Fantasy_Football

Kemon Lardas

2/20/2024

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.0     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
getwd() #find working directory
```

```
## [1] "C:/Users/13475/OneDrive/Documents/Side Stuff/Basics_of_R_notes"
```

```
players <- read_csv("2019projections.csv") #use the read_csv command to import data
```

```
## New names:
## Rows: 2577 Columns: 9
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (3): Player, Pos, Status dbl (6): ...1, Week, Team, Slot, Proj, Actual
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...1`
```

```
glimpse(players) #check the variables and observations of the fantasy football data
```

```
## Rows: 2,577
## Columns: 9
## $ ...1   <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1…
## $ Week   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ Team   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, …
## $ Player <chr> "Saquon Barkley", "Alvin Kamara", "James White", "Tyler Boyd", …
## $ Slot   <dbl> 2, 2, 23, 4, 0, 6, 20, 20, 23, 4, 20, 20, 20, 2, 4, 4, 2, 0, 20…
## $ Pos    <chr> "RB", "RB", "Flex", "WR", "QB", "TE", "Bench", "Bench", "Flex",…
## $ Status <chr> "ACTIVE", "ACTIVE", "ACTIVE", "ACTIVE", "ACTIVE", "ACTIVE", "AC…
## $ Proj   <dbl> 23.115309, 21.168970, 13.946106, 14.135564, 17.244130, 9.328158…
## $ Actual <dbl> 17.90, 23.90, 13.20, 14.30, 9.44, 16.70, 9.20, 7.50, 3.70, 22.8…
```

```
players
```

```
## # A tibble: 2,577 × 9
##     ...1  Week  Team Player                    Slot Pos   Status     Proj Actual
##    <dbl> <dbl> <dbl> <chr>                    <dbl> <chr> <chr>     <dbl>  <dbl>
##  1     0     1     1 Saquon Barkley               2 RB    ACTIVE    23.1   17.9
##  2     1     1     1 Alvin Kamara                 2 RB    ACTIVE    21.2   23.9
##  3     2     1     1 James White                 23 Flex  ACTIVE    13.9   13.2
##  4     3     1     1 Tyler Boyd                   4 WR    ACTIVE    14.1   14.3
##  5     4     1     1 Jared Goff                   0 QB    ACTIVE    17.2    9.44
##  6     5     1     1 Austin Hooper                6 TE    ACTIVE     9.33  16.7
##  7     6     1     1 Marquez Valdes-Scantling    20 Bench ACTIVE     8.85   9.2
##  8     7     1     1 Jordan Howard               20 Bench QUESTION…  8.84   7.5
##  9     8     1     1 Donte Moncrief              23 Flex  ACTIVE     9.86   3.7
## 10     9     1     1 Michael Gallup               4 WR    ACTIVE    10.0   22.8
## # ℹ 2,567 more rows
```

```
players |>
    distinct(Slot) #tried to find what the Slot variable meant but I could not figure it out
```

```
## # A tibble: 7 × 1
##    Slot
##   <dbl>
## 1     2
## 2    23
## 3     4
## 4     0
## 5     6
## 6    20
## 7    21
```

```
tidy_data <- players |>
    select(!(...1 | Slot))  #removed the row number and slot variable to focus on the interested data

tidy_data
```

```
## # A tibble: 2,577 × 7
##     Week  Team Player                    Pos   Status        Proj Actual
##    <dbl> <dbl> <chr>                     <chr> <chr>        <dbl>  <dbl>
##  1     1     1 Saquon Barkley            RB    ACTIVE       23.1   17.9
##  2     1     1 Alvin Kamara              RB    ACTIVE       21.2   23.9
##  3     1     1 James White               Flex  ACTIVE       13.9   13.2
##  4     1     1 Tyler Boyd                WR    ACTIVE       14.1   14.3
##  5     1     1 Jared Goff                QB    ACTIVE       17.2    9.44
##  6     1     1 Austin Hooper             TE    ACTIVE        9.33  16.7
##  7     1     1 Marquez Valdes-Scantling Bench ACTIVE        8.85   9.2
##  8     1     1 Jordan Howard             Bench QUESTIONABLE  8.84   7.5
##  9     1     1 Donte Moncrief            Flex  ACTIVE        9.86   3.7
## 10     1     1 Michael Gallup            WR    ACTIVE       10.0   22.8
## # ℹ 2,567 more rows
```

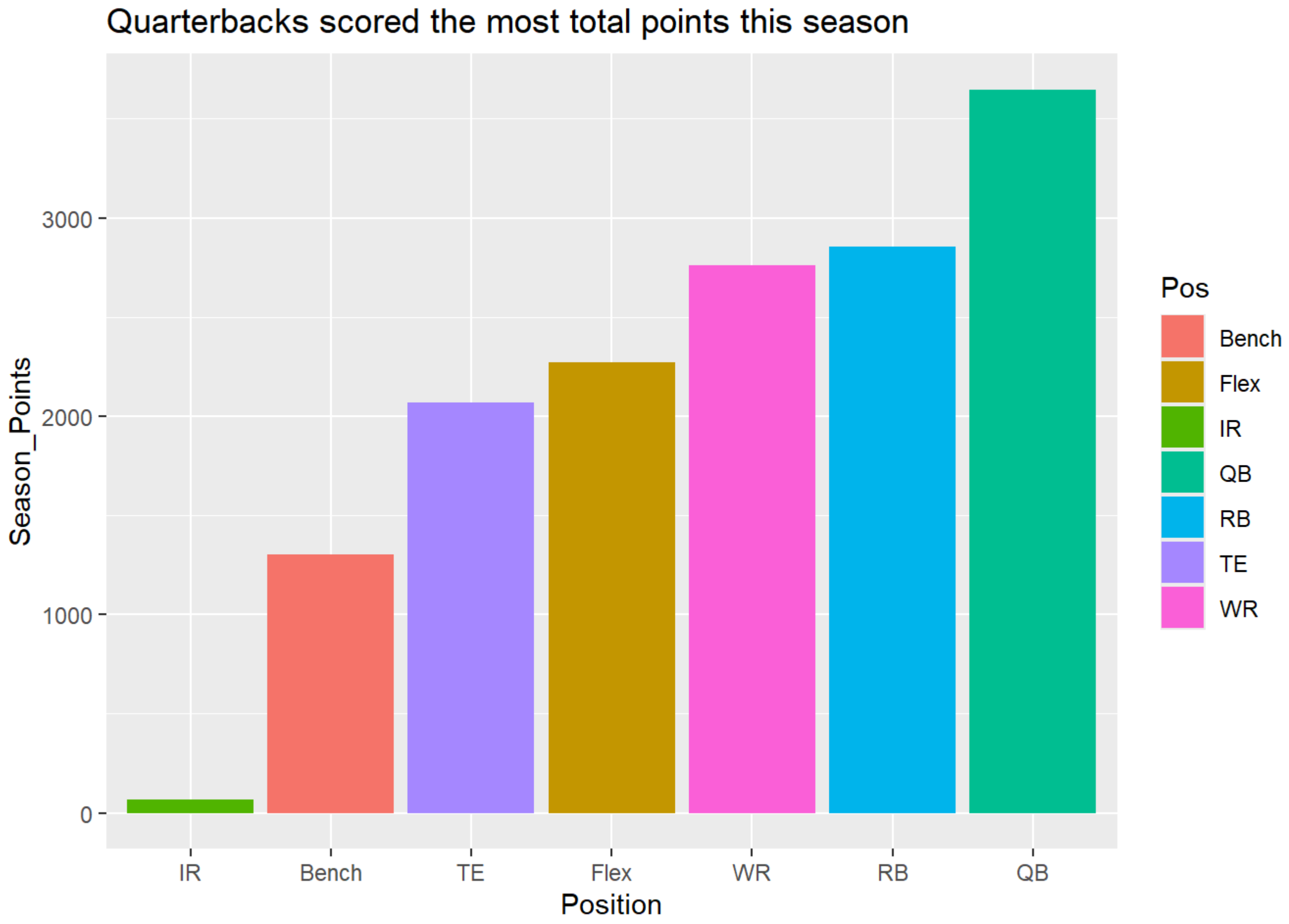
```
#which positions scored the most points during the season
tidy_data |>
    filter(Pos == "Bench") |>
    group_by(Week, Team) |>
    count(Pos) #average of 5 bench spots per team
```

