# Problem Set 3

## Univariate Visualization

[YOUR NAME]

Due Date: 2023-02-10

# Getting Set Up

Open `RStudio` and create a new RMarkDown file ( `.Rmd` ) by going to `File -> New File -> R Markdown...`. Accept defaults and save this file as `[LAST NAME]_ps3.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_ps3.Rmd` file. Then change the `author: [YOUR NAME]` (line 4) to your name.

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 10 total points, plus four extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must both have the correct code **and include a comment describing what each line does**. In addition, some questions ask you to provide a written response in addition to the code. Unlike the first two problem sets, some of the code chunks are totally empty, requiring you to try writing the code from scratch. Make sure to comment each line, explaining what it is doing!

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace by midnight on 2023/02/10.

**Good luck!**

# Question 0

Require `tidyverse` and load the `nba_players_2018.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/nba_players_2018.Rds?raw=true') data to an object called `nba` . (Tip: use the `read_rds()` function with the link to the raw data.)

```
require(tidyverse)
```

```
## Loading required package: tidyverse
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.5.0
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

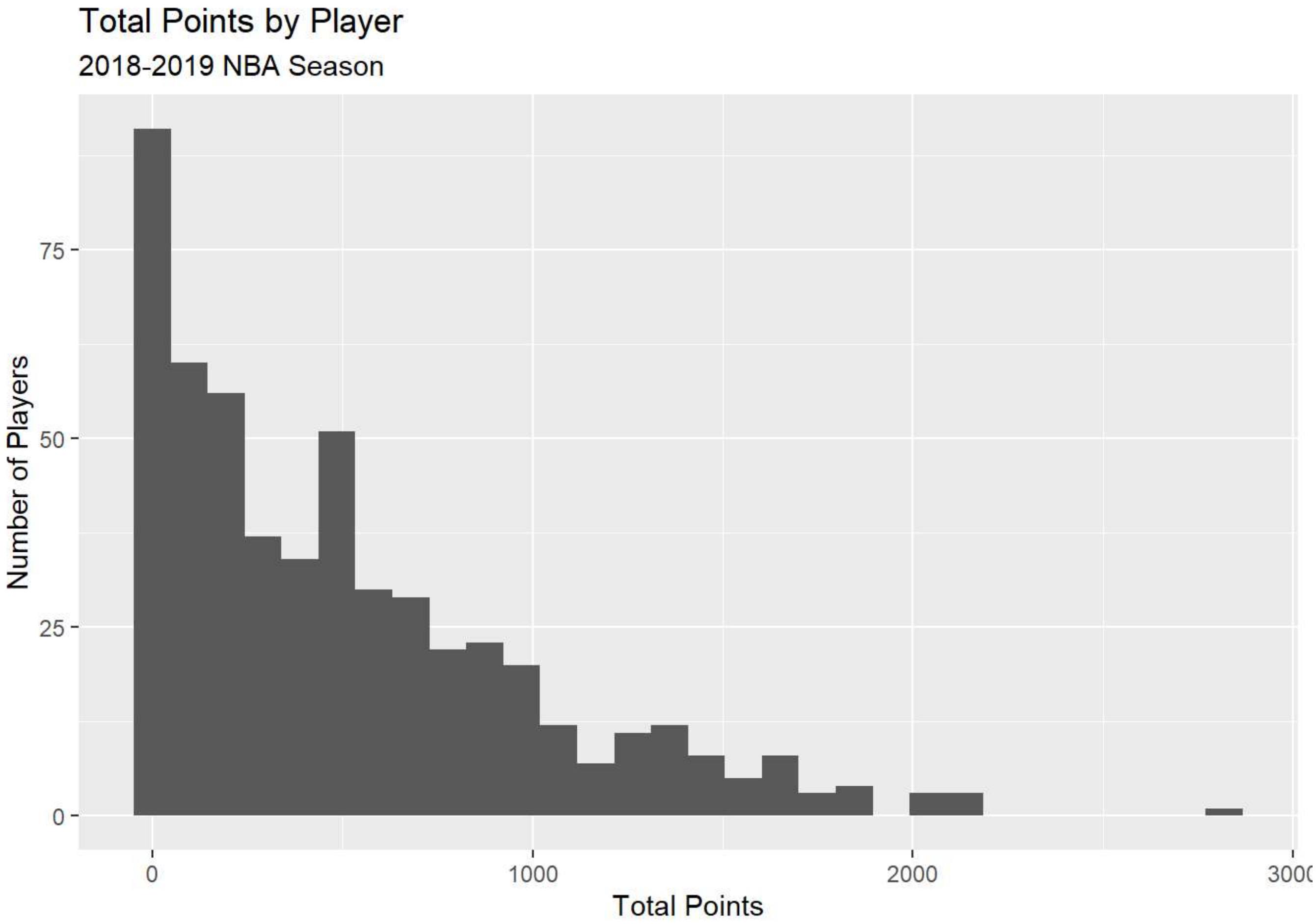
```
nba <- read_rds('../data/nba_players_2018.Rds') #https://github.com/jbisbee1/DS1000_S2023/blob/m
ain/Lectures/4_Uni_Multivariate/data/nba_players_2018.Rds?raw=true')
```

# Question 1 [1 point]

Plot the distribution of points scored by all NBA players in the 2018-2019 season. Explain why you chose the visualization that you did.

