Univariate Analysis

Description and Visualization

Prof. Bisbee

Vanderbilt University

Slides Updated: 2024-01-08

Agenda

1. Definitions and scope
2. Opening and defining the data
3. Variable classes
4. Univariate description

Definition

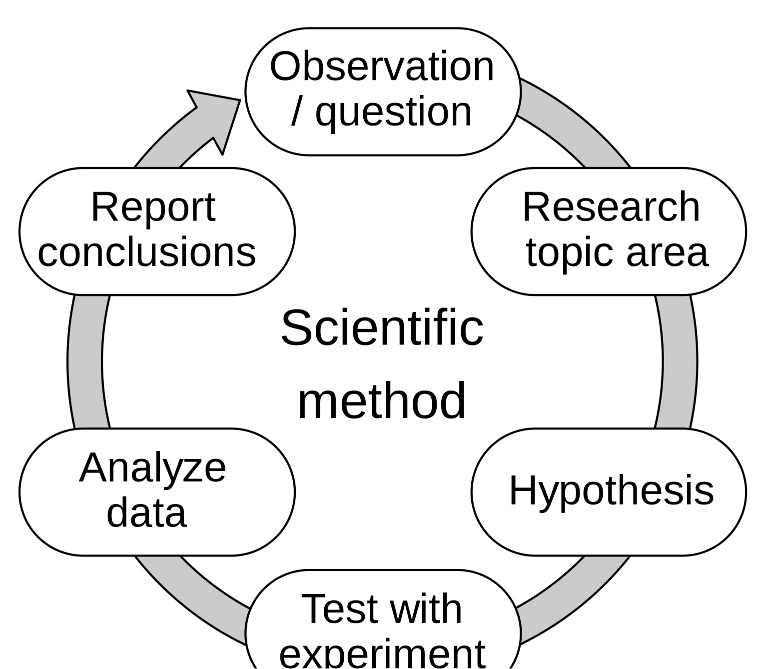
- Uni + variate
  - One + variable
  - Analysis of one variable

Scope

- How to analyze a single variable?
- How to think scientifically?
  - Typically, scientific theories concern more than one variable
  - I.e., education + wages; gender + voting
  - What might be a theory about education in isolation?
- Is there no point to univariate analysis?

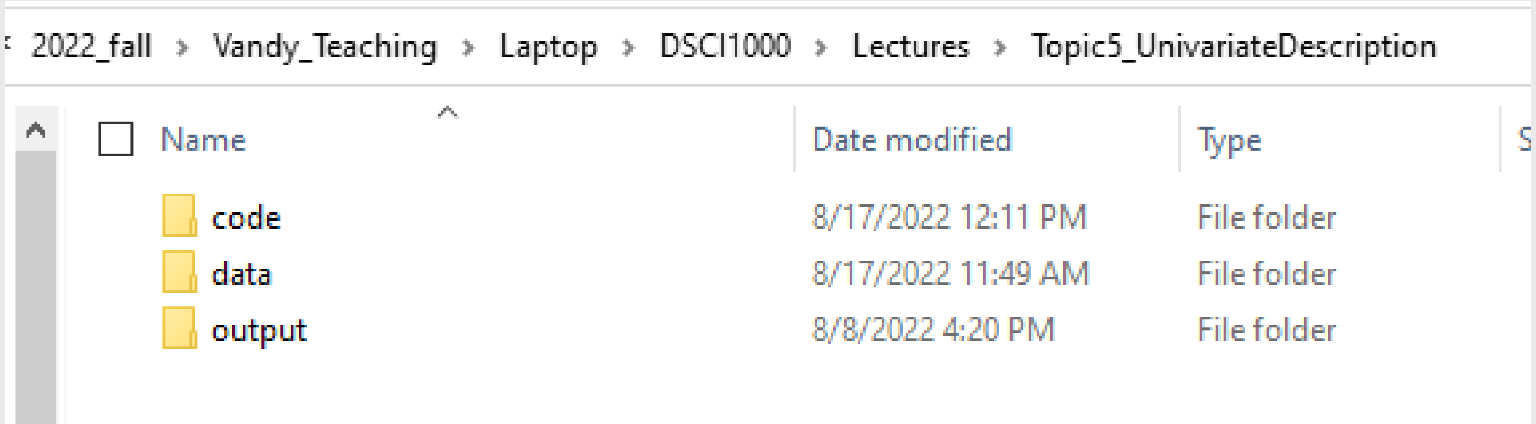
Univariate Analysis is **ESSENTIAL**

- Both from a practical data perspective...
  - Informs how we "wrangle" the data
- ...and from a scientific theory perspective
  - Generates hypotheses



Set-up and Load Data

- As always, create your topic folder first



- Open `R` via `RStudio` and `require(tidyverse)`

```r
require(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
```

Introducing the data

- Data on every NBA player active in the 2018-2019 season

Name	Definition
namePlayer	Player name
idPlayer	Unique player id
slugSeason	Season start and end
numberPlayerSeason	Which season for this player
isRookie	Rookie season, true or false
slugTeam	Team short name
idTeam	Unique team id
gp	Games Played
…	…

- `ds1000_hw_5.pdf` has the full codebook

Thinking like a scientist

- What questions do we have? What hypotheses might we want answered?
- Overwhelming? Let's start simpler
- Total points (`pts`)
  - What does this measure?
  - What kind of variable is it?

```
glimpse(nba %>% select(pts))
```

```
## Rows: 530
## Columns: 1
## $ pts <dbl> 1727, 17, 1108, 165, 729, 37, 211, 32, 108, 7,…
```

