Problem Set 3

Univariate Visualization

Alex Tomack

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   1.0.0
## ✓ 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()
```

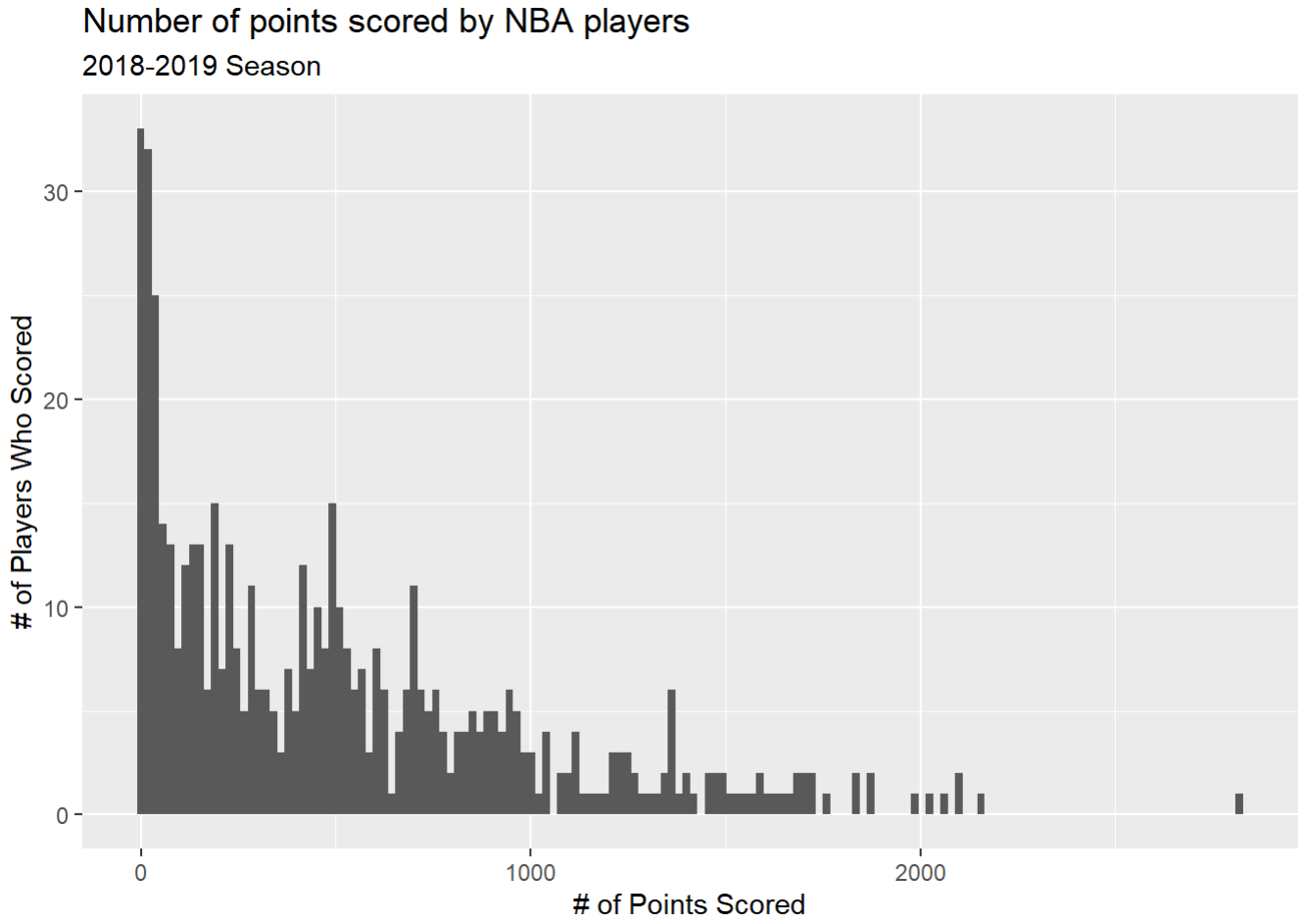
```
nba<-read_rds('https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/nba_players_2018.Rds?raw=true')
glimpse(nba)
```