```
nba %>%
  ggplot(aes(x = pts)) + # Put the pts variable on the x-axis of a ggplot.
   geom_histogram() + # Choose the appropriate geom function to visualize.
  labs(title = 'Total Points by Player',# Write a clear title explaining the plot
       subtitle = '2018-2019 NBA Season',# Write a clear subtitle describing the data
       x = 'Total Points',# Write a clear x-axis label
       y = 'Number of Players') # Write a clear y-axis label
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



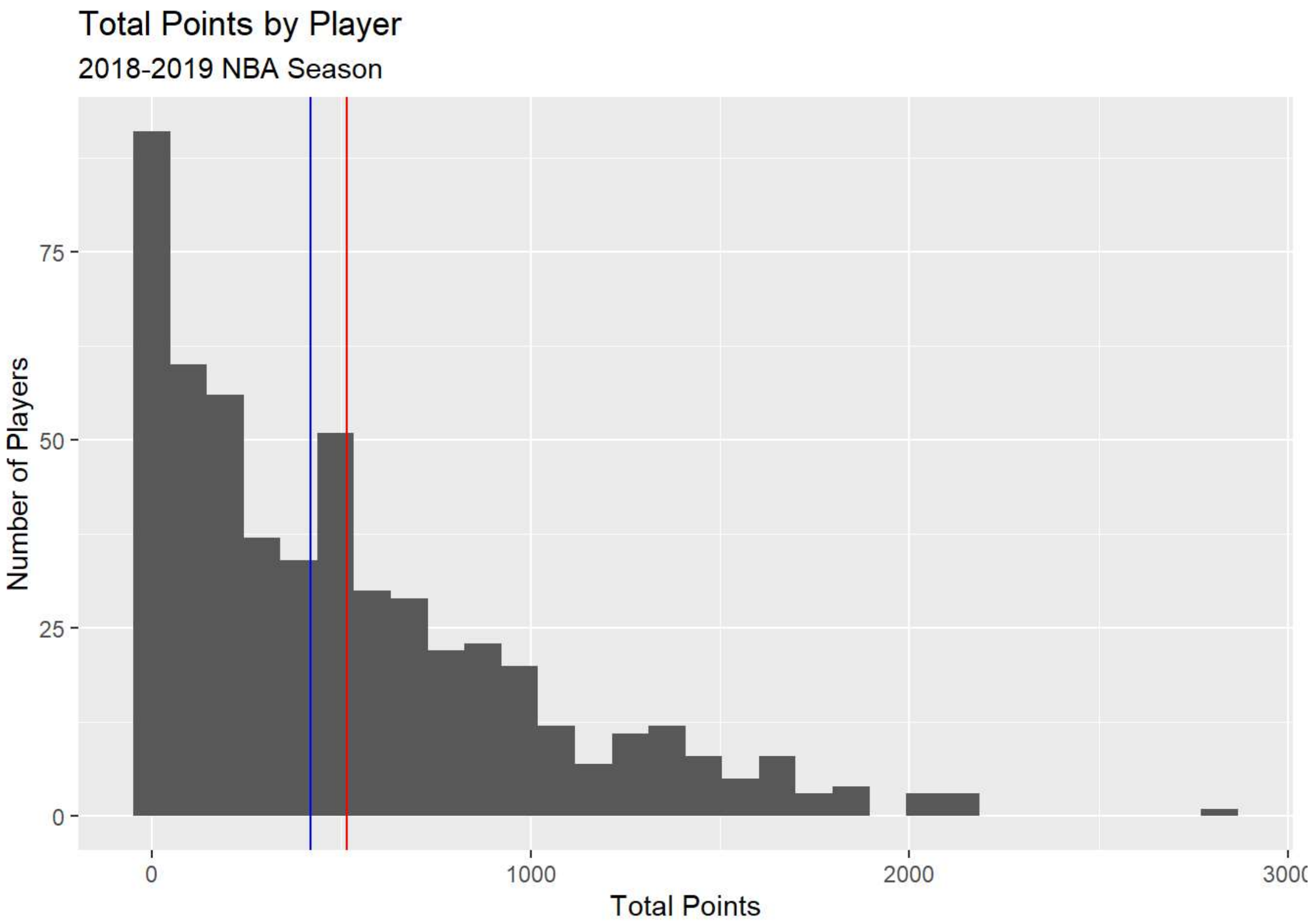
> I chose to use a geom_histogram because `pts` is a continuous variable.