Thinking like a scientist

- How can we analyze a single variable?
- Want to **summarize** it somehow
  - For example, look at the `mean()` and the `median()`

```
nba %>%
  summarise(mean_pts = mean(pts,na.rm=T),
            med_pts = median(pts,na.rm=T))
```

```
## # A tibble: 1 × 2
##   mean_pts med_pts
##      <dbl>   <dbl>
## 1     516.     419
```

Thinking like a scientist

- Or we could summarise the overall distribution with `summary()`

```
summary(nba$pts)
```

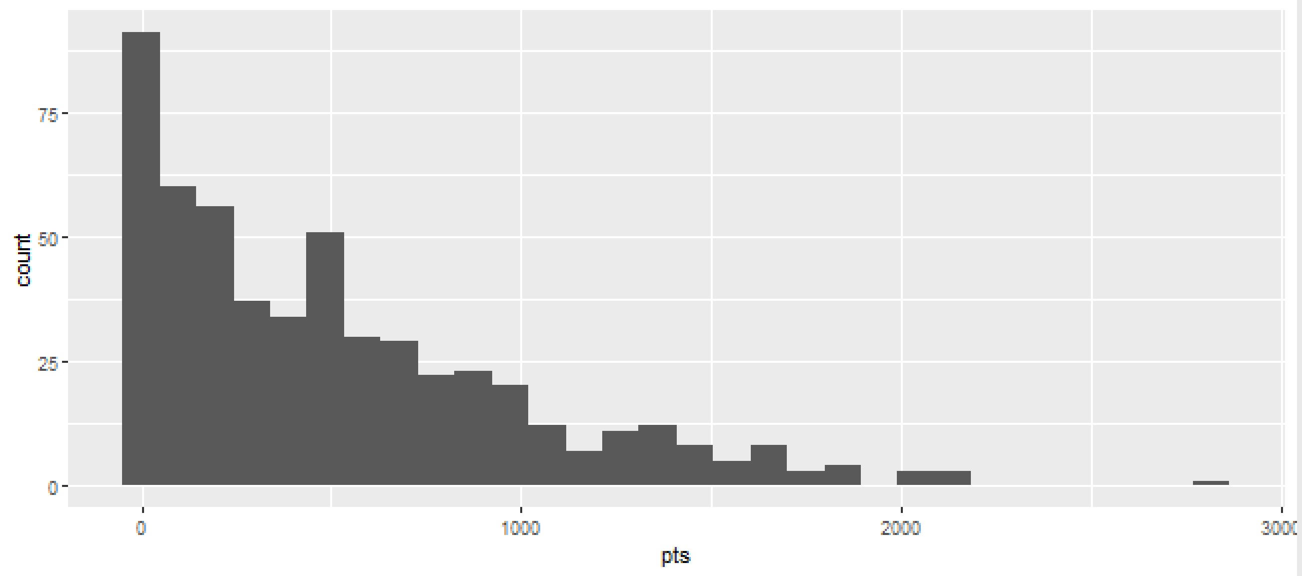
```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   115.0   419.0   516.2   759.5  2818.0
```

- In English:
  - There is at least one player who didn't score at all (`Min.`)
  - At least one player scored 2,818 points (`Max.`)
  - 25% of players scored less than 115 points (`1st Qu.`)
  - 25% of players scored more than ???
- What does a decimal mean here?

Visualization

- We could try and remember all these statements
- Or we could just visualize the data

```
nba %>%
  ggplot(aes(x = pts)) +
  geom_histogram()
```

