

# STAT184 NBAPROJECT

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
options(repos = c(CRAN = "https://cran.rstudio.com"))  
  
install.packages("tidyverse")
```

The downloaded binary packages are in  
/var/folders/vk/zxgkdbnn3l18n7gbh7r3z1z00000gn/T//RtmpiU0LvT/downloaded\_packages

```
install.packages("ggrepel")
```

The downloaded binary packages are in  
/var/folders/vk/zxgkdbnn3l18n7gbh7r3z1z00000gn/T//RtmpiU0LvT/downloaded\_packages

```
library(dplyr)
```

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

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
v forcats   1.0.0      v readr       2.1.5
```

```
v ggplot2   3.5.1      v stringr     1.5.1
```

```
v lubridate 1.9.3      v tibble      3.2.1
```

```
v purrr     1.0.2      v tidyr       1.3.1
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()     masks stats::lag()
```

```
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(base)
```

```
library(tidyr)
```

```
library(ggplot2)
```

```
library(ggplot2)
```

```
twenty23player <- read.csv("/Users/christianmungal/Desktop/STAT184/2023-24NBASTATS.csv")
```

```
twenty20player <- read.csv("/Users/christianmungal/Desktop/STAT184/2020-21NBASTATS.csv")
```

```
draft_class <- read.csv("/Users/christianmungal/Desktop/STAT184/2020DRAFTCLASS.csv")
```

```
#Getting the 2020 draft class stats from the 2020-2021 NBA Season
```

```
keep2020 <- c("Anthony Edwards", "James Wiseman", "Lamelo Ball", "Patrick Williams",  
              "Issac Okoro", "Onyeka Okongwu", "Killian Hayes",  
              "Obi Toppin", "Deni Avdija", "Jalen Smith", "Devin Vassell",  
              "Tyrese Haliburton", "Kira Lewis Jr.",  
              "Aaron Nesmith", "Cole Anthony", "Isaiah Stewart", "Aleksej Pokusevski",  
              "Josh Green", "Saddiq Bey", "Precious Achiuwa", "Tyrese Maxey",  
              "Zeke Nnaji", "Leandro Bolmaro",  
              "R.J. Hampton", "Immanuel Quickley", "Payton Pritchard",  
              "Udoka Azubuike", "Jaden McDaniels", "Malachi Flynn",  
              "Desmond Bane", "Tyrell Terry", "Vernon Carey Jr.", "Daniel Oturu",
```

```

      "Théo Maledon", "Xavier Tillman Sr.", "Tyler Bey", "Vit Krejci",
      "Saben Lee", "Elijah Hughes",
      "Robert Woodard II", "Tre Jones", "Nick Richards", "Jahmi'us Ramsey",
      " Marko Simonovic", "Jordan Nwora", "CJ Elleby", "Yam Madar",
      "Nico Mannion", "Isaiah Joe", "Skylar Mays", "Justinian Jessup", "KJ Martin",
      "Cassius Winston", "Jay Scrubb", "Grant Riller",
      "Paul Reed", "Jalen Harris", "Sam Merrill")

twenty20_stats <- twenty20player %>%
  filter(Player %in% keep2020)

#Getting the 2020 draft class stats from the 2023-2024 NBA Season
twenty23_stats <- twenty23player %>%
  filter(NAME %in% keep2020)

#Remove unwanted rows
twenty23_stats <- twenty23_stats[-c(1, 10, 12, 14, 17, 23, 25)]
twenty20_stats <- twenty20_stats[-c(1,7,9,10,11,12,13,15,16,19,20,22,23,28)]

#Remove two of the cases for RJ Hampton and keep the one that averaged the two in 2020stats
twenty20_stats <- twenty20_stats[-c(19,20), ]

#Put datasets in alphabetical order to make it easier to see errors and for easier binding
twenty20_stats <- twenty20_stats[order(twenty20_stats$Player), ]
twenty23_stats <- twenty23_stats[order(twenty23_stats$NAME), ]

#Remove duplicates
twenty23_stats <- twenty23_stats[-c(2,3,9,10,16,17,20,21,22,24,25,26,33,34,39,40), ]

#Add the players back
IQ <- data.frame(NAME = "Immanuel Quickley", TEAM = "2TMS", POS = "G", AGE = 24,
  GP = 68, MPG = 28.7, USG. = 23.9, TO. = 9.8, FT. = 0.866, X2P. = 0.479,
  X3P. = 0.395, eFG. = 0.534, PPG = 16.8, RPG = 3.7, APG = 4.65,
  SPG = 0.7, BPG = 0.1, P.R = 20.5, P.R.A = 25, VI = 8.9, ORtg = 120.5, DRtg = 112.6)
twenty23_stats <- rbind(twenty23_stats, IQ)

AP <- data.frame(NAME = "Aleksej Pokusevski", TEAM = "2TMS", POS = "F", AGE = 22,
  GP = 28, MPG = 14.5, USG. = 16.3, TO. = 14.4, FT. = 0.629,
  X2P. = 0.438, X3P. = 0.273, eFG. = 0.409, PPG = 4.3, RPG = 2.7,
  APG = 1.1, SPG = 0.5, BPG = 0.4, P.R = 7.0, P.R.A = 8.1, VI = 6.7,
  ORtg = 95.1, DRtg = 112.6)