# Question 2 [1 point]

Now recreate this plot but add a vertical line indicating the mean and median number of points in the data. Color the median line blue and the mean line red. Why is the median lower than the mean?

```
nba %>%
  ggplot(aes(x = pts)) + # Put the pts variable on the x-axis of a ggplot.
   geom_histogram() + # Choose the appropriate geom function to visualize.
  labs(title = 'Total Points by Player',# Write a clear title explaining the plot
       subtitle = '2018-2019 NBA Season',#  Write a clear subtitle describing the data
       x = 'Total Points',# Write a clear x-axis label
       y = 'Number of Players') + # Write a clear y-axis label
  geom_vline(xintercept = c(median(nba$pts)),color = 'blue') + # Median vertical line (blue)
  geom_vline(xintercept = c(mean(nba$pts)),color = 'red') # Mean vertical line (red)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

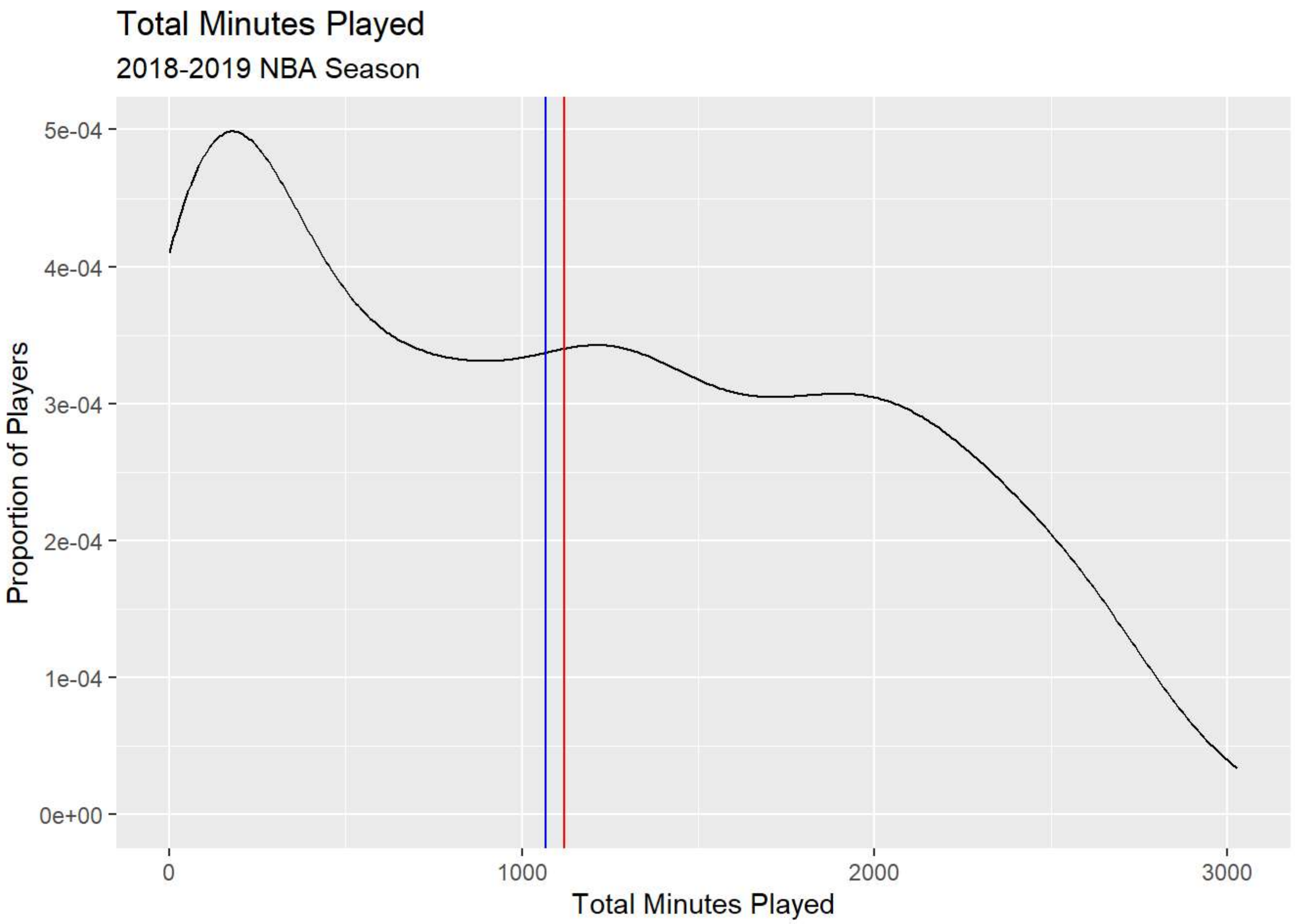


The mean is larger than the median because the data is right-skewed, reflecting the fact that there are a few players who score many points, and many players who do not.

# Question 3 [1 point + 1 EC]

Now visualize the distribution of the total minutes played ( `minutes` ). Again, justify your choice for the `geom_...` and compare the mean and median, again using blue and red lines. EC: Propose a theory for why the data looks this way.