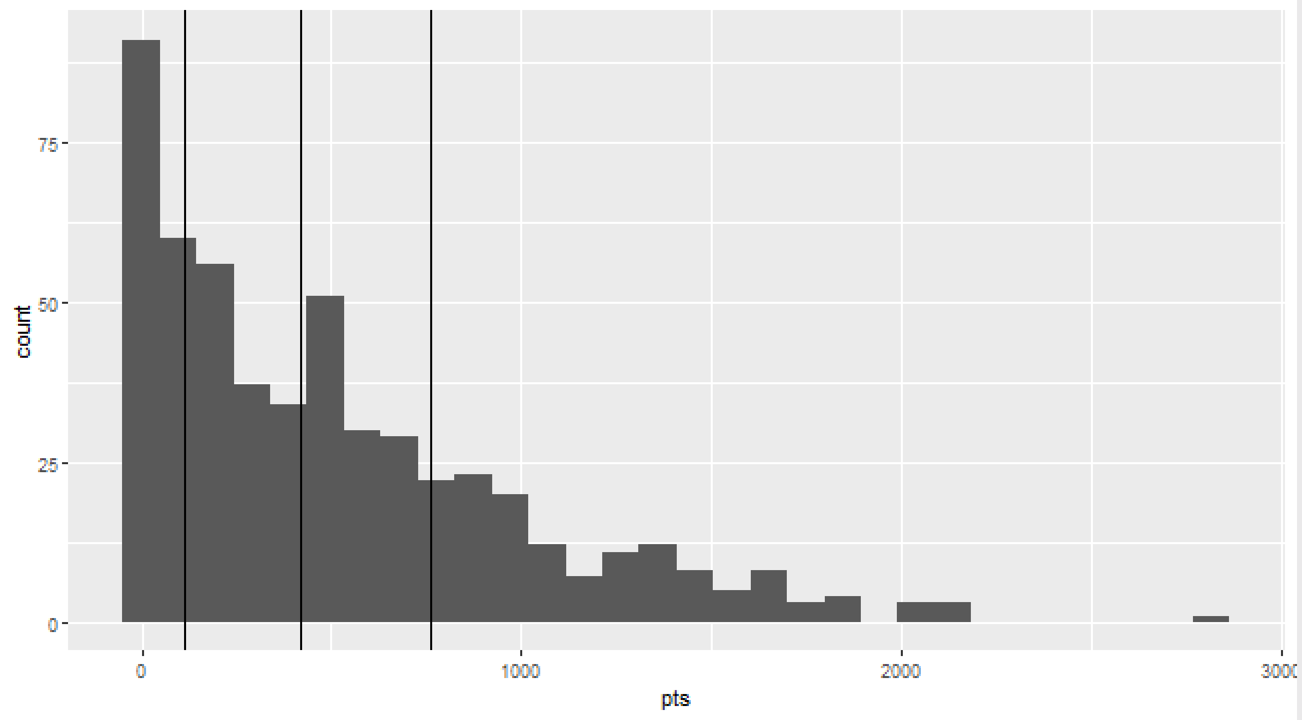


Visualization

- Plotting the histogram reveals some things!
  - There are **MANY** players who didn't score any points
  - There are **VERY FEW** who scored many
- We can combine the substantive interpretation with the visualization by plotting vertical lines for the quartiles
  - A "quartile" is 25% increments
  - A "decile" is 10% increments, a "quantile" is 20% increments
  - A "percentile" is 1% increments

Visualization

```
nba %>%
  ggplot(aes(x = pts)) +
  geom_histogram() +
  geom_vline(xintercept = quantile(nba$pts,c(.25,.5,.75)))
```



