

CVDI2 Project

Collin

2025-02-18

packages

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
```

```
## 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(tidyr)
```

```
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.4.2
```

load in team data

```
T16 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2015-2016_NBA_Box_Score_Team-Stats.x")
```

```
## New names:
```

```
## * '' -> '...39'
```

```
## * '' -> '...40'
```

```
## * '' -> '...41'
```

```
## * '' -> '...42'
```

```
T17 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2016-2017_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T18 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2017-2018_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T19 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2018-2019_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T20 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2019-2020_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T21 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2020-2021_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T22 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2021-2022_NBA_Box_Score_Team-Stats.xls")
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T23 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2022-2023_NBA_Box_Score_Team-Stats.x
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

```
T24 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2023-2024_NBA_Box_Score_Team-Stats.x
```

```
## New names:  
## * ' ' -> '...39'  
## * ' ' -> '...40'  
## * ' ' -> '...41'  
## * ' ' -> '...42'
```

load in player data

```
P16 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2015-2016-NBA-Player-BoxScore-Datase  
P17 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2016-2017-NBA-Player-BoxScore-Datase  
P18 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2017-2018-NBA-Player-BoxScore-Datase  
P19 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2018-2019_NBA_Player-BoxScore-Datase  
P20 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2019-2020_NBA_Player-BoxScore-Datase  
P21 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2020-2021_NBA_Player-BoxScore-Datase  
P22 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2021-2022_NBA_Player-BoxScore-Datase  
P23 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2022-2023_NBA_Player-BoxScore-Datase  
P24 <- read_excel("C:/Users/cstra/projects/NBAExpansionRosterBuild/2023-2024-NBA-Player-BoxScore-Datase
```

Add year column

```
T16$Year <- 2016  
T17$Year <- 2017  
T18$Year <- 2018  
T19$Year <- 2019  
T20$Year <- 2020  
T21$Year <- 2021  
T22$Year <- 2022  
T23$Year <- 2023  
T24$Year <- 2024  
  
P16$Year <- 2016  
P17$Year <- 2017  
P18$Year <- 2018  
P19$Year <- 2019  
P20$Year <- 2020  
P21$Year <- 2021  
P22$Year <- 2022  
P23$Year <- 2023  
P24$Year <- 2024
```

named differently

```

#T16 <- T16 %>%
  #rename(DATASET = `BIGDATABALL\r\nDATASET`)
#T17 <- T17 %>%
  #rename(DATASET = `BIGDATABALL\r\nDATASET`)
#T18 <- T18 %>%
  #rename(DATASET = `BIGDATABALL\r\nDATASET`)
#T19 <- T19 %>%
  #rename(DATASET = `BIGDATABALL\r\nDATASET`)
T20 <- T20 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
T21 <- T21 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
T22 <- T22 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
T23 <- T23 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
T24 <- T24 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)

P16 <- P16 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P17 <- P17 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P18 <- P18 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
#P19 <- P19 %>%
  #rename(DATASET = `BIGDATABALL\r\nDATASET`)
P20 <- P20 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P21 <- P21 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P22 <- P22 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P23 <- P23 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)
P24 <- P24 %>%
  rename(DATASET = `BIGDATABALL\r\nDATASET`)

```

combine datasets

```

TeamData <- bind_rows(T16, T17, T18, T19, T20, T21, T22, T23, T24)

PlayerData <- bind_rows(P16, P17, P18, P19, P20, P21, P22, P23, P24)

```

Clean Team

```

TeamData <- TeamData %>%
  rename(
    Minutes = MIN,
    FieldGoals = FG,
    FieldGoalsAttempted = FGA,
    ThreePointers = `3P`,

```

```

    ThreePointAttempts = `3PA`,
    FreeThrows = FT,
    FreeThrowAttempts = FTA,
    OffensiveRebounds = OR,
    DefensiveRebounds = DR,
    TotalRebounds = TOT,
    Assists = A,
    PersonalFouls = PF,
    Steals = ST,
    Turnovers = TO,
    Blocks = BL,
    Points = PTS,
    Final = F,
    TeamRestDays = `TEAM\r\nREST DAYS`
  )

TeamData <- TeamData %>% select(Year, everything())

TeamData <- TeamData %>% select(1:38)

TeamData <- TeamData %>%
  select(-"TO\r\nTO")

```

Clean Player

```

PlayerData <- PlayerData %>%
  rename(
    Player = `PLAYER \r\nFULL NAME`,
    Team = `OWN \r\nTEAM`,
    Opponent = `OPPONENT \r\nTEAM`,
    VenueHR = `VENUE\r\n(R/H)`,
    Starter = `STARTER\r\n(Y/N)`,
    Minutes = MIN,
    FieldGoals = FG,
    FieldGoalsAttempted = FGA,
    ThreePointers = `3P`,
    ThreePointAttempts = `3PA`,
    FreeThrows = FT,
    FreeThrowAttempts = FTA,
    OffensiveRebounds = OR,
    DefensiveRebounds = DR,
    TotalRebounds = TOT,
    Assists = A,
    PersonalFouls = PF,
    Steals = ST,
    Turnovers = TO,
    Blocks = BL,
    Points = PTS,
    UsageRate = `USAGE \r\nRATE (%)`,
    DaysRest = `DAYS\r\nREST`,
    VenueHRN = `VENUE\r\n(R/H/N)`
  )

```

```

PlayerData <- PlayerData %>% select(DATASET, everything())

PlayerData <- PlayerData %>% select(Year, everything())

PlayerData <- PlayerData %>% select(-VenueHR, everything(), VenueHR)

PlayerData <- PlayerData %>% mutate(POSITION = substr(POSITION, 1, 1))

```

Add to Teams

```

TeamData <- TeamData %>%
  mutate(
    PRA = Points + TotalRebounds + Assists,
    PA = Points + Assists,
    PR = Points + TotalRebounds,
    RA = TotalRebounds + Assists,
    Stocks = Steals + Blocks
  )

TeamData <- TeamData %>%
  mutate(TeamYear = paste(Year, TEAM, sep = " "))

TeamData <- TeamData %>%
  mutate(
    FieldGoalPercentage = FieldGoals / FieldGoalsAttempted,
    TwoPointPercentage = (FieldGoals - ThreePointers) / (FieldGoalsAttempted - ThreePointAttempts),
    ThreePointPercentage = ThreePointers / ThreePointAttempts,
    FreeThrowPercentage = FreeThrows / FreeThrowAttempts
  )

```

Add to Players

```

PlayerData <- PlayerData %>%
  mutate(
    PRA = Points + TotalRebounds + Assists,
    PA = Points + Assists,
    PR = Points + TotalRebounds,
    RA = TotalRebounds + Assists,
    Stocks = Steals + Blocks
  )

PlayerData <- PlayerData %>%
  mutate(TeamYear = paste(Year, Team, sep = " "),
         PlayerYear = paste(Year, Player, sep = " "))

```

Make all stats per minute

```

# ensure there are no divisions by zero
PlayerData$Minutes[PlayerData$Minutes == 0] <- NA

```

```

# Create new per-minute columns
PlayerData$PointsPerMinute <- PlayerData$Points / PlayerData$Minutes
PlayerData$FieldGoalsPerMinute <- PlayerData$FieldGoals / PlayerData$Minutes
PlayerData$FieldGoalsAttemptedPerMinute <- PlayerData$FieldGoalsAttempted /
  PlayerData$Minutes
PlayerData$ThreePointersPerMinute <- PlayerData$ThreePointers / PlayerData$Minutes
PlayerData$ThreePointAttemptsPerMinute <- PlayerData$ThreePointAttempts /
  PlayerData$Minutes
PlayerData$FreeThrowsPerMinute <- PlayerData$FreeThrows / PlayerData$Minutes
PlayerData$FreeThrowAttemptsPerMinute <- PlayerData$FreeThrowAttempts /
  PlayerData$Minutes
PlayerData$OffensiveReboundsPerMinute <- PlayerData$OffensiveRebounds /
  PlayerData$Minutes
PlayerData$DefensiveReboundsPerMinute <- PlayerData$DefensiveRebounds /
  PlayerData$Minutes
PlayerData$TotalReboundsPerMinute <- PlayerData$TotalRebounds / PlayerData$Minutes
PlayerData$AssistsPerMinute <- PlayerData$Assists / PlayerData$Minutes
PlayerData$PersonalFoulsPerMinute <- PlayerData$PersonalFouls / PlayerData$Minutes
PlayerData$StealsPerMinute <- PlayerData$Steals / PlayerData$Minutes
PlayerData$TurnoversPerMinute <- PlayerData$Turnovers / PlayerData$Minutes
PlayerData$BlocksPerMinute <- PlayerData$Blocks / PlayerData$Minutes
PlayerData$PRAPerMinute <- PlayerData$PRA / PlayerData$Minutes
PlayerData$PAPerMinute <- PlayerData$PA / PlayerData$Minutes
PlayerData$PRPerMinute <- PlayerData$PR / PlayerData$Minutes
PlayerData$RAPerMinute <- PlayerData$RA / PlayerData$Minutes
PlayerData$StocksPerMinute <- PlayerData$Stocks / PlayerData$Minutes