```
## Rows: 530
## Columns: 37
## $ namePlayer         <chr> "LaMarcus Aldridge", "Quincy Acy", "Steven Adams", …
## $ idPlayer           <dbl> 200746, 203112, 203500, 203518, 1628389, 1628959, 1…
## $ slugSeason         <chr> "2018-19", "2018-19", "2018-19", "2018-19", "2018-1…
## $ numberPlayerSeason <dbl> 12, 6, 5, 2, 1, 0, 0, 0, 0, 0, 8, 5, 4, 3, 1, 1, 1,…
## $ isRookie           <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE…
## $ slugTeam           <chr> "SAS", "PHX", "OKC", "OKC", "MIA", "CHI", "UTA", "C…
## $ idTeam             <dbl> 1610612759, 1610612756, 1610612760, 1610612760, 161…
## $ gp                 <dbl> 81, 10, 80, 31, 82, 10, 38, 19, 34, 7, 81, 72, 43, …
## $ gs                 <dbl> 81, 0, 80, 2, 28, 1, 2, 3, 1, 0, 81, 72, 40, 4, 80,…
## $ fgm                <dbl> 684, 4, 481, 56, 280, 13, 67, 11, 38, 3, 257, 721, …
## $ fga                <dbl> 1319, 18, 809, 157, 486, 39, 178, 36, 110, 10, 593,…
## $ pctFG              <dbl> 0.519, 0.222, 0.595, 0.357, 0.576, 0.333, 0.376, 0.…
## $ fg3m               <dbl> 10, 2, 0, 41, 3, 3, 32, 6, 25, 0, 96, 52, 9, 24, 6,…
## $ fg3a               <dbl> 42, 15, 2, 127, 15, 12, 99, 23, 74, 4, 280, 203, 34…
## $ pctFG3             <dbl> 0.2380952, 0.1333333, 0.0000000, 0.3228346, 0.20000…
## $ pctFT              <dbl> 0.847, 0.700, 0.500, 0.923, 0.735, 0.667, 0.750, 1.…
## $ fg2m               <dbl> 674, 2, 481, 15, 277, 10, 35, 5, 13, 3, 161, 669, 1…
## $ fg2a               <dbl> 1277, 3, 807, 30, 471, 27, 79, 13, 36, 6, 313, 1044…
## $ pctFG2             <dbl> 0.5277995, 0.6666667, 0.5960347, 0.5000000, 0.58811…
## $ agePlayer          <dbl> 33, 28, 25, 25, 21, 21, 23, 22, 23, 26, 28, 24, 25,…
## $ minutes            <dbl> 2687, 123, 2669, 588, 1913, 120, 416, 194, 428, 22,…
## $ ftm                <dbl> 349, 7, 146, 12, 166, 8, 45, 4, 7, 1, 150, 500, 37,…
## $ fta                <dbl> 412, 10, 292, 13, 226, 12, 60, 4, 9, 2, 173, 686, 6…
## $ oreb               <dbl> 251, 3, 391, 5, 165, 11, 3, 3, 11, 1, 112, 159, 48,…
## $ dreb               <dbl> 493, 22, 369, 43, 432, 15, 20, 16, 49, 3, 498, 739,…
## $ treb               <dbl> 744, 25, 760, 48, 597, 26, 23, 19, 60, 4, 610, 898,…
## $ ast                <dbl> 194, 8, 124, 20, 184, 13, 25, 5, 65, 6, 104, 424, 1…
## $ stl                <dbl> 43, 1, 117, 17, 71, 1, 6, 1, 14, 2, 68, 92, 54, 22,…
## $ blk                <dbl> 107, 4, 76, 6, 65, 0, 6, 4, 5, 0, 33, 110, 37, 13, …
## $ tov                <dbl> 144, 4, 135, 14, 121, 8, 33, 6, 28, 2, 72, 268, 58,…
## $ pf                 <dbl> 179, 24, 204, 53, 203, 7, 47, 13, 45, 4, 143, 232, …
## $ pts                <dbl> 1727, 17, 1108, 165, 729, 37, 211, 32, 108, 7, 760,…
## $ urlNBAAPI          <chr> "https://stats.nba.com/stats/playercareerstats?Leag…
## $ n                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ org                <fct> Texas, NA, Other, FC Barcelona Basquet, Kentucky, N…
## $ country            <chr> NA, NA, NA, "Spain", NA, NA, NA, NA, NA, NA, NA, "S…
## $ idConference       <int> 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, …
```

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(bins=150) + # Choose the appropriate geom function to visualize.
  labs(title = 'Number of points scored by NBA players', # Write a clear title explaining the pl
ot
       subtitle = '2018-2019 Season', # Write a clear subtitle describing the data
       x = '# of Points Scored', # Write a clear x-axis label
       y = '# of Players Who Scored') # Write a clear y-axis label
```

