Problem Set 4

Univariate Analysis

[YOUR NAME]

Due Date: 2024-02-09

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]_ps4.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[LAST NAME]_ps4.Rmd` file. Then change the `author: [Your Name]` to your name.

We will be using the `nba_players_2018.Rds` file from the course github page (https://github.com/jbisbee1/DS1000_S2024/blob/main/data/nba_players_2018.Rds).

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 8 total points, plus two extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

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 on Friday by midnight. Also note that the TAs and professors will not respond to Campuswire posts after 5PM on Friday, so don't wait until the last minute to get started!

Good luck!

ChatGPT Link [Optional]

*Copy the link to ChatGPT you used here: _____

Question 0

Require `tidyverse` and an additional package called `haven` (remember to `install.packages("haven")` if you don't have it yet), and then load the `nba_players_2018.Rds` (https://github.com/jbisbee1/DS1000_S2024/blob/main/data/nba_players_2018.Rds?raw=true') data to an object called `nba` .

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
require(haven)
```

```
## Loading required package: haven
```

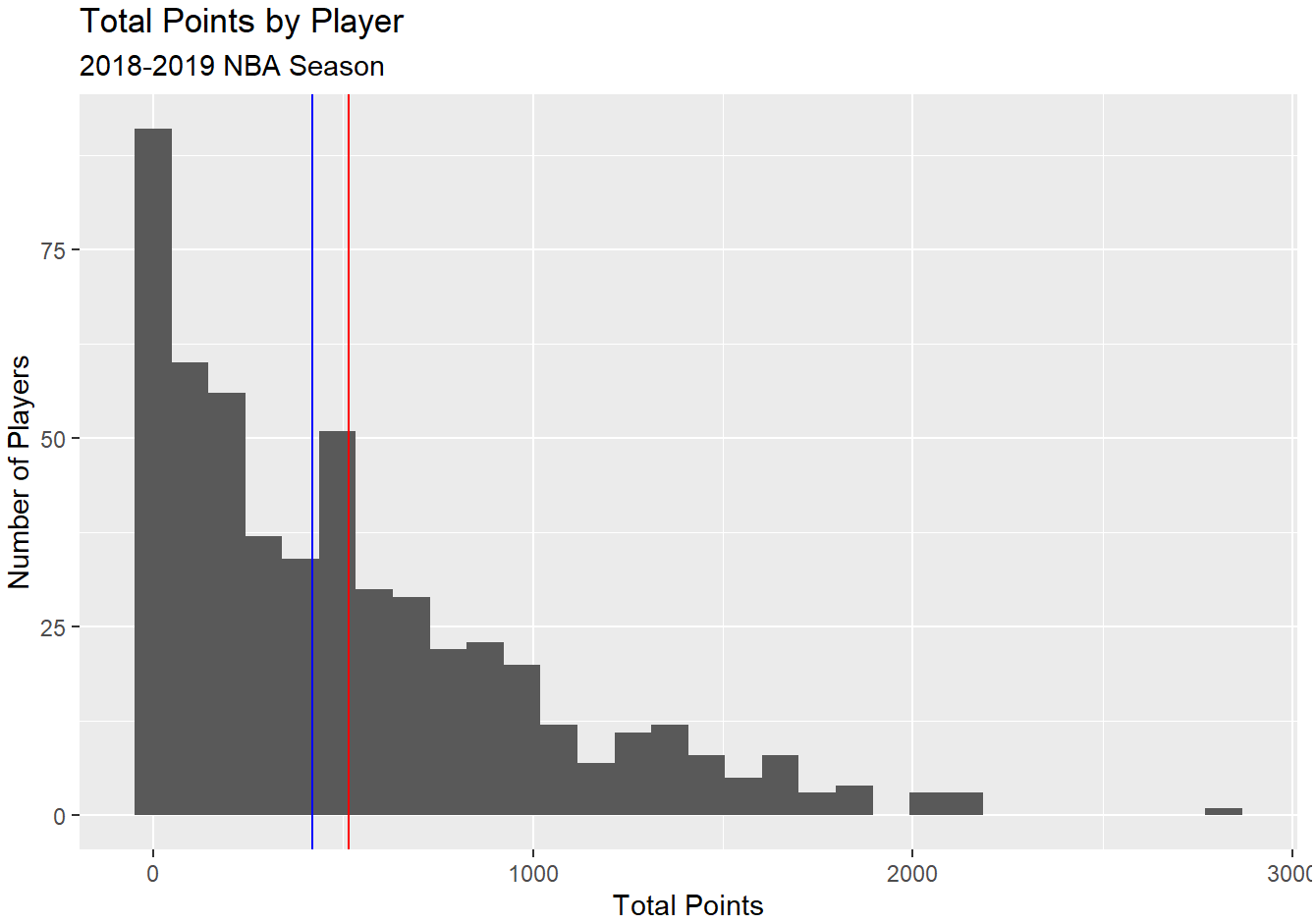
```
nba <- read_rds('https://github.com/jbisbee1/DS1000_S2024/blob/main/data/nba_players_2018.Rds?ra
w=true')
#https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/nba_players
_2018.Rds?raw=true')
```

Question 1 [2 points]

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

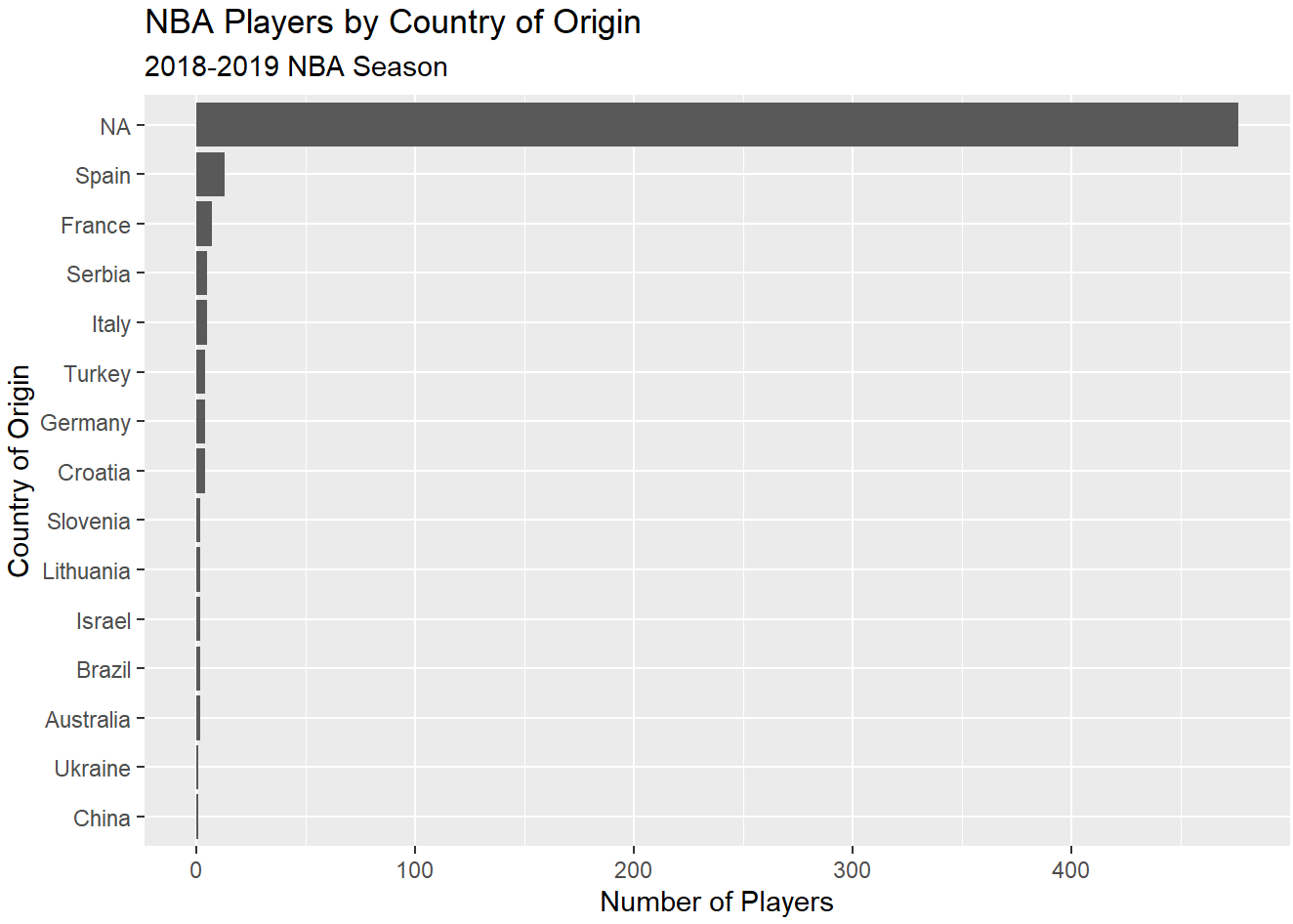


I chose to use a geom_histogram because `pts` is a continuous variable. The mean is larger than the median because the data is right-skewed, reflecting the fact that there are a few players who score many points, and many players who do not.

Question 2 [2 points]

Now examine the `country` variable. Visualize this variable using the appropriate `geom_...`, and justify your reason for choosing it. Tweak the plot to put the country labels on the y-axis, ordered by frequency. Which country are most NBA players from? What is weird about your answer, and what might explain it?