```

points per 48

```

# Replace zero minutes with NA to avoid division by zero
PlayerData$Minutes[PlayerData$Minutes == 0] <- NA

# Create new per-48-minute columns
PlayerData$PointsPer48 <- (PlayerData$Points / PlayerData$Minutes) * 48
PlayerData$FieldGoalsPer48 <- (PlayerData$FieldGoals / PlayerData$Minutes) * 48
PlayerData$FieldGoalsAttemptedPer48 <- (PlayerData$FieldGoalsAttempted / PlayerData$Minutes) * 48
PlayerData$ThreePointersPer48 <- (PlayerData$ThreePointers / PlayerData$Minutes) * 48
PlayerData$ThreePointAttemptsPer48 <- (PlayerData$ThreePointAttempts / PlayerData$Minutes) * 48
PlayerData$FreeThrowsPer48 <- (PlayerData$FreeThrows / PlayerData$Minutes) * 48
PlayerData$FreeThrowAttemptsPer48 <- (PlayerData$FreeThrowAttempts / PlayerData$Minutes) * 48
PlayerData$OffensiveReboundsPer48 <- (PlayerData$OffensiveRebounds / PlayerData$Minutes) * 48
PlayerData$DefensiveReboundsPer48 <- (PlayerData$DefensiveRebounds / PlayerData$Minutes) * 48
PlayerData$TotalReboundsPer48 <- (PlayerData$TotalRebounds / PlayerData$Minutes) * 48
PlayerData$AssistsPer48 <- (PlayerData$Assists / PlayerData$Minutes) * 48
PlayerData$PersonalFoulsPer48 <- (PlayerData$PersonalFouls / PlayerData$Minutes) * 48
PlayerData$StealsPer48 <- (PlayerData$Steals / PlayerData$Minutes) * 48
PlayerData$TurnoversPer48 <- (PlayerData$Turnovers / PlayerData$Minutes) * 48
PlayerData$BlocksPer48 <- (PlayerData$Blocks / PlayerData$Minutes) * 48
PlayerData$PRAPer48 <- (PlayerData$PRA / PlayerData$Minutes) * 48
PlayerData$PAPer48 <- (PlayerData$PA / PlayerData$Minutes) * 48
PlayerData$PRPer48 <- (PlayerData$PR / PlayerData$Minutes) * 48
PlayerData$RAPer48 <- (PlayerData$RA / PlayerData$Minutes) * 48
PlayerData$StocksPer48 <- (PlayerData$Stocks / PlayerData$Minutes) * 48

```

Get rid of playoffs/playin/ist game data

```
NonRegSzn <- c("Playoffs", "Play-in", "In-Season Tournament")

TeamData <- TeamData %>%
  filter(!grepl(paste(NonRegSzn, collapse="|"), DATASET))

PlayerData <- PlayerData %>%
  filter(!grepl(paste(NonRegSzn, collapse="|"), DATASET))
```

Remove OT Games

```
TeamData <- TeamData %>%
  select(
    -"OT1",
    -"OT2",
    -"OT3",
    -"OT4",
    -"OT5",
  )
```

Get rid of bottom teams

```
#remove all teams that missed playoffs

non_playoff_teams <- c(
  "2024 Brooklyn", "2024 Toronto", "2024 Charlotte", "2024 Washington", "2024 Detroit",
  "2024 Houston", "2024 Utah", "2024 Memphis", "2024 San Antonio", "2024 Portland",
  "2023 Indiana", "2023 Washington", "2023 Orlando", "2023 Charlotte", "2023 Detroit",
  "2023 Dallas", "2023 Utah", "2023 Portland", "2023 Houston", "2023 San Antonio",
  "2022 New York", "2022 Washington", "2022 Indiana", "2022 Detroit", "2022 Orlando",
  "2022 LA Lakers", "2022 Sacramento", "2022 Portland", "2022 Oklahoma City", "2022 Houston",
  "2021 Cleveland", "2021 Miami", "2021 Orlando", "2021 Washington", "2021 Charlotte",
  "2021 Minnesota", "2021 Phoenix", "2021 Portland", "2021 Sacramento", "2021 Memphis",
  "2020 Washington", "2020 Charlotte", "2020 Chicago", "2020 New York", "2020 Detroit",
  "2020 Atlanta", "2020 Cleveland", "2020 Memphis", "2020 Phoenix", "2020 San Antonio",
  "2020 Sacramento", "2020 New Orleans", "2020 Minnesota", "2020 Golden State",
  "2019 Charlotte", "2019 Miami", "2019 Washington", "2019 Atlanta", "2019 Chicago",
  "2019 Cleveland", "2019 New York", "2019 Sacramento", "2019 LA Lakers", "2019 Minnesota",
  "2019 Memphis", "2019 New Orleans", "2019 Dallas", "2019 Phoenix",
  "2018 Detroit", "2018 Charlotte", "2018 New York", "2018 Brooklyn", "2018 Chicago",
  "2018 Orlando", "2018 Atlanta", "2018 Denver", "2018 LA Clippers", "2018 LA Lakers",
  "2018 Sacramento", "2018 Dallas", "2018 Memphis", "2018 Phoenix",
  "2017 Miami", "2017 Detroit", "2017 Charlotte", "2017 New York", "2017 Orlando",
  "2017 Philadelphia", "2017 Brooklyn", "2017 Denver", "2017 New Orleans", "2017 Dallas",
  "2017 Sacramento", "2017 Minnesota", "2017 LA Lakers", "2017 Phoenix",
  "2016 Chicago", "2016 Washington", "2016 Orlando", "2016 Milwaukee", "2016 New York",
  "2016 Brooklyn", "2016 Philadelphia", "2016 Utah", "2016 Sacramento", "2016 Denver",
  "2016 New Orleans", "2016 Minnesota", "2016 Phoenix", "2016 LA Lakers"
)

TeamData <- TeamData %>%
  filter(!`TeamYear` %in% non_playoff_teams)
```


Get rid of top players

```
#remove all stars, all nba team players, etc
```

```
all_nba_team_players <- c(
```

```
  # 2023-24
```

```
  "Giannis Antetokounmpo", "Luka Doncic", "Shai Gilgeous-Alexander", "Nikola Jokic", "Nikola Jokic", "Jalen Brunson", "Anthony Davis", "Kevin Durant", "Anthony Edwards", "Kawhi Leonard", "Devin Booker", "Stephen Curry", "Tyrese Haliburton", "LeBron James", "Domantas Sabonis",
```

```
  # 2022-23
```

```
  "Joel Embiid", "Jimmy Butler", "Jaylen Brown", "Donovan Mitchell", "Julius Randle", "De'Aaron Fox", "Damian Lillard",
```

```
  # 2021-22
```

```
  "DeMar DeRozan", "Kevin Durant", "Ja Morant", "Pascal Siakam", "Karl-Anthony Towns", "Chris Paul", "Trae Young",
```

```
  # 2020-21
```

```
  "Kawhi Leonard", "Julius Randle", "Jimmy Butler", "Paul George", "Rudy Gobert", "Bradley Beal", "Kyrie Irving",
```

```
  # 2019-20
```

```
  "Anthony Davis", "James Harden", "Chris Paul", "Ben Simmons", "Russell Westbrook",
```

```
  # 2018-19
```

```
  "Paul George", "Blake Griffin", "Kemba Walker",
```

```
  # 2017-18
```

```
  "Anthony Davis", "LaMarcus Aldridge", "Victor Oladipo",
```

```
  # 2016-17
```

```
  "Isaiah Thomas", "Draymond Green", "DeAndre Jordan", "John Wall",
```

```
  # 2015-16
```

```
  "DeMarcus Cousins", "Andre Drummond", "Klay Thompson", "Kyle Lowry"
```

```
)
```

```
all_rookie_players <- c(
```

```
  # 2023-24
```

```
  "Chet Holmgren", "Victor Wembanyama", "Brandon Miller", "Jaime Jaquez Jr.", "Brandin Podziemski", "Dereck Lively II", "Amen Thompson", "Keyonte George", "Cason Wallace", "GG Jackson II",
```

```
  # 2022-23
```

```
  "Paolo Banchero", "Walker Kessler", "Jalen Williams", "Bennedict Mathurin", "Keegan Murray", "Jalen Duren", "Jaden Ivey", "Tari Eason", "Jabari Smith Jr.", "Jeremy Sochan",
```

```
  # 2021-22
```

```
  "Scottie Barnes", "Cade Cunningham", "Evan Mobley", "Franz Wagner", "Jalen Green", "Herbert Jones", "Josh Giddey", "Bones Hyland", "Ayo Dosunmu", "Chris Duarte",
```

```
  # 2020-21
```

```
  "LaMelo Ball", "Anthony Edwards", "Tyrese Haliburton", "Saddiq Bey", "Jae'Sean Tate", "Immanuel Quickley", "Desmond Bane", "Isaiah Stewart", "Isaac Okoro", "Patrick Williams",
```

```
  # 2019-20
```

```
  "Ja Morant", "Kendrick Nunn", "Brandon Clarke", "Zion Williamson", "Eric Paschall", "Tyler Herro", "Terence Davis II", "Coby White", "P.J. Washington Jr.", "Rui Hachimura",
```

```
  # 2018-19
```

```
  "Luka Doncic", "Trae Young", "Deandre Ayton", "Jaren Jackson Jr.", "Marvin Bagley III", "Shai Gilgeous-Alexander", "Collin Sexton", "Landry Shamet", "Mitchell Robinson", "Kevin Huerter",
```

```
  # 2017-18
```

```
  "Kyle Kuzma", "Lauri Markkanen", "Donovan Mitchell", "Ben Simmons", "Jayson Tatum", "Lonzo Ball", "Bogdan Bogdanovic", "John Collins", "Josh Jackson", "Dennis Smith Jr.",
```

```
  # 2016-17
```

```
  "Malcolm Brogdon", "Dario Saric", "Joel Embiid", "Buddy Hield", "Willy Hernangomez", "Jamal Murray", "Jaylen Brown", "Marquese Chriss", "Brandon Ingram", "Yogi Ferrell",
```

```
  # 2015-16
```

```

"Karl-Anthony Towns", "Kristaps Porzingis", "Devin Booker", "Nikola Jokic", "Jahlil Okafor",
"Justise Winslow", "D'Angelo Russell", "Emmanuel Mudiay", "Myles Turner", "Willie Cauley-Stein"
)

all_defensive_players <- c(
  # 2023-24
  "Bam Adebayo", "Anthony Davis", "Rudy Gobert", "Herb Jones", "Victor Wembanyama",
  "Alex Caruso", "Jrue Holiday", "Jaden McDaniels", "Jalen Suggs", "Derrick White",
  # 2022-23
  "Jaren Jackson Jr.", "Brook Lopez", "Evan Mobley", "O.G. Anunoby", "Dillon Brooks",
  # 2021-22
  "Marcus Smart", "Mikal Bridges", "Giannis Antetokounmpo", "Matisse Thybulle", "Robert Williams III",
  # 2020-21
  "Ben Simmons", "Jimmy Butler", "Joel Embiid", "Kawhi Leonard",
  # 2019-20
  "Marcus Smart", "Patrick Beverley", "Eric Bledsoe",
  # 2018-19
  "Paul George", "Klay Thompson",
  # 2017-18
  "Victor Oladipo", "Robert Covington", "Al Horford", "Dejounte Murray",
  # 2016-17
  "Chris Paul", "Patrick Beverley", "Tony Allen", "Danny Green", "Andre Roberson",
  # 2015-16
  "DeAndre Jordan", "Avery Bradley", "Paul Millsap", "Hassan Whiteside"
)

