

Final Project

Ope Abegunde

2024-02-22

##Part 1 - Introduction #The topic of study is the characteristics of NBA players from 1947 to 2022. We are studying the effect of height on free throw percentage and also finding the changes over time of basketball players and their demographics. This includes a variety of variables including height, weight, BMI, free throw percentage, and etc. We can learn about the changes over time of NBA players who qualify and see whether certain characteristics affect player level and quality.

##Part 2 - Background #Our group used three datasets 1) contained the physical attributes of all of the people to ever play in the NBA dataset 2) and 3) had the game stats for all the active players in the league for the 22-23 and 23-24 season. The datasets provided our group with the needed information to observe the phenomenon where the tallest players are the worst at free throws compared to their shorter team members. We also observed how the average height, weight, and BMI of players have changed since the start of the league in 1948 to today. All of the datasets were Excel/CSV files. The unit of observation for the all_player_bios was the individual player's name. The stats data frame unit of observation was also the individual player in a given season, which was either 22-23 or 23-24. The most important variables for our project were the player's height, weight, BMI, position, and free throw percentage(FT%). These variables were the most useful for us to try and observe a correlation between physical attributes such as height and free throw accuracy. When reading the data visualizations it is important to remember that a player's position is usually related to the player's height. For example, the tallest and biggest players generally play center, and the smaller and faster players are more likely to play a guard position. It is also important to note that some data points for the free throws lie on the asymptote 1 or 0.0. This does mean that the player that the point represents has 100% success at the line or zero success. It more likely indicates that that player did not have the opportunity to play more than once or twice that season. Finally, for the age visualization, the graph appears counterintuitive in that as players get older they would get worse. This is a collider bias issue because only the players who remain high performers in their later years stay in the league. Thus the graph might be misleading because only the best players can play into their late thirties and early forties it doesn't show that the majority of players retire around 33 years old.

##Part 3 - Data Wrangling

#Load tidyverse, stringr, ggribges, and csv data files.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2     3.4.4      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
```

```
## -- 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 errors
```

```
library(stringr)
library(ggridges)

a_initial_Season_22_23 <- read_csv("NBA Stats 202223 All Stats NBA Player Props Tool.csv")
```

```
## Rows: 609 Columns: 29
## -- Column specification -----
## Delimiter: ","
## chr (3): NAME, TEAM, POS
## dbl (25): AGE, GP, MPG, USG%, TO%, FTA, FT%, 2PA, 2P%, 3PA, 3P%, eFG%, TS%, ...
## lgl (1): RANK
##
## 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.
```

```
a_initial_Season_23_24 <- read_csv("NBA Stats 202324 All Stats NBA Player Props Tool.csv")
```

```
## Rows: 598 Columns: 29
## -- Column specification -----
## Delimiter: ","
## chr (3): NAME, TEAM, POS
## dbl (25): AGE, GP, MPG, USG%, TO%, FTA, FT%, 2PA, 2P%, 3PA, 3P%, eFG%, TS%, ...
## lgl (1): RANK
##
## 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.
```

```
a_initial_playerset <- read_csv("common_player_info.csv")
```

```
## Rows: 4171 Columns: 33
## -- Column specification -----
## Delimiter: ","
## chr (26): first_name, last_name, display_first_last, display_last_comma_fir...
## dbl (6): person_id, weight, season_exp, team_id, from_year, to_year
## dtm (1): birthdate
##
## 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.
```

```
all_player_bios <- a_initial_playerset %>%
```

```
#Clean data by removing players who have never been drafted or have no body measurement data.
```

```
  filter(draft_year != "Undrafted") %>%
```

```
  filter_at(vars(height,weight), all_vars(!is.na(.))) %>%
```

```
#Select variables for first name, last name, body measurements, season experience, position, year of dr
```

```
  subset(select= c(first_name,
                    last_name,
                    weight,
                    height,
                    season_exp,
```

```

        position,
        draft_year,
        to_year)) %>%

#Concatenate first and last name of players. Create a new variable, "NAME", and alphabetically order pl
  unite("NAME",
        c(first_name,last_name),
        sep=" ",
        remove=TRUE) %>%
  arrange(NAME) %>%

#Convert "height" from "feet, inches" into strictly inches.
#Separate non-numeric variable, "height", into two numeric variables, "Feet" and "Inches".
  separate(height, into = c("Feet", "Inches"),
            sep = "-") %>%
  mutate(Feet = as.numeric(Feet)) %>%
  mutate(Inches = as.numeric(Inches)) %>%

#Recalculate "height" in strictly inches and remove the "Feet" and "Inches" variables.
  mutate(height = 12 * Feet + Inches) %>%
  subset(select = -c(Feet,Inches)) %>%
  relocate(height, .after = NAME) %>%

#Create and calculate BMI
  mutate(BMI =
        round((weight*0.453592)/(height*0.0254)^2, 2)) %>%
  relocate(BMI, .after = NAME) %>%

#Edit "position" for later joining and simplify "position" to strictly Forward, Center, or Guard.
  mutate(position = gsub("Forward","F",position)) %>%
  mutate(position = gsub("Center","C",position)) %>%
  mutate(position = gsub("Guard","G",position)) %>%
  mutate(position =
        case_when(
          position == "F-C" | position == "F-G" | position == "F" ~ "F",
          position == "G-C" | position == "G-F" | position == "G" ~ "G",
          position == "C-F" | position == "C-G" | position == "C" ~ "C")) %>%

#Change draft_year and to_year into a numeric for later analysis.
  mutate(draft_year = as.numeric(draft_year)) %>%
  mutate(to_year = as.numeric(to_year)) %>%

#Remove punctuation that could hinder later joining.
  mutate(NAME = gsub("\\\\.", "",NAME)) %>%
  mutate(NAME = gsub("\\\\'", "",NAME))

#Create a new dataset that reports general info of players in the 2022-2023 and 2023-2024 season.
player_bios_22_24 <- all_player_bios %>%
  filter(to_year >= 2022)

#-----
print(all_player_bios)

```

```
## # A tibble: 3,037 x 8
##   NAME      BMI height weight season_exp position draft_year to_year
##   <chr>      <dbl> <dbl> <dbl>      <dbl> <chr>      <dbl> <dbl>
## 1 AC Green    24.1    81   225        16 F        1985    2000
## 2 AJ Bramlett 23.7    82   227         1 C        1999    1999
## 3 AJ English   21.9    75   175         2 G        1990    1991
## 4 AJ Griffin   25.4    78   220         1 F        2022    2023
## 5 AJ Price     25.0    74   195         6 G        2009    2014
## 6 Aaron Brooks 21.8    72   161        11 G        2007    2017
## 7 Aaron Gordon 25.8    80   235         9 F        2014    2023
## 8 Aaron Gray   26.9    84   270         7 C        2007    2013
## 9 Aaron Holiday 25.1    72   185         5 G        2018    2023
## 10 Aaron McKie 24.8    77   209        14 G        1994    2006
## # i 3,027 more rows
```

```
print(player_bios_22_24)
```

```
## # A tibble: 356 x 8
##   NAME      BMI height weight season_exp position draft_year to_year
##   <chr>      <dbl> <dbl> <dbl>      <dbl> <chr>      <dbl> <dbl>
## 1 AJ Griffin   25.4    78   220         1 F        2022    2023
## 2 Aaron Gordon 25.8    80   235         9 F        2014    2023
## 3 Aaron Holiday 25.1    72   185         5 G        2018    2023
## 4 Aaron Nesmith 25.5    77   215         3 G        2020    2023
## 5 Aaron Wiggins 22.5    77   190         2 G        2021    2023
## 6 Admiral Schofield 28.6    77   241         3 F        2019    2023
## 7 Al Horford    25.7    81   240        16 C        2007    2023
## 8 Alec Burks    24.7    78   214        12 G        2011    2023
## 9 Aleksej Pokusevski 20.9    84   210         3 F        2020    2023
## 10 Alex Len      23.8    86   250        10 C        2013    2023
## # i 346 more rows
```

```
#Create "SEASON" that reports the season that the data originates from.
a_initial_Season_22_23 <- a_initial_Season_22_23 %>%
  mutate(SEASON = "2022-23") %>%
  relocate(SEASON)

a_initial_Season_23_24 <- a_initial_Season_23_24 %>%
  mutate(SEASON = "2023-24") %>%
  relocate(SEASON)

#Join 2022-2023 and 2023-2024 season stats. Remove unnecessary variable, "RANK".
player_gamestats_22_24 <- full_join(a_initial_Season_22_23, a_initial_Season_23_24) %>%
  arrange(NAME) %>%

#Rename certain stats with "%" for later analysis
  rename(FTp = "FT%") %>%
  rename(P3p = "3P%") %>%
  subset(select = c(SEASON, NAME, TEAM, POS, AGE, FTp, P3p)) %>%

#Remove punctuation that could hinder later joining.
  mutate(NAME = gsub("\\.", "", NAME)) %>%
  mutate(NAME = gsub("\\'", "", NAME))
```

```
## Joining with 'by = join_by(SEASON, RANK, NAME, TEAM, POS, AGE, GP, MPG, 'USG%',
## 'TO%', FTA, 'FT%', '2PA', '2P%', '3PA', '3P%', 'eFG%', 'TS%', PPG, RPG, APG,
## SPG, BPG, TPG, 'P+R', 'P+A', 'P+R+A', VI, ORtg, DRTg)'
```