```
nba %>%
  ggplot(aes(x = minutes)) + # Put the minutes variable on the x-axis
  geom_density() + # Chose density because it is continuous (histogram also works)
  labs(title = 'Total Minutes Played', # Good title
       subtitle = '2018-2019 NBA Season', # Good subtitle
       x = 'Total Minutes Played', # Good x-axis label
       y = 'Proportion of Players') +  # Good y-axis label
  geom_vline(xintercept = mean(nba$minutes),color = 'red') + # Mean vertical line
  geom_vline(xintercept = median(nba$minutes),color = 'blue') # Median vertical line
```

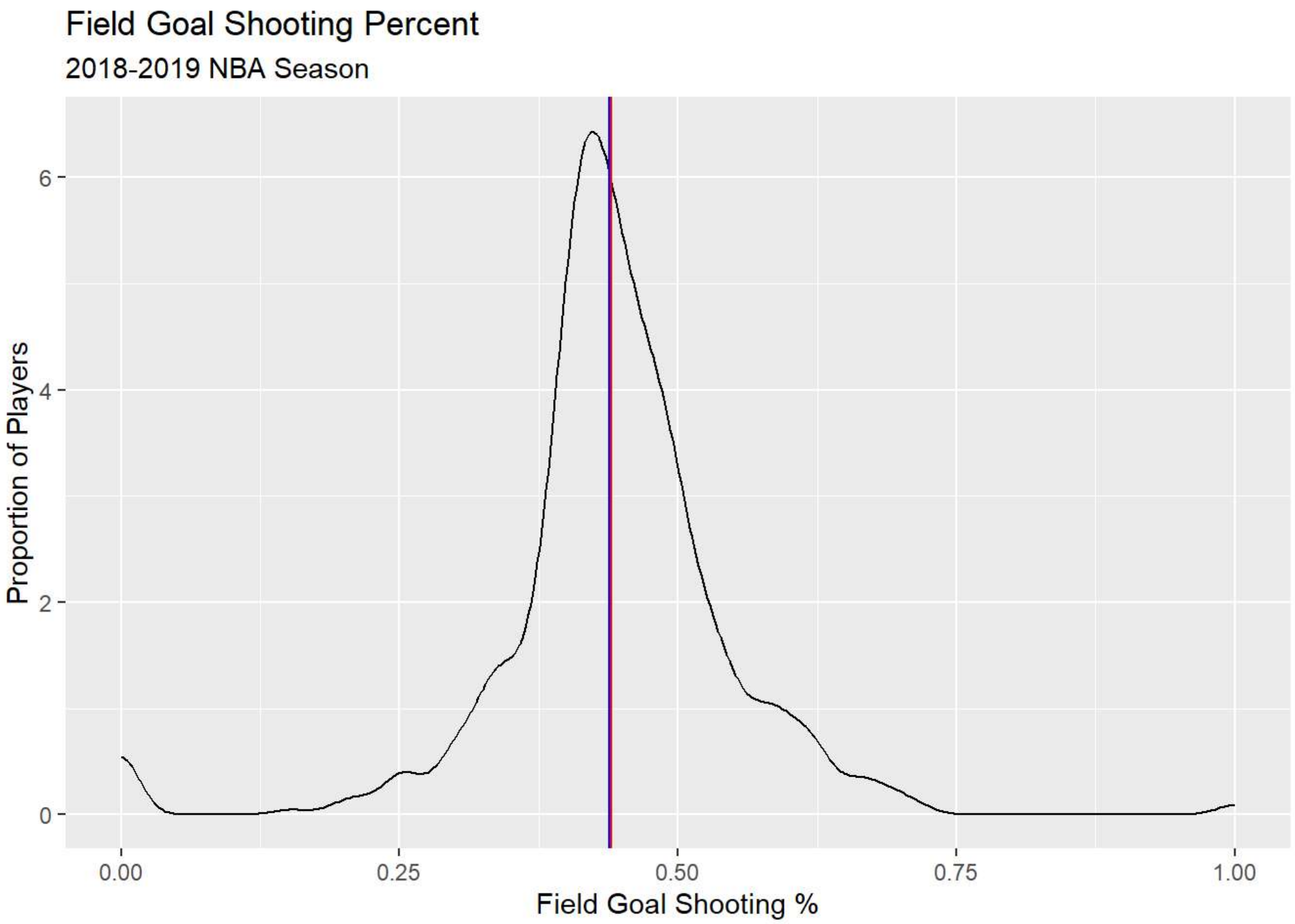


> I chose a geom_density this time, which is the other appropriate choice for continuous variables. The mean is larger than the median again, reflecting the fact that the data are right-skewed, meaning that a few players play many minutes and many players play only a few minutes. EC: I theorize that this reflects the style of building a competitive NBA team, which are typically organized around a few superstar players who play many minutes, and who are supported by many role players who do not play many minutes.

# Question 4 [1 point + 1 EC]

Now visualize the distribution of the field goal shooting percent ( `pctFG` ). Again, justify your choice for the `geom_...` and compare the mean and median, again using blue and red lines. EC: Propose an explanation for why this variable is **not** right-skewed, unlike the `pts` variable from Q2.