```

#every all star ever - a little redundant but was easier and does the same thing

```

all_stars <- c(
  "LeBron James", "Kareem Abdul-Jabbar", "Kobe Bryant", "Julius Erving", "Tim Duncan",
  "Kevin Garnett", "Shaquille O'Neal", "Kevin Durant", "Michael Jordan", "Karl Malone",
  "Dirk Nowitzki", "Jerry West", "Wilt Chamberlain", "Bob Cousy", "John Havlicek",
  "Moses Malone", "Dwyane Wade", "Rick Barry", "Larry Bird", "George Gervin",
  "Elvin Hayes", "Magic Johnson", "Hakeem Olajuwon", "Chris Paul", "Oscar Robertson",
  "Bill Russell", "Dolph Schayes", "Isiah Thomas", "Charles Barkley", "Elgin Baylor",
  "Chris Bosh", "Patrick Ewing", "Artis Gilmore", "Allen Iverson", "Bob Pettit",
  "Ray Allen", "Carmelo Anthony", "Paul Arizin", "Stephen Curry", "Clyde Drexler",
  "Hal Greer", "James Harden", "Jason Kidd", "Paul Pierce", "David Robinson",
  "John Stockton", "Anthony Davis", "Paul George", "Robert Parish", "Gary Payton",
  "Russell Westbrook", "Lenny Wilkens", "Dominique Wilkins", "Giannis Antetokounmpo",
  "Vince Carter", "Dave Cowens", "Dave DeBusschere", "Alex English", "Larry Foust",
  "Dwight Howard",
  "Kyrie Irving", "Bob Lanier", "Damian Lillard", "Yao Ming", "Dikembe Mutombo",
  "Steve Nash", "Bill Sharman", "LaMarcus Aldridge", "Dave Bing", "Louie Dampier",
  "Mel Daniels", "Joel Embiid", "Walt Frazier", "Harry Gallatin", "Grant Hill",
  "Dan Issel", "Joe Johnson", "Jerry Lucas", "Ed Macauley", "Slater Martin",
  "Tracy McGrady", "Dick McGuire", "Kevin McHale", "Alonzo Mourning", "Scottie Pippen",
  "Willis Reed", "Jack Sikma", "Nate Thurmond", "Chet Walker", "Jo Jo White",
  "James Worthy", "Tiny Archibald", "Jimmy Butler", "Larry Costello", "Adrian Dantley",
  "Walter Davis", "DeMar DeRozan", "Joe Dumars", "Pau Gasol", "Blake Griffin",
  "Richie Guerin", "Cliff Hagan", "Connie Hawkins", "Tom Heinsohn", "Bailey Howell",
  "Lou Hudson", "Neil Johnston", "Nikola Jokić", "Jimmy Jones", "Shawn Kemp",
  "Kawhi Leonard", "Kyle Lowry", "George McGinnis", "Vern Mikkelsen", "Jermaine O'Neal",