```

```

twenty23_stats <- rbind(twenty23_stats, AP)

JN <- data.frame(NAME = "Jordan Nwora", TEAM = "2TMS", POS = "F", AGE = 25.6, GP = 52,
  MPG = 13.7, USG. = 22.5, TO. = 14.3, FT. = 0.825, X2P. = 0.561, X3P. = 0.32,
  eFG. = 0.530, PPG = 6.6, RPG = 2.6, APG = 1.15, SPG = 0.5, BPG = 0.3, P.R. =
  P.R.A = 10.4, VI = 8.3, ORtg = 105.3, DRtg = 113.2)
twenty23_stats <- rbind(twenty23_stats, JN)

KLJ <- data.frame(NAME = "Kira Lewis Jr.", TEAM = "3TMS", POS = "G", AGE = 23,
  GP = 28, MPG = 9.5, USG. = 18.6, TO. = 13.0, FT. = 0.850,
  X2P. = 0.508, X3P. = 0.121,
  eFG. = 0.391, PPG = 3.2, RPG = 0.9, APG = 1.3, SPG = 0.3,
  BPG = 0.1, P.R. = 4.7, P.R.A = 6.0, VI = 6.7, ORtg = 100.9, DRtg = 117.4)
twenty23_stats <- rbind(twenty23_stats, KLJ)

MF <- data.frame(NAME = "Malachi Flynn", TEAM = "3TMS", POS = "G", AGE = 25.9,
  GP = 69, MPG = 12.7, USG. = 21.4, TO. = 14.9, FT. = 0.730,
  X2P. = 0.513, X3P. = 0.331, eFG. = 0.505,
  PPG = 5.5, RPG = 1.7, APG = 1.9, SPG = 0.6, BPG = 0.1, P.R. = 6.5,
  P.R.A = 8.2, VI = 7.7, ORtg = 108.4, DRtg = 113.8)
twenty23_stats <- rbind(twenty23_stats, MF)

PA <- data.frame(NAME = "Precious Achiuwa", TEAM = "2TMS", POS = "F", AGE = 24,
  GP = 74, MPG = 21.9, USG. = 17.6, TO. = 14.9, FT. = 0.616,
  X2P. = 0.562, X3P. = 0.268, eFG. = 0.529,
  PPG = 7.6, RPG = 6.6, APG = 1.3, SPG = 0.6, BPG = 0.8, P.R. = 14,
  P.R.A = 15.3, VI = 8, ORtg = 111.1, DRtg = 108.1)
twenty23_stats <- rbind(twenty23_stats, PA)

SM <- data.frame(NAME = "Skylar Mays", TEAM = "2TMS", POS = "G", AGE = 26,
  GP = 38, MPG = 11.4, USG. = 16.2, TO. = 10.1, FT. = 0.765,
  X2P. = 0.469, X3P. = 0.296, eFG. = 0.447,
  PPG = 4.1, RPG = 1.1, APG = 2.2, SPG = 0.6, BPG = 0.1,
  P.R. = 5.1, P.R.A = 6.9, VI = 7, ORtg = 109.7, DRtg = 112.6)
twenty23_stats <- rbind(twenty23_stats, SM)

#Reorder the new cases in alphabetical order
twenty23_stats <- twenty23_stats[order(twenty23_stats$NAME), ]

#Only keep players that played in both seasons
twenty23_stats <- twenty23_stats[-c(