```
nba %>%
  ggplot(aes(x = pctFG)) + # Put field goal % on the x-axis
  geom_density() + # Chose density again (could have been histogram)
  labs(title = 'Field Goal Shooting Percent', # Clear title
       subtitle = '2018-2019 NBA Season', # Clear subtitle
       x = 'Field Goal Shooting %', # clear x-axis label
       y = 'Proportion of Players') +  # clear y-axis label
  geom_vline(xintercept = mean(nba$pctFG),color = 'red') +  # mean vertical line
  geom_vline(xintercept = median(nba$pctFG),color = 'blue') # median vertical line
```

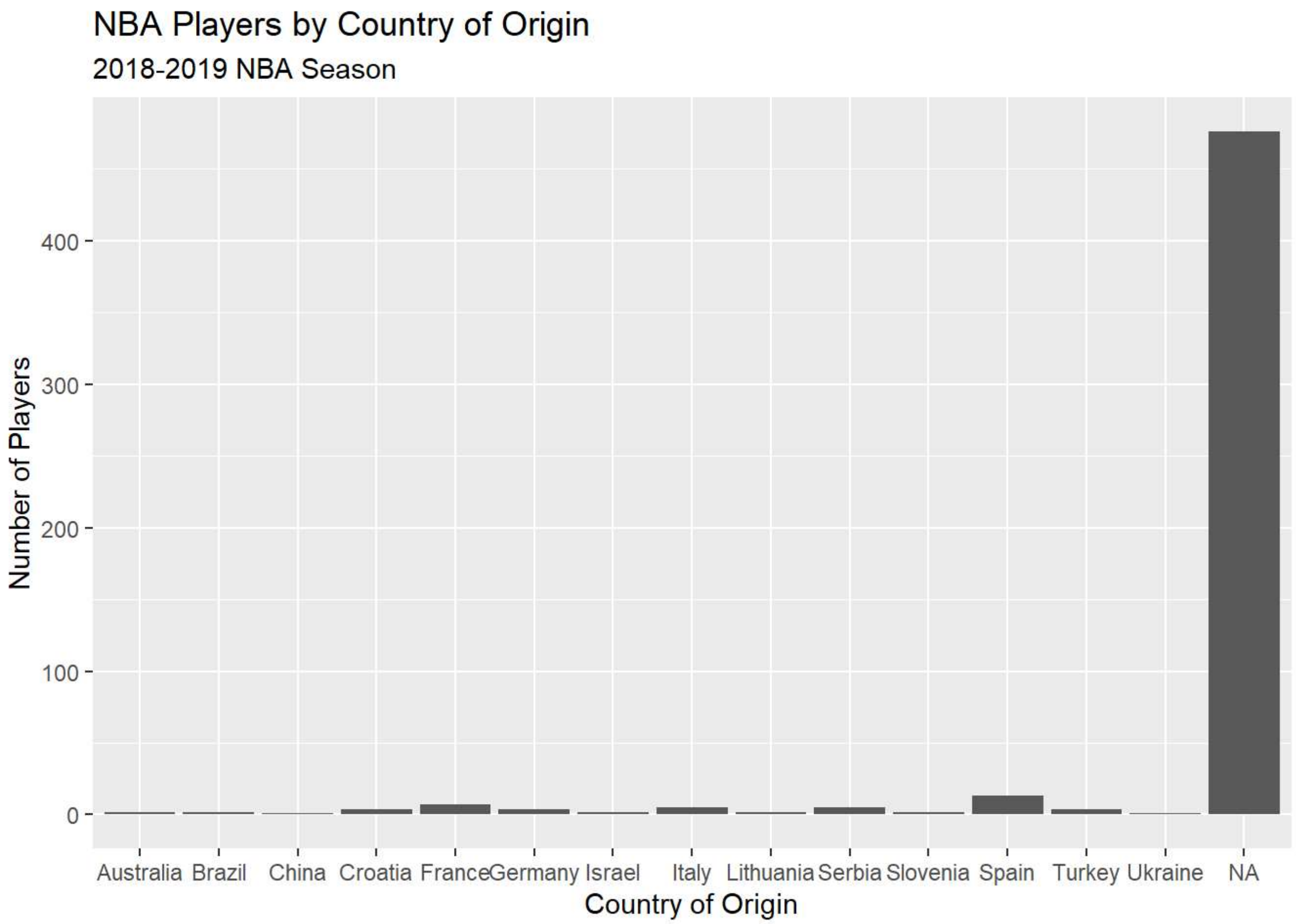


I chose a geom_density this time, which is the other appropriate choice for continuous variables. The mean and median are basically identical for the field goal shooting percent, meaning that the data is not skewed in either direction. EC: I theorize that this data is not right-skewed like the `pts` variable, because it measures an innate ability which is not affected by the management strategies of the NBA that are organized around a few superstars and many supporting players.

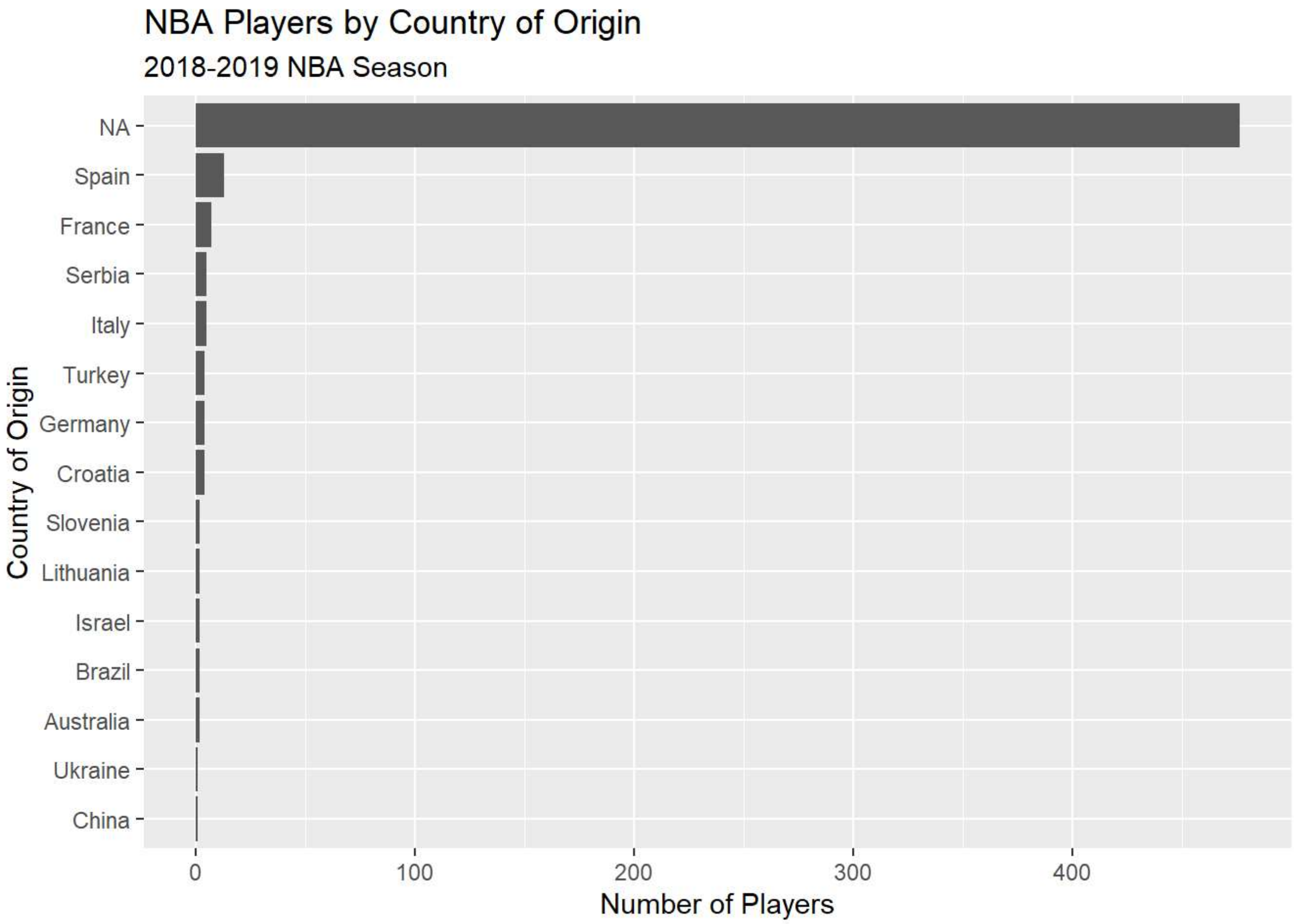
# Question 5 [1 point + 1 EC]

Now examine the `country` variable. Which country are most NBA players from? Visualize this variable using the appropriate `geom_...`, and justify your reason for choosing it. EC: Tweak the plot to put the country labels on the y-axis, ordered by frequency.