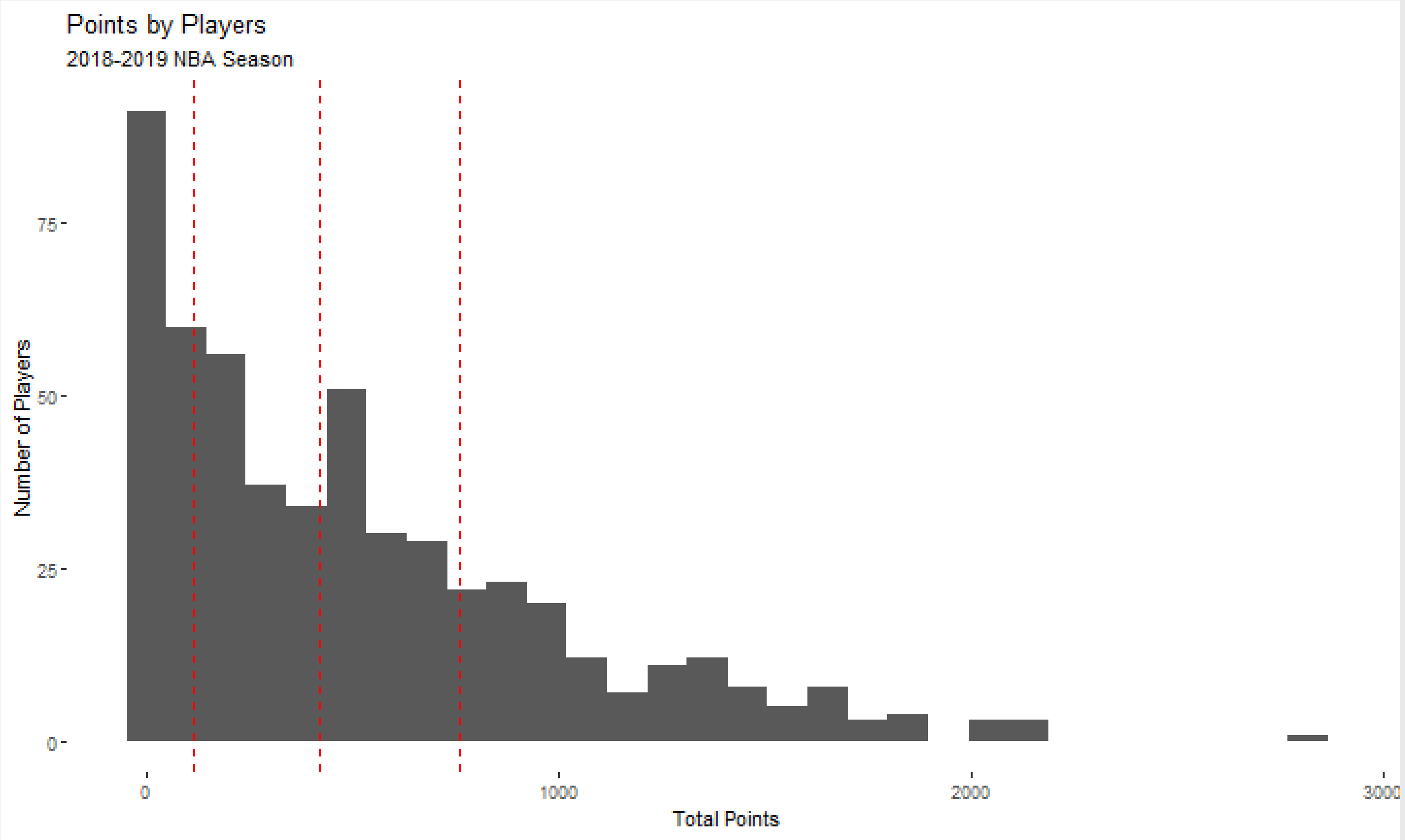
Visualization

- We can save and update plots using the object assignment operator `<-`

```
p <- nba %>%
  ggplot(aes(x = pts)) +
  geom_histogram()

p <- p + geom_vline(xintercept = quantile(nba$pts,c(.25,.5,.75)),linetype = 'dashed',color =
'red')

p <- p + xlab('Total Points') + ylab('Number of Players')

p <- p + theme(panel.background = element_rect(fill = 'white'))

p <- p + labs(title = 'Points by Players',subtitle = '2018-2019 NBA Season')
```

Visualization

```
p
```



Visualization informs science

- Looking at the data can help generate research questions, theories, and hypotheses
  - **Question:** Why do some players not score any points?
  - **Theory:** Players need minutes to score points.
  - **Hypothesis:** The number of points a player scores should be positively correlated with their minutes.

Univariate Description

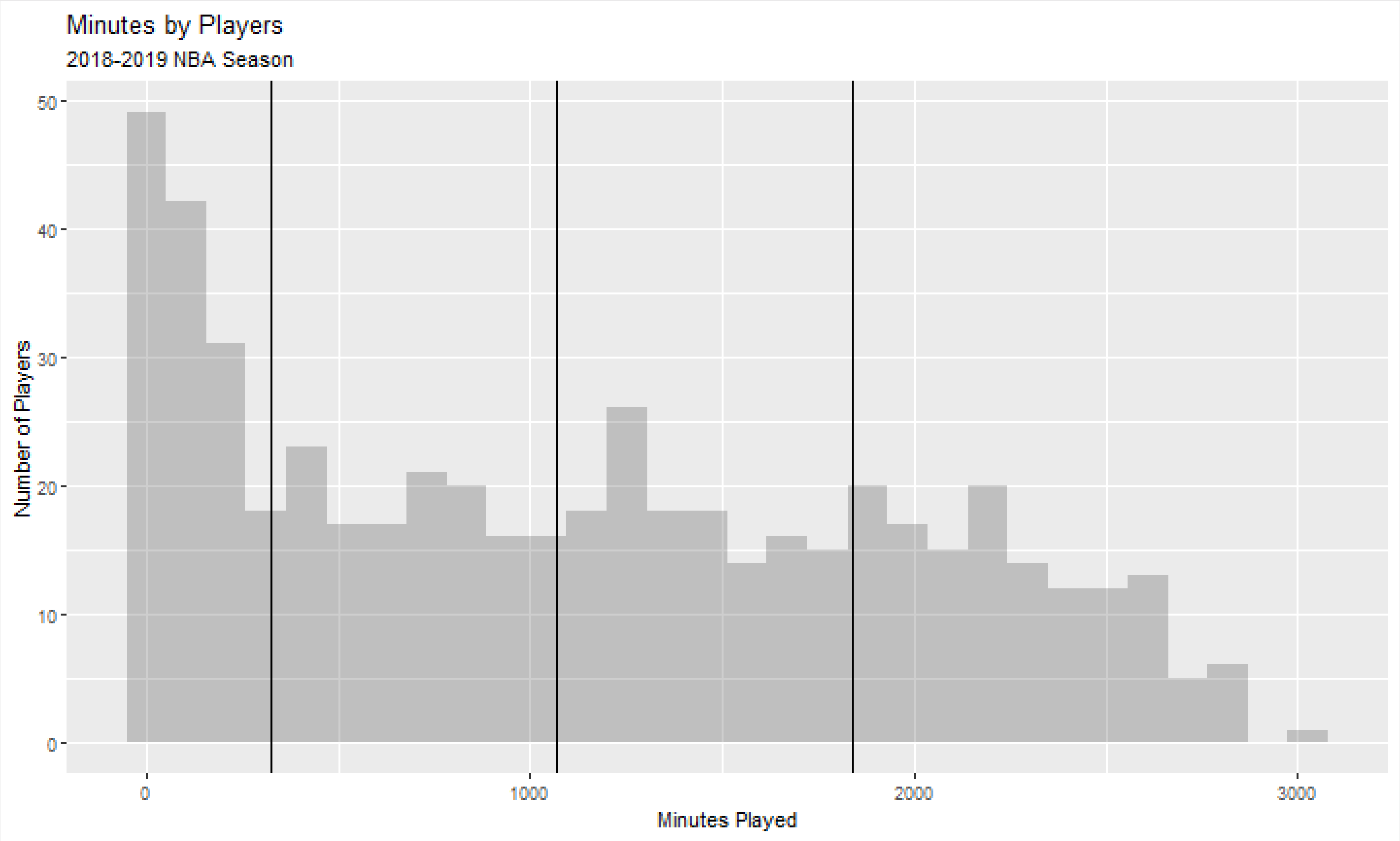
- Testing this hypothesis comes later
- For now, let's also describe the minutes variable

```
summary(nba$minutes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   322.8  1069.0  1121.6  1836.5  3028.0
```

- At minimum, every player played at least 1 minute
- Does the distribution of this variable look similar to the points?

```
nba %>%
  ggplot(aes(x = minutes)) +
  geom_histogram(alpha = .3) +
  geom_vline(xintercept = quantile(nba$minutes,c(.25,.5,.75))) +
  labs(title = 'Minutes by Players',subtitle = '2018-2019 NBA Season',x = 'Minutes Played',y =
'Number of Players')
```



Other Variables

- Thus far, `pts` and `minutes` are both `dbl`

```
glimpse(nba %>% select(pts,minutes))
```

```
## Rows: 530
## Columns: 2
## $ pts     <dbl> 1727, 17, 1108, 165, 729, 37, 211, 32, 108…
## $ minutes <dbl> 2687, 123, 2669, 588, 1913, 120, 416, 194,…
```

- What about other variable types?