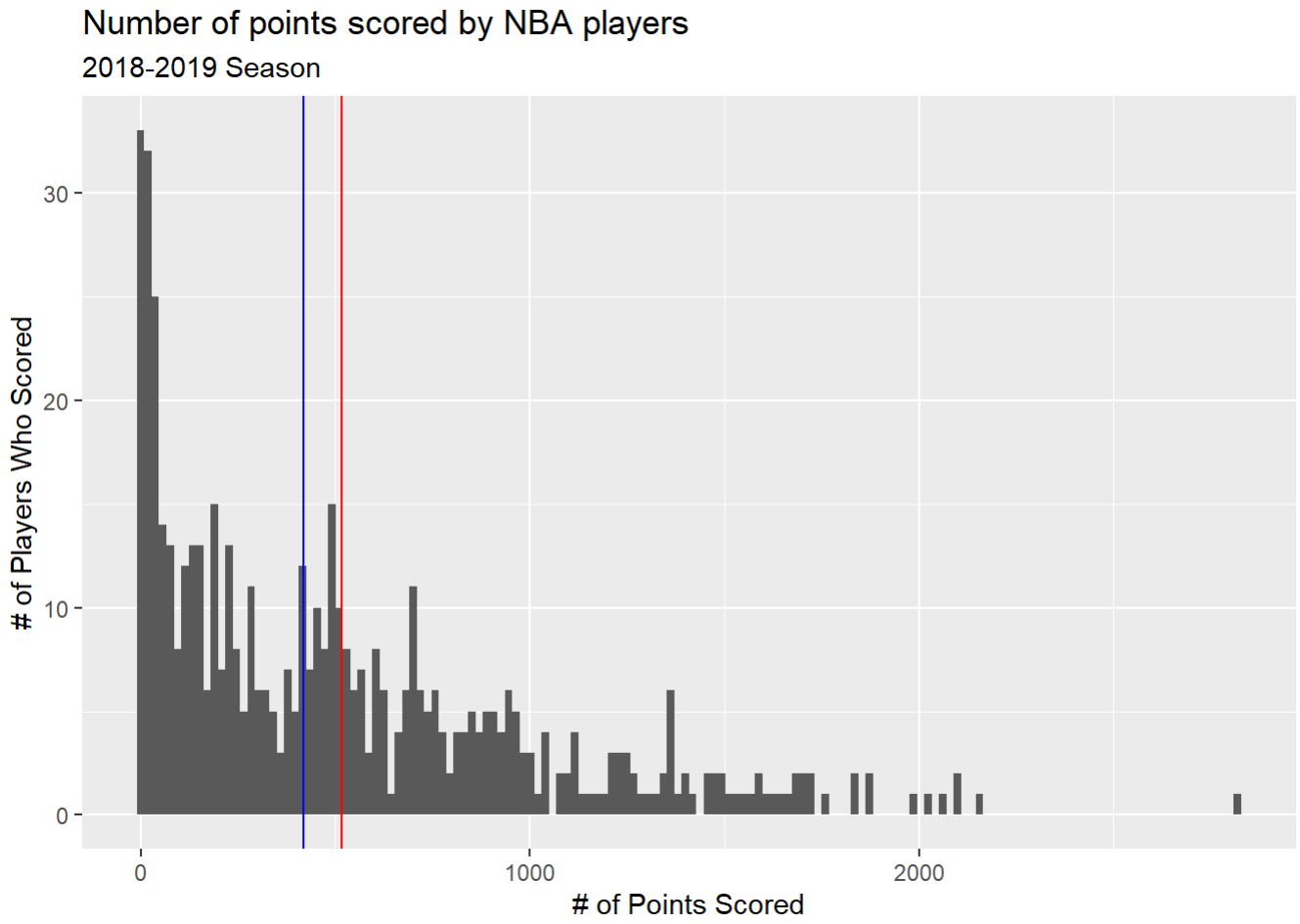


Since we're visualizing a continuous variable, not a categorical one, we want to plot with either a histogram or a density function. Histogram gives a nice visualization because there aren't any gaps between the bins.

Question 2 [1 point]

Now recreate this plot but add two vertical lines 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(bins=150) + # Choose the appropriate geom function to visualize.
  labs(title = 'Number of points scored by NBA players', # Write a clear title explaining the pl
ot
       subtitle = '2018-2019 Season', # Write a clear subtitle describing the data
       x = '# of Points Scored', # Write a clear x-axis label
       y = '# of Players Who Scored')+ # Write a clear y-axis label
  geom_vline(xintercept = median(nba$pts),color = 'blue') + # Median vertical line (blue)