```

"Tony Parker", "Mitch Richmond", "Amar'e Stoudemire", "Jack Twyman", "George Yardley",
 "Zelmo Beaty", "Chauncey Billups", "Carl Braun", "Mack Calvin", "Billy Cunningham",
 "Brad Daugherty", "Luka Dončić", "Wayne Embry", "Donnie Freeman", "Tom Gola",
 "Gail Goodrich", "Tim Hardaway", "Spencer Haywood", "Al Horford", "Dennis Johnson",
 "Gus Johnson", "Marques Johnson", "Bobby Jones", "Sam Jones", "Larry Kenon",
 "Rudy LaRusso", "Kevin Love", "Maurice Lucas", "Pete Maravich", "Bob McAdoo",
 "Reggie Miller", "Donovan Mitchell", "Sidney Moncrief", "Chris Mullin", "Don Ohl",
 "Andy Phillip", "Charlie Scott", "Gene Shue", "Ralph Simpson", "Jayson Tatum",
 "David Thompson", "Klay Thompson", "Rudy Tomjanovich", "Wes Unseld", "John Wall",
 "Bobby Wanzer", "Chris Webber", "Paul Westphal", "Vin Baker", "Walt Bellamy",
 "Otis Birdsong", "Rolando Blackman", "Devin Booker", "Ron Boone", "Roger Brown",
 "Joe Caldwell", "Tom Chambers", "Maurice Cheeks", "Doug Collins", "DeMarcus Cousins",
 "Bob Dandridge", "Bob Davies", "Dick Garmaker", "Draymond Green", "Johnny Green",
 "Anfernee Hardaway", "Mel Hutchins", "Warren Jabali", "Larry Jones", "Bernard King",
 "Bill Laimbeer", "Clyde Lovellette", "Shawn Marion", "George Mikan", "Paul Millsap",
 "Earl Monroe", "Willie Naulls", "Bob Netolicky", "Billy Paultz", "Jim Pollard",
 "Micheal Ray Richardson", "Arnie Risen", "Red Robbins", "Alvin Robertson", "Guy Rodgers",
 "Rajon Rondo", "Ralph Sampson", "Latrell Sprewell", "Karl-Anthony Towns", "Kemba Walker",
 "Ben Wallace", "Rasheed Wallace", "Sidney Wicks",
 "Bam Adebayo", "Mark Aguirre", "Gilbert Arenas", "Bradley Beal", "John Beasley",
 "Bill Bridges", "Jaylen Brown", "Larry Brown", "Darel Carrier", "Phil Chenier",
 "Glen Combs", "Terry Dischinger", "Steve Francis", "Marc Gasol", "Rudy Gobert",
 "Richard Hamilton", "Kevin Johnson", "Stew Johnson", "Eddie Jones", "Steve Jones",
 "Bob Kauffman", "Red Kerr", "Billy Knight", "Freddie Lewis", "Bob Love",
 "Dan Majerle", "Bill Melchionni", "Khris Middleton", "Doug Moe", "Jeff Mullins",
 "Larry Nance", "Julius Randle", "Glen Rice", "Derrick Rose", "Dan Roundfield",
 "Brandon Roy", "Domantas Sabonis", "Detlef Schrempf", "Paul Seymour", "Ben Simmons",
 "Peja Stojaković", "Maurice Stokes", "George Thompson", "Dick Van Arsdale",
 "Tom Van Arsdale",
 "Norm Van Lier", "Antoine Walker", "Jamaal Wilkes", "Buck Williams", "Deron Williams",
 "Willie Wise", "Trae Young", "Marvin Barnes", "Leo Barnhorst", "Byron Beck",
 "Art Becker", "Carlos Boozer", "Elton Brand", "Terrell Brandon", "Frankie Brian",
 "John Brisker", "Don Buse", "Caron Butler", "Archie Clark", "Terry Cummings",
 "Baron Davis", "Warren Davis", "Luol Deng", "John Drew", "Andre Drummond",
 "Kevin Duckworth", "Walter Dukes", "Dike Eddleman", "Anthony Edwards", "Sean Elliott",
 "Michael Finley", "Joe Fulks", "Jack George", "Shai Gilgeous-Alexander", "Manu Ginóbili",
 "Tyrese Haliburton", "Roy Hibbert", "Jrue Holiday", "Allan Houston", "Hot Rod Hundley",
 "Les Hunter", "Zydrunas Ilgauskas", "Antawn Jamison", "Eddie Johnson", "John Johnson",
 "Larry Johnson", "Rich Jones", "Don Kojis", "Wendell Ladner", "Zach LaVine",
 "David Lee", "Fat Lever", "Mike Lewis", "Rashard Lewis", "Jeff Malone",
 "Danny Manning", "Stephon Marbury", "Jack Marin", "Brad Miller", "Ja Morant",
 "Sven Nater", "Norm Nixon", "Joakim Noah", "Victor Oladipo", "Jim Paxson",
 "Geoff Petrie", "Terry Porter", "Cincy Powell", "Zach Randolph", "Glenn Robinson",
 "Truck Robinson", "Red Rocha", "Dennis Rodman", "Jeff Ruland", "Fred Scolari",
 "Kenny Sears", "Frank Selvy", "Pascal Siakam", "James Silas", "Paul Silas",
 "Jerry Sloan", "Phil Smith", "Randy Smith", "Jerry Stackhouse", "Levern Tart",
 "Brian Taylor", "Reggie Theus", "Isaiah Thomas", "Andrew Toney", "Kelly Tripucka",
 "Kiki Vandeweghe", "Bob Verga", "Nikola Vučević", "Nikola Vucevic", "Jimmy Walker",
 "Bill Walton",
 "Scott Wedman", "David West", "Charlie Williams", "Chuck Williams", "Gus Williams",
 "Zion Williamson", "Brian Winters",
 "Shareef Abdur-Rahim", "Alvan Adams", "Michael Adams", "Danny Ainge", "Jarrett Allen",
 "Kenny Anderson", "B.J. Armstrong", "LaMelo Ball", "Paolo Banchero", "Don Barksdale",

```

"Scottie Barnes", "Dick Barnett", "Dana Barros", "Butch Beard", "Ralph Beard",
"Mookie Blaylock", "John Block", "Bob Boozer", "Vince Boryla", "Bill Bradley",
"Fred Brown", "Roger Brown", "Jalen Brunson", "Larry Bunce", "Andrew Bynum",
"Austin Carr", "Joe Barry Carroll", "George Carter", "Bill Cartwright", "Sam Cassell",
"Cedric Ceballos", "Tyson Chandler", "Len Chappell", "Nat Clifton", "Derrick Coleman",
"Jack Coleman", "Mike Conley", "Antonio Davis", "Dale Davis", "Vlade Divac",
"James Donaldson", "Goran Dragić", "Jim Eakins", "Mark Eaton", "Dale Ellis",
"Ray Felix", "Sleepy Floyd", "Jimmy Foster", "De'Aaron Fox", "World B. Free",
"Bill Gabor", "Darius Garland", "Chris Gatling", "Gus Gerard", "Gerald Govan",
"Danny Granger", "Horace Grant", "A.C. Green", "Mike Green", "Rickey Green",
"Alex Groza", "Tom Gugliotta", "Devin Harris", "Bob Harrison", "Hersey Hawkins",
"Gordon Hayward", "Walt Hazzard", "Art Heyman", "Wayne Hightower", "Tyrone Hill",
"Lionel Hollins", "Jeff Hornacek", "Josh Howard", "Juwan Howard", "Andre Iguodala",
"Darrall Imhoff", "Brandon Ingram", "Jaren Jackson Jr.", "Luke Jackson", "Mark Jackson",
"Merv Jackson", "Tony Jackson", "Neil Johnson", "Steve Johnson", "Caldwell Jones",
"Wil Jones", "DeAndre Jordan", "Chris Kaman", "Julius Keye", "Jim King",
"Andrei Kirilenko", "Kyle Korver", "Sam Lacey", "Christian Laettner", "Clyde Lee",
"Reggie Lewis", "Goose Ligon", "Brook Lopez", "Jamaal Magloire", "Randy Mahaffey",
"Lauri Markkanen", "Kenyon Martin", "Jamal Mashburn", "Anthony Mason", "Tyrese Maxey",
"Ted McClain", "Xavier McDaniel", "Jim McDaniels", "Antonio McDyess", "Jon McGlocklin",
"Dewitt Menyard", "Tom Meschery", "Eddie Miles", "Mike Mitchell", "Steve Mix",
"Jack Molinas", "Gene Moore", "Calvin Murphy", "Dejounte Murray", "Calvin Natt",
"Jameer Nelson", "Chuck Noble", "Charles Oakley", "Mehmet Okur", "Ricky Pierce",
"Kristaps Porziņģis", "Jim Price", "Theo Ratliff", "Michael Redd", "Richie Regan",
"Doc Rivers", "Clifford Robinson", "Flynn Robinson", "Curtis Rowe", "Bob Rule",
"Campy Russell", "Cazzie Russell", "D'Angelo Russell", "Woody Sauldsberry", "Fred Schaus",
"Lee Shaffer", "Lonnie Shelton", "Walt Simon", "Adrian Smith", "Steve Smith",
"Rik Smits", "Willie Somerset", "John Starks", "Don Sunderlage", "Wally Szczerbiak",
"Jeff Teague", "Claude Terry", "Skip Thoren", "Otis Thorpe", "Monte Towe",
"Dave Twardzik", "Nick Van Exel", "Fred VanVleet", "Chico Vaughn", "Gerald Wallace",
"Paul Walther", "Ben Warley", "Kermit Washington", "Trooper Washington", "Andrew Wiggins",
"Jayson Williams", "Mo Williams", "Kevin Willis", "Metta World Peace", "Max Zaslofsky"
)

```

filter available players

```

library(dplyr)

# Step 1: Filter out games where players played fewer than 10 minutes
PlayerData <- PlayerData %>%
  filter(Minutes >= 10)

# Step 2: Filter players active in 2024
players_in_2024 <- PlayerData %>%
  filter(Year == 2024) %>%
  select(~PLAYER-ID~) %>%
  distinct()

PlayerData <- PlayerData %>%
  filter(~PLAYER-ID~ %in% players_in_2024$`PLAYER-ID~)

# Step 3: Calculate games played by each player in 2024
games_played_2024 <- PlayerData %>%

```

```

filter(Year == 2024) %>%
group_by(`PLAYER-ID`) %>%
summarize(Games = n_distinct(`GAME-ID`))

# Step 4: Filter players with at least 41 games in 2024
eligible_players <- games_played_2024 %>%
  filter(Games >= 41) %>%
  select(`PLAYER-ID`)

# Apply the final filter to the PlayerData
PlayerData <- PlayerData %>%
  filter(`PLAYER-ID` %in% eligible_players$`PLAYER-ID`)

```

give players a single position

```

# Calculate the most frequent position for each player
player_position_frequency <- PlayerData %>%
  group_by(`PLAYER-ID`, Player, POSITION) %>%
  summarize(Frequency = n(), .groups = 'drop') %>%
  arrange(`PLAYER-ID`, desc(Frequency)) %>%
  group_by(`PLAYER-ID`, Player) %>%
  slice_max(Frequency, with_ties = FALSE) %>%
  select(`PLAYER-ID`, Player, MostFrequentPosition = POSITION) %>%
  ungroup()

# Merge the most frequent position back to the original dataframe
PlayerData <- PlayerData %>%
  left_join(player_position_frequency, by = c("PLAYER-ID", "Player")) %>%
  mutate(POSITION = MostFrequentPosition) %>%
  select(-MostFrequentPosition) # Remove the helper column

```

create player availability lists all - __ alien - A role - V V + <3 years - S

```

PlayerDataA <-
  PlayerData %>% filter(Year == 2024)

PlayerDataV <- PlayerData %>%
  filter(!`Player` %in% c(all_stars, all_nba_team_players, all_rookie_players, all_defensive_players))

# Identify players who played before the 2021-2022 season
players_to_remove <- PlayerDataV %>%
  filter(Year < 2022) %>%
  pull(`PLAYER-ID`) %>%
  unique()

# Remove these players from PlayerData
PlayerDataS <- PlayerDataV %>%
  filter(!(`PLAYER-ID` %in% players_to_remove))

```

#write as excel

```

write.xlsx(PlayerData, "PlayerData.xlsx")
write.xlsx(PlayerDataA, "PlayerDataA.xlsx")
write.xlsx(PlayerDataS, "PlayerDataS.xlsx")
write.xlsx(PlayerDataV, "PlayerDataV.xlsx")
write.xlsx(TeamData, "TeamData.xlsx")