Other Variabes

```
glimpse(nba)
```

```
## Rows: 530
## Columns: 37
## $ namePlayer         <chr> "LaMarcus Aldridge", "Quincy Ac…
## $ idPlayer           <dbl> 200746, 203112, 203500, 203518,…
## $ slugSeason         <chr> "2018-19", "2018-19", "2018-19"…
## $ numberPlayerSeason <dbl> 12, 6, 5, 2, 1, 0, 0, 0, 0, 0, …
## $ isRookie           <lgl> FALSE, FALSE, FALSE, FALSE, FAL…
## $ slugTeam           <chr> "SAS", "PHX", "OKC", "OKC", "MI…
## $ idTeam             <dbl> 1610612759, 1610612756, 1610612…
## $ gp                 <dbl> 81, 10, 80, 31, 82, 10, 38, 19,…
## $ gs                 <dbl> 81, 0, 80, 2, 28, 1, 2, 3, 1, 0…
## $ fgm                <dbl> 684, 4, 481, 56, 280, 13, 67, 1…
## $ fga                <dbl> 1319, 18, 809, 157, 486, 39, 17…
## $ pctFG              <dbl> 0.519, 0.222, 0.595, 0.357, 0.5…
## $ fg3m               <dbl> 10, 2, 0, 41, 3, 3, 32, 6, 25, …
## $ fg3a               <dbl> 42, 15, 2, 127, 15, 12, 99, 23,…
## $ pctFG3             <dbl> 0.2380952, 0.1333333, 0.0000000…
## $ pctFT              <dbl> 0.847, 0.700, 0.500, 0.923, 0.7…
## $ fg2m               <dbl> 674, 2, 481, 15, 277, 10, 35, 5…
## $ fg2a               <dbl> 1277, 3, 807, 30, 471, 27, 79, …
```

Categorical Variables

- Already introduced you to `dbl`, `fct`, `chr` and `int`
- Taking a step back: Outside `R`, data science uses "categorical" variables

   1. Mutually exclusive: observations can only be in one category
   2. Exhaustive: every observation is assigned to a category

- For example, `isRookie`

   1. Mutually exclusive: Players are either in their rookie season in 2018-2019, or are not
   2. Exhaustive: these categories define every player in the data

Categorical Variables

- Categorical variables can be divided into the following sub-types
- **Ordered:** There is a sensible order (i.e., education)
    - Should be arranged intuitively (i.e., LTHS, HS Degree, Some coll, etc.)
    - To summarize, calculate the proportions for each category.
    - If there are too many categories, use the "mode"

Categorical Variables

- Categorical variables can be divided into the following sub-types
- **Ordered, Binary:** An ordered categorical variable with just two levels
  - Should be arranged in intuitive order (i.e., is not a rookie / is a rookie)
  - To summarize, just convert to a [0,1] number and take the mean

Categorical Variables

- Categorical variables can be divided into the following sub-types
- **Unordered**: No sensible order of categories (i.e., major degree)
  - Order by most commonly occurring categories
  - As before, use the mode for too many categories

Categorical Variables

- Categorical variables can be divided into the following sub-types
- **Unordered, Binary**: No sensible order and only two levels (i.e., edible)

Categorical Variables

- Categorical variables are meaningfully different from continuous variables
    - Continuous variables are ordered and can theoretically be divided into arbitrarily small measures
    - Technically can be defined as either **interval** or **ratio** variables
    - In practice, we rarely worry about this distinction, but we **DO** care about continuous versus categorical variables

Categorical Variables

- `fct` is a class that is unique to `R`
  - Meant for ordered categorical variables
  - `fct` stores the order and assigns a numeric value + a definition
  - Most of the time, better to store as a `chr` (but not always)

Variables

- `R` may store categorical variables as `chr`, `fct`, `lgl`, `int`, or even `dbl`
- Continuous variables typically stored as `int` or `dbl`
- Up to the data scientist to look at the data and determine
- Simple **process**

  1. Look at a few observations and make a guess about the variable type
  2. Create a plot or table based on that guess
  3. If the result is sensible, proceed. OTW go back to #1.

In Practice

- Let's look at field goals (`fgm`)
- What type of variable should this be?
  - *Technically* not continuous, since it can't be divided into fractions (i.e., what is 35.5 field goals?)
  - But we typically don't care about this distinction
  - We just want to make sure it is not a categorical variable (i.e., less than 20 FGs, 20-40 FGs...etc. would be categorical)
- To check, follow the process!