```
#-----
print(player_gamestats_22_24)
```

```
## # A tibble: 1,207 x 7
##   SEASON NAME      TEAM POS    AGE  FTp  P3p
##   <chr>  <chr>    <chr> <chr> <dbl> <dbl> <dbl>
## 1 2022-23 AJ Green  Mil  G    23.5 1    0.419
## 2 2022-23 AJ Lawson Dal  G    22.7 0.25 0.4
## 3 2022-23 AJ Lawson Min  G    22.7 0    0
## 4 2023-24 AJ Lawson Dal  G    23.6 0.632 0.325
## 5 2023-24 AJ Green  Mil  G    24.4 1    0.415
## 6 2022-23 AJ Griffin Atl  F    19.6 0.894 0.39
## 7 2023-24 AJ Griffin Atl  F    20.5 1    0.273
## 8 2022-23 Aaron Gordon Den  F    27.6 0.609 0.347
## 9 2023-24 Aaron Gordon Den  F    28.4 0.641 0.315
## 10 2022-23 Aaron Holiday Atl  G    26.5 0.844 0.409
## # i 1,197 more rows
```

```
#Join data from 2022-2024 season stats and player general info.
```

```
player_info_22_24 <- player_gamestats_22_24 %>%
  left_join(player_bios_22_24, by=c("NAME", "POS"="position")) %>%
```

```
#Remove players with no height or weight data (approx. 300 players).
```

```
filter_at(vars(height,weight), all_vars(!is.na(.)))
```

```
#Group players by position and draft year. Create variables for the mean height, weight, and BMI of players.
```

```
mean_all_player_bios <- all_player_bios %>%
```

```
group_by(position, draft_year) %>%
  summarize(mean_height = mean(height),
            mean_weight = mean(weight),
            mean_BMI = mean(BMI))
```

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

```
#-----
print(player_info_22_24)
```

```
## # A tibble: 556 x 13
##   SEASON NAME      TEAM POS    AGE  FTp  P3p  BMI height weight season_exp
##   <chr>  <chr>    <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2022-23 AJ Grif~ Atl  F    19.6 0.894 0.39 25.4 78 220 1
## 2 2023-24 AJ Grif~ Atl  F    20.5 1    0.273 25.4 78 220 1
## 3 2022-23 Aaron G~ Den  F    27.6 0.609 0.347 25.8 80 235 9
## 4 2023-24 Aaron G~ Den  F    28.4 0.641 0.315 25.8 80 235 9
## 5 2022-23 Aaron H~ Atl  G    26.5 0.844 0.409 25.1 72 185 5
## 6 2023-24 Aaron H~ Hou  G    27.4 0.902 0.409 25.1 72 185 5
## 7 2023-24 Aaron W~ Okc  G    25.1 0.75 0.519 22.5 77 190 2
```

```
## 8 2022-23 Admiral~ Orl F 26 0.913 0.324 28.6 77 241 3
## 9 2023-24 Admiral~ Orl F 26.9 0 0.3 28.6 77 241 3
## 10 2022-23 Alec Bu~ Det G 31.7 0.814 0.414 24.7 78 214 12
## # i 546 more rows
## # i 2 more variables: draft_year <dbl>, to_year <dbl>
```

```
print(mean_all_player_bios)
```

```
## # A tibble: 227 x 5
## # Groups:   position [3]
##   position draft_year mean_height mean_weight mean_BMI
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 C        1947        79.3        205        23.0
## 2 C        1948        80         224        24.6
## 3 C        1949        80         207        22.7
## 4 C        1950        80.2        219.        23.9
## 5 C        1951        81         210        22.5
## 6 C        1952        80.5        220.        23.8
## 7 C        1953        81.6        220        23.3
## 8 C        1954        78.8        221.        24.9
## 9 C        1956        81.6        219        23.2
## 10 C       1957        80         230        25.3
## # i 217 more rows
```

##Part 4 - Exploratory Analysis

#All of these histograms were used to show the viability of our data and accuracy of each variable.

#Histogram for Height vs. Draft Year

```
histogram_height_year <- mean_all_player_bios
```

#Mutating histogram to create groups by 10 year intervals

```
histogram_height_year <- mutate (histogram_height_year,
  year_group = cut(draft_year,
    breaks = seq(1940, 2022, by = 10)))
```

#Creating Average Heights for each 10 year interval

```
histogram_height <- histogram_height_year %>%
  group_by (year_group) %>%
  summarise (avg_height = mean(mean_height))
```

#

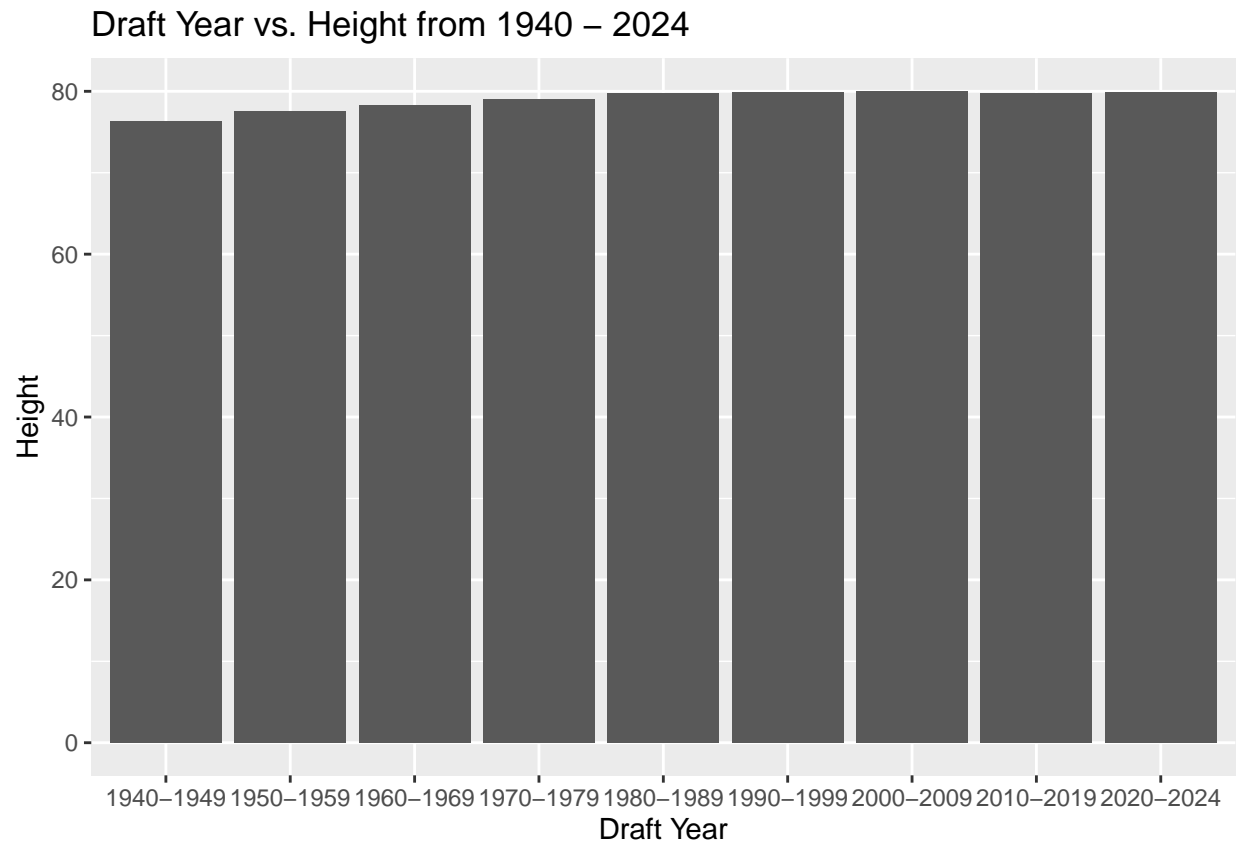
#Creating Histogram

```
ggplot(histogram_height,
  aes(y = avg_height,
    x = year_group)) +

  geom_col() +

  labs (x = "Draft Year",
    y = "Height",
    title = "Draft Year vs. Height from 1940 - 2024") +
```

```
scale_x_discrete (labels =
  c("1940-1949", "1950-1959", "1960-1969", "1970-1979", "1980-1989", "1990-1999", "2000-2009", "2010-2019", "2020-2024")
```



```
#-----
#Histogram for Weight vs. Position

#Creating Average Weights for each position
histogram_weight <- mean_all_player_bios %>%
  group_by (position) %>%
  summarise (avg_weight = mean(mean_weight))