  geom_vline(xintercept = mean(nba$pts),color = 'red') # Mean vertical line (red)
```

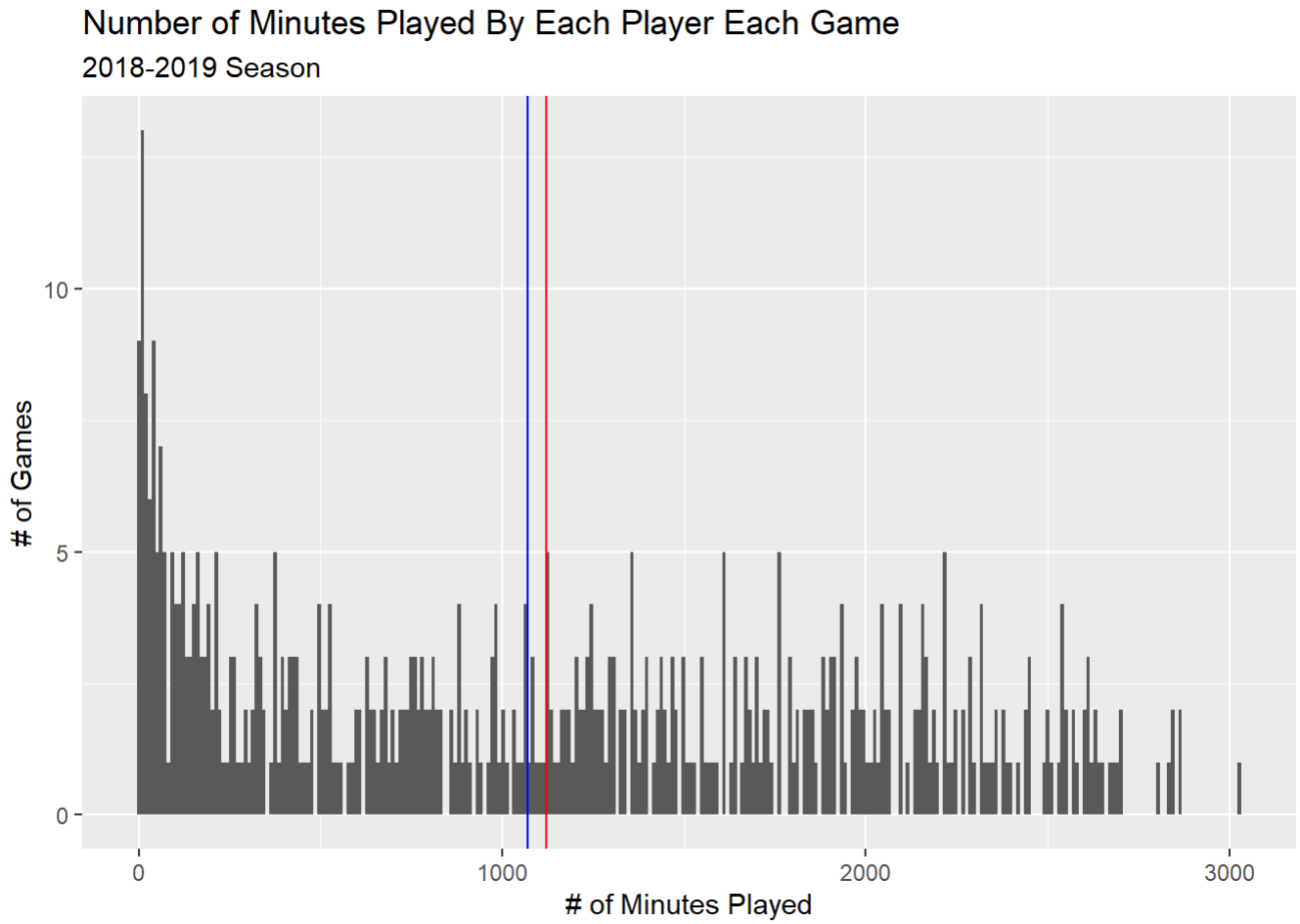


Mean is skewed by outliers– that one person who scored more than 2000 points is skewing the mean to be much higher.

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)) + #look inside nba data, plot minutes on the x-axis
  geom_histogram(bins=300) +# using a histogram since we have a continuous variable
  labs(title='Number of Minutes Played By Each Player Each Game', #title
       subtitle='2018-2019 Season', # subtitle
       x= '# of Minutes Played', #x-axis label
       y='# of Games')+#y-axis label
  geom_vline(xintercept=mean(nba$minutes), color="red") + # vertical line drawn at mean
  geom_vline(xintercept=median(nba$minutes), color="blue") # vertical line drawn at median
```