```
## # A tibble: 192 × 4
## # Groups:   Week, Team [192]
##     Week  Team Pos       n
##    <dbl> <dbl> <chr> <int>
##  1     1     1 Bench     5
##  2     1     2 Bench     5
##  3     1     3 Bench     5
##  4     1     5 Bench     5
##  5     1     6 Bench     5
##  6     1     7 Bench     5
##  7     1     8 Bench     5
##  8     1     9 Bench     5
##  9     1    10 Bench     5
## 10     1    11 Bench     5
## # ℹ 182 more rows
```

```
most_pos_pts_scored <- tidy_data |>
    group_by(Pos)|>
    summarize(Total_points = sum(Actual, na.rm = TRUE)) |>
    mutate(Total_points = if_else(Pos == "Bench", Total_points/5, Total_points),
           Total_points = if_else(Pos == "Flex" | Pos == "RB" | Pos == "WR", Total_points/2, Tot
al_points)) |>
    arrange(desc(Total_points))

most_pos_pts_scored
```

```
## # A tibble: 7 × 2
##   Pos   Total_points
##   <chr>        <dbl>
## 1 QB          3647.
## 2 RB          2855.
## 3 WR          2759.
## 4 Flex        2272.
## 5 TE          2068.
## 6 Bench       1304.
## 7 IR            65.6
```

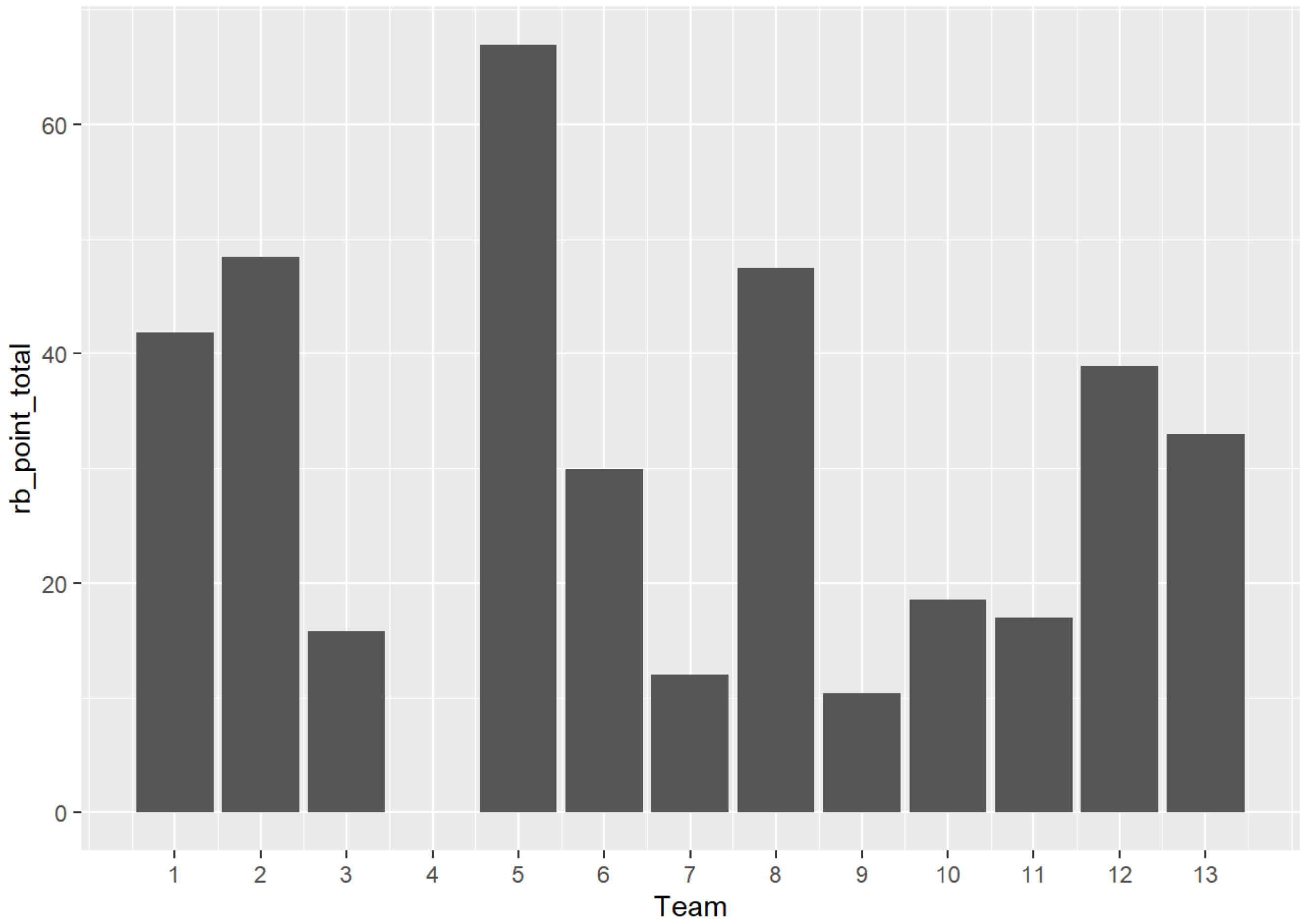
```
ggplot(most_pos_pts_scored, aes(fct_reorder(Pos, Total_points), Total_points)) +
    geom_col(aes(fill = Pos)) +
    labs(
        x = "Position",
        y = "Season_Points",
        title = "Quarterbacks scored the most total points this season"
    )
```



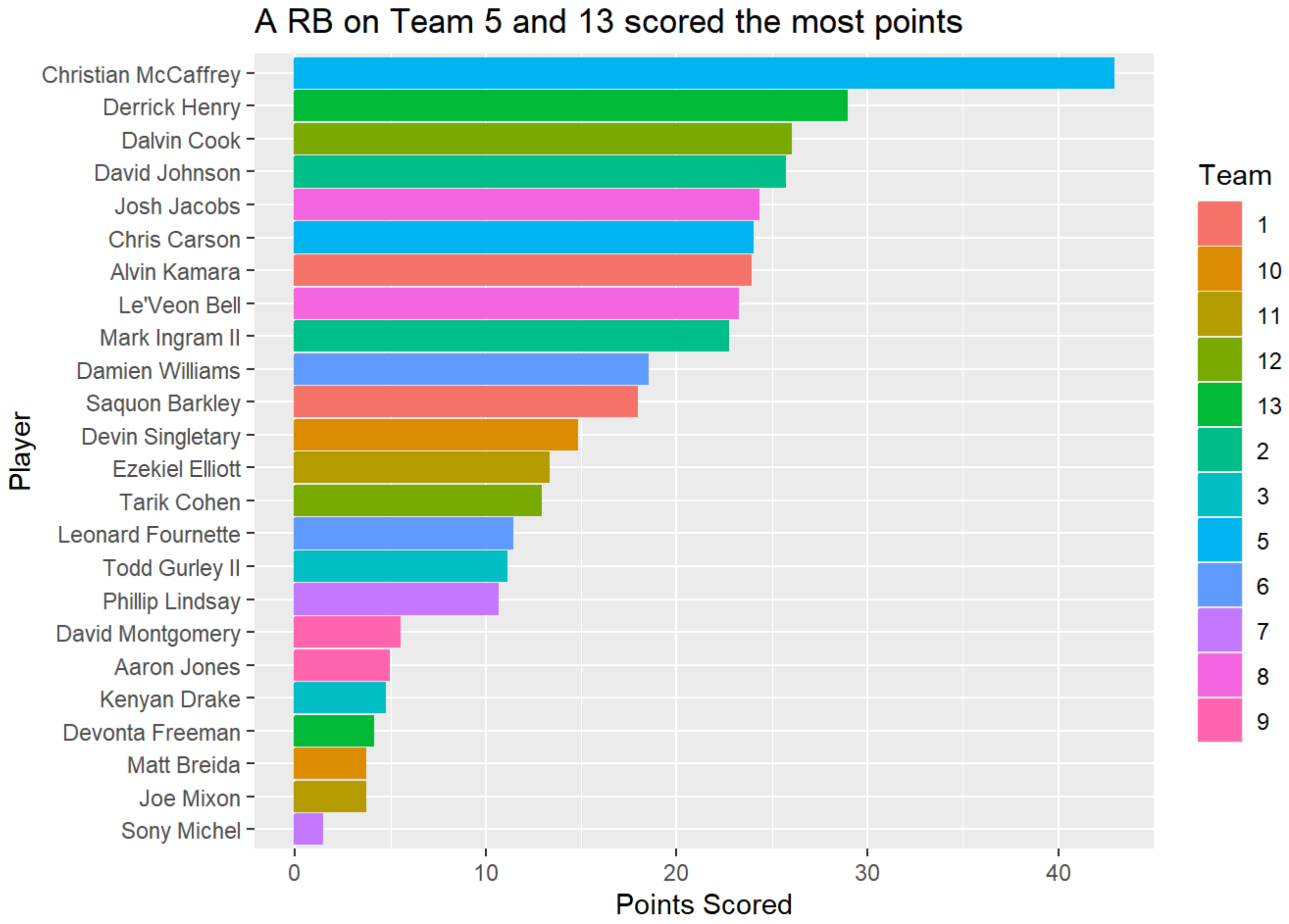
```
#top 5 scorers in week 1
week_one_best <- tidy_data |>
    filter(Week == 1) |>
    arrange(desc(Actual)) |>
    head(5)

