-17 Manchester City Tottenham       2-2
##  6     2 2019-08-18 Chelsea         Leicester City  1-1
##  7     3 2019-08-24 Norwich City    Chelsea         2-3
##  8     3 2019-08-24 Liverpool       Arsenal         3-1
##  9     3 2019-08-25 Bournemouth     Manchester City 1-3
## 10     4 2019-08-31 Chelsea         Sheffield Utd   2-2
## # i 98 more rows
```

Reference: https://rdocumentation.org/packages/dplyr/versions/1.0.10/topics/slice

Reference: https://dplyr.tidyverse.org/reference/pull.html

9. Find the 3 dates with the most games played. Your R output should only show three dates. Then obtain all the games that were played on these days (you can keep all the columns).

```
dates <- table(scoresfixtures$Date)
sorted_dates <- sort(dates, decreasing = TRUE)
top_3_dates <- names(sorted_dates[1:3])
top_3_date <- data.frame(Date = top_3_dates)
print(top_3_date)
```

```
##         Date
## 1 2020-07-26
## 2 2019-12-26
## 3 2020-01-01
```

```
scoresfixtures$Date <- as.character(scoresfixtures$Date)
top_3_dates_table <- left_join(top_3_date, scoresfixtures, by = "Date")
print(top_3_dates_table)
```

```
##          Date Wk Day          Time            Home xG...6 Score xG...8
## 1  2020-07-26 38 Sun 16:00 (11:00)        West Ham    0.7   1-1    1.2
## 2  2020-07-26 38 Sun 16:00 (11:00)         Burnley    1.8   1-2    1.1
## 3  2020-07-26 38 Sun 16:00 (11:00)     Southampton    2.2   3-1    1.4
## 4  2020-07-26 38 Sun 16:00 (11:00)         Chelsea    1.3   2-0    0.2
## 5  2020-07-26 38 Sun 16:00 (11:00)  Crystal Palace    0.8   1-1    0.4
## 6  2020-07-26 38 Sun 16:00 (11:00)         Everton    2.2   1-3    1.8
## 7  2020-07-26 38 Sun 16:00 (11:00)         Arsenal    1.7   3-2    3.4
## 8  2020-07-26 38 Sun 16:00 (11:00) Manchester City    3.2   5-0    1.1
## 9  2020-07-26 38 Sun 16:00 (11:00)  Leicester City    1.4   0-2    1.9
## 10 2020-07-26 38 Sun 16:00 (11:00)   Newcastle Utd    0.4   1-3    0.6
## 11 2019-12-26 19 Thu 12:30 (07:30)       Tottenham    1.5   2-1    0.6
## 12 2019-12-26 19 Thu 15:00 (10:00)         Everton    1.6   1-0    0.4
## 13 2019-12-26 19 Thu 15:00 (10:00)     Bournemouth    0.9   1-1    1.5
```