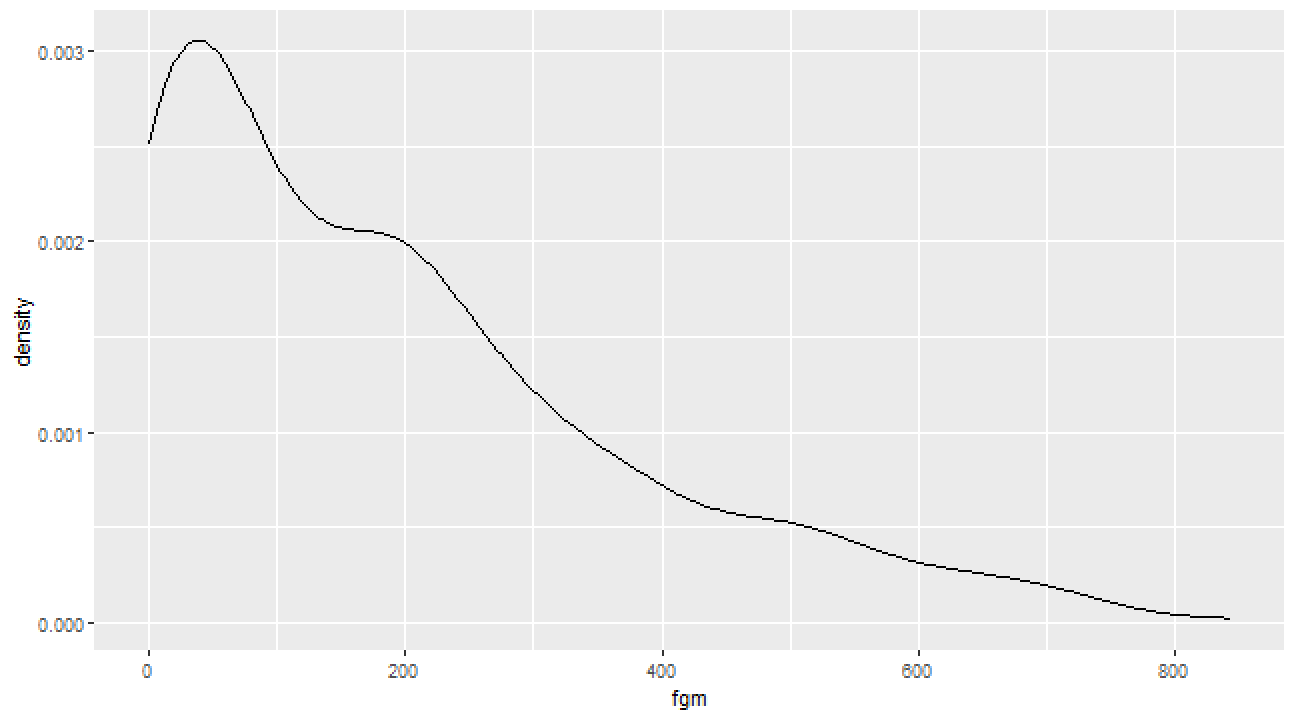
The Process: #1 Look

```
nba %>%
  select(namePlayer,slugTeam,fgm) %>%
  arrange(-fgm)
```

```
## # A tibble: 530 × 3
##    namePlayer              slugTeam   fgm
##    <chr>                   <chr>    <dbl>
##  1 James Harden            HOU        843
##  2 Bradley Beal            WAS        764
##  3 Kemba Walker            CHA        731
##  4 Giannis Antetokounmpo MIL        721
##  5 Kevin Durant            GSW        721
##  6 Paul George             OKC        707
##  7 Nikola Vucevic          ORL        701
##  8 LaMarcus Aldridge       SAS        684
##  9 Damian Lillard          POR        681
## 10 Karl-Anthony Towns      MIN        681
## # ℹ 520 more rows
```

The Process: #2 Create

```
nba %>%
  ggplot(aes(x = fgm)) +
  geom_density()
```



The Process: #3 Evaluate

- Looks like a continuous variable to me!
- Summarize it!

```
nba %>%
  summarise(mean_fg = mean(fgm,na.rm=T),
            med_fg = median(fgm,na.rm=T))
```

```
## # A tibble: 1 × 2
##   mean_fg med_fg
##     <dbl>  <dbl>
## 1    191.    157
```

- `mean()` is more easily understood, but more sensitive to outliers
- `median()` is harder to explain to a general audience, but more sensible when there are outliers

Other Variables: Use the process!

- What kind of variable is field goal percentage?
- Follow the process!

```
# INSERT CODE HERE
```

Another example

- Player age
- What kind of variable do we think this might be?
    - Continuous? It is ordered and divisible to arbitrary fractions! (Just ask any 6 and three quarters year old)
    - But is it also useful to think of it as a categorical? In the context of NBA players, there aren't many categories!
- Time for the **process**!