#top 5 scorers throughout the season
season_best <- tidy_data |>
    group_by(Player)|>
    summarize(total_szn_points = sum(Actual, na.rm = TRUE))|>
    arrange(desc(total_szn_points)) |>
    head(5)

#only Christian McCaffrey and Lamar Jackson were top 5 in week 1 and the entire season
week_one_best |>
    inner_join(season_best)
```

```
## Joining with `by = join_by(Player)`
## # A tibble: 2 × 8
##    Week  Team Player               Pos   Status  Proj Actual total_szn_points
##   <dbl> <dbl> <chr>                <chr> <chr>  <dbl>  <dbl>            <dbl>
## 1     1     5 Christian McCaffrey RB    ACTIVE  21.0   42.9             448.
## 2     1     5 Lamar Jackson       Bench ACTIVE  18.9   33.6             416.
```

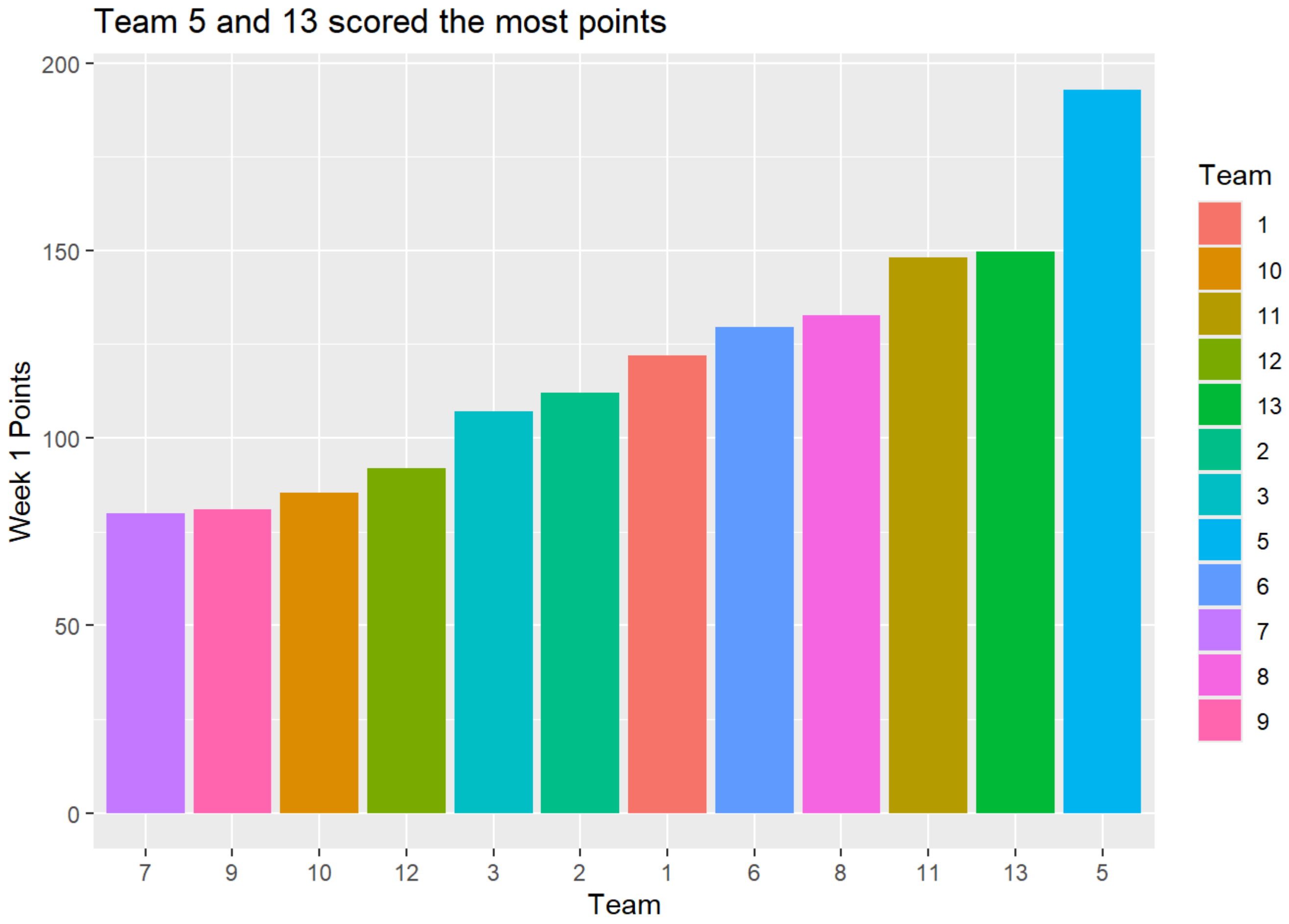
```
#visualize the rbs total points for each team in week 1
team_graph <- tidy_data |>
    filter(Pos == "RB", Week == 1) |>
    group_by(Team) |>
    summarise(rb_point_total = sum(Actual)) |>
    ggplot(aes(x = Team, y = rb_point_total)) +
    geom_col() +
    scale_x_continuous(breaks = seq(1,13, by = 1))