```
## 14 2019-12-26 19 Thu 15:00 (10:00)   Sheffield Utd    2.1   1-1    0.7
## 15 2019-12-26 19 Thu 15:00 (10:00)     Aston Villa    0.6   1-0    1.1
## 16 2019-12-26 19 Thu 15:00 (10:00)  Crystal Palace    2.0   2-1    0.9
## 17 2019-12-26 19 Thu 15:00 (10:00)         Chelsea    0.5   0-2    0.7
## 18 2019-12-26 19 Thu 17:30 (12:30)  Manchester Utd    1.6   4-1    0.9
## 19 2019-12-26 19 Thu 20:00 (15:00)  Leicester City    0.1   0-4    3.1
## 20 2020-01-01 21 Wed 12:30 (07:30)        Brighton    0.9   1-1    1.6
## 21 2020-01-01 21 Wed 12:30 (07:30)         Burnley    2.0   1-2    0.9
## 22 2020-01-01 21 Wed 15:00 (10:00)         Watford    0.9   2-1    1.4
## 23 2020-01-01 21 Wed 15:00 (10:00)   Newcastle Utd    0.7   0-3    1.7
## 24 2020-01-01 21 Wed 15:00 (10:00)     Southampton    1.4   1-0    0.5
## 25 2020-01-01 21 Wed 17:30 (12:30)    Norwich City    1.6   1-1    1.5
## 26 2020-01-01 21 Wed 17:30 (12:30)        West Ham    1.8   4-0    0.3
## 27 2020-01-01 21 Wed 17:30 (12:30) Manchester City    1.6   2-1    1.2
## 28 2020-01-01 21 Wed 20:00 (15:00)         Arsenal    1.5   2-0    0.8
##              Away Attendance                                    Venue
## 1     Aston Villa         NA                         London Stadium
## 2         Brighton         NA                              Turf Moor
## 3    Sheffield Utd         NA                    St. Mary's Stadium
## 4           Wolves         NA                       Stamford Bridge
## 5        Tottenham         NA                         Selhurst Park
## 6      Bournemouth         NA                         Goodison Park
## 7          Watford         NA                       Emirates Stadium
## 8     Norwich City         NA                         Etihad Stadium
## 9   Manchester Utd         NA                    King Power Stadium
## 10       Liverpool         NA                       St. James' Park
## 11        Brighton      56308             Tottenham Hotspur Stadium
## 12         Burnley      39177                         Goodison Park
## 13         Arsenal      10234                      Vitality Stadium
## 14         Watford      30222                          Bramall Lane
## 15    Norwich City      41289                            Villa Park
## 16        West Ham      25462                         Selhurst Park
## 17     Southampton      40651                      Stamford Bridge
## 18   Newcastle Utd      73206                         Old Trafford
## 19       Liverpool      32211                    King Power Stadium
## 20         Chelsea      30559 The American Express Community Stadium
## 21     Aston Villa      19561                              Turf Moor
## 22          Wolves      20584                 Vicarage Road Stadium
## 23 Leicester City      52178                       St. James' Park
## 24       Tottenham      30976                    St. Mary's Stadium
## 25 Crystal Palace      27021                            Carrow Road
## 26     Bournemouth      59917                         London Stadium
## 27         Everton      54407                         Etihad Stadium
## 28 Manchester Utd      60328                       Emirates Stadium
##            Referee
## 1   Michael Oliver
## 2    Jonathan Moss
## 3     Peter Bankes
## 4   Stuart Attwell
## 5   Andre Marriner
## 6   Chris Kavanagh
## 7        Mike Dean
## 8     Craig Pawson
## 9  Martin Atkinson
```

```
## 10  Anthony Taylor
## 11    Graham Scott
## 12  Anthony Taylor
## 13  Stuart Attwell
## 14     David Coote
## 15  Chris Kavanagh
## 16  Andre Marriner
## 17   Jonathan Moss
## 18    Kevin Friend
## 19  Michael Oliver
## 20  Stuart Attwell
## 21  Michael Oliver
## 22     Andy Madley
## 23 Martin Atkinson
## 24       Mike Dean
## 25   Jonathan Moss
## 26    Graham Scott
## 27  Andre Marriner
## 28  Chris Kavanagh
```

10. Find the 3 dates with the most goals scored on those days (i.e. adding up all the goals scored in all the games played on that day). Your resulting tibble should only have 3 rows.

```
most_goals_scored <- tibble(scoresfixtures %>% separate(Score,into = c("Goal_Team_1", "Goal_Team_2"), s
scores <- as.integer(most_goals_scored$Goal_Team_1) + as.integer(most_goals_scored$Goal_Team_2)
score <- cbind(most_goals_scored, scores)

top_3_dates_max_score<- score %>% arrange(desc(scores)) %>% slice(1:3)
print(top_3_dates_max_score)
```

```
##   Wk Day       Date          Time            Home xG...6 Goal_Team_1
## 1 10 Fri 2019-10-25 20:00 (15:00)     Southampton    0.4           0
## 2  6 Sat 2019-09-21 15:00 (10:00) Manchester City    5.8           8
## 3 37 Wed 2020-07-22 20:15 (15:15)       Liverpool    1.4           5
##   Goal_Team_2 xG...8           Away Attendance              Venue
## 1           9    3.7 Leicester City      28762 St. Mary's Stadium
## 2           0    0.3        Watford      54273     Etihad Stadium
## 3           3    2.3        Chelsea         NA            Anfield
##          Referee scores
## 1 Andre Marriner      9
## 2      Mike Dean      8
## 3 Andre Marriner      8
```