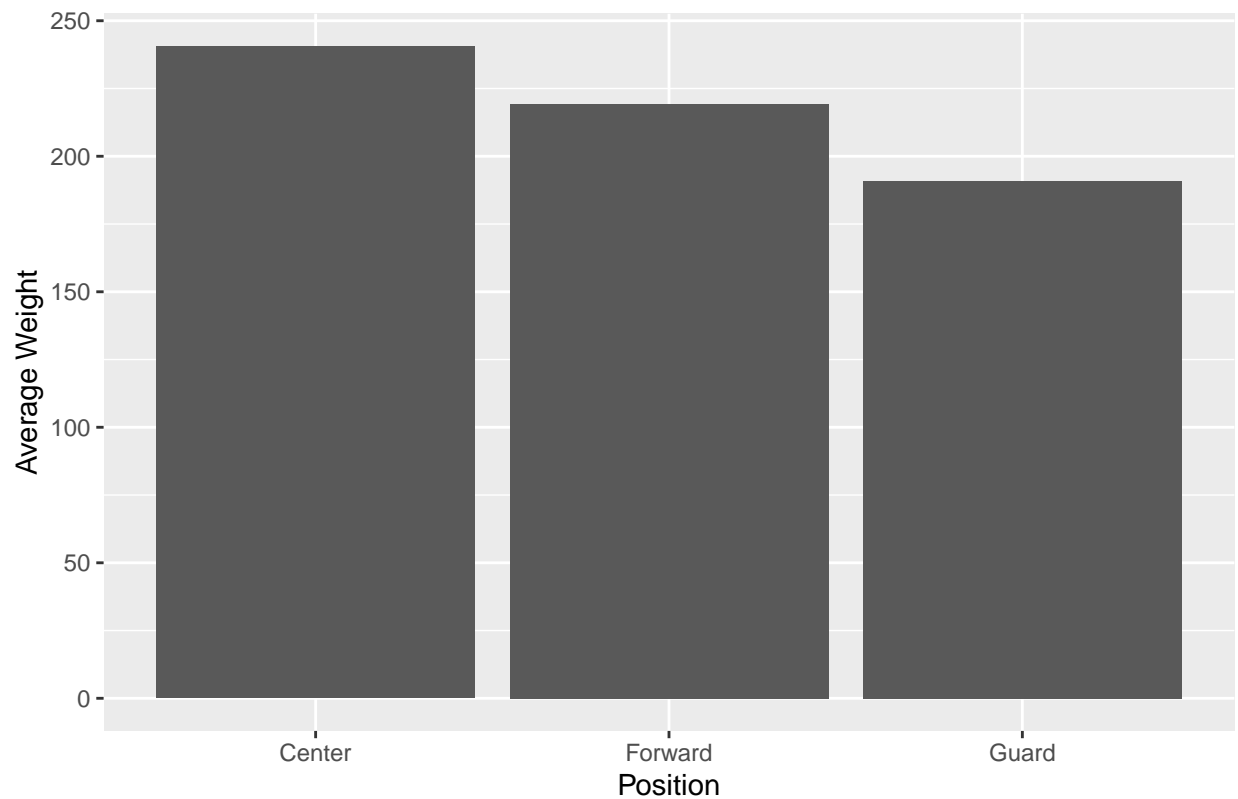
#Creating Histogram
ggplot(histogram_weight,
  aes(x = position,
    y = avg_weight)) +

  geom_col() +

  labs (x = "Position",
    y = "Average Weight",
    title = "Position vs. Weight for Players from 1940 - 2024") +

  scale_x_discrete (labels =
    c ("Center", "Forward", "Guard"))
```

Position vs. Weight for Players from 1940 – 2024



```
#-----
#Histogram for BMI vs. Draft Year

#Creating Average Weights for each 10 year interval
histogram_bmi <- histogram_height_year %>%
group_by (year_group) %>%
  summarise (avg_bmi = mean(mean_BMI))

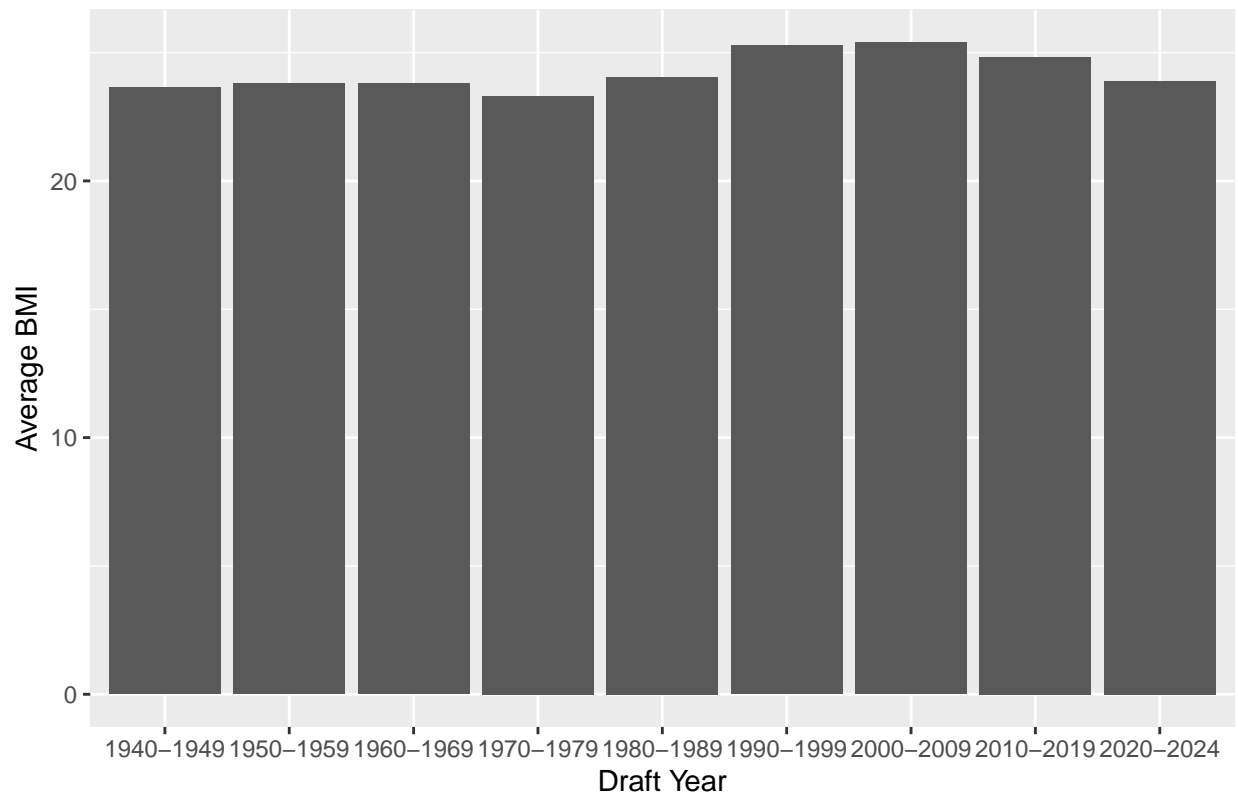
#Creating graph
ggplot (histogram_bmi,
        aes(x = year_group, y = avg_bmi)) +

  geom_col () +

  labs (x = "Draft Year",
        y = "Average BMI",
        title = "Draft Year vs. Average BMI for Players from 1940 - 2024") +

  scale_x_discrete (labels = c("1940-1949", "1950-1959", "1960-1969", "1970-1979", "1980-1989", "1990-1999"))
```


Draft Year vs. Average BMI for Players from 1940 – 2024



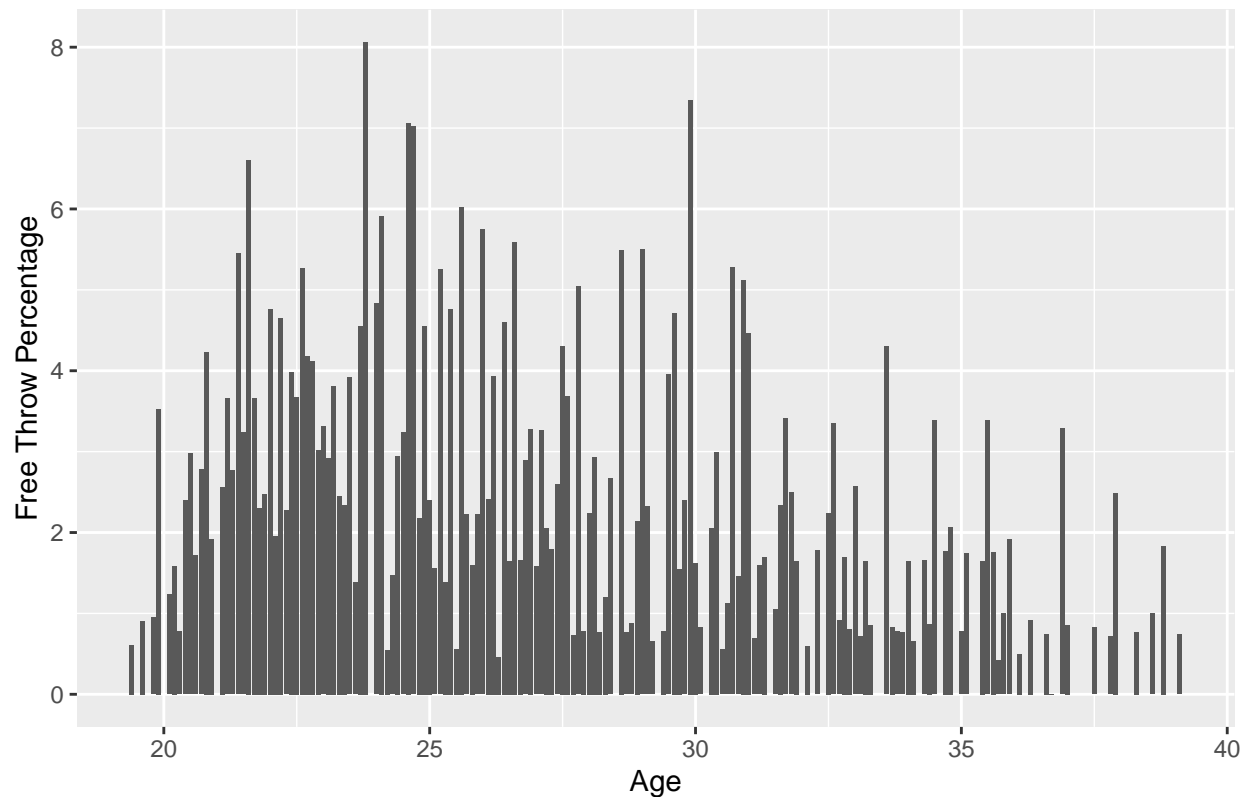
```
#-----
#Histogram for Age vs. Free Throw Percentage