#visualize the rbs total points in week 1
player_graph <- tidy_data |>
    filter(Pos == "RB", Week == 1) |>
    arrange(desc(Actual)) |>
    mutate(Team = as.character(Team)) |>
    ggplot(aes(fct_reorder(Player, Actual), Actual, color = Team, fill = Team)) +
    geom_col() +
    coord_flip() +
    labs(
        x = "Player",
        y = "Points Scored",
        title = "A RB on Team 5 and 13 scored the most points"
    )

#Team with most points in week 1
best_week1_team <- tidy_data |>
    filter(Pos != "Bench" & Week == 1) |>
    group_by(Team) |>
    summarize(week_one_points = sum(Actual)) |>
    arrange(desc(week_one_points)) |>
    mutate(Team = as.character(Team))

best_week1_team_graph <-
    ggplot(best_week1_team, aes(fct_reorder(Team, week_one_points), week_one_points)) +
    geom_col(aes(fill = Team)) +
    labs(
        x = "Team",
        y = "Week 1 Points",
        title = "Team 5 and 13 scored the most points"
    )

team_graph
```



```
player_graph
```



```
best_week1_team_graph
```



```
#Team 5 and 13 seem to have scored very well b/c of their RBs
```

```
#Using ch. 5 notes to see if active players are more likely to have more proj points or question
able players

#Active players are projected more points than questionable players
tidy_data |>
    pivot_wider(
        names_from = Status,
        values_from = Proj
    ) |>
    select(!(OUT:SUSPENSION)) |>
    summarize(
        active_proj_points = mean(ACTIVE, na.rm = TRUE),
        questionable_proj_points = mean(QUESTIONABLE, na.rm = TRUE)
    )
```

```
## # A tibble: 1 × 2
##   active_proj_points questionable_proj_points
##                <dbl>                    <dbl>
## 1               11.1                     10.6
```

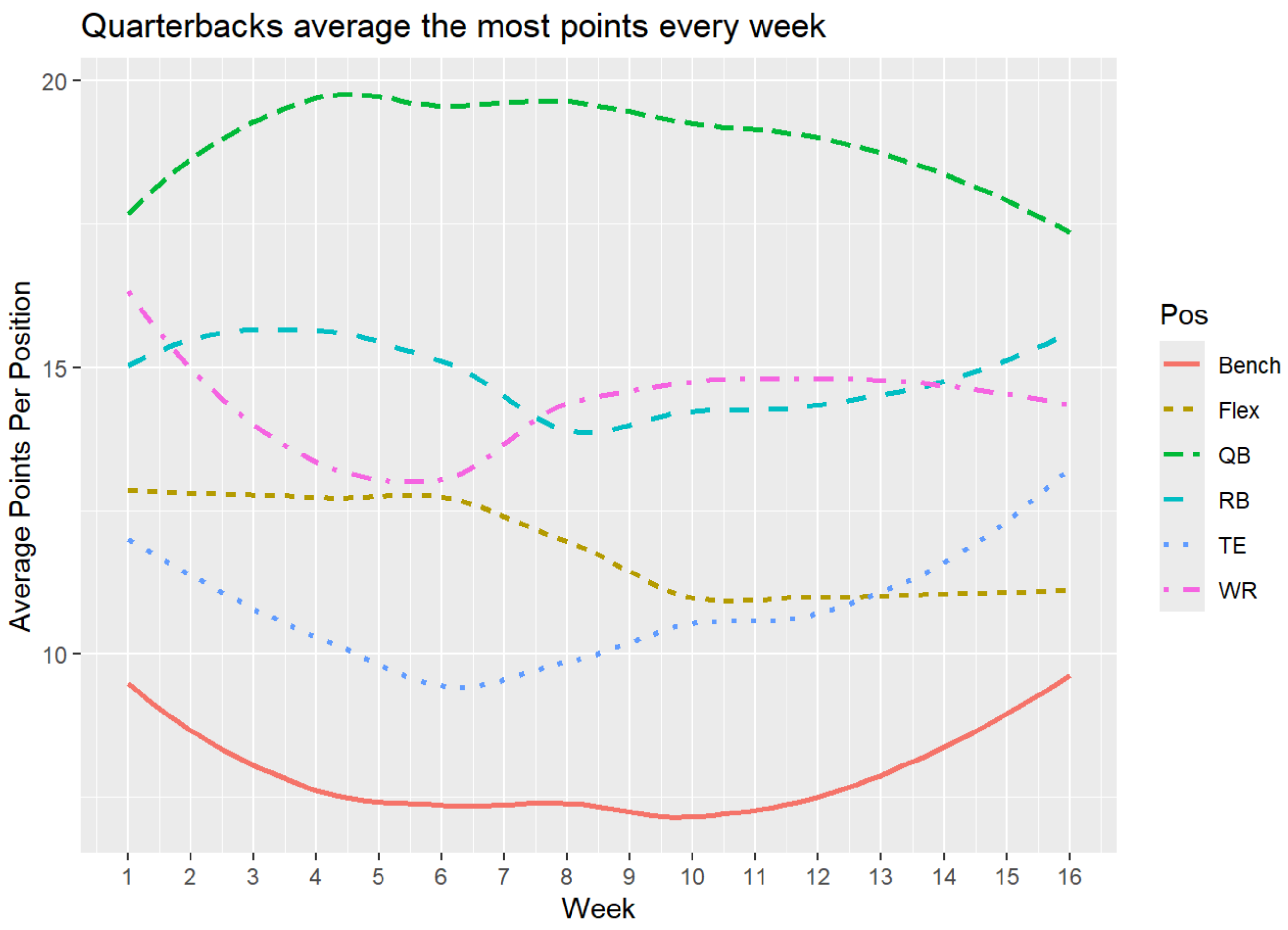
```
tidy_data |>
    filter(Status == "SUSPENSION")
```

```
## # A tibble: 2 × 7
##    Week  Team Player      Pos   Status      Proj Actual
##   <dbl> <dbl> <chr>       <chr> <chr>      <dbl>  <dbl>
## 1     8     2 Mark Walton RB    SUSPENSION  9.16    6.4
## 2     9     2 Mark Walton RB    SUSPENSION 12.2     7.1
```

```
#who even is mark walton?  A rb for the Dolphins that was suspended four games for violating the
conduct and substance abuse policy
```