```

```

1, 9, 11, 12, 15, 20, 21, 22, 24, 30, 31, 32, 35, 36, 37), ]
twenty20_stats <- twenty20_stats[-c(9, 11, 23, 26), ]

#POINTS PROGRESSION

PTS_PROG <- bind_cols(
  select(twenty20_stats, Player),
  select(twenty20_stats, PTS),
  select(twenty23_stats, PPG)
)
PTS_PROG <- PTS_PROG %>%
  rename(
    "2020" = PTS,
    "2023" = PPG
  )

PTS_PROG_FORMED <- pivot_longer(
  PTS_PROG,
  cols = c("2020", "2023"),
  names_to = "Year",
  values_to = "PPG"
)
PTS_PROG_FORMED$Year <- as.numeric(PTS_PROG_FORMED$Year)
ggplot(
  PTS_PROG_FORMED,
  aes(
    x = Year,
    y = PPG,
    color = Player,
    group = Player
  )
) +
  geom_line(size = 0.7) +
  geom_point(size = 2) +
  labs(
    title = "Points Per Game Over Time",
    x = "Year",
    y = "Points Per Game (PPG)"
  ) +
  scale_x_continuous(
    breaks = c(2020, 2023),

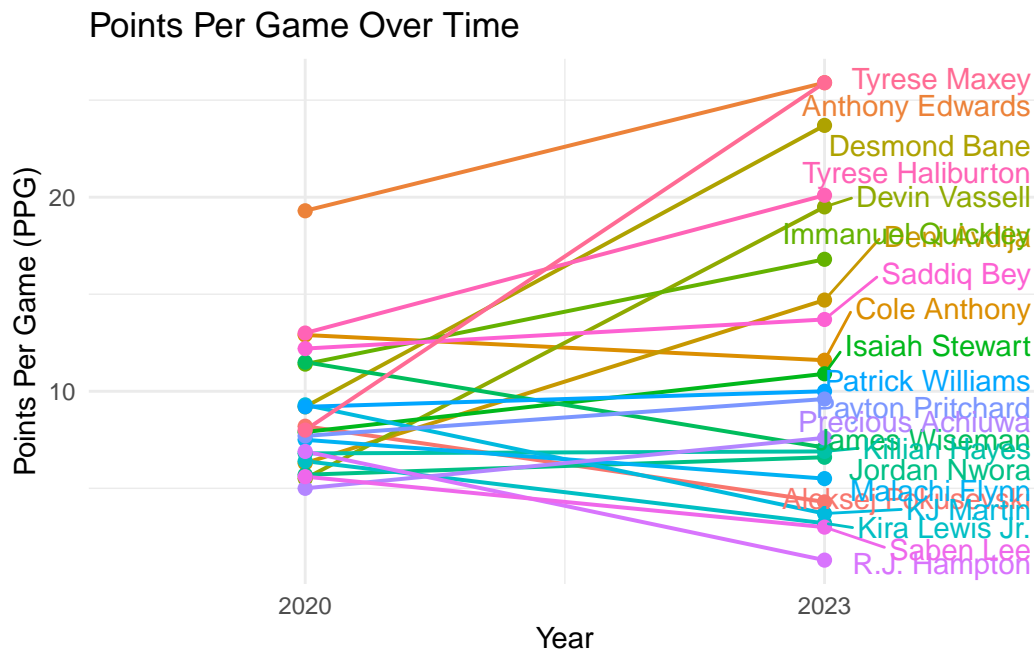
```

```

limits = c(2019,2024)
) +
geom_text_repel(
  data = PTS_PROG_FORMED[PTS_PROG_FORMED$Year == 2023, ],
  aes(
    label = Player
  ),
  nudge_x = 0.5,
  direction = "y",
  hjust = 0
) +
theme_minimal() +
theme(
  legend.position = "none"
)

```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
 i Please use `linewidth` instead.



```

#Defensive Stats
twenty23_stats <- twenty23_stats %>%
  mutate(DefStats23 = SPG + BPG)