#Creating histogram
ggplot(player_info_22_24,
       aes(x = AGE, y = FTp)) +

  geom_col() +

  labs(x = "Age",
       y = "Free Throw Percentage",
       title = "The Age of Players vs. Free Throw Percentage for 2022-2023",
       color = "Position")
```

The Age of Players vs. Free Throw Percentage for 2022–2023



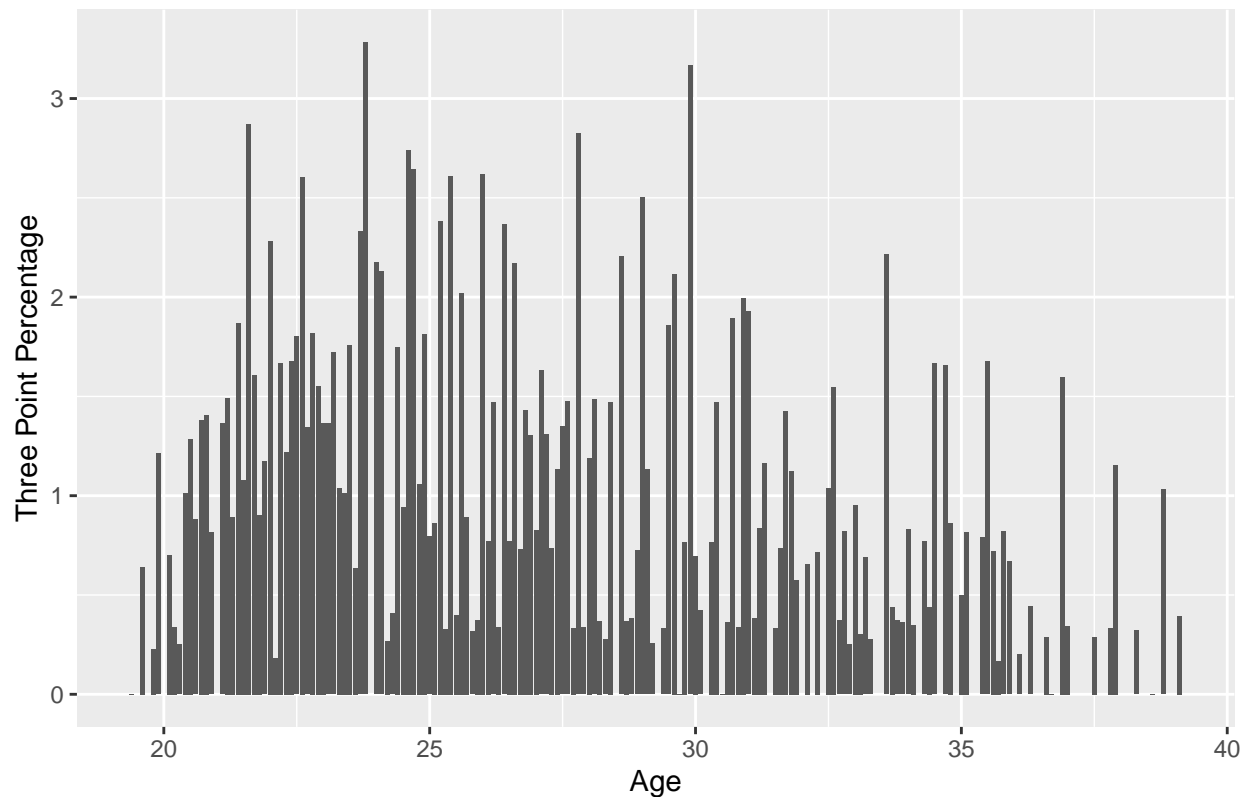
```
#-----
#Histogram for Age vs. Three Point Percentage

#Creating histogram
ggplot(player_info_22_24,
       aes(x = AGE, y = P3p)) +

  geom_col() +

  labs(x = "Age",
       y = "Three Point Percentage",
       title = "The Age of Players vs. Three Point Percentage for 2022-2023",
       color = "Position")
```

The Age of Players vs. Three Point Percentage for 2022–2023



#How has height and weight changed throughout the year - “common_player_info”?

```
ggplot(mean_all_player_bios,
  aes(y=mean_height,
    x=draft_year,
    color = position,
    group = position)) +

  geom_line() +

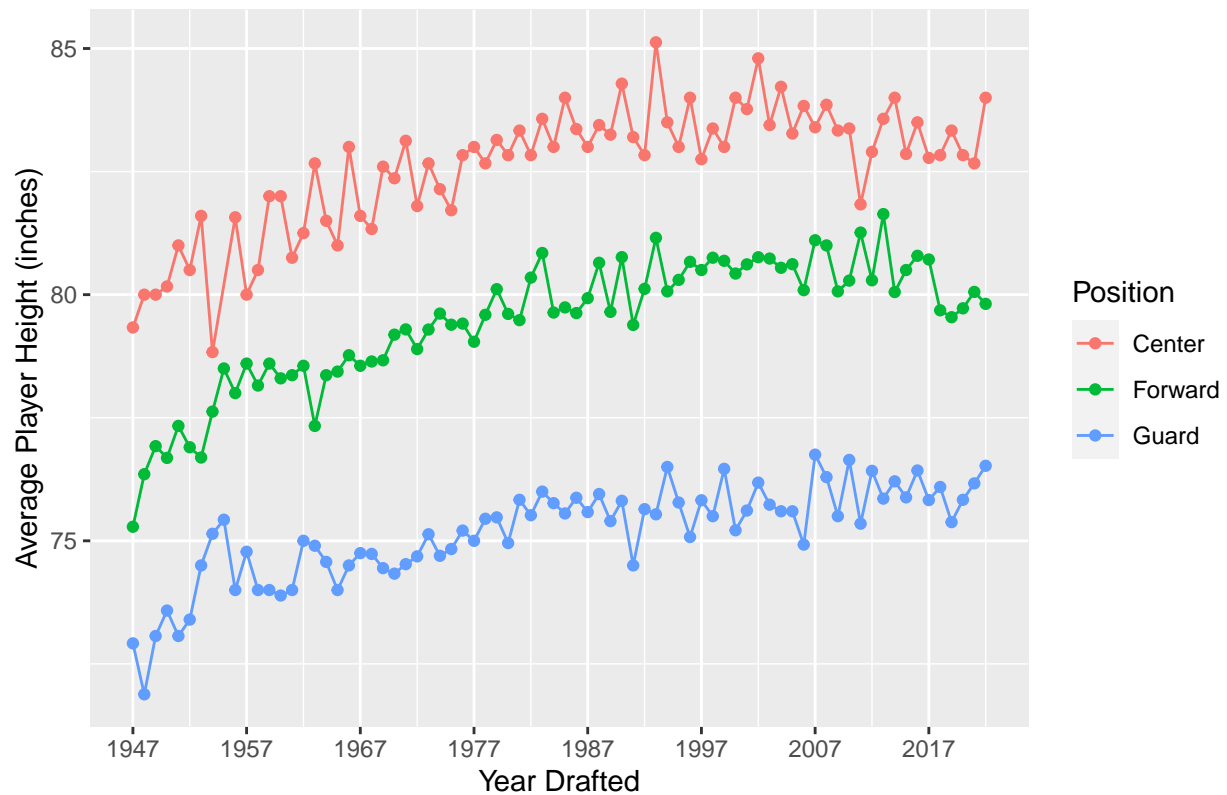
  geom_point() +

  scale_x_continuous(breaks =
    seq(1947, 2022,
      by = 10)) +

  scale_color_discrete(labels =
    c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Year Drafted",
    y = "Average Player Height (inches)",
    title = "Average Player Height vs. Year Drafted between 1947 to 2022",
    color = "Position")
```

Average Player Height vs. Year Drafted between 1947 to 2022



```
# -----
ggplot(mean_all_player_bios,
  aes(y=mean_weight,
    x=draft_year,
    color = position,
    group = position)) +