```
#Which position averages the most points over all weeks
tidy_data |>
    group_by(Pos, Week) |>
    summarise(
        average_pos_pts = mean(Actual, na.rm = TRUE)
    ) |>
    filter(Pos != "IR") |>
    ggplot(aes(Week, average_pos_pts, linetype = Pos, color = Pos)) +
    geom_smooth(se = FALSE) +
    labs(
        y = "Average Points Per Position",
        title = "Quarterbacks average the most points every week"
    ) +
    scale_x_continuous("Week", breaks=seq(0,16,1))
```

```
## `summarise()` has grouped output by 'Pos'. You can override using the `.groups`
## argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
#Which team has the most active roster throughout the szn
active <- tidy_data |>
    filter(Status == "ACTIVE") |>
    group_by(Team) |>
    summarize(
        healthiest = length(Status)
    ) |>
    arrange(Team)

active
```

```
## # A tibble: 12 × 2
##     Team healthiest
##    <dbl>      <int>
##  1     1        176
##  2     2        189
##  3     3        192
##  4     5        154
##  5     6        118
##  6     7        196
##  7     8        137
##  8     9        185
##  9    10        174
## 10    11        152
## 11    12        163
## 12    13        159
```

```
#Who scored the most points throughout the season
most_pts <- tidy_data |>
    group_by(Team) |>
    filter(Pos != "Bench") |>
    summarize(
        most_pts_scored = sum(Actual, na.rm = TRUE)
    ) |>
    arrange(Team)

most_pts
```

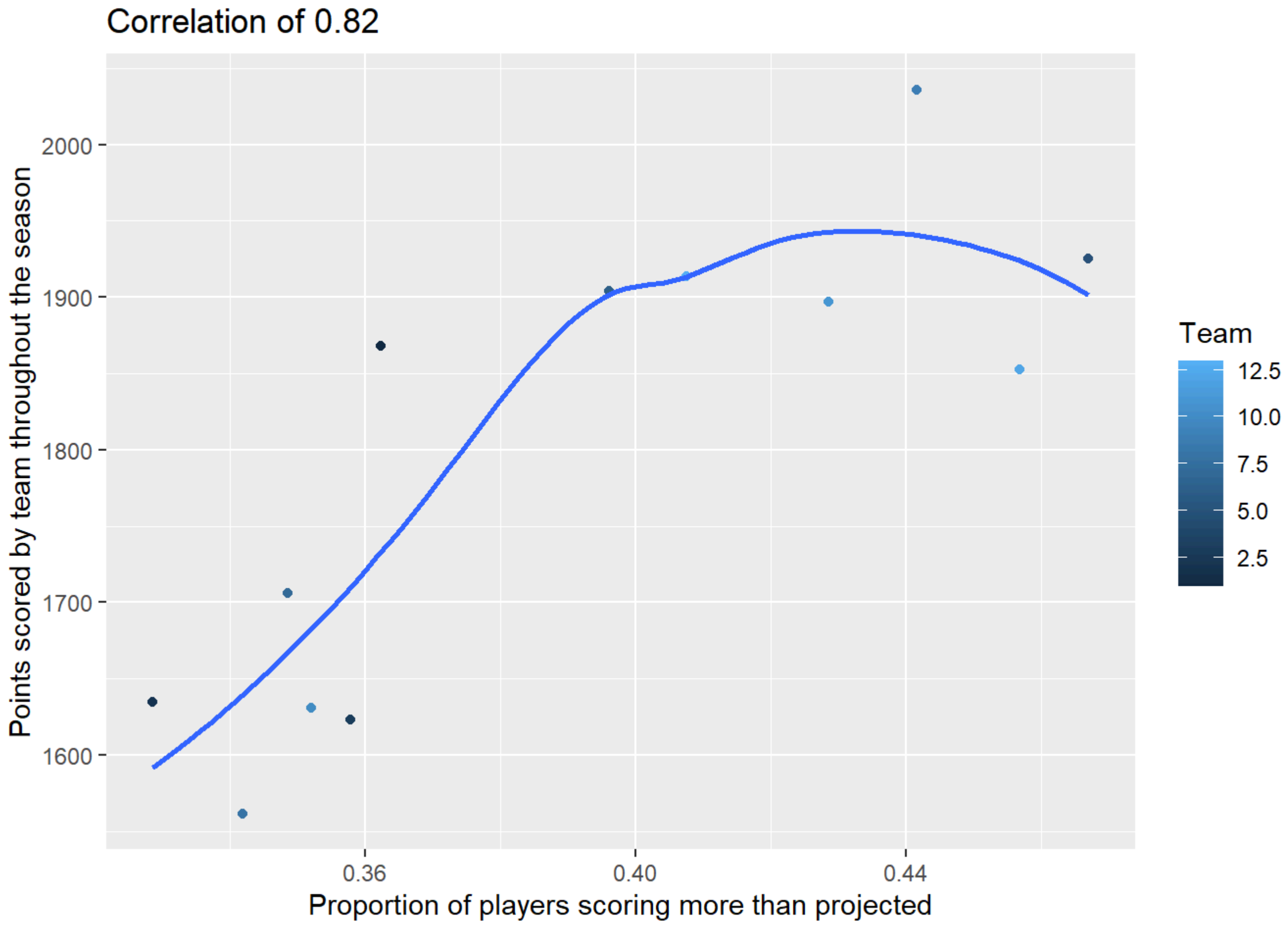
```
## # A tibble: 12 × 2
##     Team most_pts_scored
##    <dbl>           <dbl>
##  1     1           1868.
##  2     2           1635.
##  3     3           1623.
##  4     5           1925.
##  5     6           1904.
##  6     7           1706.
##  7     8           1562.
##  8     9           2036.
##  9    10           1631.
## 10    11           1897.
## 11    12           1853.
## 12    13           1913.
```

```
#Who had the best proportion of players scoring more than projected
prop_boom <- tidy_data |>
    mutate(boom = Actual > Proj) |>
    group_by(Team) |>
    summarize(
        prop = mean(boom, na.rm = TRUE)
    ) |>
    arrange(Team)

prop_boom
```

```
## # A tibble: 12 × 2
##     Team  prop
##    <dbl> <dbl>
##  1     1 0.362
##  2     2 0.329
##  3     3 0.358
##  4     5 0.467
##  5     6 0.396
##  6     7 0.348
##  7     8 0.342
##  8     9 0.442
##  9    10 0.352
## 10    11 0.429
## 11    12 0.457
## 12    13 0.408
```

```
#There is not a strong correlation between points scored and proportion of players scoring more than projected
prop_boom |>
    right_join(most_pts) |>
    ggplot(aes(prop, most_pts_scored)) +
    geom_point(aes(color = Team)) +
    geom_smooth(se = FALSE) +
    labs(
        x = "Proportion of players scoring more than projected",
        y = "Points scored by team throughout the season",
        title = "Correlation of 0.82"
    )
```

```
## Joining with `by = join_by(Team)`
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
cor(most_pts$most_pts_scored, active$healthiest) #correlation of -0.23
```

```
## [1] -0.229958
```

```
cor(most_pts$most_pts_scored, prop_boom$prop) #correlation of 0.82
```

```
## [1] 0.8150369
```

```
#much stronger correlation for proportion of players scoring more than projected than a healthy roster
```

```
#Split the First and Last name of each player into two columns
library("stringr") #load this package for string functions

#This seems to be the most effective way to split up their names since it splits there names up based on the first space
tidy_data[c("First_Name", "Last_Name")] <- str_split_fixed(tidy_data$Player, " ", 2)