```
# Basic Plot (NOT EC)
nba %>%
  ggplot(aes(x = country)) + # Put the country variable on the x-axis of a ggplot.
  geom_bar() + # Choose the appropriate geom function to visualize.
  labs(title = 'NBA Players by Country of Origin', # Write a clear title explaining the plot
       subtitle = '2018-2019 NBA Season', # Write a clear subtitle describing the data
       y = 'Number of Players', # Write a clear x-axis label
       x = 'Country of Origin') # Write a clear y-axis label
```



```
# EC Plot: Insert code below and comment each line!
nba %>%
  count(country) %>% # count the number of players by country
  ggplot(aes(y = reorder(country,n),x = n)) + # place the country on the y-axis, reordered by th
e number of players. Put the number of players on the x-axis
  geom_bar(stat = 'identity') + # Set stat = 'identity' because we are setting both x and y axes
  labs(title = 'NBA Players by Country of Origin', # Clear title
       subtitle = '2018-2019 NBA Season', # Clear subtitle
       x = 'Number of Players', # Clear x-axis label
       y = 'Country of Origin') # Clear y-axis label
```



> The majority of NBA players are from `NA`, which is likely just the United States. I chose the `geom_bar()` visualization since the `country` variable is categorical.

# Question 6 [3 points]

Perform a thorough univariate description of the variable `agePlayer`. Start by determining what type of measure it is (i.e., continuous, ordered categorical, etc.). Then, based on this conclusion, summarize it with either `summary()` or `count()`. Finally, visualize it. In the write-up, explain each part of this process and defend your choice of the `geom_...` used to visualize the data. Make sure to label the plot!

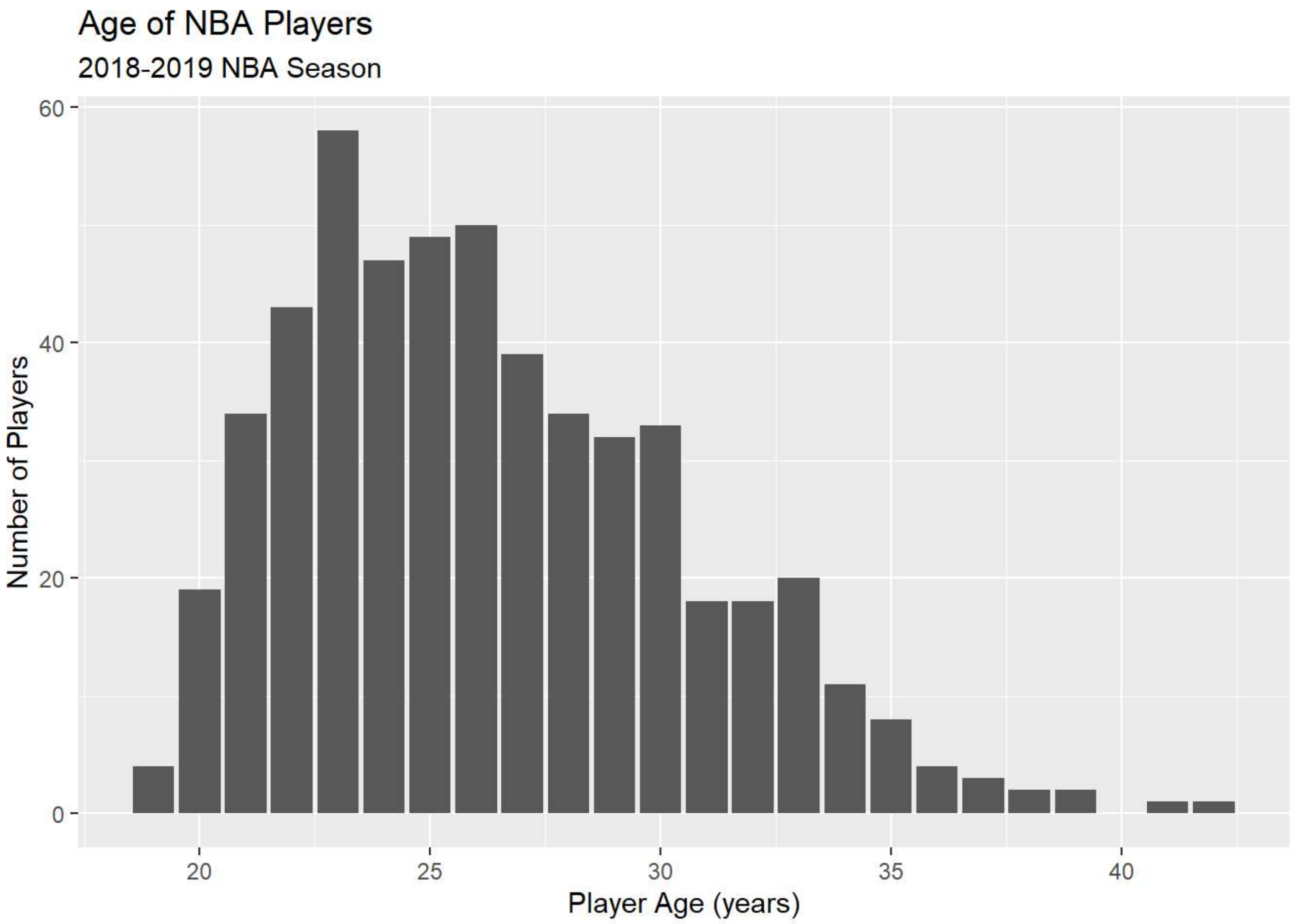
```
glimpse(nba %>% select(agePlayer)) # Look at the variable first
```