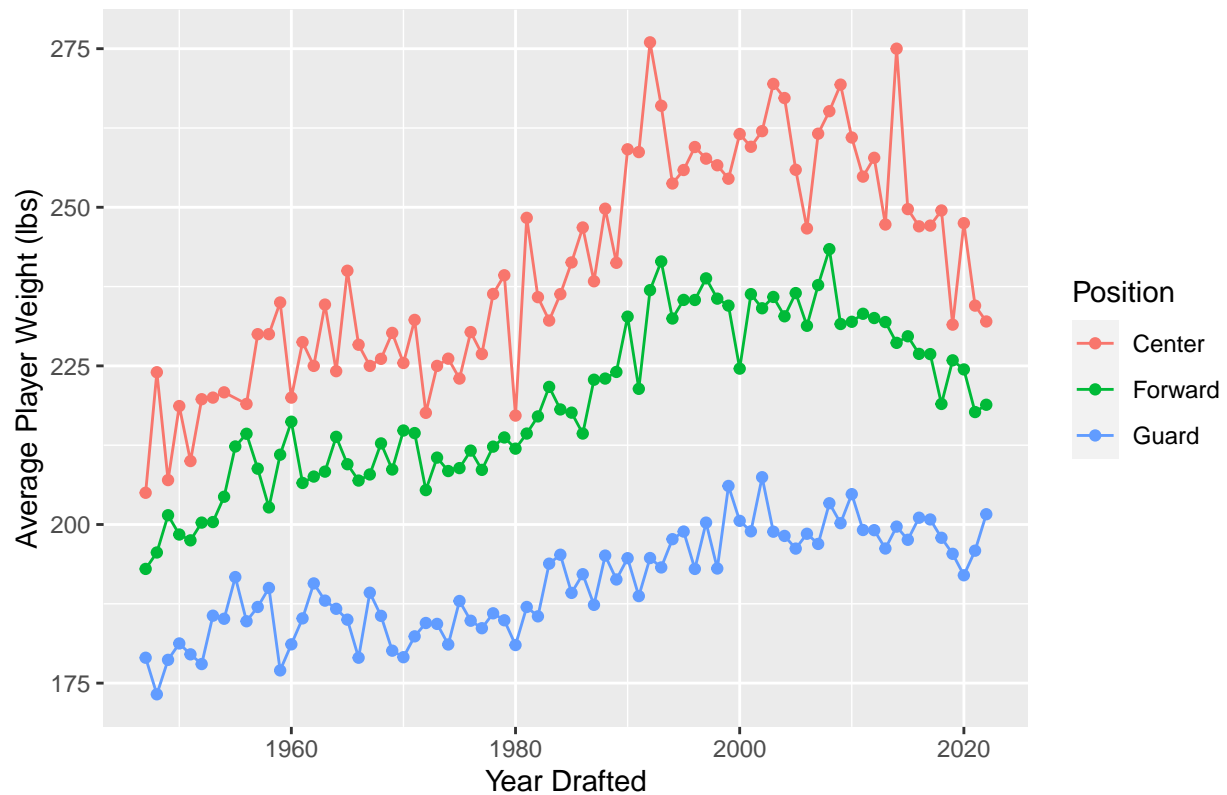
  geom_line() +

  geom_point() +

  scale_color_discrete(labels =
    c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Year Drafted",
    y = "Average Player Weight (lbs)",
    title = "Average Player Weight vs. Year Drafted between 1947 to 2022",
    color = "Position")
```

Average Player Weight vs. Year Drafted between 1947 to 2022



```
# -----
ggplot(mean_all_player_bios,
  aes(y=mean_BMI,
    x=draft_year,
    color = position,
    group = position)) +

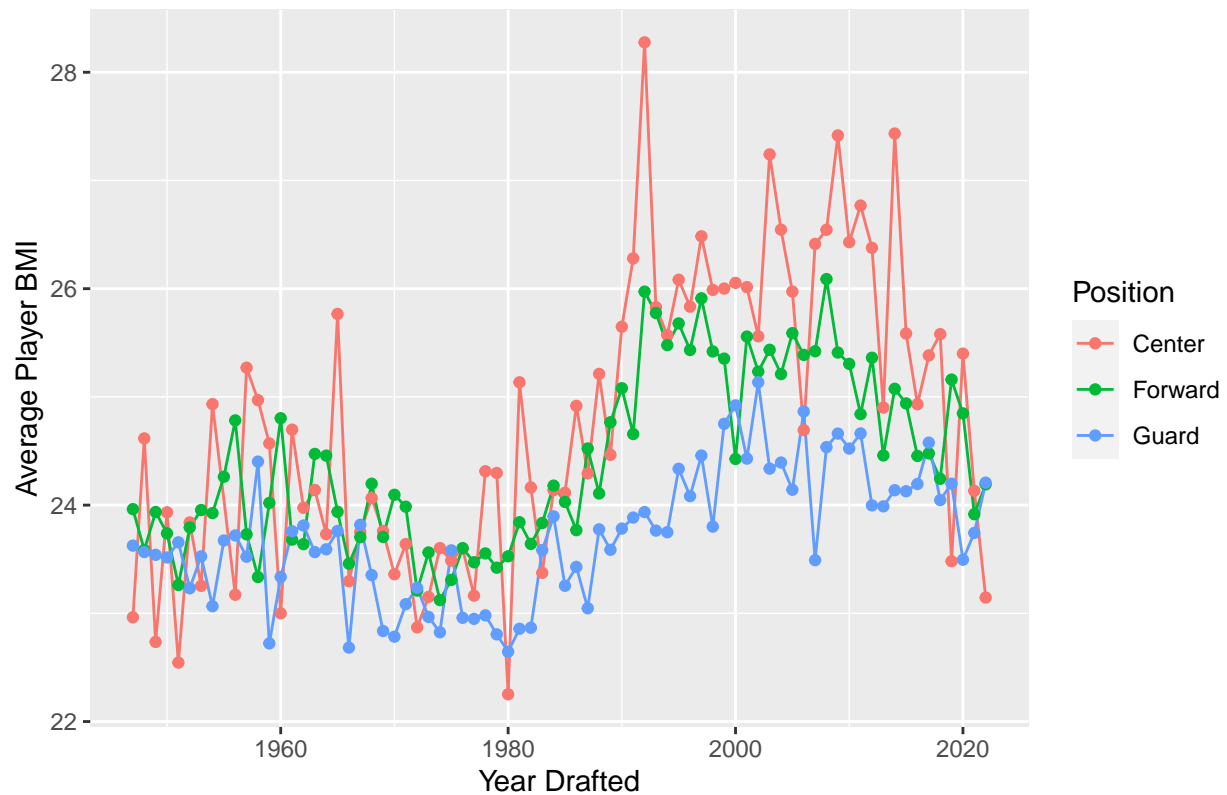
  geom_line() +

  geom_point() +

  scale_color_discrete(labels =
    c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Year Drafted",
    y = "Average Player BMI",
    title = "Average Player BMI vs. Year Drafted between 1947 to 2022",
    color = "Position")
```

Average Player BMI vs. Year Drafted between 1947 to 2022



#How does height and weight differ btwn teams?

```
#-----
#Age vs Free Throw Percentage

#Filtering to include only players who played in 2023.
df3 <- all_player_bios %>%
  filter(to_year >= 2023)
#join_by(all_player_bios, a_initial_Season_23_24, "NAME")
#merge(all_player_bios, a_initial_Season_22_23[, c("NAME", "FT%")], by = "NAME", all.x = TRUE)

#Merges bio info with the statistics to create one data frame for visual.
info_22_23 <- merge(df3,
  a_initial_Season_22_23[,c("NAME", "FT%", "3P%", "AGE")],
  by = "NAME",
  all.x = TRUE) %>%

  rename(FTP = "FT%") %>%
  rename(tPP = "3P%")

#-----
#Height vs. Free Throw Percentage

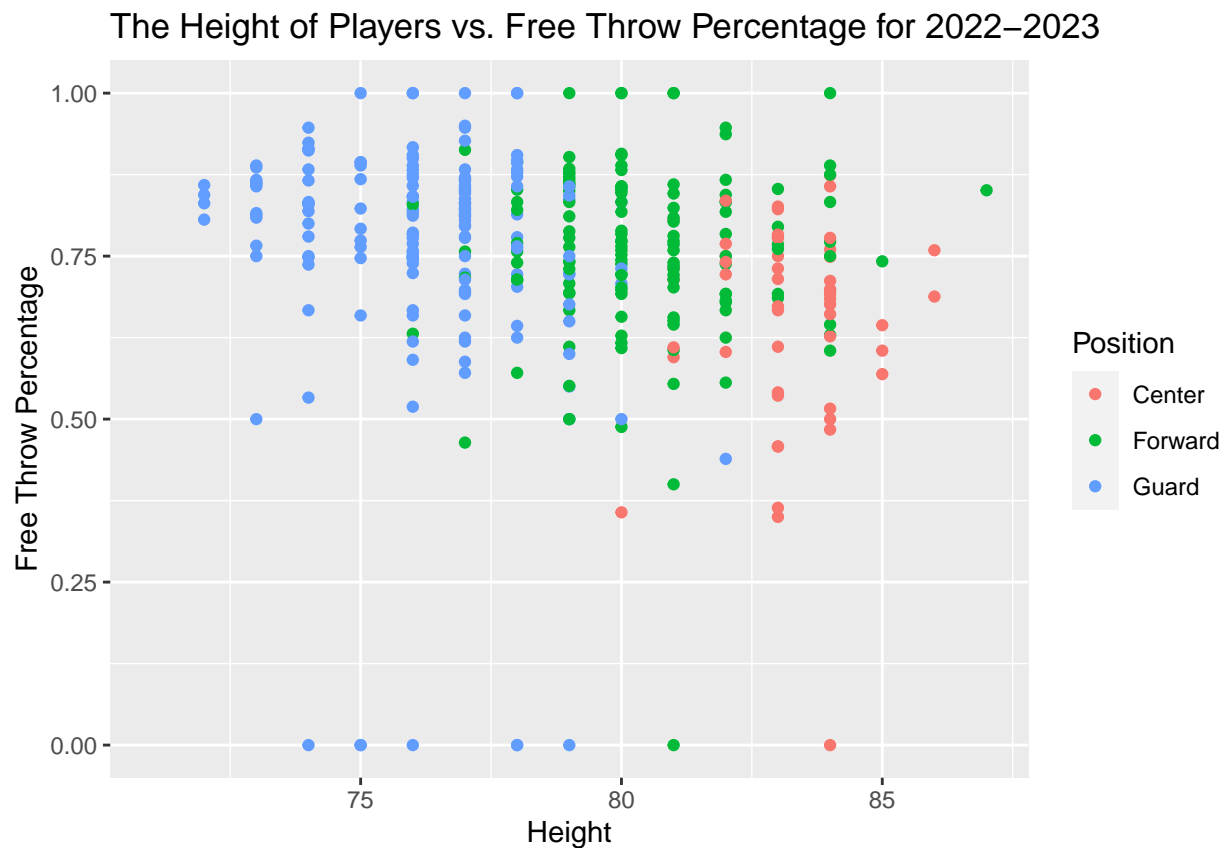
#Creating point graph only allowing values between 0 and 1 for percentages
ggplot(info_22_23, aes(x = height,
  y = FTP,
```

```

        color = position)) +
geom_point() +
scale_y_continuous(limits = c(0, 1)) +
scale_color_discrete (labels =
                      c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
labs(x = "Height",
     y = "Free Throw Percentage",
     title = "The Height of Players vs. Free Throw Percentage for 2022-2023",
     color = "Position")

```

Warning: Removed 33 rows containing missing values (‘geom_point()’).