```
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 3 [2 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!

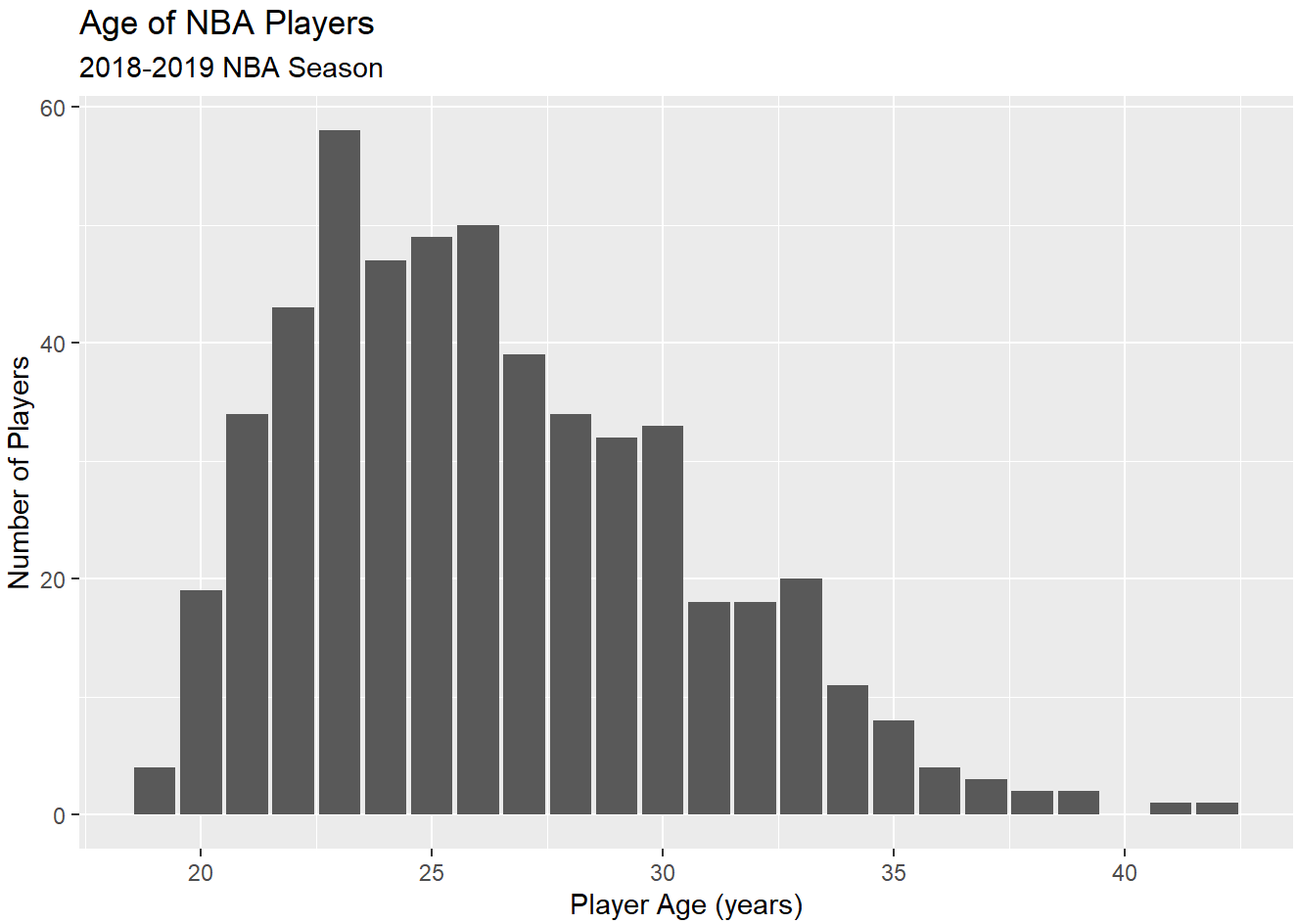
```
glimpse(nba %>% select(agePlayer)) # Look at the variable first
```

```
## 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
```



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 4 [2 points]

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).

