Problem Set 2

Data Wrangling

[YOUR NAME]

Due Date: 2024-07-09

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 `[YOUR NAME]_ps2.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[YOUR NAME]_ps2.Rmd` file. Then change the `author: [Your Name]` on line 2 to your name.

We will be using two different files. First is the `MI2020_ExitPoll.rds` data from the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/data/MI2020_ExitPoll.rds). Second is the `nba_players_2018.Rds` data, which is also on the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/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 10 total points, plus **two** extra credit questions, each worth **two** 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, email the knitted output to Eun Ji Kim (kej990804@snu.ac.kr (mailto:kej990804@snu.ac.kr)) **as a PDF** by the start of class on Tuesday, July 9th. If you need help converting to a PDF, see this tutorial (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Psets/ISP_pset_0_HELPER.pdf).

Good luck!

*Copy the link to ChatGPT you used here: _____

Question 0

Require `tidyverse` and load the `MI2020_ExitPoll.Rds` data to an object called `MI_raw`. ALSO load a new package called `labelled`, which will allow us to read the labels for our variables. Remember, if you don't have this package yet, you need to use `install.packages("labelled")` in the `Console` window.

```
require(tidyverse)
```

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

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ 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 t
o become errors
```

```
require(labelled)
```

```
## Loading required package: labelled
```

```
MI_raw <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/MI20
20_ExitPoll.rds")
```

Question 1 [1 point]

How many units of observation are there in the raw dataset? How many variables are there?

> There are 1,231 observations in the raw dataset, and 63 variables. (You can see this by looking in the environment, or by just looking at the data.)

Now create a new object called `MI_clean` that contains only the following variables:

- AGE10
- SEX
- PARTYID
- EDUC18
- PRSMI20
- QLT20
- LGBT
- BRNAGAIN
- LATINOS
- QRACEAI
- WEIGHT

```
MI_clean <- MI_raw %>%
  select(AGE10,SEX,PARTYID,EDUC18,PRSMI20,QLT20,LGBT,BRNAGAIN,LATINOS,QRACEAI,WEIGHT)
```

How many units of observation are there in the new dataset? How many variables are there?

There are still 1,231 observations, but now we only have 11 variables.

Question 2 [1 point]

How many **unit non-responses** *are there in the* `PRSMI20` *variable? What is the numeric code for unit non-response in this data? Remember, unit non-response refers to people who were asked a question but refused to answer.*

```
MI_clean %>%
  count(PRSMI20)
```

```
## # A tibble: 6 × 2
##   PRSMI20                                      n
##   <dbl+lbl>                                <int>
## 1 0 (NA) [Will/Did not vote for president]     6
## 2 1 [Joe Biden, the Democrat]                723
## 3 2 [Donald Trump, the Republican]           459
## 4 7 [Undecided/Don't know]                     4
## 5 8 [Refused]                                 14
## 6 9 [Another candidate]                       25
```

There are 14 people who refused to tell us who they voted for. The numeric code for unit non-response is 8 in this data.

Question 3 [1 point]

Let's create a new variable called `preschoice` *that converts* `PRSMI20` *to a simpler categorical variable that indicates whether the respondent voted for Trump, Biden, or neither. To do this, use* `mutate()` *and* `ifelse()` *to replace the numeric values of* `PRSMI20` *with their text labels. Remember to replace the unit non-response code with* `NA`*.*

```
MI_clean <- MI_clean %>%
  mutate(preschoice = ifelse(PRSMI20 == 1, # Test for Biden
                             "Biden",
                             ifelse(PRSMI20 == 2, # Test for Trump
                                    "Trump",
                                    ifelse(PRSMI20 == 8, # Test for unit non-response
                                           NA,
                                           "Neither"))))

# Use this code to confirm you did it correctly
MI_clean %>%
  count(PRSMI20,preschoice)
```

```
## # A tibble: 6 × 3
##   PRSMI20                                    preschoice     n
##   <dbl+lbl>                                  <chr>      <int>
## 1 0 (NA) [Will/Did not vote for president] Neither        6
## 2 1 [Joe Biden, the Democrat]              Biden        723
## 3 2 [Donald Trump, the Republican]         Trump        459
## 4 7 [Undecided/Don't know]                 Neither        4
## 5 8 [Refused]                              <NA>          14
## 6 9 [Another candidate]                    Neither       25
```

Now `count()` the number of respondents who reported voting for each candidate using the `preschoice` variable. How many respondents didn't vote for either Biden or Trump in 2020?