#This was probably the second best option because we were able to extract both the first and last name, but unfortunately if there was an extra suffix like Jr, or II it would leave that out
tidy_data |>
    mutate(
        first_name = str_split(Player, " ", simplify = TRUE)[ ,1],
        last_name = str_split(Player, " ", simplify = TRUE)[ ,2]
    )
```

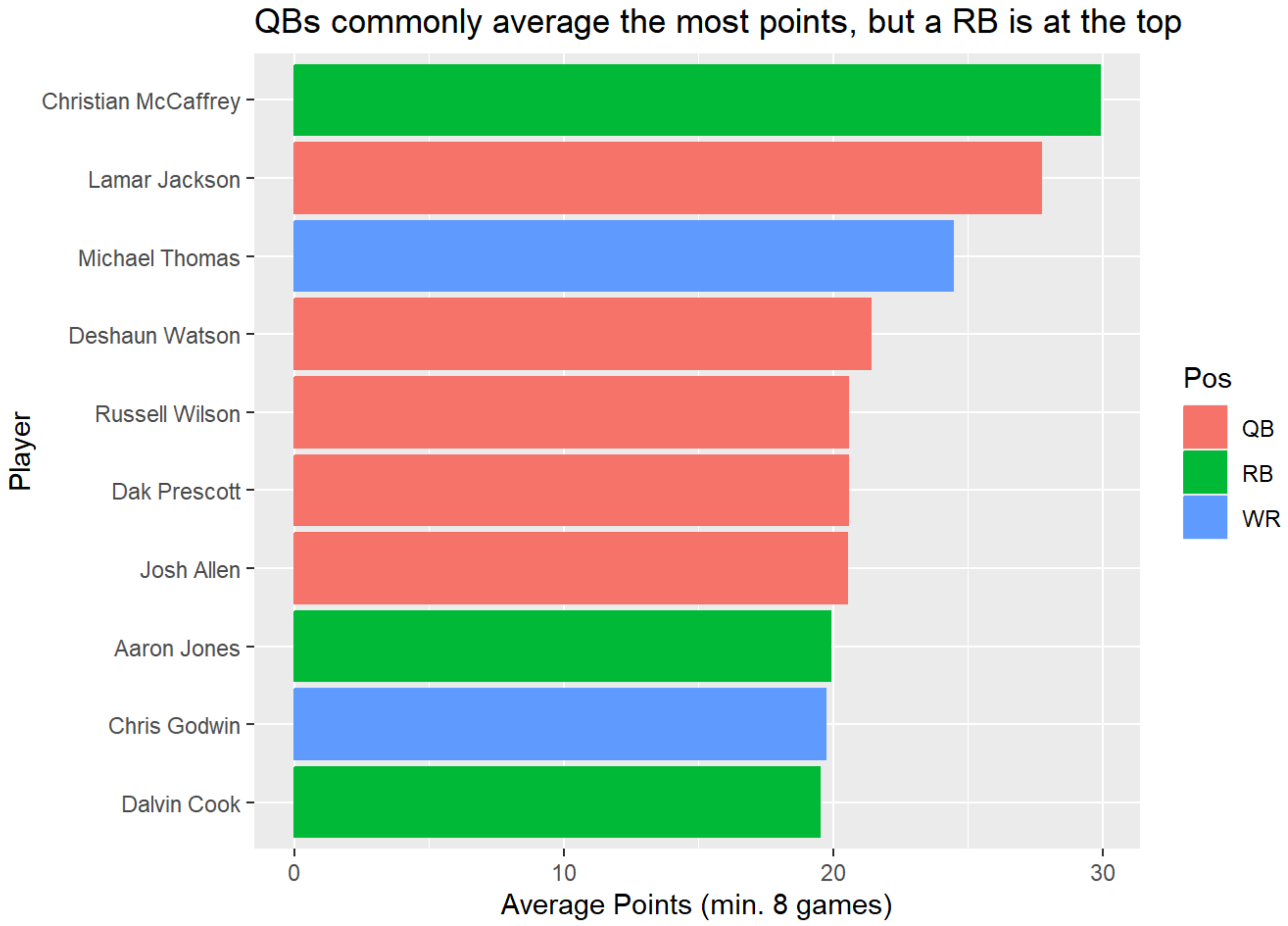
```
## # A tibble: 2,577 × 11
##     Week  Team Player  Pos   Status  Proj Actual First_Name Last_Name first_name
##    <dbl> <dbl> <chr>   <chr> <chr>  <dbl>  <dbl> <chr>      <chr>     <chr>
##  1     1     1 Saquon… RB    ACTIVE 23.1   17.9  Saquon     Barkley   Saquon
##  2     1     1 Alvin … RB    ACTIVE 21.2   23.9  Alvin      Kamara    Alvin
##  3     1     1 James … Flex  ACTIVE 13.9   13.2  James      White     James
##  4     1     1 Tyler … WR    ACTIVE 14.1   14.3  Tyler      Boyd      Tyler
##  5     1     1 Jared … QB    ACTIVE 17.2    9.44 Jared      Goff      Jared
##  6     1     1 Austin… TE    ACTIVE  9.33  16.7  Austin     Hooper    Austin
##  7     1     1 Marque… Bench ACTIVE  8.85   9.2  Marquez    Valdes-S… Marquez
##  8     1     1 Jordan… Bench QUEST…  8.84   7.5  Jordan     Howard    Jordan
##  9     1     1 Donte … Flex  ACTIVE  9.86   3.7  Donte      Moncrief  Donte
## 10     1     1 Michae… WR    ACTIVE 10.0   22.8  Michael    Gallup    Michael
## # ℹ 2,567 more rows
## # ℹ 1 more variable: last_name <chr>
```

```
#This was the least effective method because it automatically separated based on the .'s for abbreviated names ex: T.J. Hockenson had T as first name and J as last name
tidy_data |>
    separate(Player, c("First_Name", "Last_Name"), remove = FALSE) |>
    filter((str_length(First_Name) < 3)) |>
    filter((str_length(Last_Name) < 3))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 277 rows [7, 13, 17, 24, 26,
## 28, 39, 45, 51, 61, 80, 89, 107, 108, 109, 126, 150, 164, 167, 169, ...].
## # A tibble: 65 × 9
##     Week  Team Player         First_Name Last_Name Pos   Status      Proj Actual
##    <dbl> <dbl> <chr>          <chr>      <chr>     <chr> <chr>      <dbl>  <dbl>
##  1     1     1 T.J. Hockenson T          J         Bench QUESTIONA…  7.05   25.1
##  2     1     2 T.Y. Hilton    T          Y         Flex  ACTIVE     12.8    28.7
##  3     1     5 O.J. Howard    O          J         TE    ACTIVE     10.3     5.2
##  4     1    13 A.J. Green     A          J         Bench QUESTIONA…  0       0
##  5     2     1 T.J. Hockenson T          J         TE    QUESTIONA…  9.14    1.7
##  6     2     2 T.Y. Hilton    T          Y         Flex  ACTIVE     13.3    14.3
##  7     2     5 O.J. Howard    O          J         TE    ACTIVE      8.45    0
##  8     2    13 A.J. Green     A          J         IR    QUESTIONA…  0       0
##  9     3     1 T.J. Hockenson T          J         Bench QUESTIONA…  9.07    1.1
## 10     3     2 T.Y. Hilton    T          Y         Flex  ACTIVE     13.3    20.5
## # ℹ 55 more rows
```