```

playoff teams averages

```

# Load dplyr package
library(dplyr)

# Calculate the team averages for the specified variables
team_averages <- TeamData %>%
  group_by(TeamYear) %>%
  summarize(
    AVGFieldGoals = mean(FieldGoals, na.rm = TRUE),
    AVGFieldGoalsAttempted = mean(FieldGoalsAttempted, na.rm = TRUE),
    AVGThreePointers = mean(ThreePointers, na.rm = TRUE),
    AVGThreePointAttempts = mean(ThreePointAttempts, na.rm = TRUE),
    AVGFreeThrows = mean(FreeThrows, na.rm = TRUE),
    AVGFreeThrowAttempts = mean(FreeThrowAttempts, na.rm = TRUE),
    AVGOffensiveRebounds = mean(OffensiveRebounds, na.rm = TRUE),
    AVGDefensiveRebounds = mean(DefensiveRebounds, na.rm = TRUE),
    AVGTotalsRebounds = mean(TotalRebounds, na.rm = TRUE),
    AVGAssists = mean(Assists, na.rm = TRUE),
    AVGPPersonalFouls = mean(PersonalFouls, na.rm = TRUE),
    AVGSteals = mean(Steals, na.rm = TRUE),
    AVGTurnovers = mean(Turnovers, na.rm = TRUE),
    AVGBlocks = mean(Blocks, na.rm = TRUE),
    AVGPpoints = mean(Points, na.rm = TRUE),
    AVGPossessions = mean(POSS, na.rm = TRUE),
    AVGPace = mean(PACE, na.rm = TRUE),
    AVGOEFF = mean(OEFF, na.rm = TRUE),
    AVGDEFF = mean(DEFF, na.rm = TRUE),
    AVGPRA = mean(PRA, na.rm = TRUE),
    AVGPA = mean(PA, na.rm = TRUE),
    AVGPGR = mean(PR, na.rm = TRUE),
    AVGRA = mean(RA, na.rm = TRUE),
    AVGSTocks = mean(Stocks, na.rm = TRUE)
  )

print(team_averages)

```

```

## # A tibble: 160 x 25
##   TeamYear      AVGFieldGoals AVGFieldGoalsAttempted AVGThreePointers
##   <chr>          <dbl>          <dbl>          <dbl>
## 1 2016 Atlanta      38.6            84.4            9.94
## 2 2016 Boston      39.2            89.2            8.74
## 3 2016 Charlotte    37.0            84.4           10.6
## 4 2016 Cleveland    38.7            84             10.7
## 5 2016 Dallas       37.4            84.1            9.83
## 6 2016 Detroit      37.9            86.4            9.02
## 7 2016 Golden State 42.5            87.3           13.1

```

```
## 8 2016 Houston          37.7          83.5          10.7
## 9 2016 Indiana          38.3          85.2           8.09
## 10 2016 LA Clippers     38.3          82.4           9.72
## # i 150 more rows
## # i 21 more variables: AVGThreePointAttempts <dbl>, AVGFreeThrows <dbl>,
## #   AVGFreeThrowAttempts <dbl>, AVGOffensiveRebounds <dbl>,
## #   AVGDefensiveRebounds <dbl>, AVGTotalsRebounds <dbl>, AVGAssists <dbl>,
## #   AVGPPersonalFouls <dbl>, AVGSteals <dbl>, AVGTurnovers <dbl>,
## #   AVGBlocks <dbl>, AVGPPoints <dbl>, AVGPossessions <dbl>, AVGPace <dbl>,
## #   AVGOEFF <dbl>, AVGDEFF <dbl>, AVGPRA <dbl>, AVGPA <dbl>, AVGP <dbl>, ...
```

playoff team league average

```
# Calculate the total averages for the specified variables
total_averages <- TeamData %>%
  summarize(
    AVGFieldGoals = mean(FieldGoals, na.rm = TRUE),
    AVGFieldGoalsAttempted = mean(FieldGoalsAttempted, na.rm = TRUE),
    AVGThreePointers = mean(ThreePointers, na.rm = TRUE),
    AVGThreePointAttempts = mean(ThreePointAttempts, na.rm = TRUE),
    AVGFreeThrows = mean(FreeThrows, na.rm = TRUE),
    AVGFreeThrowAttempts = mean(FreeThrowAttempts, na.rm = TRUE),
    AVGOffensiveRebounds = mean(OffensiveRebounds, na.rm = TRUE),
    AVGDefensiveRebounds = mean(DefensiveRebounds, na.rm = TRUE),
    AVGTotalsRebounds = mean(TotalRebounds, na.rm = TRUE),
    AVGAssists = mean(Assists, na.rm = TRUE),
    AVGPPersonalFouls = mean(PersonalFouls, na.rm = TRUE),
    AVGSteals = mean(Steals, na.rm = TRUE),
    AVGTurnovers = mean(Turnovers, na.rm = TRUE),
    AVGBlocks = mean(Blocks, na.rm = TRUE),
    AVGPPoints = mean(Points, na.rm = TRUE),
    AVGPossessions = mean(POSS, na.rm = TRUE),
    AVGPRA = mean(PRA, na.rm = TRUE),
    AVGPA = mean(PA, na.rm = TRUE),
    AVGP <= mean(PR, na.rm = TRUE),
    AVGRA = mean(RA, na.rm = TRUE),
    AVGSTocks = mean(Stocks, na.rm = TRUE)
  )
print(total_averages)
```

```
## # A tibble: 1 x 21
##   AVGFieldGoals AVGFieldGoalsAttempted AVGThreePointers AVGThreePointAttempts
##   <dbl>          <dbl>          <dbl>          <dbl>
## 1      41.0        87.4        11.8        32.3
## # i 17 more variables: AVGFreeThrows <dbl>, AVGFreeThrowAttempts <dbl>,
## #   AVGOffensiveRebounds <dbl>, AVGDefensiveRebounds <dbl>,
## #   AVGTotalsRebounds <dbl>, AVGAssists <dbl>, AVGPPersonalFouls <dbl>,
## #   AVGSteals <dbl>, AVGTurnovers <dbl>, AVGBlocks <dbl>, AVGPPoints <dbl>,
## #   AVGPossessions <dbl>, AVGPRA <dbl>, AVGPA <dbl>, AVGP <dbl>, AVGRA <dbl>,
## #   AVGSTocks <dbl>
```

Player stats - easier to run in tab


```

#need minimum minutes

# Function to calculate the top 10 players for each numerical category
top_players <- function(data, category) {
  data %>%
    group_by(Player) %>%
    summarize(Average = mean(!is.na(category), na.rm = TRUE)) %>%
    arrange(desc(Average)) %>%
    top_n(10, Average)
}

top_FieldGoals_players <- top_players(PlayerData, "FieldGoals")
top_FieldGoalsAttempted_players <- top_players(PlayerData, "FieldGoalsAttempted")
top_ThreePointers_players <- top_players(PlayerData, "ThreePointers")
top_ThreePointAttempts_players <- top_players(PlayerData, "ThreePointAttempts")
top_FreeThrows_players <- top_players(PlayerData, "FreeThrows")
top_FreeThrowAttempts_players <- top_players(PlayerData, "FreeThrowAttempts")
top_OffensiveRebounds_players <- top_players(PlayerData, "OffensiveRebounds")
top_DefensiveRebounds_players <- top_players(PlayerData, "DefensiveRebounds")
top_TotalRebounds_players <- top_players(PlayerData, "TotalRebounds")
top_Assists_players <- top_players(PlayerData, "Assists")
top_PersonalFouls_players <- top_players(PlayerData, "PersonalFouls")
top_Steals_players <- top_players(PlayerData, "Steals")
top_Turnovers_players <- top_players(PlayerData, "Turnovers")
top_Blocks_players <- top_players(PlayerData, "Blocks")
top_Points_players <- top_players(PlayerData, "Points")
#top_UsageRate_players <- top_players(PlayerData, "UsageRate")
#top_PRA_players <- top_players(PlayerData, "PRA")
#top_PA_players <- top_players(PlayerData, "PA")
#top_PR_players <- top_players(PlayerData, "PR")
#top_RA_players <- top_players(PlayerData, "RA")
#top_Stocks_players <- top_players(PlayerData, "Stocks")

top_PointsPerMinute_players <- top_players(PlayerData, "PointsPerMinute")
top_FieldGoalsPerMinute_players <- top_players(PlayerData, "FieldGoalsPerMinute")
top_FieldGoalsAttemptedPerMinute_players <- top_players(PlayerData, "FieldGoalsAttemptedPerMinute")
top_ThreePointersPerMinute_players <- top_players(PlayerData, "ThreePointersPerMinute")
top_ThreePointAttemptsPerMinute_players <- top_players(PlayerData, "ThreePointAttemptsPerMinute")
top_FreeThrowsPerMinute_players <- top_players(PlayerData, "FreeThrowsPerMinute")
top_FreeThrowAttemptsPerMinute_players <- top_players(PlayerData, "FreeThrowAttemptsPerMinute")
top_OffensiveReboundsPerMinute_players <- top_players(PlayerData, "OffensiveReboundsPerMinute")
top_DefensiveReboundsPerMinute_players <- top_players(PlayerData, "DefensiveReboundsPerMinute")
top_TotalReboundsPerMinute_players <- top_players(PlayerData, "TotalReboundsPerMinute")
top_AssistsPerMinute_players <- top_players(PlayerData, "AssistsPerMinute")
top_PersonalFoulsPerMinute_players <- top_players(PlayerData, "PersonalFoulsPerMinute")
top_StealsPerMinute_players <- top_players(PlayerData, "StealsPerMinute")
top_TurnoversPerMinute_players <- top_players(PlayerData, "TurnoversPerMinute")
top_BlocksPerMinute_players <- top_players(PlayerData, "BlocksPerMinute")
#top_PRAPerMinute_players <- top_players(PlayerData, "PRAPerMinute")
#top_PAPerMinute_players <- top_players(PlayerData, "PAPerMinute")