```

#-----
#Faceted Height vs. Free Throw Percentage

#Creating graph with a facet for each position
ggplot(info_22_23, aes(x = height,
                       y = FTP,
                       color = position)) +

geom_point() +
scale_y_continuous(limits = c(0, 1)) +
facet_wrap(~ position) +
geom_smooth() +
scale_color_discrete (labels = c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
labs(x = "Height",

```

```
y = "Free Throw Percentage",
title = "The Height of Players vs. Free Throw Percentage for 2022-2023",
color = "Position")
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 33 rows containing non-finite values ('stat_smooth()').

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at 83

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 0

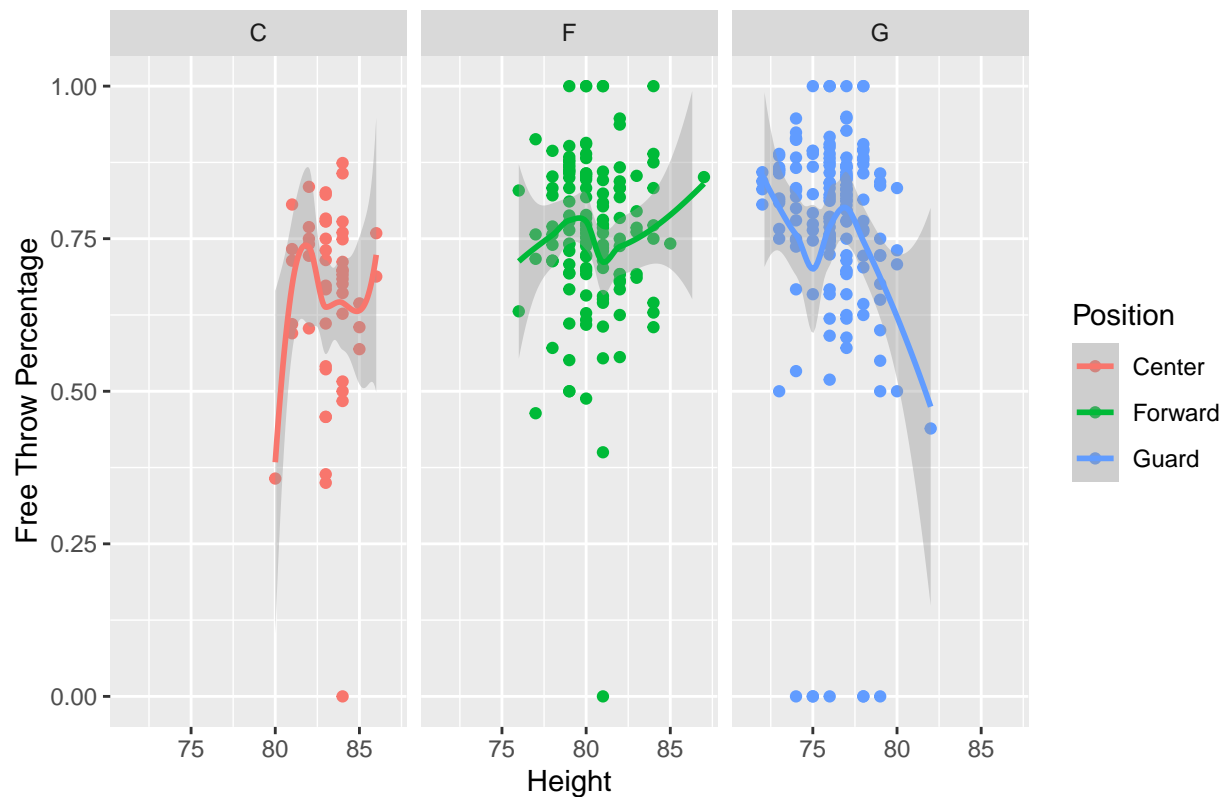
## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## 83

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius 1

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x
## else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 0

## Warning: Removed 33 rows containing missing values ('geom_point()').
```


The Height of Players vs. Free Throw Percentage for 2022–2023

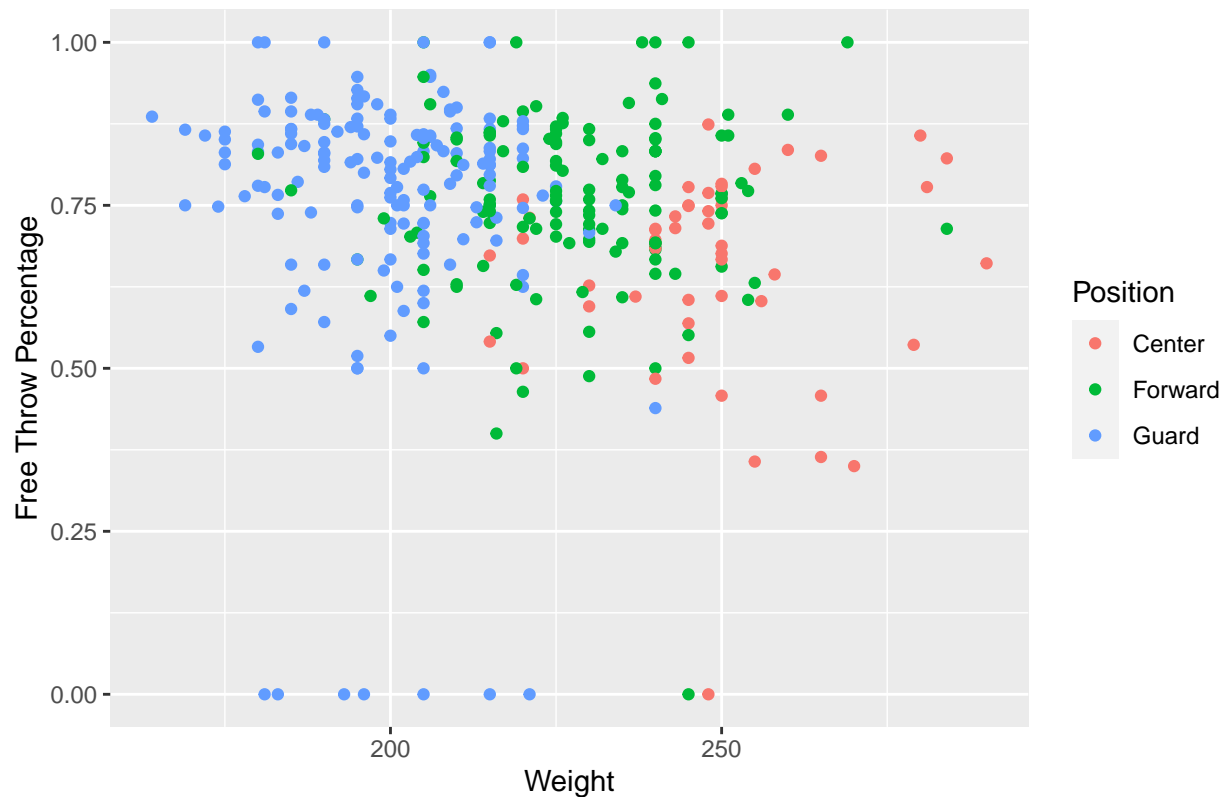


```
#-----
#Weight vs. Free Throw Percentage

#Creating graph showing weight and free throw percentage
ggplot(info_22_23, aes(x = weight,
                       y = FTP,
                       color = position)) +
  geom_point() +
  scale_color_discrete (labels =
                        c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
  labs(x = "Weight",
       y = "Free Throw Percentage",
       title = "The Weight of Players vs. Free Throw Percentage for 2022-2023",
       color = "Position")
```

```
## Warning: Removed 33 rows containing missing values ('geom_point()').
```

The Weight of Players vs. Free Throw Percentage for 2022–2023



```
#-----
#Weight vs. Free Throw Percentage Faceted

#Creating graph with a facet for each percentage and smooth line through points to show trend
ggplot(info_22_23, aes(x = weight,
                        y = FTP,
                        color = position)) +