```
MI_clean %>%
  count(preschoice)
```

```
## # A tibble: 4 × 2
##   preschoice     n
##   <chr>      <int>
## 1 Biden        723
## 2 Neither       35
## 3 Trump        459
## 4 <NA>          14
```

35 respondents didn't vote for either Biden or Trump in 2020. This is a combination of those who voted for another candidate, did not vote at all, or didn't know who they would vote for at the time of the survey.

Question 4 [1 point]

Now do the same for the `SEX` variable and the `LGBT` variable. Name the the text version `Gender` for `SEX` and `Lgbt_clean` for `LGBT`. Remember to replace the unit non-response code with `NA`, and be aware that different variables use different codes!

```
# Follow these steps for each variable
# 1. count() to see the unit non-response code (if it exists)
MI_clean %>%
  count(SEX)
```

```
## # A tibble: 2 × 2
##   SEX            n
##   <dbl+lbl>  <int>
## 1 1 [Male]     579
## 2 2 [Female]   652
```

```
# 2. Create the cleaned variable
MI_clean <- MI_clean %>%
  mutate(Gender = ifelse(SEX == 1, # Test for value 1
                         "Male",
                         "Female"))

# 1. count() to see the unit non-response code (if it exists)
MI_clean %>%
  count(LGBT)
```

```
## # A tibble: 4 × 2
##   LGBT                                      n
##   <dbl+lbl>                             <int>
## 1  1 [Yes]                                 23
## 2  2 [No]                                 570
## 3  9 [[DON'T READ] Don’t know/Refused]     23
## 4 NA                                      615
```

```
# 2. Create the cleaned variable
MI_clean <- MI_clean %>%
  mutate(Lgbt_clean = ifelse(LGBT == 1,
                             'LGBT',
                             ifelse(LGBT == 9,
                                    NA,
                                    'Not LGBT')))
```

Question 5 [1 point]

What proportion of women supported Trump? What proportion of LGBTQ-identifying respondents supported Trump?

```
# Apply the following for each variable
# Method 1: more lines of code
MI_clean %>%
  count(Gender,preschoice) %>%
  group_by(Gender) %>%
  mutate(totn = sum(n)) %>%
  mutate(proportion = n / totn)
```

```
## # A tibble: 8 × 5
## # Groups:   Gender [2]
##   Gender preschoice     n  totn proportion
##   <chr>  <chr>      <int> <int>      <dbl>
## 1 Female Biden        419   652     0.643
## 2 Female Neither       14   652     0.0215
## 3 Female Trump        212   652     0.325
## 4 Female <NA>           7   652     0.0107
## 5 Male   Biden        304   579     0.525
## 6 Male   Neither       21   579     0.0363
## 7 Male   Trump        247   579     0.427
## 8 Male   <NA>           7   579     0.0121
```

```
# Method 2: prop.table()
MI_clean %>%
  count(Gender,preschoice) %>%
  group_by(Gender) %>%
  mutate(proportion = prop.table(n))
```

```
## # A tibble: 8 × 4
## # Groups:   Gender [2]
##   Gender preschoice     n proportion
##   <chr>  <chr>      <int>      <dbl>
## 1 Female Biden        419     0.643
## 2 Female Neither       14     0.0215
## 3 Female Trump        212     0.325
## 4 Female <NA>           7     0.0107
## 5 Male   Biden        304     0.525
## 6 Male   Neither       21     0.0363
## 7 Male   Trump        247     0.427
## 8 Male   <NA>           7     0.0121
```

```
# LGBT
MI_clean %>%
  count(Lgbt_clean,preschoice) %>%
  group_by(Lgbt_clean) %>%
  mutate(proportion = prop.table(n))
```