```

```

twenty20_stats <- twenty20_stats %>%
  mutate(DefStats20 = STL + BLK)

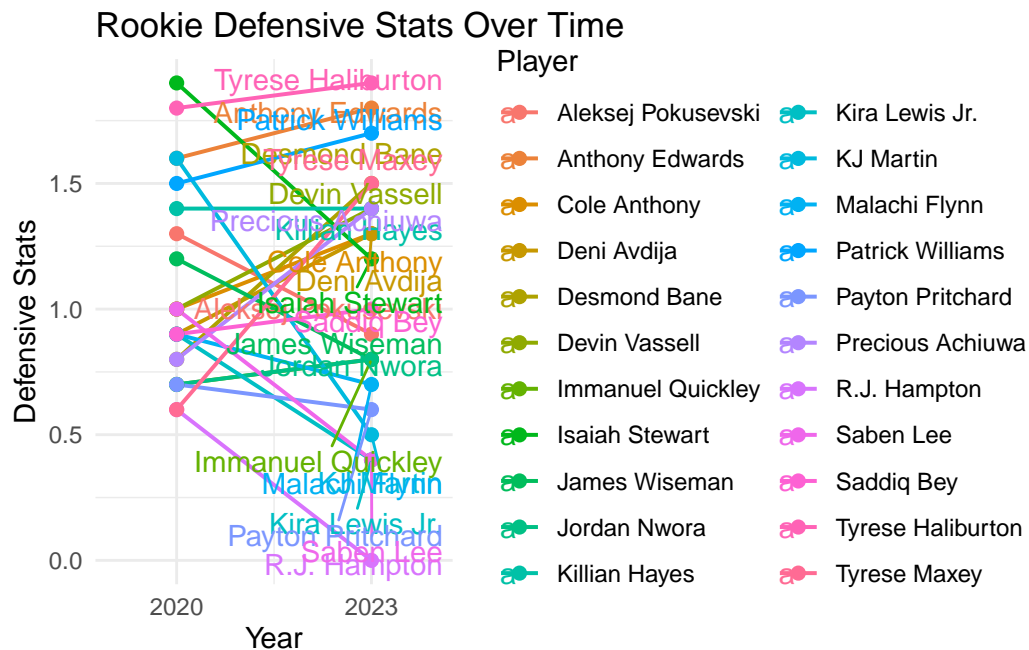
Def_PROG <- bind_cols(
  select(twenty20_stats, Player),
  select(twenty20_stats, DefStats20),
  select(twenty23_stats, DefStats23)
)
Def_PROG <- Def_PROG %>%
  rename(
    "2020" = DefStats20,
    "2023" = DefStats23)

Def_PROG_FORMED <- Def_PROG %>%
  pivot_longer(
    cols = c("2020", "2023"),
    names_to = "Year",
    values_to = "Defensive Stats (BLKS +STLS)"
  )
Def_PROG_FORMED$Year <- as.numeric(Def_PROG_FORMED$Year)

ggplot(
  Def_PROG_FORMED,
  aes(x = Year,
      y = `Defensive Stats (BLKS +STLS)`,
      color = Player,
      group = Player)
) +
  geom_line(size = 0.7) +
  geom_point(size = 2) +
  labs(
    title = "Rookie Defensive Stats Over Time",
    x = "Year",
    y = "Defensive Stats"
  ) +
  scale_x_continuous(
    breaks = c(2020, 2023),
    limits = c(2019, 2024)
  ) +
  geom_text_repel(
    data = Def_PROG_FORMED[Def_PROG_FORMED$Year == "2023", ],
    aes(label = Player),

```

```
nudge_x = 0.5,
direction = "y",
hjust = 0
) +
theme_minimal()
```



```
# Combine columns for eFG%
eFG_PROG <- bind_cols(
  dplyr::select(twenty20_stats, Player),
  dplyr::select(twenty20_stats, eFG.),
  dplyr::select(twenty23_stats, eFG.)
)
```