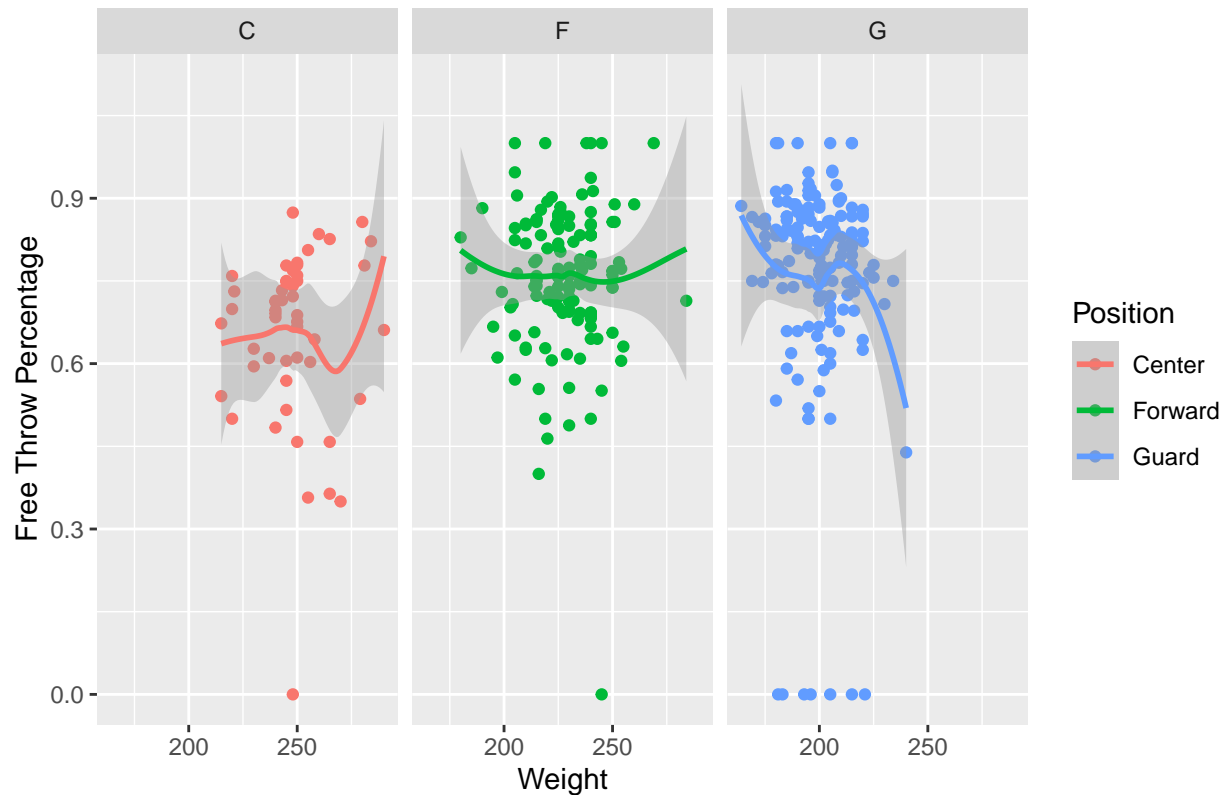
  geom_point() +
  facet_wrap(~ position) +
  geom_smooth() +
  scale_color_discrete (labels =
                        c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
  labs(x = "Weight",
       y = "Free Throw Percentage",
       title = "The Weight of Players vs. Free Throw Percentage for 2022-2023",
       color = "Position")
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 33 rows containing non-finite values ('stat_smooth()').
```

```
## Removed 33 rows containing missing values ('geom_point()').
```

The Weight of Players vs. Free Throw Percentage for 2022–2023



```
#-----
#Age vs. Free Throw Percentage

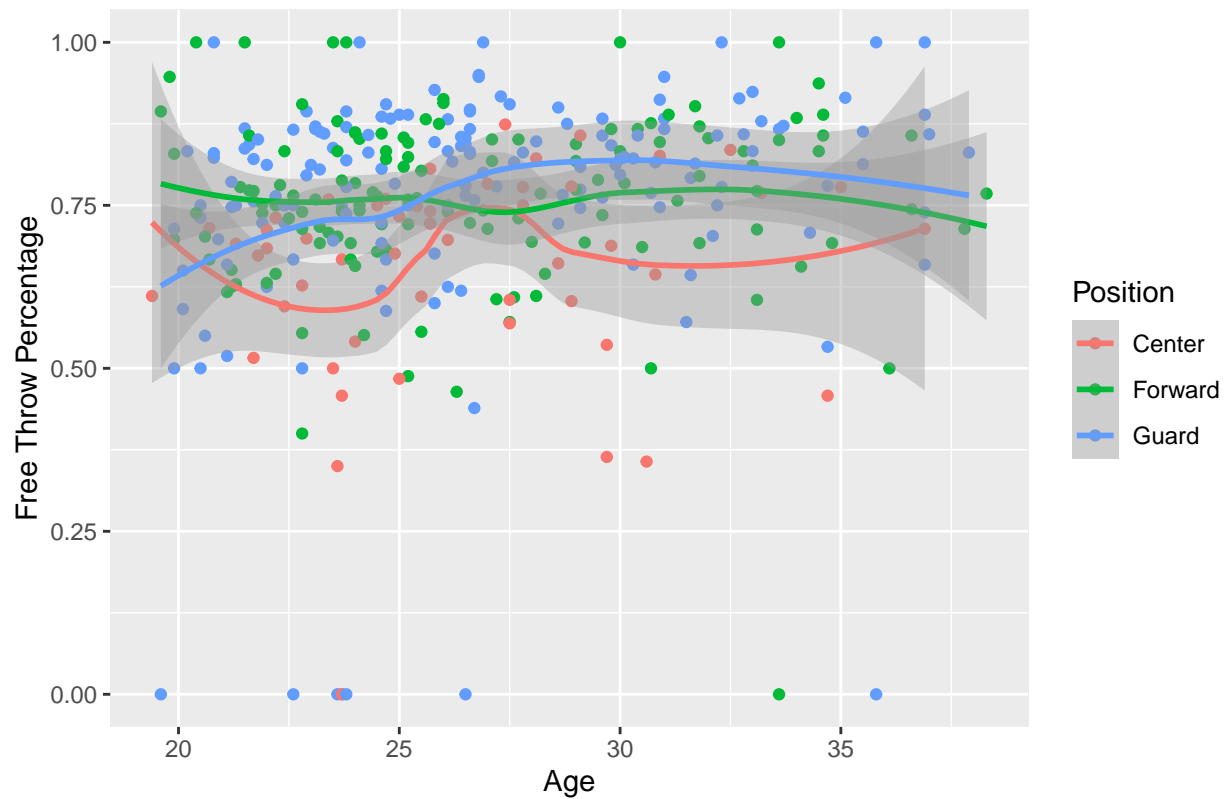
#Creating graph with smooth line showing trend for each position
ggplot(info_22_23, aes(x = AGE,
                        y = FTP,
                        color = position)) +
  geom_point() +
  geom_smooth() +
  scale_color_discrete (labels =
                        c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
  labs(x = "Age",
       y = "Free Throw Percentage",
       title = "The Age of Players vs. Free Throw Percentage for 2022-2023",
       color = "Position")
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 33 rows containing non-finite values ('stat_smooth()').
```

```
## Removed 33 rows containing missing values ('geom_point()').
```

The Age of Players vs. Free Throw Percentage for 2022–2023

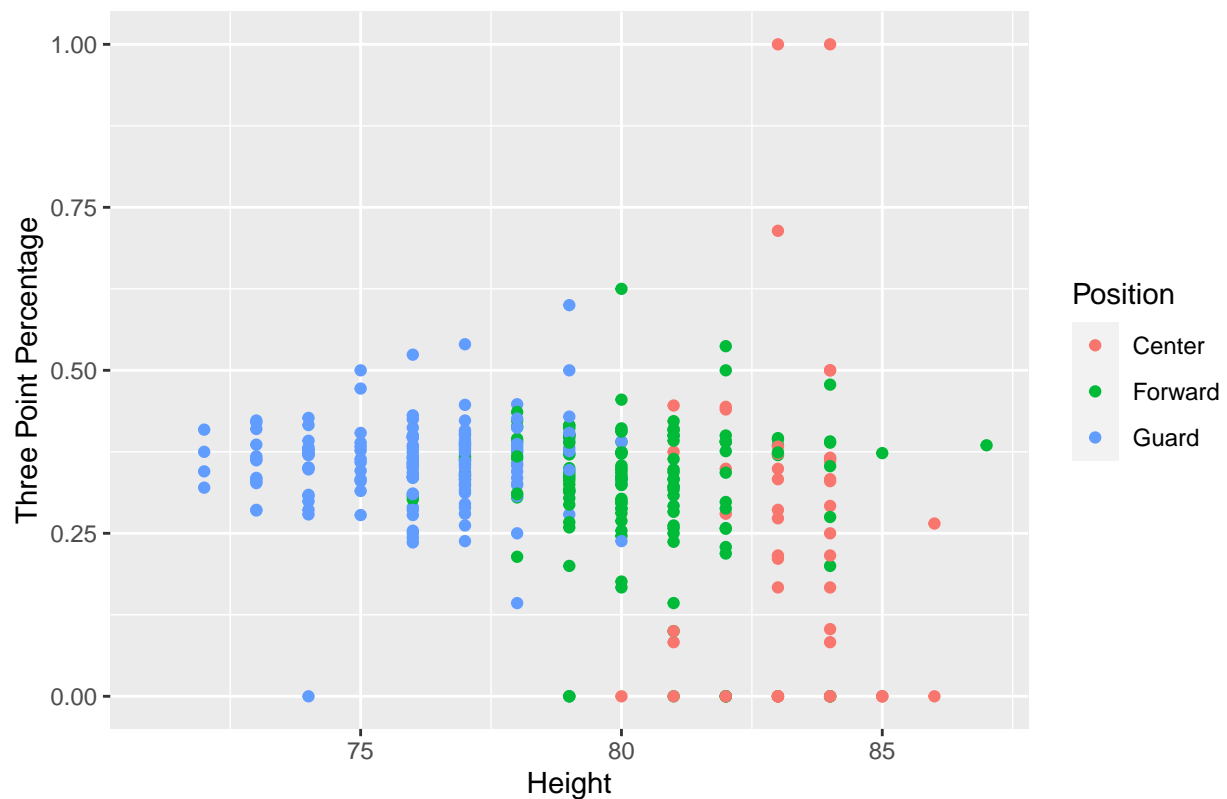