```
tidy_data <- tidy_data |>
    select(!First_Name:Last_Name)
```

```
#which players averaged the most points with minimum 8 weeks played
most_avg_pts <- tidy_data |>
    group_by(Player) |>
    mutate(
        n = n(),
        avg_pts = mean(Actual, na.rm = TRUE)) |>
    filter(n >= 8) |>
    arrange(desc(avg_pts)) |>
    distinct(Player, .keep_all = TRUE) |>
    head(n = 10L)

most_avg_pts$Pos[2] = "QB"
most_avg_pts$Pos[6] = "QB"

ggplot(most_avg_pts, aes(reorder(Player, avg_pts), avg_pts, color = Pos, fill = Pos)) +
    geom_col() +
    coord_flip() +
    labs(
        title = "QBs commonly average the most points, but a RB is at the top",
        x = "Player",
        y = "Average Points (min. 8 games)"
    )
```



```
#which team scored the most points each week
total_team_points_per_week <- tidy_data |>
    filter(Pos != "Bench") |>
    group_by(Week, Team) |>
    summarize(
        total = sum(Actual)
    )
```

```
## `summarise()` has grouped output by 'Week'. You can override using the
## `.groups` argument.
```

```
total_team_points_per_week
```

```
## # A tibble: 192 × 3
## # Groups:   Week [16]
##     Week  Team total
##    <dbl> <dbl> <dbl>
##  1     1     1 122. 
##  2     1     2 112. 
##  3     1     3 107. 
##  4     1     5 193. 
##  5     1     6 129. 
##  6     1     7  80.0
##  7     1     8 133. 
##  8     1     9  81.0
##  9     1    10  85.3
## 10     1    11 148. 
## # ℹ 182 more rows
```

```
total_team_points_per_week |>
    group_by(Week) |>
    arrange(desc(total), .by_group = TRUE) |>
    mutate(
        best = max(total, na.rm = TRUE)
    ) |>
    distinct(best, .keep_all = TRUE) |>
    ungroup() |>
    group_by(Team) |>
    summarize(n = n()) |>
    arrange(desc(n))
```

```
## # A tibble: 10 × 2
##     Team     n
##    <dbl> <int>
##  1     9     4
##  2     1     2
##  3     5     2
##  4     6     2
##  5     2     1
##  6     7     1
##  7    10     1
##  8    11     1
##  9    12     1
## 10    13     1
```

```
most_pts |>
    arrange(desc(most_pts_scored))#team 9 had the most points in 4 different weeks (the most out
of all teams), and also scored the most points throughout the season
```

```
## # A tibble: 12 × 2
##     Team most_pts_scored
##    <dbl>           <dbl>
##  1     9           2036.
##  2     5           1925.
##  3    13           1913.
##  4     6           1904.
##  5    11           1897.
##  6     1           1868.
##  7    12           1853.
##  8     7           1706.
##  9     2           1635.
## 10    10           1631.
## 11     3           1623.
## 12     8           1562.
```

```
#use a function in a data frame to find the distribution of points scored amongst all players
best_player <- function(df, group_var, sum_var) {
    df |>
        group_by({{ group_var }}) |>
        summarize(tot_pts = sum({{ sum_var }}, na.rm = TRUE)) |>
        arrange(desc(tot_pts))
}

best_player(tidy_data, Player, Actual) #I idd this earlier but wanted to try this using a functi
on
```

```
## # A tibble: 250 × 2
##    Player              tot_pts
##    <chr>                 <dbl>
##  1 Christian McCaffrey    448.
##  2 Lamar Jackson          416.
##  3 Michael Thomas         367.
##  4 Deshaun Watson         321.
##  5 Russell Wilson         308.
##  6 Dak Prescott           308.
##  7 Aaron Jones            298.
##  8 Dalvin Cook            292.
##  9 Austin Ekeler          291.
## 10 Ezekiel Elliott        284.
## # ℹ 240 more rows
```

```
performances <- function(df, var) {
    df |>
        summarize(
            min = min({{ var }}, na.rm = TRUE),
            max = max({{ var }}, na.rm = TRUE),
            median = median({{ var }}, na.rm = TRUE),
            mean = mean({{ var }}, na.rm = TRUE)
        )
}

#Give the min, max, median, and mean
performances(tidy_data, Actual)
```

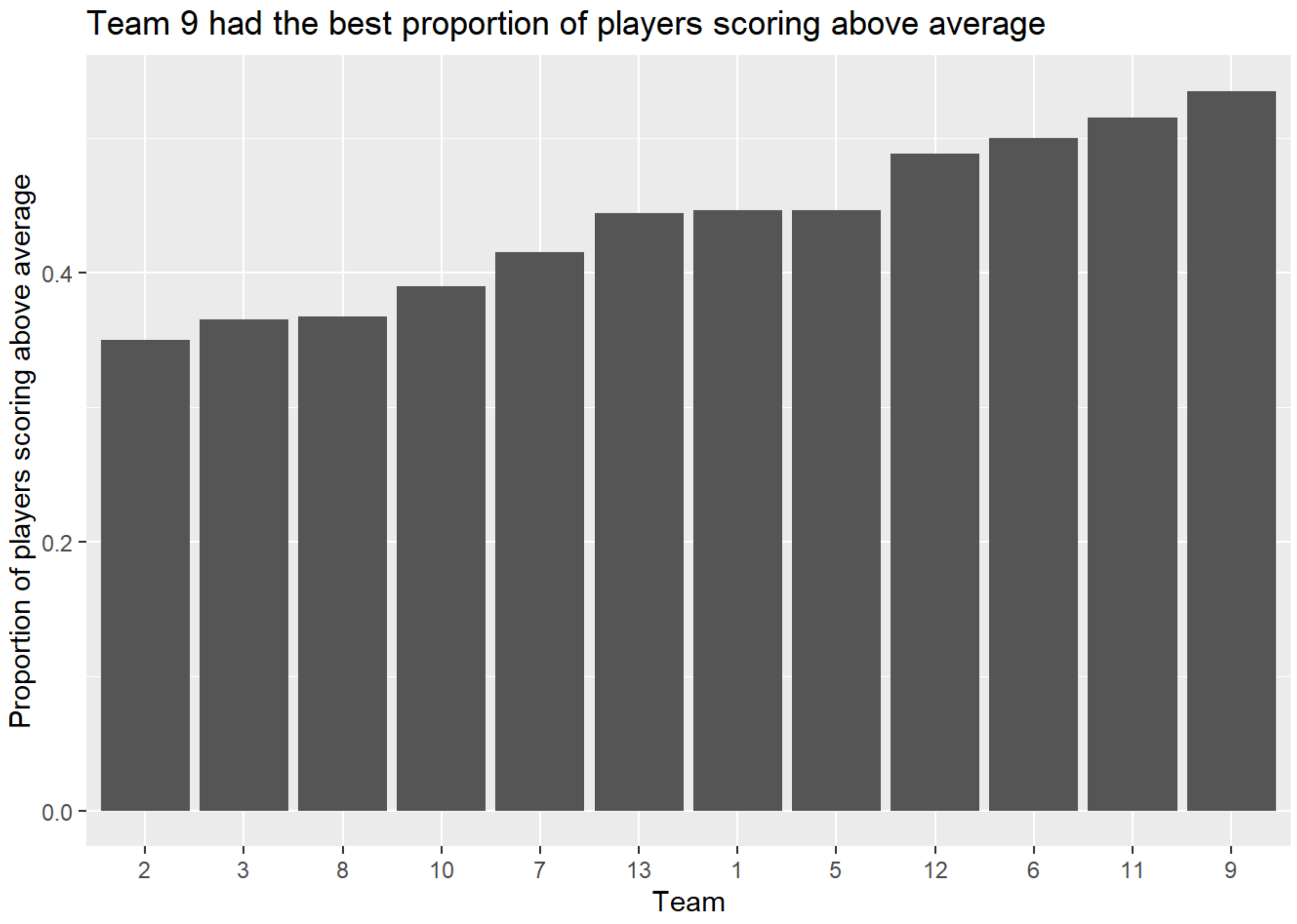
```
## # A tibble: 1 × 4
##     min   max median  mean
##   <dbl> <dbl>  <dbl> <dbl>
## 1    -2  53.7   10.3  11.6
```