New names:

```
* `eFG.` -> `eFG....2`
* `eFG.` -> `eFG....3`
```

```
# Rename columns
eFG_PROG <- eFG_PROG %>%
  rename(
    eFG2020 = eFG....2,
```



```

    eFG2023 = eFG....3 # Ensure correct column selection if duplicates exist
  )

# Transform data into long format
eFG_PROG_FORMED <- pivot_longer(
  eFG_PROG,
  cols = c(eFG2020, eFG2023),
  names_to = "Season",
  values_to = "eFG_Percentage"
)

# Format Player names for better display
eFG_PROG_FORMED$Player <- gsub(" ", "\\n", eFG_PROG_FORMED$Player)

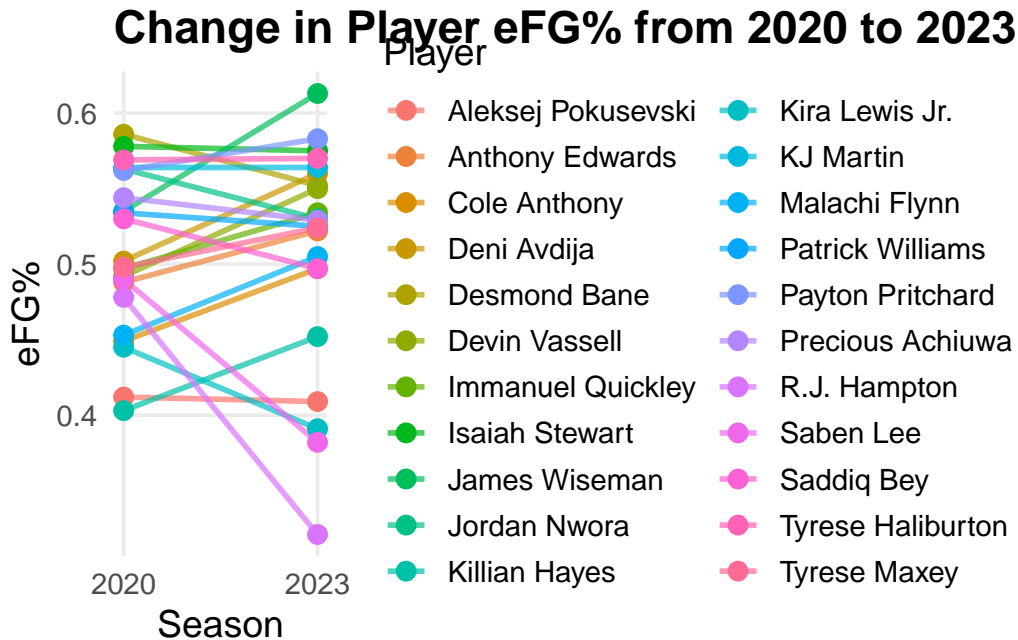
# Assume your data frame is called eFG_PROG with columns: Player, eFG2020, eFG2023
# First, pivot to long format
eFG_PROG_FORMED <- eFG_PROG %>%
  pivot_longer(cols = c("eFG2020", "eFG2023"),
    names_to = "Season",
    values_to = "eFG") %>%
  mutate(
    # Convert Season names to numeric years
    Season = recode(Season, "eFG2020" = "2020", "eFG2023" = "2023"),
    Season = as.integer(Season)
  )

# Plot the time plot (slope chart)
ggplot(eFG_PROG_FORMED, aes(x = Season, y = eFG, group = Player, color = Player)) +
  geom_line(size = 1, alpha = 0.7) +
  geom_point(size = 3) +
  scale_x_continuous(breaks = c(2020, 2023)) +
  labs(
    title = "Change in Player eFG% from 2020 to 2023",
    x = "Season",
    y = "eFG%"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(hjust = 0.5),

```

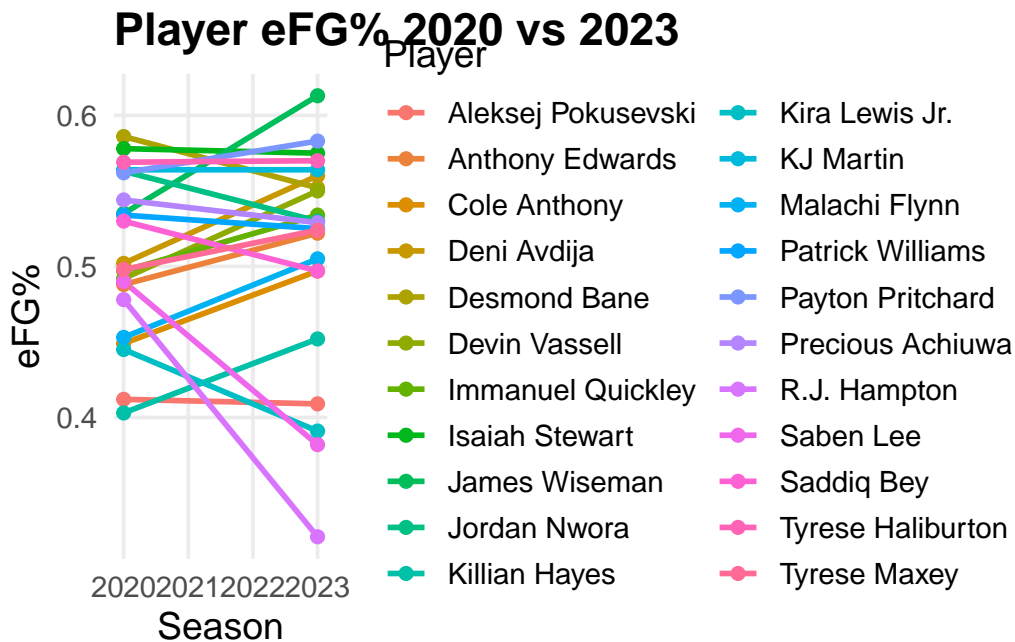
```
plot.title = element_text(face = "bold")
```

```
)
```



```
# Assuming PTS_PROG_FORMED has the columns: Player, Season, and Points
# and Season takes values "PPG2020" and "PPG2023".
```

```
ggplot(eFG_PROG_FORMED, aes(x = Season, y = eFG, group = Player, color = Player)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Player eFG% 2020 vs 2023",
    x = "Season",
    y = "eFG%"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(hjust = 0.5),
    plot.title = element_text(face = "bold")
  )
```



```
# Combine columns for eFG%
Minutes_PROG <- bind_cols(
  dplyr::select(twenty20_stats, Player),
  dplyr::select(twenty20_stats, MP),
  dplyr::select(twenty23_stats, MPG)
)

# Rename columns
Minutes_PROG <- Minutes_PROG %>%
  rename(
    Minutes_2020 = MPG,
    Minutes_2023 = MP # Ensure correct column selection if duplicates exist
  )

# Transform data into long format
Minutes_FORMED <- pivot_longer(
  Minutes_PROG,
  cols = c(Minutes_2020, Minutes_2023),
  names_to = "Season",
  values_to = "Minutes"
)

# Format Player names for better display
```