```



```

#top_PRRPerMinute_players <- top_players(PlayerData, "PRPerMinute")
#top_RAPerMinute_players <- top_players(PlayerData, "RAPerMinute")
#top_StocksPerMinute_players <- top_players(PlayerData, "StocksPerMinute")

# Print the results
list(
  # FieldGoals = top_FieldGoals_players,
  FieldGoalsAttempted = top_FieldGoalsAttempted_players,
  # ThreePointers = top_ThreePointers_players,
  # ThreePointAttempts = top_ThreePointAttempts_players,
  # FreeThrows = top_FreeThrows_players,
  # FreeThrowAttempts = top_FreeThrowAttempts_players,
  # OffensiveRebounds = top_OffensiveRebounds_players,
  # DefensiveRebounds = top_DefensiveRebounds_players,
  # TotalRebounds = top_TotalRebounds_players,
  # Assists = top_Assists_players,
  # PersonalFouls = top_PersonalFouls_players,
  # Steals = top_Steals_players,
  # Turnovers = top_Turnovers_players,
  # Blocks = top_Blocks_players,
  Points = top_Points_players,
  # UsageRate = top_UsageRate_players,
  # PRA = top_PRA_players,
  # PA = top_PA_players,
  # PR = top_PR_players,
  # RA = top_RA_players,
  # Stocks = top_Stocks_players,
  PointsPerMinute = top_PointsPerMinute_players
  # FieldGoalsPerMinute = top_FieldGoalsPerMinute_players,
  # FieldGoalsAttemptedPerMinute = top_FieldGoalsAttemptedPerMinute_players,
  # ThreePointersPerMinute = top_ThreePointersPerMinute_players,
  # ThreePointAttemptsPerMinute = top_ThreePointAttemptsPerMinute_players,
  # FreeThrowsPerMinute = top_FreeThrowsPerMinute_players,
  # FreeThrowAttemptsPerMinute = top_FreeThrowAttemptsPerMinute_players,
  # OffensiveReboundsPerMinute = top_OffensiveReboundsPerMinute_players,
  # DefensiveReboundsPerMinute = top_DefensiveReboundsPerMinute_players,
  # TotalReboundsPerMinute = top_TotalReboundsPerMinute_players,
  # AssistsPerMinute = top_AssistsPerMinute_players,
  # PersonalFoulsPerMinute = top_PersonalFoulsPerMinute_players,
  # StealsPerMinute = top_StealsPerMinute_players,
  # TurnoversPerMinute = top_TurnoversPerMinute_players,
  # BlocksPerMinute = top_BlocksPerMinute_players,
  #PRAPerMinute = top_PRAPerMinute_players
  # PAPERMinute = top_PAPERMinute_players,
  # PRPerMinute = top_PRRPerMinute_players,
  # RAPerMinute = top_RAPerMinute_players,
  # StocksPerMinute = top_StocksPerMinute_players
)

## $FieldGoalsAttempted
## # A tibble: 10 x 2
##   Player      Average
##   <chr>      <dbl>

```

```
## 1 Luka Doncic          20.8
## 2 Donovan Mitchell     19.6
## 3 Damian Lillard       19.5
## 4 Stephen Curry        19.5
## 5 LeBron James         19.4
## 6 Kyrie Irving         19.3
## 7 Devin Booker         18.7
## 8 Bradley Beal         18.6
## 9 Trae Young           18.6
## 10 James Harden        18.5
##
## $Points
## # A tibble: 10 x 2
##   Player          Average
##   <chr>          <dbl>
## 1 Luka Doncic      28.8
## 2 Stephen Curry    27.8
## 3 James Harden     27.6
## 4 Kevin Durant     27.2
## 5 Damian Lillard   27.1
## 6 Giannis Antetokounmpo 26.9
## 7 LeBron James     26.8
## 8 Anthony Davis    26.0
## 9 Trae Young       25.5
## 10 Kyrie Irving     25.1
##
## $PointsPerMinute
## # A tibble: 10 x 2
##   Player          Average
##   <chr>          <dbl>
## 1 Stephen Curry    0.820
## 2 Luka Doncic      0.820
## 3 Giannis Antetokounmpo 0.796
## 4 Zion Williamson  0.779
## 5 Kevin Durant     0.771
## 6 James Harden     0.752
## 7 Damian Lillard   0.751
## 8 LeBron James     0.748
## 9 Trae Young       0.741
## 10 Anthony Davis    0.738
```

Find roster

```
# Define positive and negative metrics
positive_metrics <- c("PointsPerMinute", "FieldGoalsPerMinute", "ThreePointersPerMinute", "AssistsPerMinute")

negative_metrics <- c("FieldGoalsAttemptedPerMinute")

# Normalize metrics to range [0, 1]
normalize <- function(x) {
  if (max(x, na.rm = TRUE) == min(x, na.rm = TRUE)) {
    return(rep(0.5, length(x)))
  } else {
    return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
  }
}
```

```

}
}

# Compute career averages for each player
career_averages <- PlayerData %>%
  group_by(`PLAYER-ID`, Player, POSITION) %>%
  summarize(across(all_of(positive_metrics), mean, na.rm = TRUE),
            across(all_of(negative_metrics), mean, na.rm = TRUE), .groups = 'drop')

## Warning: There was 1 warning in `summarize()`.
## i In argument: `across(all_of(positive_metrics), mean, na.rm = TRUE)`.
## i In group 1: `PLAYER-ID = 2544`, `Player = "LeBron James"`, `POSITION = "F"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
## # Previously
##   across(a:b, mean, na.rm = TRUE)
##
## # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))

# Normalize the career averages
career_averages_normalized <- career_averages %>%
  mutate(across(all_of(positive_metrics), ~ normalize(.))) %>%
  mutate(across(all_of(negative_metrics), ~ 1 - normalize(.)))

# Compute the overall career efficiency score
career_averages_normalized <- career_averages_normalized %>%
  rowwise() %>%
  mutate(Efficiency = sum(c_across(all_of(positive_metrics))) - sum(c_across(all_of(negative_metrics))))
  ungroup()

# Select top players by position
top_4_guards <- career_averages_normalized %>%
  filter(POSITION == "G") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_4_forwards <- career_averages_normalized %>%
  filter(POSITION == "F") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_2_centers <- career_averages_normalized %>%
  filter(POSITION == "C") %>%
  arrange(desc(Efficiency)) %>%
  head(2)

# Combine the top players
top_10_career_efficiency <- bind_rows(top_4_guards, top_4_forwards, top_2_centers)

# Print the results

```

```
print(top_10_career_efficiency)
```

```
## # A tibble: 10 x 14
##   'PLAYER-ID' Player      POSITION PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>
## 1 201939 Stephen Curry    G              1              0.871
## 2 201566 Russell Westbrook G              0.716          0.711
## 3 201935 James Harden    G              0.885          0.635
## 4 202681 Kyrie Irving     G              0.841          0.858
## 5 1641705 Victor Wembanyama F              0.848          0.818
## 6 1629029 Luka Doncic     F              1.00           0.894
## 7 203507 Giannis Antetokounm~ F              0.960          0.948
## 8 203076 Anthony Davis    F              0.861          0.875
## 9 203999 Nikola Jokic     C              0.729          0.781
## 10 203083 Andre Drummond   C              0.424          0.525
## # i 9 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>, Efficiency <dbl>
```