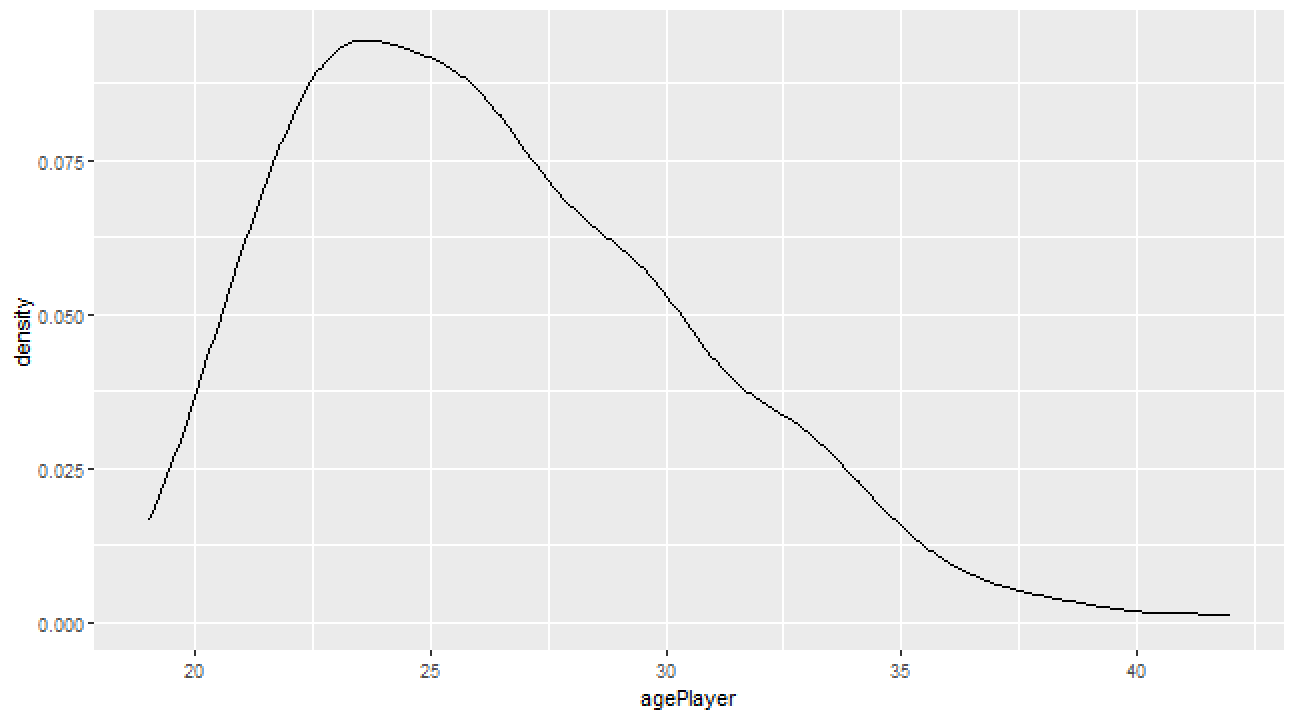
The Process: #1 Look

```
nba %>%
  select(namePlayer,agePlayer) %>%
  arrange(-agePlayer)
```

```
## # A tibble: 530 × 2
##    namePlayer     agePlayer
##    <chr>              <dbl>
##  1 Vince Carter          42
##  2 Dirk Nowitzki         41
##  3 Jamal Crawford        39
##  4 Udonis Haslem         39
##  5 Pau Gasol             38
##  6 Kyle Korver           38
##  7 Jose Calderon         37
##  8 Tony Parker           37
##  9 Dwyane Wade           37
## 10 Channing Frye         36
## # ℹ 520 more rows
```

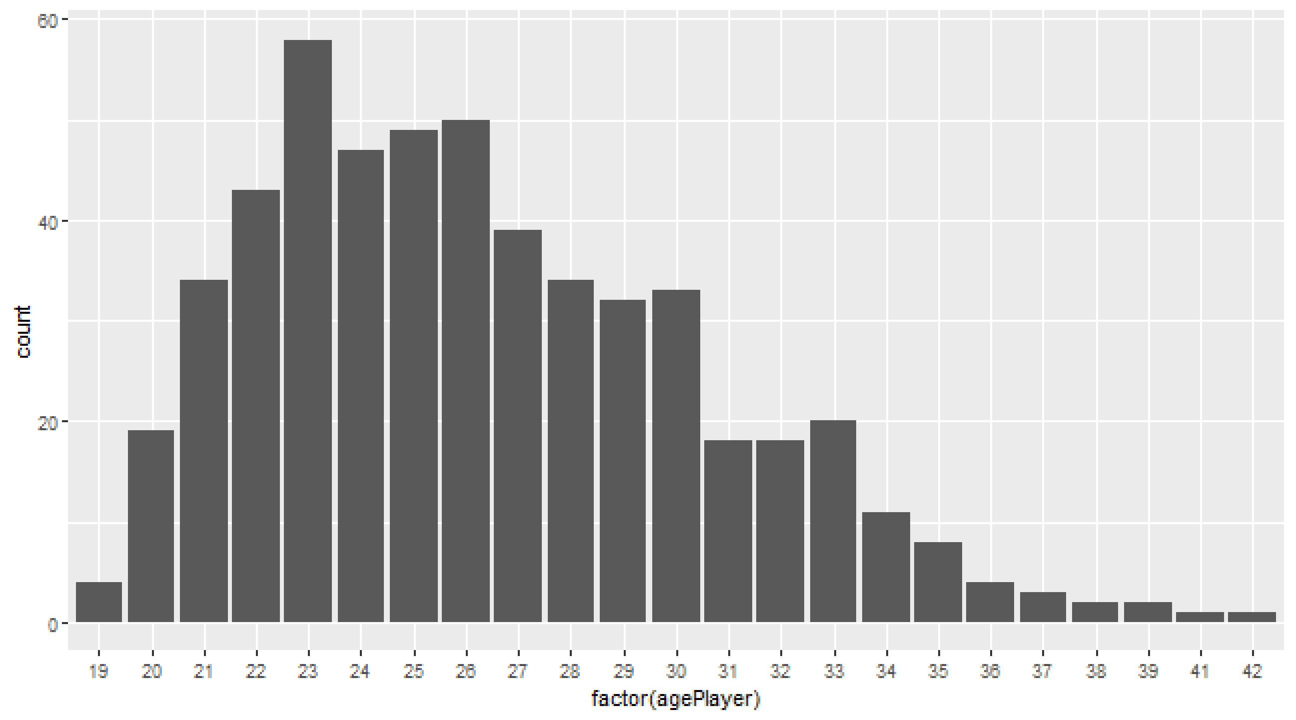
The Process: #2 Create

```
nba %>%
  ggplot(aes(x = agePlayer)) +
  geom_density()
```