```
#-----
#Height vs. Three Point Percentage

#Creating graph showing height vs. three point percentage
ggplot(info_22_23, aes(x = height, y = tPP, color = position)) +
  geom_point() +
  scale_color_discrete (labels = c("C" = "Center", "F" = "Forward", "G" = "Guard")) +
  labs(x = "Height",
       y = "Three Point Percentage",
       title = "The Height of Players vs. Three Point Percentage for 2022–2023",
       color = "Position")
```

```
## Warning: Removed 33 rows containing missing values ('geom_point()').
```

The Height of Players vs. Three Point Percentage for 2022–2023



What are the variations between different body measurements and positions?

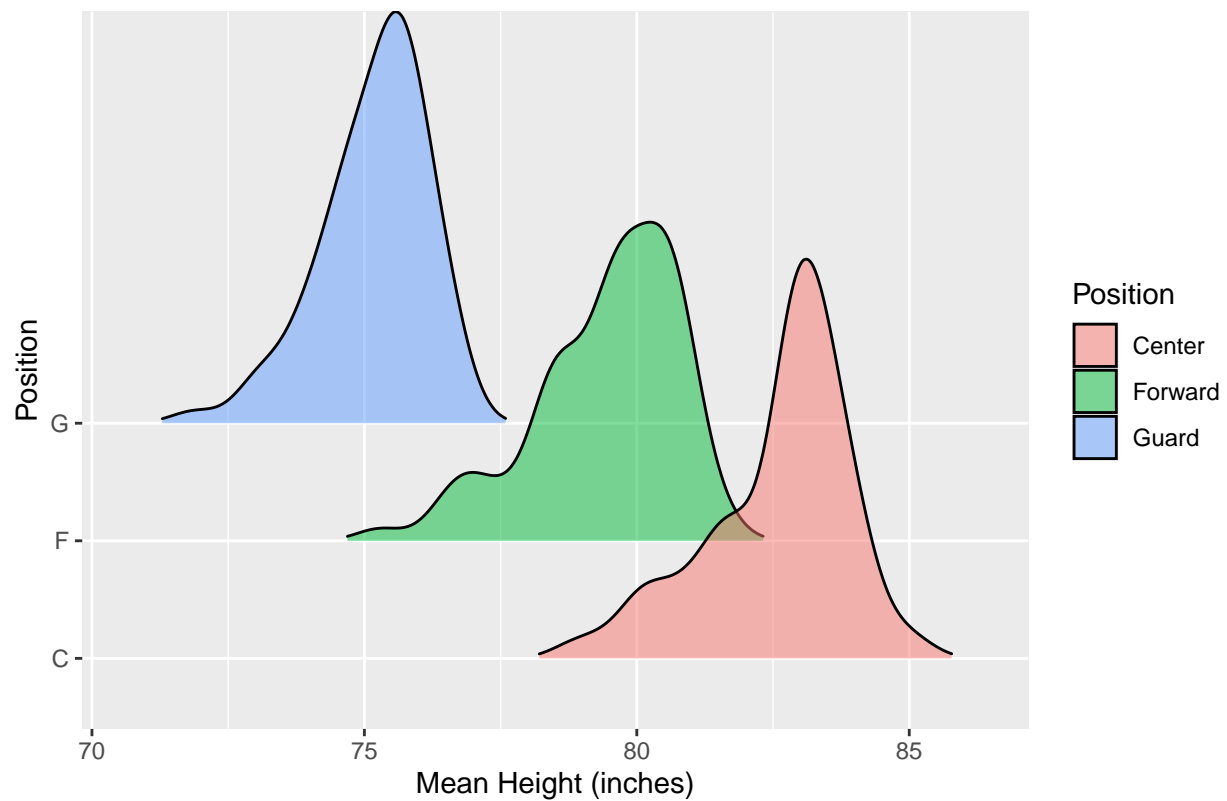
```
#
##HEIGHT
ggplot(mean_all_player_bios,
  aes(x=mean_height,
    y=position,
    fill = position)) +

  geom_density_ridges(scale = 3.5, alpha = 0.5,
    rel_min_height = 0.01) +
  scale_fill_discrete(labels =
    c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Mean Height (inches)",
    y = "Position",
    title = "Average Player Height vs. Positions",
    color = "Position",
    fill = "Position")
```

```
## Picking joint bandwidth of 0.424
```

Average Player Height vs. Positions



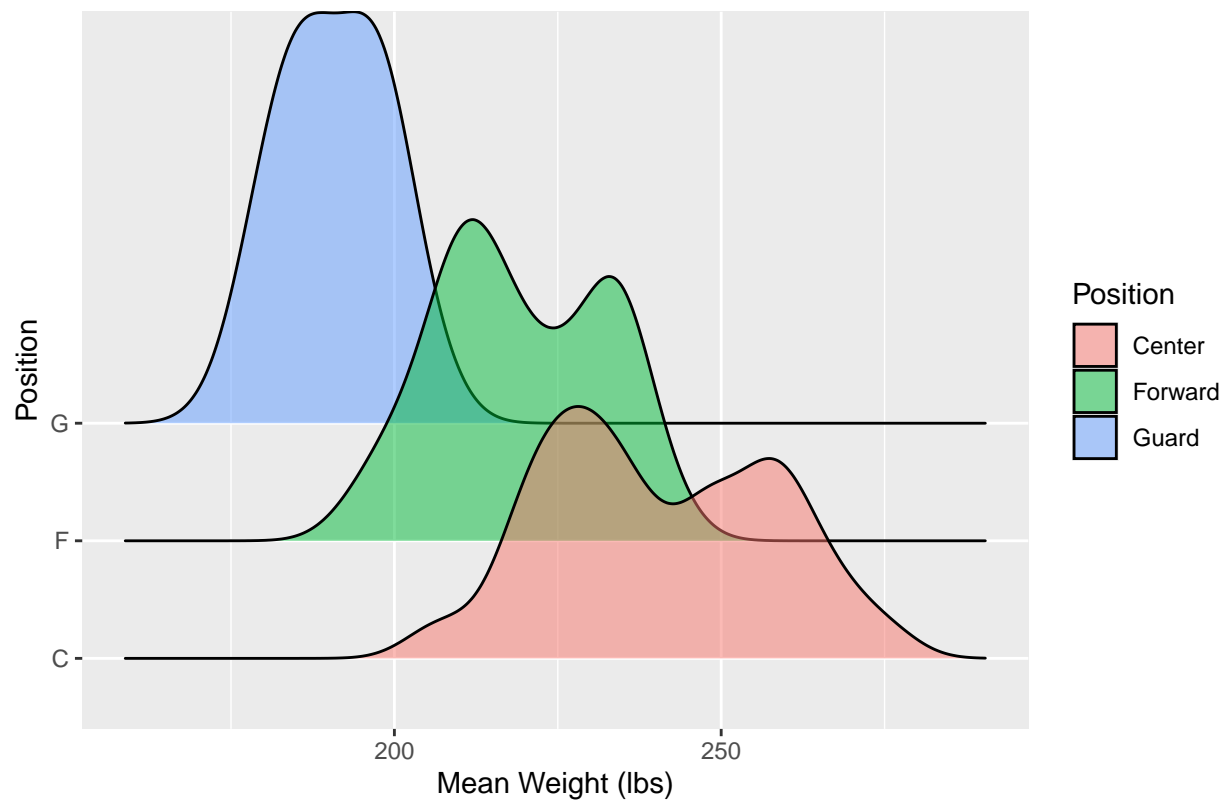
```
#
## WEIGHT
ggplot(mean_all_player_bios,
  aes(x = mean_weight,
    y = position,
    fill = position)) +

  geom_density_ridges(scale = 3.5, alpha = 0.5)+
  scale_fill_discrete(labels=c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Mean Weight (lbs)",
    y = "Position",
    title = "Average Player Weight vs. Positions",
    color = "Position",
    fill = "Position" )
```

```
## Picking joint bandwidth of 4.8
```

Average Player Weight vs. Positions



```
#-----
## BMI
ggplot(mean_all_player_bios,
  aes(x = mean_BMI,
    y = position,
    fill = position))+

  geom_density_ridges(scale = 3.5, alpha = 0.5)+
  scale_fill_discrete(labels =
    c("C" = "Center", "F" = "Forward", "G" = "Guard")) +

  labs(x = "Mean BMI", y = "Position",
    title = "Average Player BMI vs. Positions",
    color = "Position",
    fill = "Position")
```

```
## Picking joint bandwidth of 0.347
```

Average Player BMI vs. Positions