```
## # A tibble: 11 × 4
## # Groups:   Lgbt_clean [3]
##    Lgbt_clean preschoice     n proportion
##    <chr>      <chr>      <int>      <dbl>
##  1 LGBT       Biden         14    0.609
##  2 LGBT       Neither        2    0.0870
##  3 LGBT       Trump          7    0.304
##  4 Not LGBT   Biden        337    0.591
##  5 Not LGBT   Neither       11    0.0193
##  6 Not LGBT   Trump        218    0.382
##  7 Not LGBT   <NA>           4    0.00702
##  8 <NA>       Biden        372    0.583
##  9 <NA>       Neither       22    0.0345
## 10 <NA>       Trump        234    0.367
## 11 <NA>       <NA>          10    0.0157
```

32.5% of women supported Trump. 30.4% of LGBT-identifying respondents supported Trump.

Extra Credit 1 [2 points]

Calculate the proportion of women who supported Trump by age-group to determine which age-group had the highest Trump support among women. You will need to clean the AGE10 variable before completing this problem, just like we did with the PRSMI20, SEX, and LGBT variables. Call the new variable "Age". HINT: to make your life easier (and not write a 10-level nested ifelse() function), try asking ChatGPT for help with this prompt: "I have a labelled variable in R that I want to convert to text. How can I do this?"

```
MI_clean %>%
  count(AGE10)
```

```
## # A tibble: 11 × 2
##    AGE10                          n
##    <dbl+lbl>                  <int>
##  1  1 [18 and 24,]               33
##  2  2 [25 and 29,]               28
##  3  3 [30 and 34,]               42
##  4  4 [35 and 39,]               46
##  5  5 [40 and 44,]               78
##  6  6 [45 and 49,]               83
##  7  7 [50 and 59,]              274
##  8  8 [60 and 64,]              143
##  9  9 [65 and 74,]              290
## 10 10 [75 or over?]             199
## 11 99 [[DON'T READ] Refused]    15
```

```
require(labelled)
MI_clean <- MI_clean %>%
  mutate(Age = ifelse(AGE10 == 99,
                      NA,
                      as.character(to_factor(AGE10))))

MI_clean %>%
  count(AGE10,Age)
```

```
## # A tibble: 11 × 3
##    AGE10                      Age             n
##    <dbl+lbl>                  <chr>       <int>
##  1  1 [18 and 24,]            18 and 24,     33
##  2  2 [25 and 29,]            25 and 29,     28
##  3  3 [30 and 34,]            30 and 34,     42
##  4  4 [35 and 39,]            35 and 39,     46
##  5  5 [40 and 44,]            40 and 44,     78
##  6  6 [45 and 49,]            45 and 49,     83
##  7  7 [50 and 59,]            50 and 59,    274
##  8  8 [60 and 64,]            60 and 64,    143
##  9  9 [65 and 74,]            65 and 74,    290
## 10 10 [75 or over?]           75 or over?   199
## 11 99 [[DON'T READ] Refused] <NA>           15
```

```
MI_clean %>%
  count(Age,Gender,preschoice) %>%
  group_by(Age,Gender) %>%
  mutate(proportion = prop.table(n)) %>%
  filter(Gender == 'Female',
         preschoice == 'Trump')
```

```
## # A tibble: 11 × 5
## # Groups:   Age, Gender [11]
##    Age          Gender preschoice     n proportion
##    <chr>        <chr>  <chr>      <int>      <dbl>
##  1 18 and 24,   Female Trump          1     0.0588
##  2 25 and 29,   Female Trump          1     0.0714
##  3 30 and 34,   Female Trump          4     0.267
##  4 35 and 39,   Female Trump          5     0.294
##  5 40 and 44,   Female Trump         15     0.469
##  6 45 and 49,   Female Trump         16     0.372
##  7 50 and 59,   Female Trump         54     0.36
##  8 60 and 64,   Female Trump         20     0.256
##  9 65 and 74,   Female Trump         51     0.304
## 10 75 or over? Female Trump         41     0.369
## 11 <NA>        Female Trump          4     0.571
```

> The age-group with the highest support for Trump among women is the 40 to 44 year old group, with 47% supporting Trump. (Interestingly, among women who refused to give their age, 57% supported Trump.)

Question 6 [1 point]

Now let's load a different dataset to practice univariate visualization. Open `nba_players_2018.Rds` from the github page and save it to a new object called `nba`. This dataset contains information on basketball players in the NBA from the 2018-2019 season. The codebook for it can be found in homework 3, which is also on github (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Homeworks/ISP_hw_3.pdf).