```
#describe in the psych package
```

```
#which team had the most players scoring more than average (not bench)

above_average <- tidy_data |>
    filter(Pos != "Bench") |>
    mutate(
        avg = mean(Actual, na.rm = TRUE),
        above_avg = Actual > avg,
        Team = as.character(Team)
           ) |>
    group_by(Team) |>
    summarize(
        prop = mean(above_avg, na.rm = TRUE)
    ) |>
    arrange(desc(prop)) #team 9 had the best proportion of players scoring above average

above_average
```

```
## # A tibble: 12 × 2
##    Team   prop
##    <chr> <dbl>
##  1 9     0.535
##  2 11    0.515
##  3 6     0.5
##  4 12    0.488
##  5 5     0.446
##  6 1     0.446
##  7 13    0.444
##  8 7     0.415
##  9 10    0.389
## 10 8     0.367
## 11 3     0.365
## 12 2     0.350
```

```
ggplot(above_average, aes(fct_reorder(Team, prop), prop)) +
    geom_col() +
    labs(
        x = "Team",
        y = "Proportion of players scoring above average",
        title = "Team 9 had the best proportion of players scoring above average"
    )
```



```
#How did players do coming off injury

injured <- tidy_data |>
    filter(Pos == "IR", Actual == 0.00) |>
    distinct(Player, .keep_all = TRUE) |>
    ungroup()
injured #returns all players that were listed on IR
```

```
## # A tibble: 28 × 7
##     Week  Team Player          Pos   Status        Proj Actual
##    <dbl> <dbl> <chr>           <chr> <chr>        <dbl>  <dbl>
##  1     1     2 N'Keal Harry    IR    ACTIVE           0      0
##  2     2     3 Hunter Henry    IR    ACTIVE           0      0
##  3     2     6 Marqise Lee     IR    QUESTIONABLE     0      0
##  4     2     7 Tyreek Hill     IR    ACTIVE           0      0
##  5     2    13 A.J. Green      IR    QUESTIONABLE     0      0
##  6     3     1 Michael Gallup  IR    ACTIVE           0      0
##  7     3     6 Damien Williams IR    ACTIVE           0      0
##  8     4     1 Saquon Barkley  IR    ACTIVE           0      0
##  9     4     2 T.Y. Hilton     IR    ACTIVE           0      0
## 10     5     2 Kenny Stills    IR    ACTIVE           0      0
## # ℹ 18 more rows
```

```
injured_avg_pts <- tidy_data |>
    filter(Player %in% injured$Player) |>
    group_by(Player) |>
    summarize(
        injured_avg_actual = mean(Actual, na.rm = TRUE)
    )

injured_avg_pts #injured players average points throughout the season
```

```
## # A tibble: 28 × 2
##    Player          injured_avg_actual
##    <chr>                        <dbl>
##  1 A.J. Green                    0
##  2 Adam Thielen                  7.63
##  3 Alshon Jeffery                9.4
##  4 Alvin Kamara                 15.3
##  5 Austin Hooper                12.0
##  6 Brandin Cooks                 7.37
##  7 Damien Williams               7.19
##  8 David Johnson                 9.43
##  9 DeSean Jackson                2.46
## 10 Derrius Guice                 3.83
## # ℹ 18 more rows
```

```
injured2 <- tidy_data |>
    filter(Player %in% injured$Player) |>
    group_by(Player) |>
    arrange(Player) |>
    na.omit() |>
    mutate(
        difference = Actual - lag(Actual),
        comeback = if_else(Actual == difference, TRUE, FALSE)
    ) |>
    filter(Actual != 0, comeback == TRUE) |>
    ungroup()

injured2 #returns the points scored the week a player came back from injury
```

```
## # A tibble: 30 × 9
##     Week  Team Player          Pos   Status      Proj Actual difference comeback
##    <dbl> <dbl> <chr>           <chr> <chr>      <dbl>  <dbl>      <dbl> <lgl>
##  1    15     3 Adam Thielen    Flex  ACTIVE     15.8     6          6   TRUE
##  2     4     6 Alshon Jeffery  Bench QUESTIONA… 12.1    12.8       12.8 TRUE
##  3    13     6 Alshon Jeffery  IR    QUESTIONA… 12.7    28.7       28.7 TRUE
##  4    10     1 Alvin Kamara    RB    ACTIVE     22.6    15.4       15.4 TRUE
##  5    14     1 Austin Hooper   IR    ACTIVE     14.1     5.2        5.2 TRUE
##  6    12     2 Brandin Cooks   Flex  ACTIVE     11.1     5.3        5.3 TRUE
##  7    15     2 Brandin Cooks   Flex  ACTIVE      6.73    8.6        8.6 TRUE
##  8     5     6 Damien Williams Flex  ACTIVE     13.4     6.8        6.8 TRUE
##  9    16     2 Damien Williams Bench ACTIVE      9.28   18.2       18.2 TRUE
## 10    13     2 David Johnson   Bench ACTIVE      6.57    4.4        4.4 TRUE
## # ℹ 20 more rows
```

```
injured2 |>
    left_join(injured_avg_pts, by = "Player") |>
    select(Player, Pos, Actual, injured_avg_actual) |>
    mutate(
        good_return = Actual > injured_avg_actual
    )
```

```
## # A tibble: 30 × 5
##    Player          Pos   Actual injured_avg_actual good_return
##    <chr>           <chr>  <dbl>              <dbl> <lgl>
##  1 Adam Thielen    Flex     6                 7.63 FALSE
##  2 Alshon Jeffery  Bench   12.8               9.4  TRUE
##  3 Alshon Jeffery  IR      28.7               9.4  TRUE
##  4 Alvin Kamara    RB      15.4              15.3  TRUE
##  5 Austin Hooper   IR       5.2              12.0  FALSE
##  6 Brandin Cooks   Flex     5.3               7.37 FALSE
##  7 Brandin Cooks   Flex     8.6               7.37 TRUE
##  8 Damien Williams Flex     6.8               7.19 FALSE
##  9 Damien Williams Bench   18.2               7.19 TRUE
## 10 David Johnson   Bench    4.4               9.43 FALSE
## # ℹ 20 more rows
```

```
#returns whether a player scored above there average points for the season when the week they came back
```