2022-23 English Premier League Stats: Data Analysis and Visualization Project

Maxwell Fitzell

2023-05-02

Introduction & Context

In this project, I gathered large amounts of stats and data from Fbref.com for the 2022-23 English Premier League season to perform data analysis on players and teams and create visualizations to display these findings.

Having just completed the Google Data Analytics Professional Certificate, I decided to create an open-ended project that utilizes the R programming skills I was introduced to. Currently, I have only analyzed long passing stats for the league's midfielders - and wanted to share my findings - but I intend to create other analyses as well.

Packages Loaded:

```
library(tidyverse)
library(readxl)
library(ggrepel)
library(scales)
```

Converting Excel Spreadsheets to R Data Frames

```
pl.pass <- read_excel("Player Stats per90.xlsx", sheet="Pl Pass", skip = 1)[-c(1)] %>% rename('1/3'='44929')
pl.stand <- read_excel("Player Stats per90.xlsx", sheet="Pl Standard", skip = 1)[-c(1)]
pl.gk <- read_excel("Player Stats per90.xlsx", sheet="Pl GK", skip = 1)[-c(1)]
pl.gk_adv <- read_excel("Player Stats per90.xlsx", sheet="Pl GK Adv", skip = 1)[-c(1)]
pl.shoot <- read_excel("Player Stats per90.xlsx", sheet="Pl Shoot", skip = 1)[-c(1)]
pl.pass_type <- read_excel("Player Stats per90.xlsx", sheet="Pl Pass Type", skip = 1)[-c(1)]
pl.gscreate <- read_excel("Player Stats per90.xlsx", sheet="Pl G&S Create", skip = 1)[-c(1)]
pl.def <- read_excel("Player Stats per90.xlsx", sheet="Pl Def Act", skip = 1)[-c(1)]
pl.poss <- read_excel("Player Stats per90.xlsx", sheet="Pl Poss", skip = 1)[-c(1)]
pl.play_time <- read_excel("Player Stats per90.xlsx", sheet="Pl Play Time", skip = 1)[-c(1)]
pl.misc <- read_excel("Player Stats per90.xlsx", sheet="Pl Misc", skip = 1)[-c(1)]
```

Explanation: The above code uses the "read_excel" function from the *readxl* library to convert Excel spreadsheets to R data frames.

In the analysis that follows, we will only be using the **player passing (pl.pass)** data frame, which was converted from an Excel worksheet:

- From the "Player Stats per90.xlsx" workbook, specifically the "Pl Standard" worksheet
- I used **"skip = 1"** in the function's body to remove the first row, as I needed the second row to be the column names.
- Outside of the "read_excel" function, I used **[-c(1)]** to remove the first column, which was just a series of arbitrary numbers.
- I used **rename()** function to rename a column back to "1/3", as it was altered when converted to an R data frame.

Analysis: Long Passes Attempted per 90 vs. Percentage Successfully Completed

With my spreadsheet data now loaded into my R environment as data frames, I decided that I would look at long passing stats for Premier League midfielders.

Specifically, I decided that it would be interesting to look at *long passes attempted per game/per 90 minutes compared to the percentage of long passes successfully completed*. I elected to use only players who had played at least 10 games (10 "90s"), so that each player's stats would be more reliable.

Here are the steps I took to prepare and clean the data for analysis/visualization. Each step is broken into a What (to explain *what* the code does) and a Why (to explain *why* I needed to use the code):

```
pl.pass.clean <- pl.pass %>% separate(Player, into=c('first','last'),sep=' ', extra="merge")
```

What: Separated name column ('Player') into two new columns: 'first' and 'last' names.

Why: A column with just 'last' names allowed me to label the data points in my viz without overcrowding the plot.

```
pl.pass.clean$last[is.na(pl.pass.clean$last)] <- pl.pass.clean$first[is.na(pl.pass.clean$last)]
```

What: Code finds rows with a null 'last' name value, then returns that rows 'first' name value, duplicating the 'first' name value in the 'last' name, which was originally null.

Why: Many players only go by one name and their 'last' name slots were left empty in the previous line of code. This function filled in those players' 'last' name column, which was null.

```
pl.pass.mids_10_90s <- pl.pass.clean %>% filter (`90s▼`>10 & (Pos == "MF" | Pos == "MF,DF" | Pos == "MF,FW" | Pos == "DF,FW" | Pos == "FW"))
```

What: Filters the data frame by selecting ***1.*** only players with over 10 full games played and ***2.*** only players whose position is midfielder (or what I considered similar to a midfielder using my domain knowledge).