```
nba <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/nba_players_2018.Rds") # Insert link here
```

We are interested in the relationship between the player's age (`agePlayer`) and the amount of points they score (`pts`). Please answer the following research question and provide a theory supporting your answer: "Do older NBA players score more points than younger players?"

> I think that older NBA players do not score as many points as younger players because they are older and therefore more likely to get tired. Younger players are probably in better physical condition than older players, allowing them to play longer and score more points.

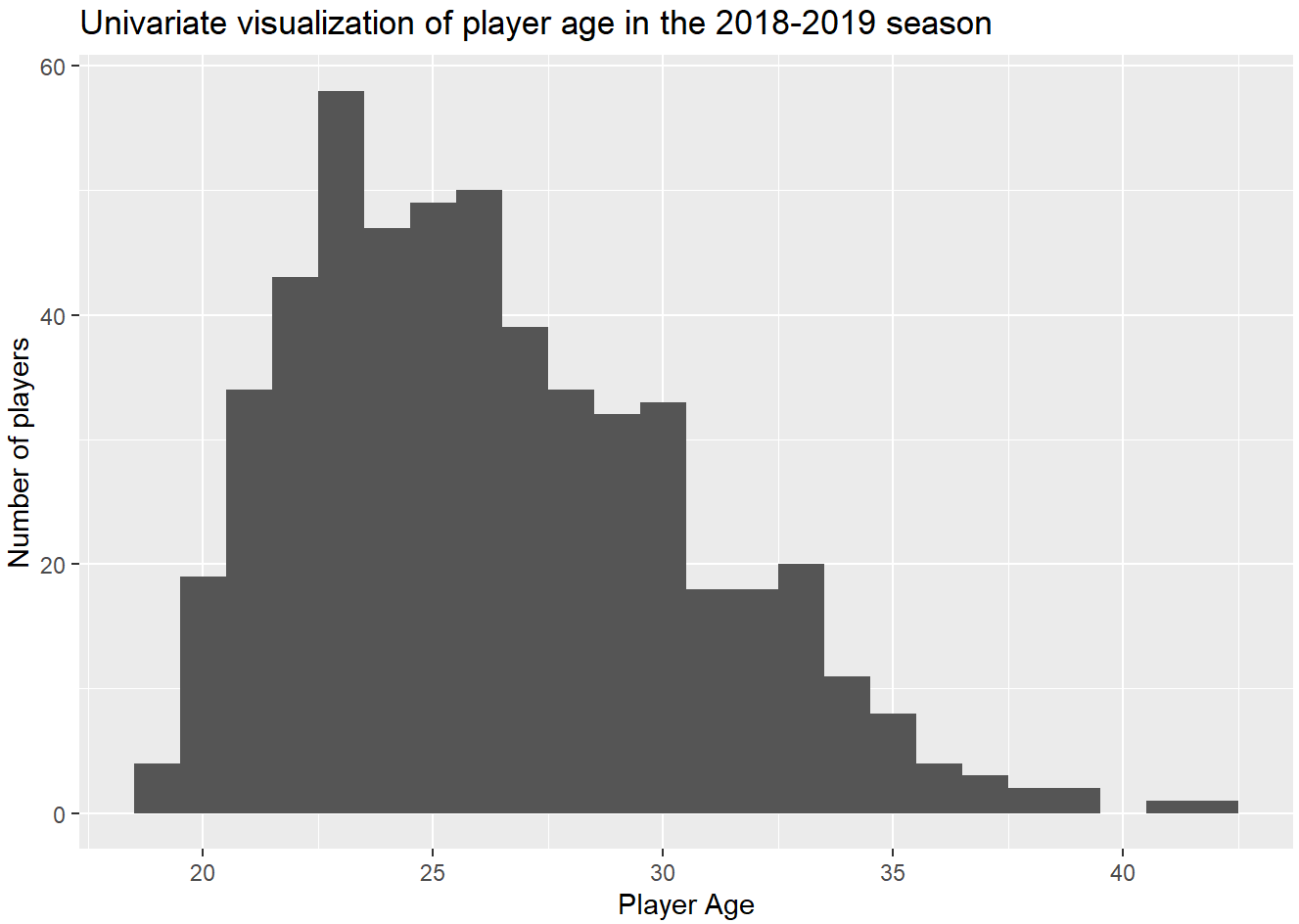
Question 7 [2 points]

Based on your answer above, what is the outcome / dependent / Y variable and what is the explanatory / independent / X variable? Why?

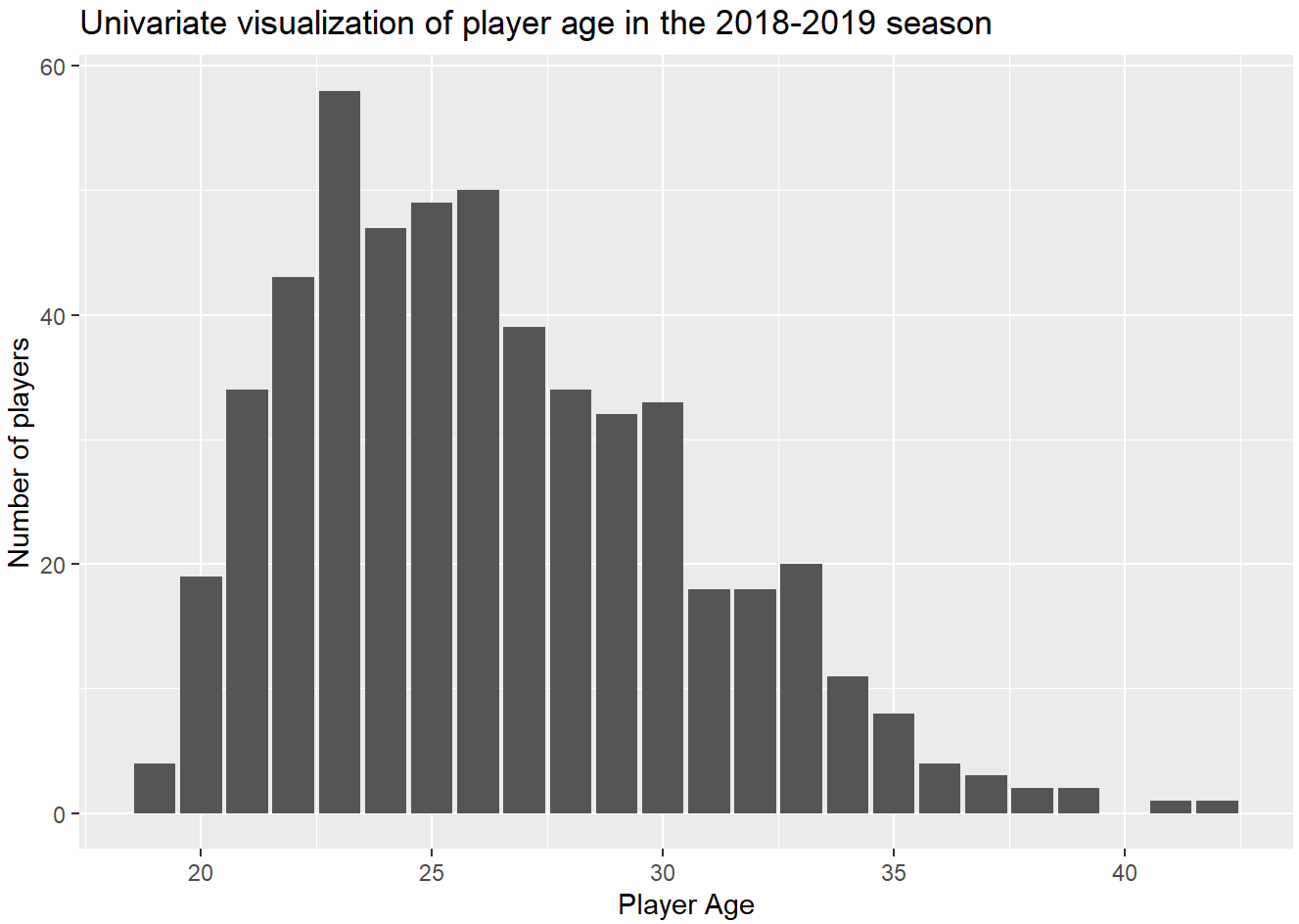
> Based on my answer, the outcome variable is points and the explanatory variable is player age.

Create a univariate visualization of both the X and Y variables. Choose the best `geom_...()` based on the variable type, and make sure to label your plots!