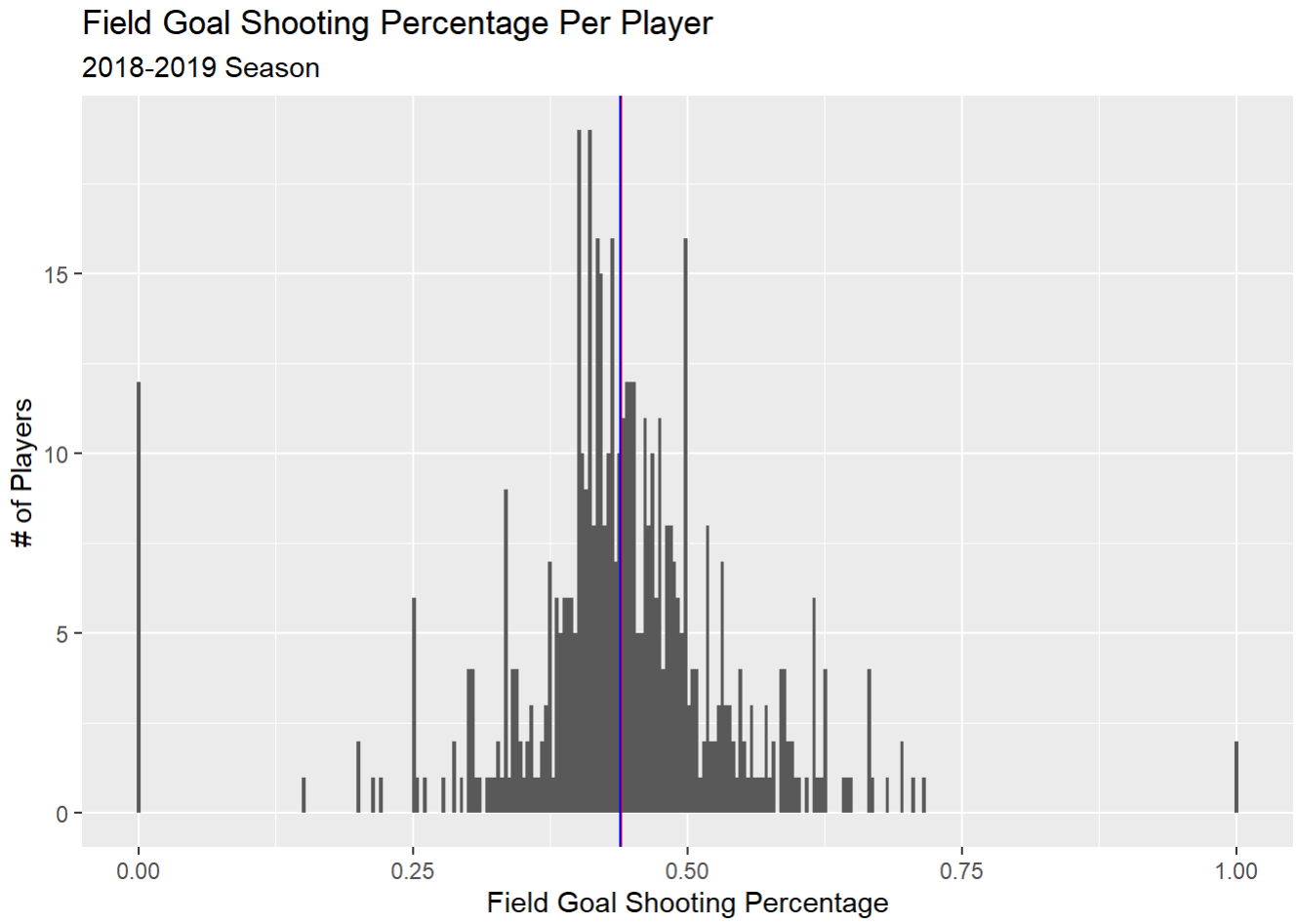


We're still using a histogram for the minutes data since it's a continuous variable. The mean is still a bit higher than the median because of the outlier around 3000 minutes, but data is less skewed for this variable. Our data looks right skewed because there are certain crowd favourites in the NBA– meaning they get more play time, while others are just benchwarmers.