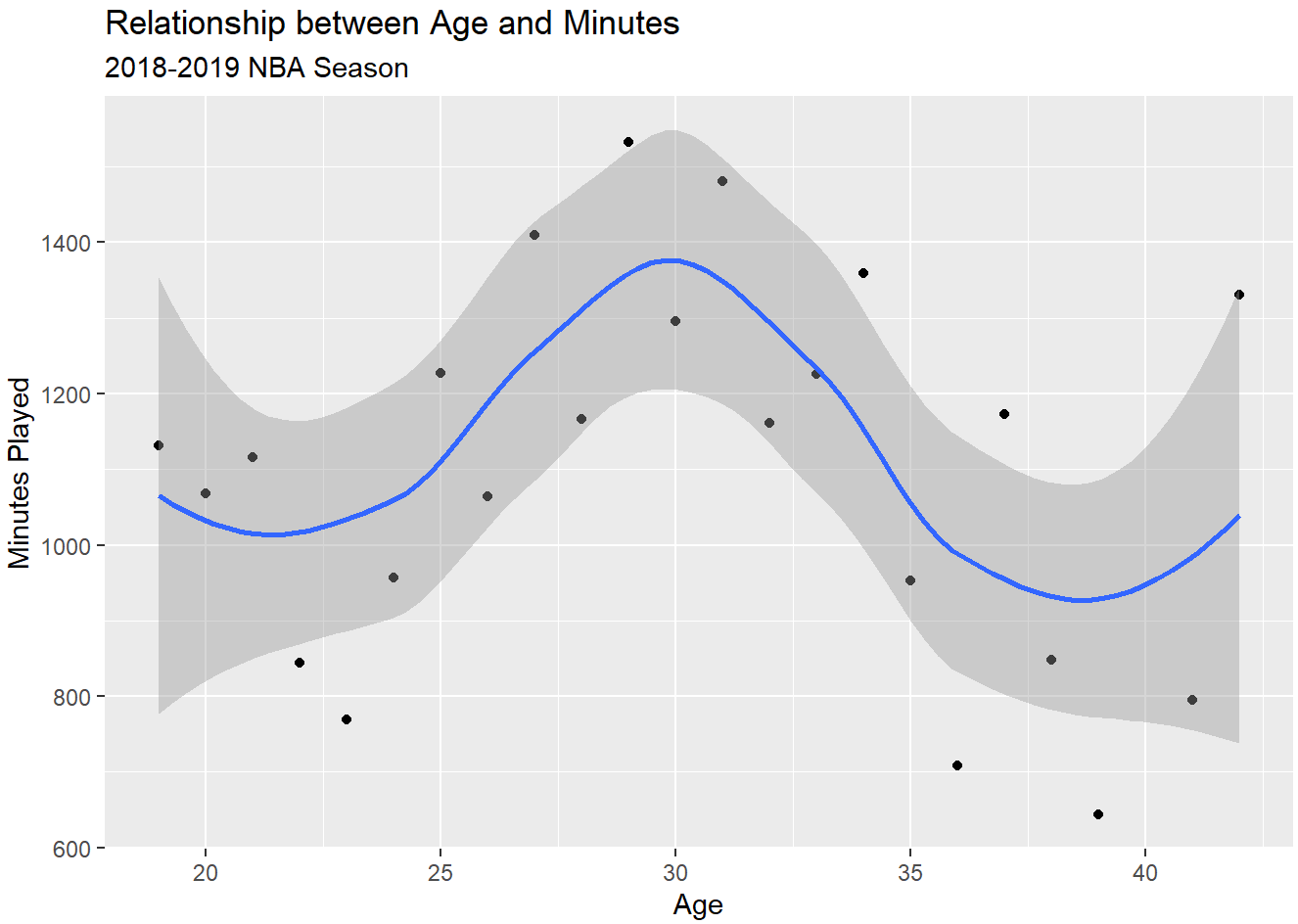
> I think that there is a trade-off between minutes and age for players. I assume that the older the player is, the more tired they get, meaning that older players should play fewer minutes than younger players. However, I also assume that young players are less experienced, which means that younger 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.

Extra Credit [2 points]

Generate a multivariate visualization that provides an answer to this question. To do this, first calculate the average minutes played by player age, then plot the resulting data. Does the data support your hypothesis?

```
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'
```



The data exactly supports my hypothesis. The youngest and oldest players get the fewest minutes, while players in their mid-to-late 20s get the most playing time.