```
## Rows: 530
## Columns: 1
## $ agePlayer <dbl> 33, 28, 25, 25, 21, 21, 23, 22, 23, 26, 28, 24, 25, 25, 21, …
```

```
summary(nba$agePlayer) # Summarize the variable with either summary() or count()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   19.00   23.00   26.00   26.35   29.00   42.00
```

```
nba %>% # Visualize with ggplot() (don't forget to label the plot!)
  ggplot(aes(x = agePlayer)) + # Put the agePlayer variable on the x-axis of a ggplot.
  geom_bar() + # Choose the appropriate geom function to visualize.
  labs(title = "Age of NBA Players", # Write a clear title explaining the plot
       subtitle = '2018-2019 NBA Season', # Write a clear subtitle describing the data
       x = 'Player Age (years)', # Write a clear x-axis label
       y = 'Number of Players') # Write a clear y-axis label
```



Age of NBA Players

2018-2019 NBA Season

> I started by looking at the data with `glimpse()`. Based on this inspection, I determined that player age is basically a continuous measure, although is expressed as a whole number (an integer). As such, I summarized it with the `summary()` command, indicating that the majority of players are less than 26 years old or younger, and that the oldest player is 42 years old. I then visualized it with `geom_bar()`. I chose this geom because there are only a few continuous measures. I could have also chosen `geom_histogram()` or `geom_density()`, since it is a continuous measure.

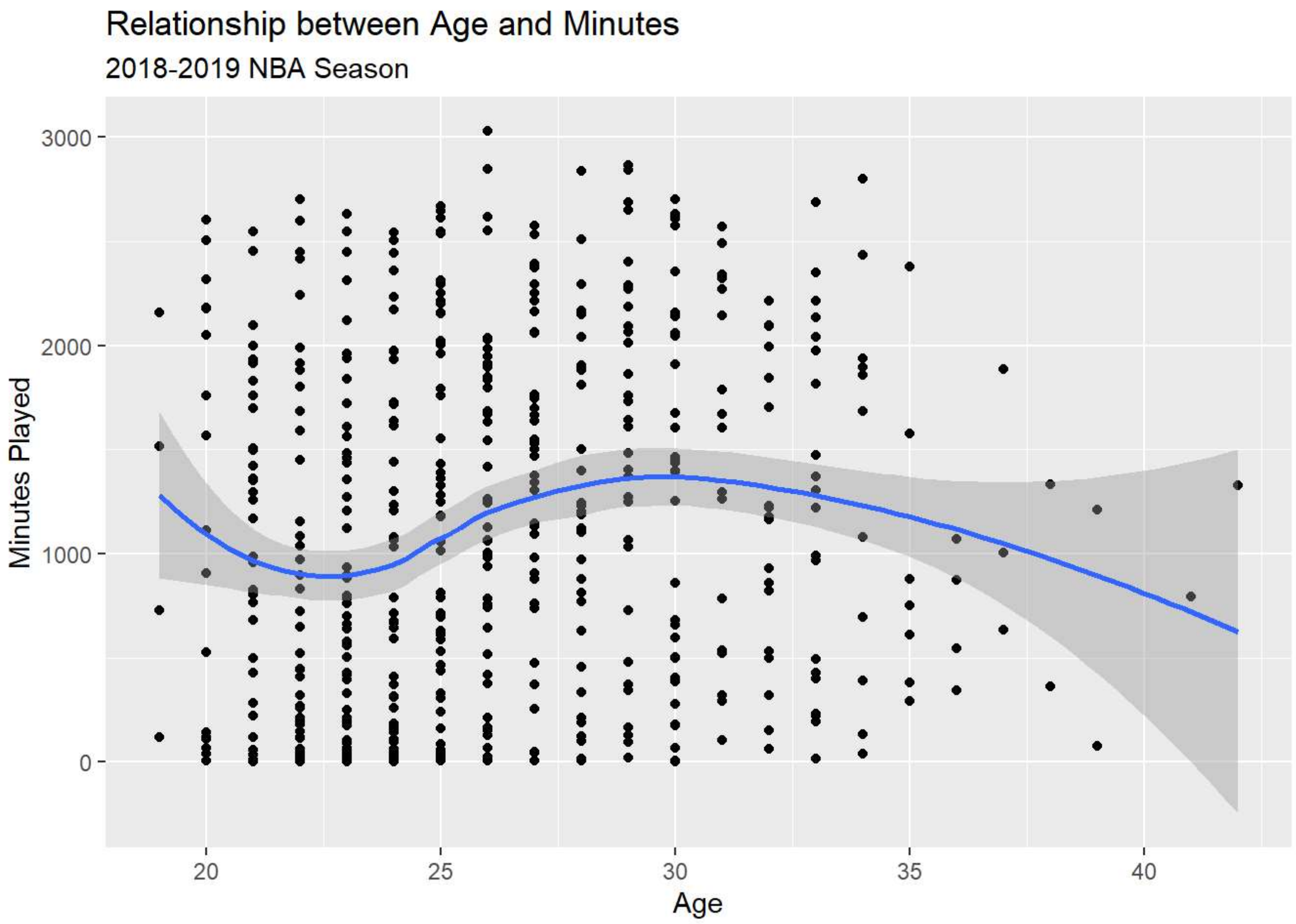
# Question 7 [2 points + 1 EC]

Consider the following research question: do coaches give more minutes to younger players? Hypothesize an answer to this question, and describe your thought process (theory). EC: generate a multivariate visualization that provides an answer to this question. Does the data support your hypothesis?