actual stats + team stats

```
library(dplyr)

# Extract the PLAYER-IDs of the top 10 most efficient players
top_10_players <- top_10_career_efficiency$`PLAYER-ID`

# Filter the original dataset to include only the top 10 players
top_10_player_stats <- PlayerData %>%
  filter(`PLAYER-ID` %in% top_10_players)

# Summarize the true stats for the top 10 players and ensure only one row per player
top_10_player_summary <- top_10_player_stats %>%
  group_by(`PLAYER-ID`) %>%
  summarize(
    Player = first(Player),
    Position = first(POSITION),
    across(c(positive_metrics, negative_metrics), mean, na.rm = TRUE),
    .groups = 'drop'
  )
```

```
## Warning: There were 2 warnings in 'summarize()'.
## The first warning was:
## i In argument: 'across(c(positive_metrics, negative_metrics), mean, na.rm =
##   TRUE)'.
## i In group 1: 'PLAYER-ID = 201566'.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
## # Was:
## data %>% select(positive_metrics)
##
```

```
## # Now:
## data %>% select(all_of(positive_metrics))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## i Run 'dplyr::last_dplyr_warnings()' to see the 1 remaining warning.

# Calculate the aggregated team metrics by summing the metrics
team_summary <- top_10_player_summary %>%
  summarize(
    `PLAYER-ID` = NA,
    Player = "TEAMxMINUTE",
    Position = "TEAMxMINUTE",
    across(c(positive_metrics, negative_metrics), sum, na.rm = TRUE) # Use sum for total values
  )

# Create the TEAMxGAME row by multiplying the TEAM row values by 24
teamxgame_summary <- team_summary %>%
  mutate(
    Player = "TEAMxGAME",
    Position = "TEAMxGAME",
    across(c(positive_metrics, negative_metrics), ~ . * 24)
  )

# Combine the summaries
combined_summary <- bind_rows(top_10_player_summary, team_summary, teamxgame_summary)

# Print the combined summary
print(combined_summary)
```

```
## # A tibble: 12 x 13
##   `PLAYER-ID` Player      Position PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>      <chr>      <dbl>      <dbl>
## 1 201566 Russell Westbrook G          0.652      0.239
## 2 201935 James Harden    G          0.752      0.222
## 3 201939 Stephen Curry   G          0.820      0.273
## 4 202681 Kyrie Irving     G          0.726      0.270
## 5 203076 Anthony Davis    F          0.738      0.274
## 6 203083 Andre Drummond    C          0.479      0.199
## 7 203507 Giannis Antetokounm~ F          0.796      0.290
## 8 203999 Nikola Jokic     C          0.654      0.253
## 9 1629029 Luka Doncic     F          0.820      0.278
## 10 1641705 Victor Wembanyama F          0.730      0.262
## 11 NA TEAMxMINUTE         TEAMxMI~    7.17      2.56
## 12 NA TEAMxGAME          TEAMxGA~   172.     61.4
## # i 8 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## # TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## # DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## # BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>
```

Alien

```
# Define positive and negative metrics
positive_metrics <- c("PointsPerMinute", "FieldGoalsPerMinute", "ThreePointersPerMinute", "AssistsPerMinute", "TotalReboundsPerMinute", "OffensiveReboundsPerMinute", "DefensiveReboundsPerMinute", "StealsPerMinute", "BlocksPerMinute", "FieldGoalsAttemptedPerMinute")
```

```

negative_metrics <- c("FieldGoalsAttemptedPerMinute")

# Normalize metrics to range [0, 1]
normalize <- function(x) {
  if (max(x, na.rm = TRUE) == min(x, na.rm = TRUE)) {
    return(rep(0.5, length(x)))
  } else {
    return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
  }
}

# Compute career averages for each player
career_averages <- PlayerDataA %>%
  group_by(`PLAYER-ID`, Player, POSITION) %>%
  summarize(across(all_of(positive_metrics), mean, na.rm = TRUE),
            across(all_of(negative_metrics), mean, na.rm = TRUE), .groups = 'drop')

# Normalize the career averages
career_averages_normalized <- career_averages %>%
  mutate(across(all_of(positive_metrics), ~ normalize(.))) %>%
  mutate(across(all_of(negative_metrics), ~ 1 - normalize(.)))

# Compute the overall career efficiency score
career_averages_normalized <- career_averages_normalized %>%
  rowwise() %>%
  mutate(Efficiency = sum(c_across(all_of(positive_metrics))) - sum(c_across(all_of(negative_metrics))))
  ungroup()

# Select top players by position
top_4_guards <- career_averages_normalized %>%
  filter(POSITION == "G") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_4_forwards <- career_averages_normalized %>%
  filter(POSITION == "F") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_2_centers <- career_averages_normalized %>%
  filter(POSITION == "C") %>%
  arrange(desc(Efficiency)) %>%
  head(2)

# Combine the top players
top_10_career_efficiency <- bind_rows(top_4_guards, top_4_forwards, top_2_centers)

# Print the results
print(top_10_career_efficiency)

```

```

## # A tibble: 10 x 14
##   'PLAYER-ID' Player      POSITION PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>

```

```
## 1      1628983 Shai Gilgeous-Alexa~ G      0.979      0.933
## 2      1628378 Donovan Mitchell    G      0.768      0.716
## 3      201939  Stephen Curry        G      0.838      0.751
## 4      1628368 De'Aaron Fox         G      0.749      0.770
## 5      1641705 Victor Wembanyama    F      0.751      0.743
## 6      1629029 Luka Doncic          F      1          0.912
## 7      203507  Giannis Antetokounm~ F      0.950      1
## 8      203076  Anthony Davis        F      0.700      0.753
## 9      203999  Nikola Jokic         C      0.779      0.886
## 10     203083  Andre Drummond       C      0.394      0.474
## # i 9 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>, Efficiency <dbl>
```

```
library(dplyr)

# Extract the PLAYER-IDs of the top 10 most efficient players
top_10_players <- top_10_career_efficiency$`PLAYER-ID`

# Filter the original dataset to include only the top 10 players
top_10_player_stats <- PlayerDataA %>%
  filter(`PLAYER-ID` %in% top_10_players)

# Summarize the true stats for the top 10 players and ensure only one row per player
top_10_player_summary <- top_10_player_stats %>%
  group_by(`PLAYER-ID`) %>%
  summarize(
    Player = first(Player),
    Position = first(POSITION),
    across(c(positive_metrics, negative_metrics), mean, na.rm = TRUE),
    .groups = 'drop'
  )

# Calculate the aggregated team metrics by summing the metrics
team_summary <- top_10_player_summary %>%
  summarize(
    `PLAYER-ID` = NA,
    Player = "TEAMxMINUTE",
    Position = "TEAMxMINUTE",
    across(c(positive_metrics, negative_metrics), sum, na.rm = TRUE) # Use sum for total values
  )

# Create the TEAMxGAME row by multiplying the TEAM row values by 24
teamxgame_summary <- team_summary %>%
  mutate(
    Player = "TEAMxGAME",
    Position = "TEAMxGAME",
    across(c(positive_metrics, negative_metrics), ~ . * 24)
  )

# Combine the summaries
combined_summary <- bind_rows(top_10_player_summary, team_summary, teamxgame_summary)
```

```
# Print the combined summary
print(combined_summary)
```

```
## # A tibble: 12 x 13
##   'PLAYER-ID' Player      Position PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>
## 1 201939 Stephen Curry      G              0.790          0.264
## 2 203076 Anthony Davis      F              0.695          0.265
## 3 203083 Andre Drummond      C              0.484          0.192
## 4 203507 Giannis Antetokounm~ F              0.868          0.328
## 5 203999 Nikola Jokic        C              0.750          0.299
## 6 1628368 De'Aaron Fox        G              0.729          0.269
## 7 1628378 Donovan Mitchell      G              0.742          0.255
## 8 1628983 Shai Gilgeous-Alexa~ G              0.888          0.311
## 9 1629029 Luka Doncic        F              0.902          0.305
## 10 1641705 Victor Wembanyama    F              0.730          0.262
## 11 NA TEAMxMINUTE            TEAMxMI~        7.58          2.75
## 12 NA TEAMxGAME             TEAMxGA~       182.          66.0
## # i 8 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>
```

Vegas

```
# Define positive and negative metrics
positive_metrics <- c("PointsPerMinute", "FieldGoalsPerMinute", "ThreePointersPerMinute", "AssistsPerMinute")

negative_metrics <- c("FieldGoalsAttemptedPerMinute")

# Normalize metrics to range [0, 1]
normalize <- function(x) {
  if (max(x, na.rm = TRUE) == min(x, na.rm = TRUE)) {
    return(rep(0.5, length(x)))
  } else {
    return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
  }
}

# Compute career averages for each player
career_averages <- PlayerDataV %>%
  group_by(`PLAYER-ID`, Player, POSITION) %>%
  summarize(across(all_of(positive_metrics), mean, na.rm = TRUE),
            across(all_of(negative_metrics), mean, na.rm = TRUE), .groups = 'drop')

# Normalize the career averages
career_averages_normalized <- career_averages %>%
  mutate(across(all_of(positive_metrics), ~ normalize(.))) %>%
  mutate(across(all_of(negative_metrics), ~ 1 - normalize(.)))

# Compute the overall career efficiency score
career_averages_normalized <- career_averages_normalized %>%
  rowwise() %>%
```



```

mutate(Efficiency = sum(c_across(all_of(positive_metrics))) - sum(c_across(all_of(negative_metrics))))
ungroup()

# Select top players by position
top_4_guards <- career_averages_normalized %>%
  filter(POSITION == "G") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_4_forwards <- career_averages_normalized %>%
  filter(POSITION == "F") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_2_centers <- career_averages_normalized %>%
  filter(POSITION == "C") %>%
  arrange(desc(Efficiency)) %>%
  head(2)

# Combine the top players
top_10_career_efficiency <- bind_rows(top_4_guards, top_4_forwards, top_2_centers)

# Print the results
print(top_10_career_efficiency)