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)) + #look inside nba data, plot pctFG on the x-axis
  geom_histogram(bins=300) +# using a histogram since we have a continuous variable
  labs(title='Field Goal Shooting Percentage Per Player', #title
       subtitle='2018-2019 Season', # subtitle
       x= 'Field Goal Shooting Percentage', #x-axis label
       y='# of Players')+#y-axis label
  geom_vline(xintercept=mean(nba$pctFG), color="red") + # vertical line drawn at mean
  geom_vline(xintercept=median(nba$pctFG), color="blue") # vertical line drawn at median
```

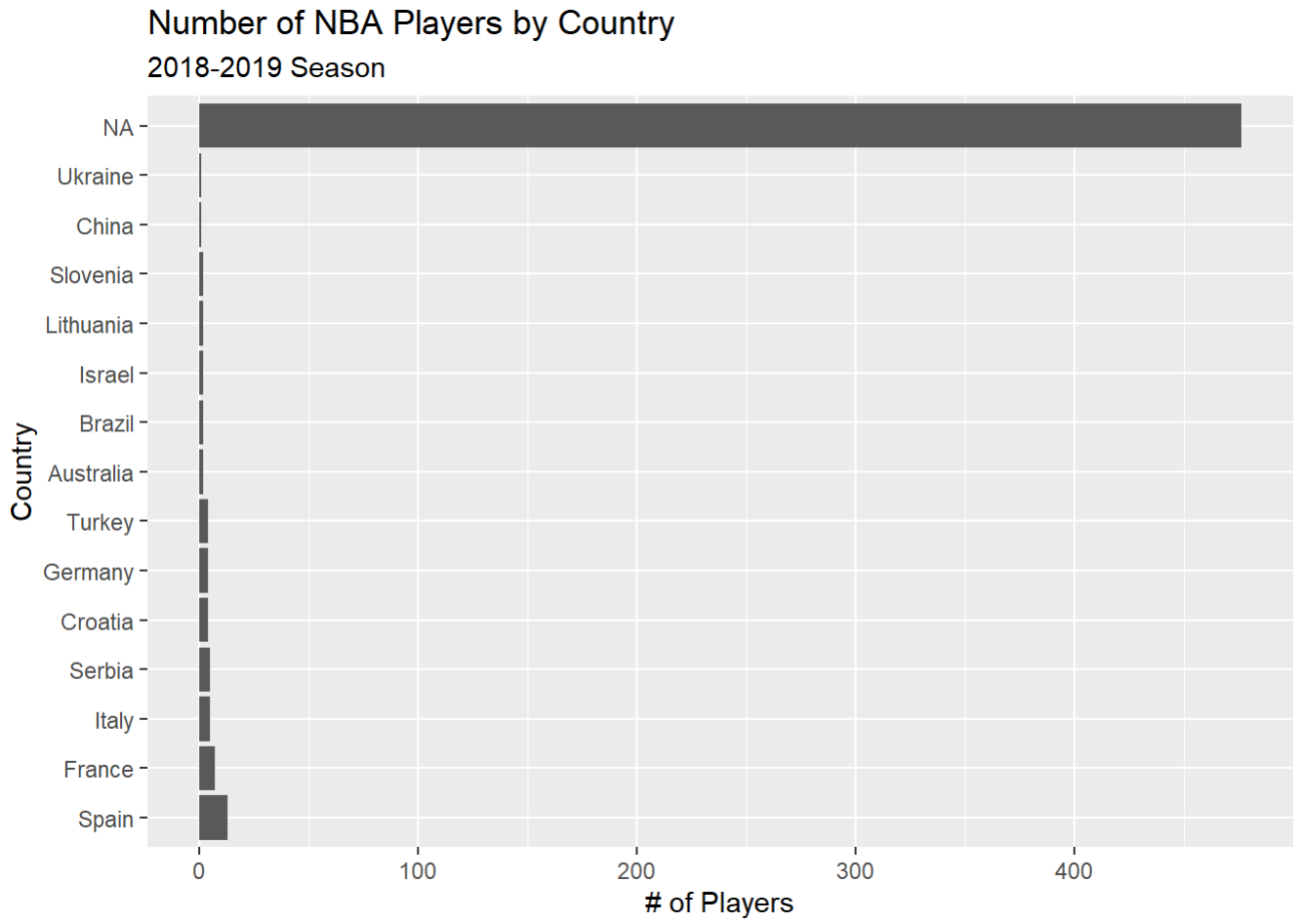


We're still using a histogram since we have a continuous variable. Now, we see the mean and median are pretty close to each other. We should expect to see a normal distribution with the mean roughly centered at 50%, since there are two outcomes– you either hit or miss the field goal shot.

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)
# EC Plot: Insert code below and comment each line!
nba %>%
  ggplot(aes(x=fct_infreq(country))) + #look inside nba data, plot country on the x-axis
  geom_bar() +# using a bar chart since categorical
  coord_flip()+ #flip categories to be on y axis and freq on x
  labs(title='Number of NBA Players by Country', #title
       subtitle='2018-2019 Season', # subtitle
       x= 'Country', #x-axis label
       y='# of Players')#y-axis label
```



Most players are from Spain, but there's a ludicrous amount of missing data.

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!

```
# 1: Look
glimpse(nba$agePlayer) # Look at the variable first
```