```
# X variable
nba %>%
  ggplot(aes(x = agePlayer)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Player Age",
       y = "Number of players",
       title = "Univariate visualization of player age in the 2018-2019 season")
```

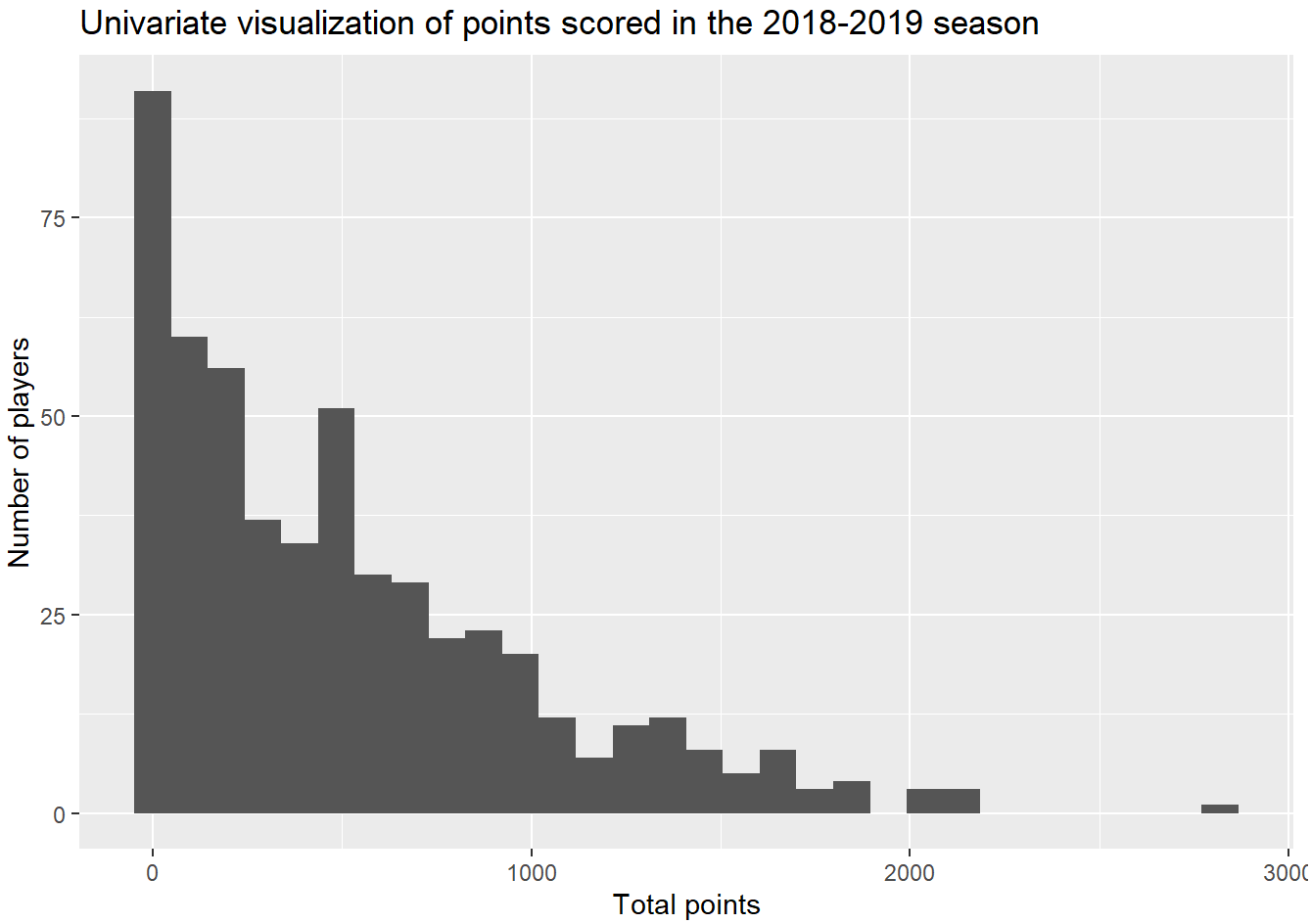