```

```

## # A tibble: 10 x 14
##   'PLAYER-ID' Player      POSITION PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>
## 1    203468 CJ McCollum    G              0.997          1
## 2    203903 Jordan Clarkson  G              0.898          0.871
## 3    1629673 Jordan Poole    G              0.892          0.721
## 4    1630175 Cole Anthony  G              0.685          0.575
## 5    1630188 Jalen Smith    F              0.781          0.776
## 6    1630194 Paul Reed    F              0.428          0.576
## 7    1626171 Bobby Portis   F              0.739          0.821
## 8    1631218 Trayce Jackson-Davis F              0.656          0.842
## 9     203994 Jusuf Nurkic    C              0.682          0.720
## 10   1630549 Day'Ron Sharpe  C              0.592          0.692
## # i 9 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>, Efficiency <dbl>

```

```

library(dplyr)

# Extract the PLAYER-IDs of the top 10 most efficient players
top_10_players <- top_10_career_efficiency$`PLAYER-ID`

# Filter the original dataset to include only the top 10 players
top_10_player_stats <- PlayerDataV %>%
  filter(`PLAYER-ID` %in% top_10_players)

# Summarize the true stats for the top 10 players and ensure only one row per player

```

```

top_10_player_summary <- top_10_player_stats %>%
  group_by(`PLAYER-ID`) %>%
  summarize(
    Player = first(Player),
    Position = first(POSITION),
    across(c(positive_metrics, negative_metrics), mean, na.rm = TRUE),
    .groups = 'drop'
  )

# Calculate the aggregated team metrics by summing the metrics
team_summary <- top_10_player_summary %>%
  summarize(
    `PLAYER-ID` = NA,
    Player = "TEAMxMINUTE",
    Position = "TEAMxMINUTE",
    across(c(positive_metrics, negative_metrics), sum, na.rm = TRUE) # Use sum for total values
  )

# Create the TEAMxGAME row by multiplying the TEAM row values by 24
teamxgame_summary <- team_summary %>%
  mutate(
    Player = "TEAMxGAME",
    Position = "TEAMxGAME",
    across(c(positive_metrics, negative_metrics), ~ . * 24)
  )

# Combine the summaries
combined_summary <- bind_rows(top_10_player_summary, team_summary, teamxgame_summary)

# Print the combined summary
print(combined_summary)

```

```

## # A tibble: 12 x 13
##   `PLAYER-ID` Player      Position PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>      <chr>      <dbl>      <dbl>
## 1 203468 CJ McCollum      G          0.618      0.236
## 2 203903 Jordan Clarkson  G          0.580      0.217
## 3 203994 Jusuf Nurkic     C          0.497      0.196
## 4 1626171 Bobby Portis    F          0.519      0.210
## 5 1629673 Jordan Poole     G          0.577      0.196
## 6 1630175 Cole Anthony     G          0.498      0.175
## 7 1630188 Jalen Smith         F          0.535      0.204
## 8 1630194 Paul Reed       F          0.399      0.175
## 9 1630549 Day'Ron Sharpe    C          0.462      0.192
## 10 1631218 Trayce Jackson-Davis F          0.487      0.213
## 11 NA TEAMxMINUTE      TEAMxMI~    5.17      2.01
## 12 NA TEAMxGAME        TEAMxGA~   124.      48.3
## # i 8 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## # TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## # DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## # BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>

```

Seattle

```

# Define positive and negative metrics
positive_metrics <- c("PointsPerMinute", "FieldGoalsPerMinute", "ThreePointersPerMinute", "AssistsPerMinute")

negative_metrics <- c("FieldGoalsAttemptedPerMinute")

# Normalize metrics to range [0, 1]
normalize <- function(x) {
  if (max(x, na.rm = TRUE) == min(x, na.rm = TRUE)) {
    return(rep(0.5, length(x)))
  } else {
    return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE)))
  }
}

# Compute career averages for each player
career_averages <- PlayerDataS %>%
  group_by(`PLAYER-ID`, Player, POSITION) %>%
  summarize(across(all_of(positive_metrics), mean, na.rm = TRUE),
            across(all_of(negative_metrics), mean, na.rm = TRUE), .groups = 'drop')

# Normalize the career averages
career_averages_normalized <- career_averages %>%
  mutate(across(all_of(positive_metrics), ~ normalize(.))) %>%
  mutate(across(all_of(negative_metrics), ~ 1 - normalize(.)))

# Compute the overall career efficiency score
career_averages_normalized <- career_averages_normalized %>%
  rowwise() %>%
  mutate(Efficiency = sum(c_across(all_of(positive_metrics))) - sum(c_across(all_of(negative_metrics))))
  ungroup()

# Select top players by position
top_4_guards <- career_averages_normalized %>%
  filter(POSITION == "G") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_4_forwards <- career_averages_normalized %>%
  filter(POSITION == "F") %>%
  arrange(desc(Efficiency)) %>%
  head(4)

top_2_centers <- career_averages_normalized %>%
  filter(POSITION == "C") %>%
  arrange(desc(Efficiency)) %>%
  head(2)

# Combine the top players
top_10_career_efficiency <- bind_rows(top_4_guards, top_4_forwards, top_2_centers)

# Print the results
print(top_10_career_efficiency)

```

```
## # A tibble: 10 x 14
##   'PLAYER-ID' Player      POSITION PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>
## 1 1630560 Cam Thomas      G              1              1
## 2 1630702 Jaden Hardy        G             0.793          0.749
## 3 1631204 Marcus Sasser    G             0.556          0.571
## 4 1641709 Ausar Thompson    G             0.261          0.360
## 5 1631218 Trayce Jackson-Davis F             0.656          0.924
## 6 1630552 Jalen Johnson        F             0.519          0.635
## 7 1630228 Jonathan Kuminga  F             0.803          0.856
## 8 1641713 GG Jackson        F             0.812          0.732
## 9 1630549 Day'Ron Sharpe    C             0.592          0.759
## 10 1630578 Alperen Sengun     C             0.778          0.869
## # i 9 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>, Efficiency <dbl>
```

```
library(dplyr)

# Extract the PLAYER-IDs of the top 10 most efficient players
top_10_players <- top_10_career_efficiency$`PLAYER-ID`

# Filter the original dataset to include only the top 10 players
top_10_player_stats <- PlayerDataS %>%
  filter(`PLAYER-ID` %in% top_10_players)

# Summarize the true stats for the top 10 players and ensure only one row per player
top_10_player_summary <- top_10_player_stats %>%
  group_by(`PLAYER-ID`) %>%
  summarize(
    Player = first(Player),
    Position = first(POSITION),
    across(c(positive_metrics, negative_metrics), mean, na.rm = TRUE),
    .groups = 'drop'
  )

# Calculate the aggregated team metrics by summing the metrics
team_summary <- top_10_player_summary %>%
  summarize(
    `PLAYER-ID` = NA,
    Player = "TEAMxMINUTE",
    Position = "TEAMxMINUTE",
    across(c(positive_metrics, negative_metrics), sum, na.rm = TRUE) # Use sum for total values
  )

# Create the TEAMxGAME row by multiplying the TEAM row values by 24
teamxgame_summary <- team_summary %>%
  mutate(
    Player = "TEAMxGAME",
    Position = "TEAMxGAME",
    across(c(positive_metrics, negative_metrics), ~ . * 24)
  )
```

```
# Combine the summaries
combined_summary <- bind_rows(top_10_player_summary, team_summary, teamxgame_summary)

# Print the combined summary
print(combined_summary)
```

```
## # A tibble: 12 x 13
##   'PLAYER-ID' Player      Position PointsPerMinute FieldGoalsPerMinute
##   <dbl> <chr>          <chr>          <dbl>          <dbl>
## 1 1630228 Jonathan Kuminga F              0.543          0.204
## 2 1630549 Day'Ron Sharpe C              0.462          0.192
## 3 1630552 Jalen Johnson   F              0.434          0.175
## 4 1630560 Cam Thomas     G              0.619          0.223
## 5 1630578 Alperen Sengun  C              0.534          0.206
## 6 1630702 Jaden Hardy      G              0.540          0.190
## 7 1631204 Marcus Sasser   G              0.448          0.167
## 8 1631218 Trayce Jackson-Davis F              0.487          0.213
## 9 1641709 Ausar Thompson   G              0.334          0.139
## 10 1641713 GG Jackson     F              0.547          0.188
## 11 NA TEAMxMINUTE    TEAMxMI~      4.95          1.90
## 12 NA TEAMxGAME      TEAMxGA~     119.          45.5
## # i 8 more variables: ThreePointersPerMinute <dbl>, AssistsPerMinute <dbl>,
## #   TotalReboundsPerMinute <dbl>, OffensiveReboundsPerMinute <dbl>,
## #   DefensiveReboundsPerMinute <dbl>, StealsPerMinute <dbl>,
## #   BlocksPerMinute <dbl>, FieldGoalsAttemptedPerMinute <dbl>
```