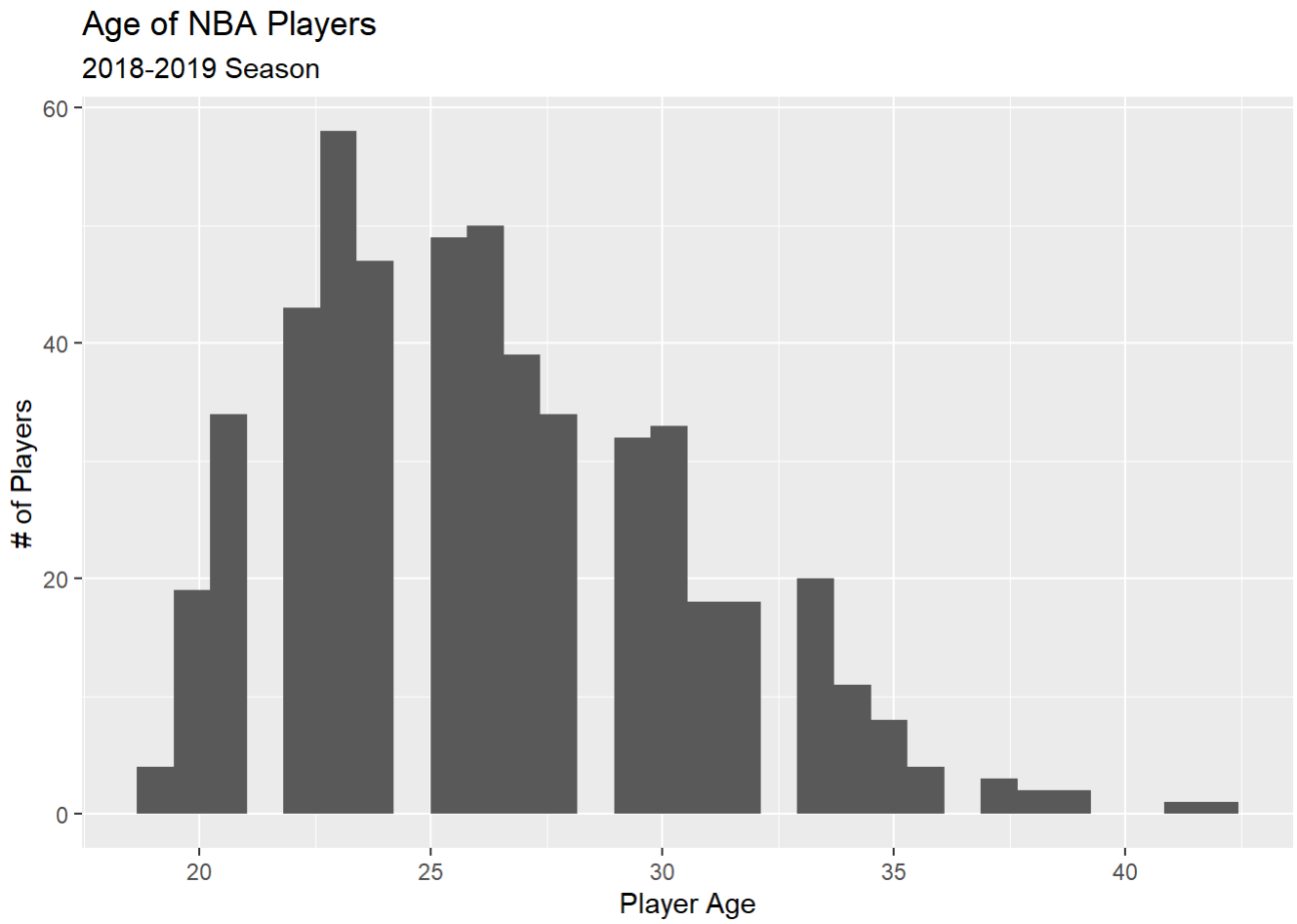
```
##  num [1:530] 33 28 25 25 21 21 23 22 23 26 ...
```

```
# 2: Summary statistics
# Summarize the variable with either summary() or count()
summary(nba$agePlayer)
```

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

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

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



First had to take a look at the variable to see what kind of data I'm working with. Type is a number, not binary. Based on that, was able to decide that a histogram would be the best visualization to use since we're working with a continuous variable.

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?

```
#took a look at the variables we have, to answer the question we need to use the agePlayer and m
inutes variables.
nba_clean <- nba %>%
  select(minutes, agePlayer) %>%
  mutate(mean_minutes=mean(minutes))
```

Do coaches give more minutes to younger players? Likely, no– games are about winning, not necessarily about giving the new players a ton of time. Play your best hand!