```
# Could also do with geom_bar(), since there are so few values
nba %>%
  ggplot(aes(x = agePlayer)) +
  geom_bar() +
  labs(x = "Player Age",
       y = "Number of players",
       title = "Univariate visualization of player age in the 2018-2019 season")
```




```
# Y variable
nba %>%
  ggplot(aes(x = pts)) +
  geom_histogram() +
  labs(x = "Total points",
       y = "Number of players",
       title = "Univariate visualization of points scored in the 2018-2019 season")
```

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

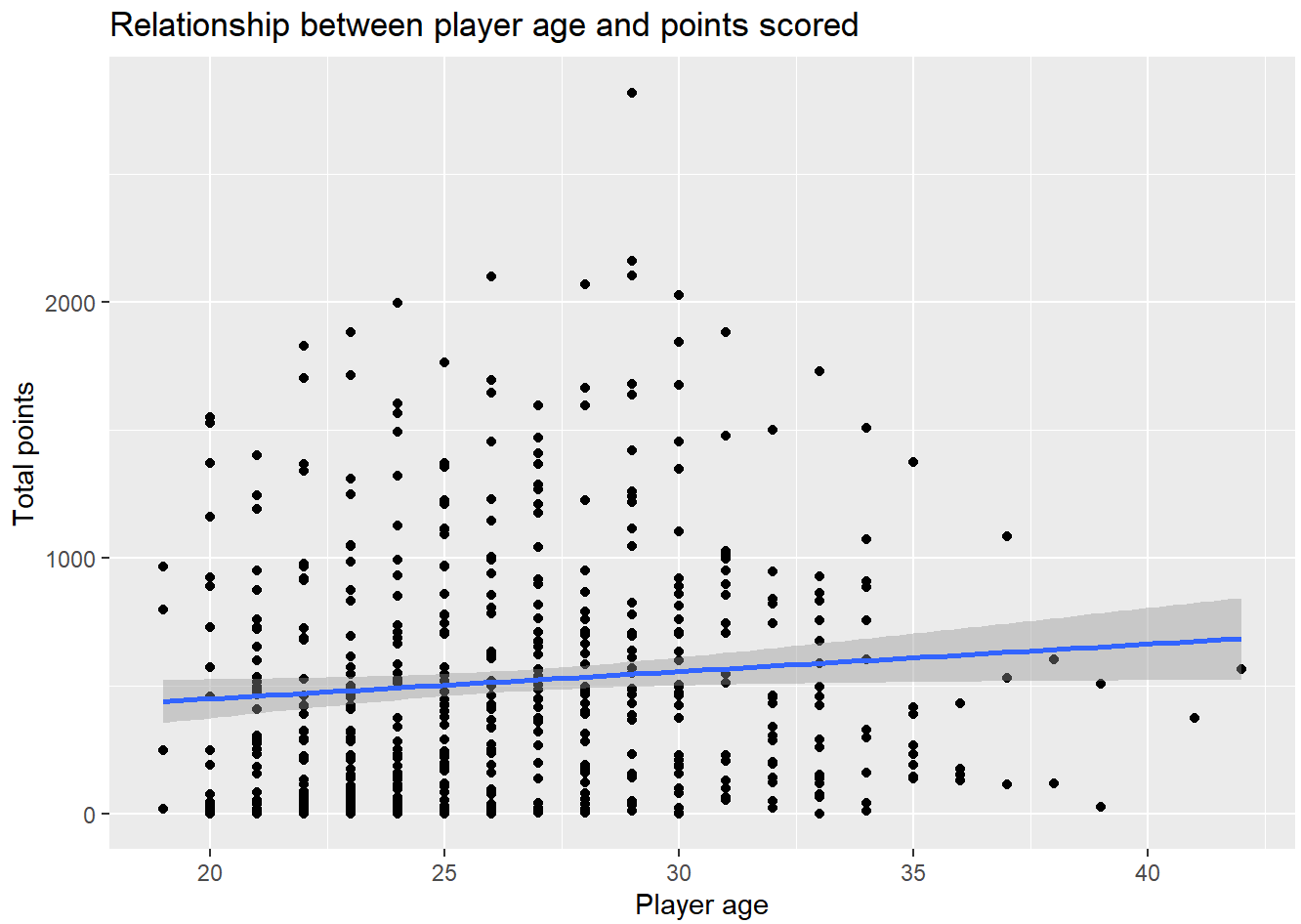


Question 8 [2 points]

Now analyze the data by creating a multivariate visualization that shows the relationship between age and points.