Why: The number of games each player has played needed to be high so the sample size was usable, and the players in question needed to also be midfielders.

Stand-Alone Calculations:

The following are two stand-alone calculations that I performed to determine the mean values of the x-axis (long passes attempted) and the y-axis (long pass completion percentage), which I would then *use in my visualization to divide the plot into quadrants*.

```
pl.pass.mids_10_90s %>% summarize(mean(Att...21))
```

```
## # A tibble: 1 × 1
##   `mean(Att...21)`
##              <dbl>
## 1             4.38
```

```
pl.pass.mids_10_90s %>% summarize(mean((Cmp...20/Att...21)))
```

```
## # A tibble: 1 × 1
##   `mean((Cmp...20/Att...21))`
##                         <dbl>
## 1                       0.590
```

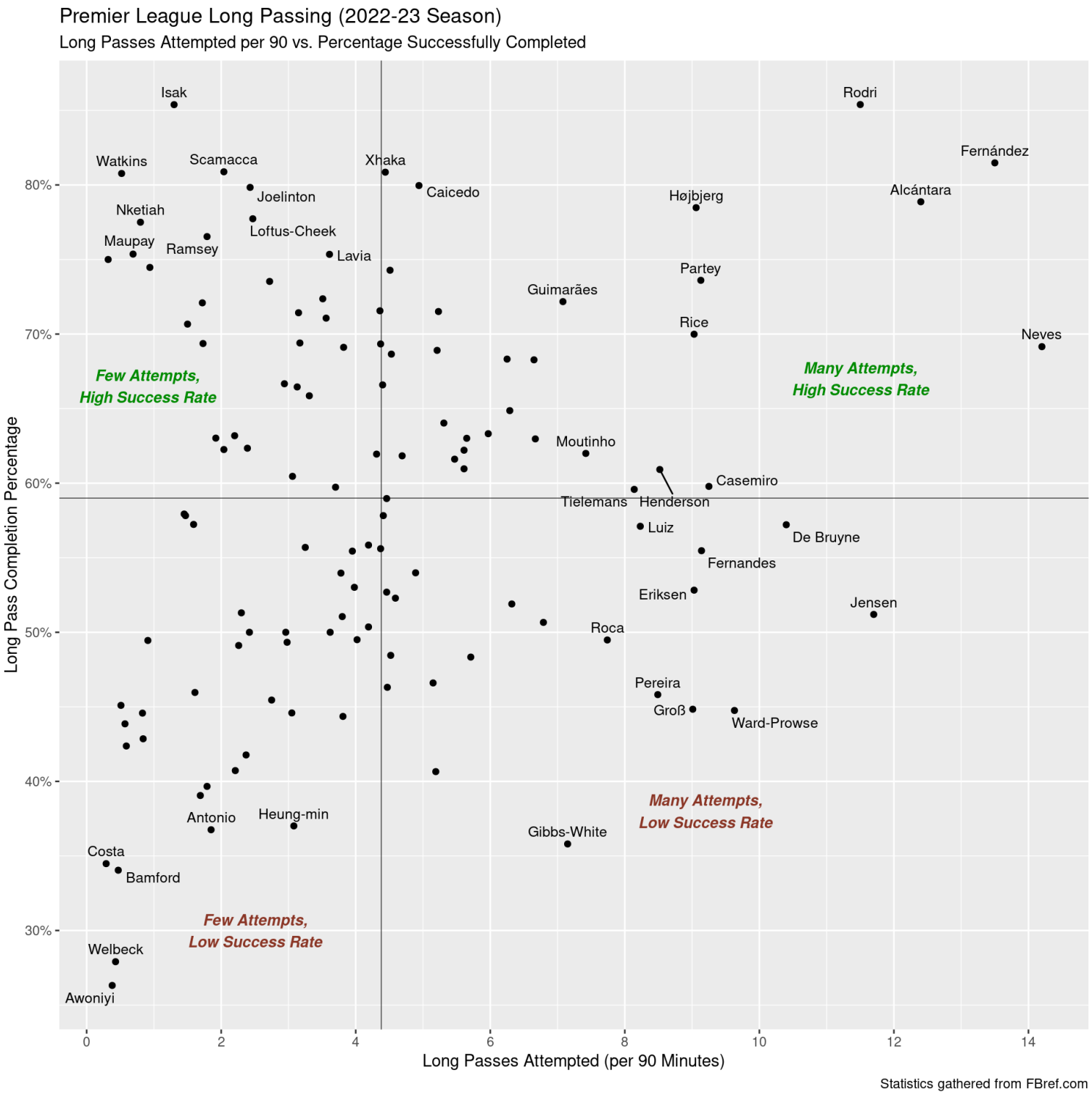
GGPlot2 Visualization Code

Explanation for the ggplot2 code below:

- *ggplot:* from the data frame created above, I chose attempted long passes per 90 as x-axis & long pass completion percent as y-axis.
- *geom_point:* to display a scatter plot.
- *geom_text_repel:* used last name column text as the label for data points that met specific criteria (if every data point was labelled, it would be too crowded).
- *xlim:* to set x-axis from 0 to 15. Allowed ylim to automatically set itself.
- *geom_hline and vline:* were set to the mean x and y values to create quadrants (values determined in the stand-alone calculations above).
- *labs:* to create title, subtitle, caption.
- *annotate:* to create 4 text labels to describe each quadrant.
- *scale_x/y_continuous:* set axis names, change unit of y-axis ticks to percent, used “pretty_breaks” to set the amount of data ticks across each axis.

```
pl.pass.mids_longpass_att.v.cmp <-
  ggplot(data=pl.pass.mids_10_90s, aes(x=Att...21, y=(Cmp...20/Att...21)))+
  geom_point()+
  geom_text_repel(data=subset(pl.pass.mids_10_90s, Att...21>7 | (Cmp...20/Att...21)>.75 | (Cmp...20/Att...21)<.375),
                  aes(x=Att...21, y=(Cmp...20/Att...21), label=last),
                  family="Helvetica", size=3.3, vjust=-.7)+
  xlim(0,15)+
  geom_hline(yintercept=.59, size=.2)+
  geom_vline(xintercept=4.38, size=.2)+
  labs(title="Premier League Long Passing (2022-23 Season)",
       subtitle="Long Passes Attempted per 90 vs. Percentage Successfully Completed",
       caption="Statistics gathered from FBref.com")+
  annotate("text", x=11.5, y=.67, label="Many Attempts,\nHigh Success Rate",
           family="Helvetica", fontface="bold.italic", size=3.5, color="green4")+
  annotate("text", x=2.5, y=.30, label="Few Attempts,\nLow Success Rate",
           family="Helvetica", fontface="bold.italic", size=3.5, color="tomato4")+
  annotate("text", x=.9, y=.665, label="Few Attempts,\nHigh Success Rate",
           family="Helvetica", fontface="bold.italic", size=3.5, color="green4")+
  annotate("text", x=9.2, y=.38, label="Many Attempts,\nLow Success Rate",
           family="Helvetica", fontface="bold.italic", size=3.5, color="tomato4")+
  scale_x_continuous(name="Long Passes Attempted (per 90 Minutes)", breaks = scales::pretty_breaks(n = 7))+
  scale_y_continuous(name="Long Pass Completion Percentage", labels=percent, breaks = scales::pretty_breaks(n = 5))

pl.pass.mids_longpass_att.v.cmp
```


Premier League Long Passing (2022-23 Season)
Long Passes Attempted per 90 vs. Percentage Successfully Completed
Long Pass Completion Percentage
80%
70%
60%
50%
40%
30%
0
2
4
6
8
10
12
14
Long Passes Attempted (per 90 Minutes)
Isak
Rodri
Watkins
Scamacca
Xhaka
Fernández
Caicedo
Joelinton
Højbjerg
Alcántara
Nketiah
Loftus-Cheek
Maupay
Ramsey
Lavia
Partey
Guimarães
Rice
Neves
Few Attempts,
High Success Rate
Many Attempts,
High Success Rate
Moutinho
Casemiro
Tielemans
Henderson
Luiz
De Bruyne
Fernandes
Eriksen
Jensen
Roca
Pereira
Groß
Ward-Prowse
Many Attempts,
Low Success Rate
Antonio
Heung-min
Gibbs-White
Costa
Bamford
Few Attempts,
Low Success Rate
Welbeck
Awoniyi
Statistics gathered from FBref.com