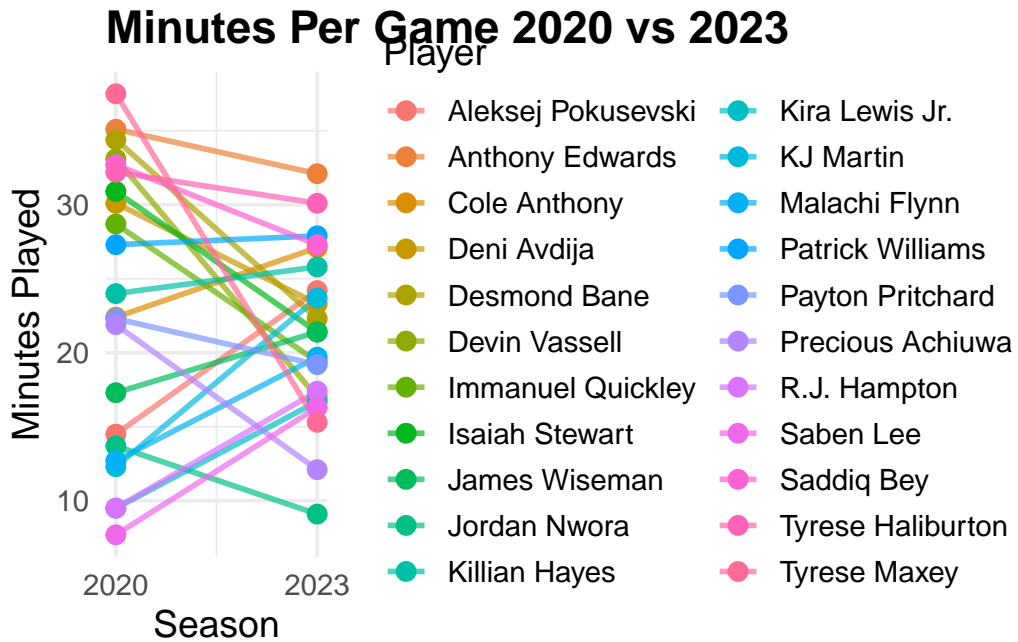
```

eFG_PROG_FORMED$Player <- gsub(" ", "\n", eFG_PROG_FORMED$Player)

# Clean Season names
Minutes_FORMED <- Minutes_FORMED %>%
  mutate(Season = recode(Season, "Minutes_2020" = "2020", "Minutes_2023" = "2023"),
         Season = as.integer(Season))

# Create the time plot
ggplot(Minutes_FORMED, aes(x = Season, y = Minutes, group = Player, color = Player)) +
  geom_line(size = 1, alpha = 0.7) +
  geom_point(size = 3) +
  scale_x_continuous(breaks = c(2020, 2023)) +
  labs(
    title = "Minutes Per Game 2020 vs 2023",
    x = "Season",
    y = "Minutes Played"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    axis.text.x = element_text(hjust = 0.5),
    plot.title = element_text(face = "bold")
  )

```



You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).