> I think that there is a trade-off between minutes and age for players. I assume that the older the player is, the more tired they get, meaning that older players should play fewer minutes than younger players. However, I also assume that young players are less experienced, which means that younger players should play fewer minutes than older players. Thus, I hypothesis that the relationship between minutes and age should be an inverted U-shape, with the most minutes being played by those in the mid to late 20s, and the fewest being played by the youngest players and the oldest players.

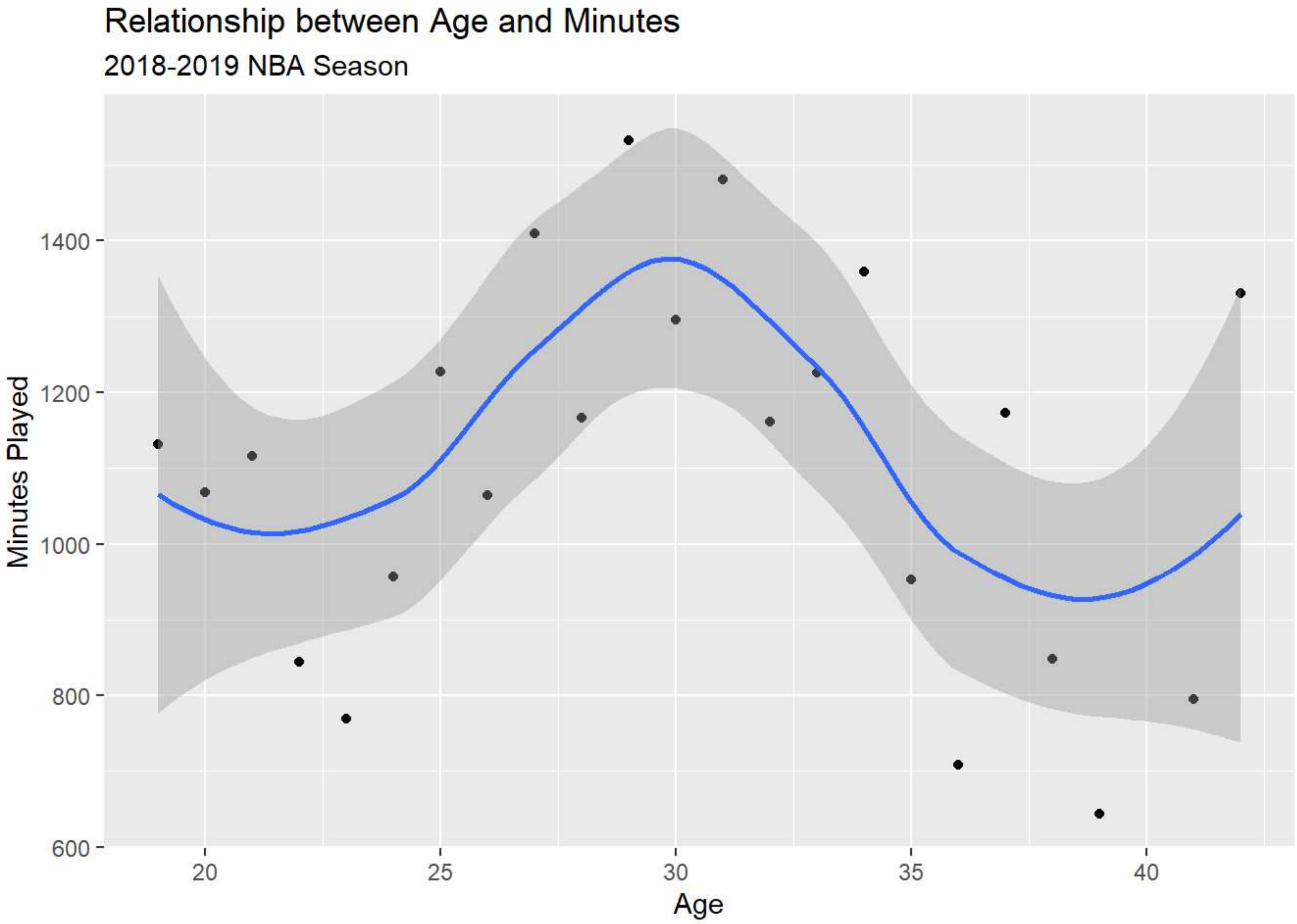
```
# EC: INSERT CODE HERE. (Don't forget to comment and add labels!)
nba %>%
  ggplot(aes(x = agePlayer,y = minutes)) + # Plotting the relationship between age and minutes
  geom_point() + # continuous by continuous so scatterplot
  geom_smooth() + # Line of best fit to help with visual clarity
  labs(title = 'Relationship between Age and Minutes', # Clear title
       subtitle = '2018-2019 NBA Season', # Clear subtitle
       x = 'Age', # Clear x-axis variable
       y = 'Minutes Played') # Clear y-axis variable
```

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



```
# EC Version 2 (easier to read)
nba %>%
  group_by(agePlayer) %>%
  summarise(minutes = mean(minutes,na.rm=T)) %>% # Calculate the average minutes by player age
  ggplot(aes(x = agePlayer,y = minutes)) +  # Plot relationship between age and average minutes
  geom_point() + # ContXcont so scatterplot
  geom_smooth() + # Line of best fit
  labs(title = 'Relationship between Age and Minutes', # Clear title
       subtitle = '2018-2019 NBA Season', # Clear subtitle
       x = 'Age', # Clear x-axis variable
       y = 'Minutes Played') # Clear y-axis variable
```

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


Relationship between Age and Minutes
2018-2019 NBA Season
Minutes Played
1400
1200
1000
800
600
20
25
30
35
40
Age