```
nba %>%
  ggplot(aes(x = agePlayer,
             y = pts)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(x = "Player age",
       y = "Total points",
       title = "Relationship between player age and points scored")
```

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



Based on your analysis, does the data support or reject your hypothesis from Question 6?

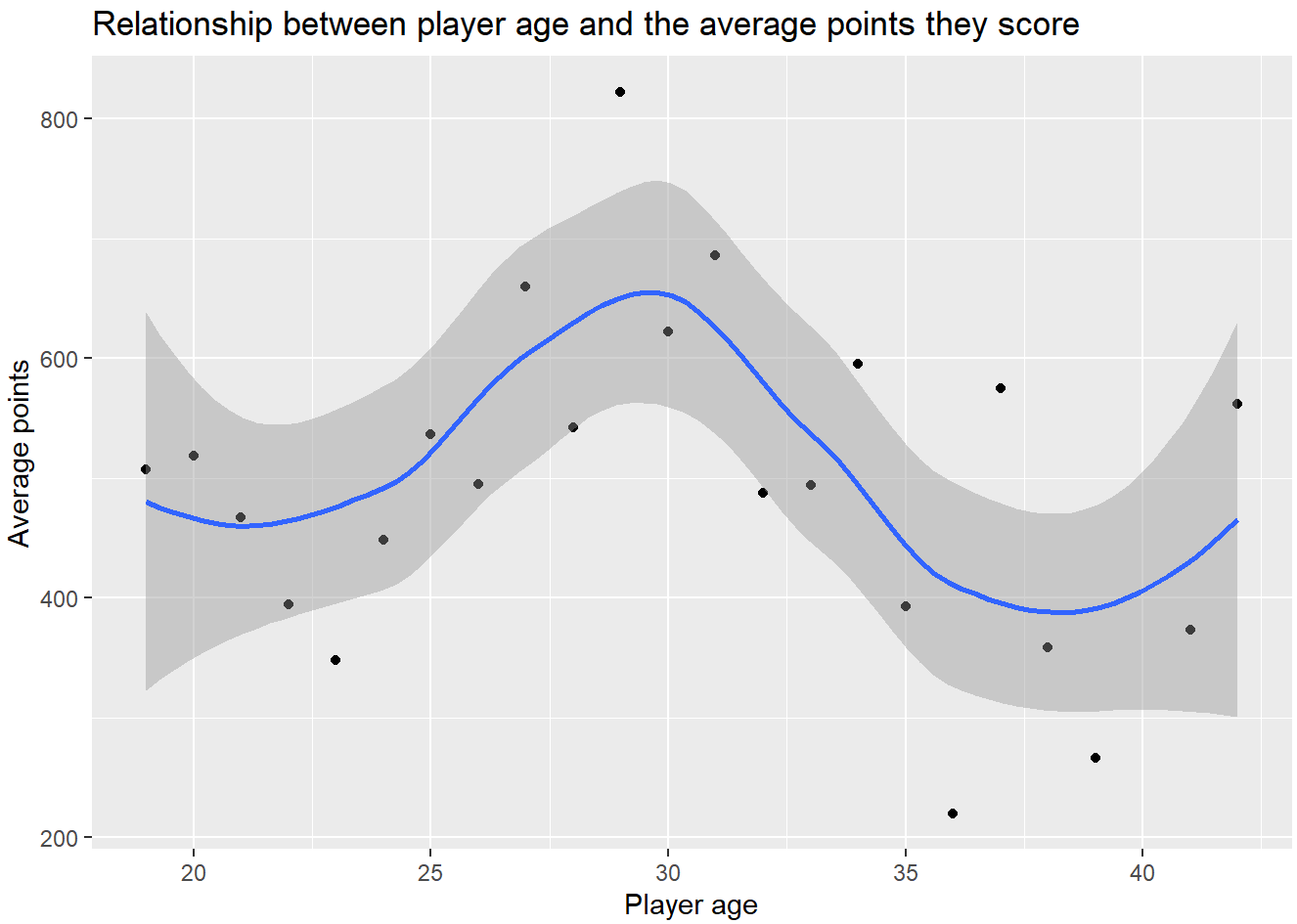
> It is difficult to say. The line of best fit is positive, suggesting that older players are able to score more points. However the relationship is very noisy, with many players located very far from the line of best fit.

Extra Credit 2 [2 points]

Let's look for evidence of a "curvelinear" relationship between player age and points scored. To do so, first calculate the average points scored by age. Then plot this relationship using a multivariate visualization. Add a line of best fit with `geom_smooth()` *but DON'T use* `method = "lm"` *. What do you conclude? Why?*

```
nba %>%
  group_by(agePlayer) %>%
  summarise(avg_points = mean(pts,na.rm=T)) %>%
  ggplot(aes(x = agePlayer,
             y = avg_points)) +
  geom_point() +
  geom_smooth() +
  labs(x = 'Player age',
       y = 'Average points',
       title = 'Relationship between player age and the average points they score')
```

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



There is evidence of a “curvelinear” relationship between player age and points scored. The plot shows that the most points are scored by neither very young nor very old players. Instead, players in their late 20s or early 30s score the most points. This might mean there are two theories: first, the oldest players are slower and have less stamina than younger players, as per my original theory. Second, the youngest players might be inexperienced, making them also score less points.