The Process: #2 Create

```
nba %>%
  ggplot(aes(x = factor(agePlayer))) +
  geom_bar(stat = 'count')
```



The Process: #2 Create

The Process: #3 Evaluate

```r
quantile(nba$agePlayer,c(.1,.25,.5,.75,.9,.95))
```

```
## 10% 25% 50% 75% 90% 95%
##  21  23  26  29  32  34
```

Some more examples!

- Which of these variables is an unordered categorical variable?
- Follow the process and calculate which category in this variable is the most commonly occurring

```
# INSERT CODE HERE
```

Career Prior to NBA (`org`)

- If you chose this as your unordered categorical variable, you probably saw something like the following in step #1 of the process

```
nba %>%
  count(org) %>%
  arrange(-n)
```

```
## # A tibble: 68 × 2
##    org                          n
##    <fct>                    <int>
##  1 <NA>                       157
##  2 Other                       85
##  3 Kentucky                    25
##  4 Duke                        17
##  5 California-Los Angeles      15
##  6 Kansas                      11
##  7 Arizona                     10
##  8 Texas                       10
##  9 North Carolina               9
## 10 Michigan                     8
## # ℹ 58 more rows
```

Career Prior to NBA (org)

- The most commonly occurring categories are NA and Other!
- Wrangle some data and re-calculate

```
nba %>%
  filter(!is.na(org)) %>%
  filter(org != 'Other') %>%
  count(org) %>%
  arrange(-n)
```

```
## # A tibble: 66 × 2
##    org                         n
##    <fct>                   <int>
##  1 Kentucky                   25
##  2 Duke                       17
##  3 California-Los Angeles     15
##  4 Kansas                     11
##  5 Arizona                    10
##  6 Texas                      10
##  7 North Carolina              9
##  8 Michigan                    8
##  9 Villanova                   7
```

Categorical: Unordered, Binary

- Which variable is an unordered binary categorical?
- Follow the process and summarize it

```
# INSERT CODE HERE
```

Categorical: Unordered, Binary (`idConference`)

- Example of the default variable class (`dbl`) not corresponding to the type of variable (unordered binary)
- Should wrangle into something better

```
nba <- nba %>%
  mutate(west_conference = ifelse(idConference == 1,1,0))

nba %>%
  summarise(propWest = mean(west_conference))
```

```
## # A tibble: 1 × 1
##   propWest
##      <dbl>
## 1    0.508
```

A Preview of Multivariate Analysis

- Let's take a "conditional mean"
  - I.e., conditional on players going to Kentucky, how many points did NBA players score in the 2018-2019 season?
  - (Simpler is just to say "how many points did NBA players who went to Kentucky score?")
- Recall the `group_by()` command

A Preview of Multivariate Analysis

```r
nba %>%
  filter(!is.na(org)) %>%
  filter(org != 'Other') %>%
  group_by(org) %>%
  summarise(tot_pts = sum(pts,na.rm=T))
```

```
## # A tibble: 66 × 2
##    org                     tot_pts
##    <fct>                     <dbl>
##  1 Anadolu Efes S.K.          1270
##  2 Arizona                    5467
##  3 Baylor                      861
##  4 Boston College             1659
##  5 Butler                     1255
##  6 California                 1942
##  7 California-Los Angeles     9061
##  8 Cincinnati                  531
##  9 Colorado                   2367
## 10 Connecticut                3634
## # ℹ 56 more rows
```

A Preview of Multivariate Analysis

- Some non-college organizations snuck in there
    - `Anadolu Efes S.K.` is a professional Turkish basketball team

A Preview of Multivariate Analysis

```
nba %>%
  filter(!is.na(org)) %>%
  filter(org != 'Other') %>%
  filter(!str_detect(org,"CB|KK|rytas|FC|B.C.|S.K.|Madrid")) %>%
  group_by(org) %>%
  summarise(tot_pts = sum(pts,na.rm=T))
```

```
## # A tibble: 57 × 2
##    org                     tot_pts
##    <fct>                     <dbl>
##  1 Arizona                    5467
##  2 Baylor                      861
##  3 Boston College             1659
##  4 Butler                     1255
##  5 California                 1942
##  6 California-Los Angeles     9061
##  7 Cincinnati                  531
##  8 Colorado                   2367
##  9 Connecticut                3634
## 10 Creighton                  1230
## # ℹ 47 more rows
```

Another Preview

- Do the same but for free throw percentage (`pctFT`)
- **NB**: should you summarise with `sum()` or `mean()`? Why?

```
# INSERT CODE HERE
```

Quiz & Homework

- Go to Brightspace and take the **5th** quiz
    - The password to take the quiz is ####
- **Homework:**
    1. Work through ds1000_hw_6.Rmd
    2. Finish Problem set 3 by Friday at midnight