Modern Dive: Chapter 1 & Chapter 2

Allyson Cameron

2022-09-02

Chapter 1

Question 1

First, I will begin by installing three packages for the examples below.

```
install.packages("causact")
install.packages("dplyr")
install.packages("igraph")
```

Question 2

Next, we will load the packages using the `library`funtion.

```
library(causact)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

Let's see what happens when we use the function `as_data_frame`!

```
df <- as_data_frame(x = c(1,2,3))
```

```
## Error in as_data_frame(x = c(1, 2, 3)): Not a graph object
```

The error says : Not a graph object. This is happening because `igraph` was the last package installed and it is using the igraph logic for the code. Let's try the code with the `dplyr` package.

```
# create dataframe using dplyr package
df <- dplyr::as_data_frame(x = c(1,2,3))
```

```
## Warning: `as_data_frame()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
glimpse(df)
```

```
## Rows: 3
## Columns: 1
## $ value <dbl> 1, 2, 3
```

Why did this one work? This code worked because we specified we wanted to use the `dplyr` package which was able to read the vector as a data frame instead of something to be graphed.

```
x <- c(5,6,2,7,9,1)
n_distinct(x)
```

```
## [1] 6
```

Can you figure out why R called the function from `igraph` instead of `dplyr`? This is happening because `igraph` was the last package installed and it is using the `igraph` logic for the code.

Question 3

Let's find out what the `n_distinct` function does.

```
?n_distinct
```

This function tells us the number, n, of distinct components within the vector we specify in the parentheses.

Question 4

Now let's get a glimpse of the `baseballData` from the package `causeact`.

```
glimpse(baseballData)
```

```
## Rows: 12,145
## Columns: 5
## $ Date         <int> 20100405, 20100405, 20100405, 20100405, 20100405, 2010040~
## $ Home         <fct> ANA, CHA, KCA, OAK, TEX, ARI, ATL, CIN, HOU, MIL, NYN, PI~
## $ Visitor      <fct> MIN, CLE, DET, SEA, TOR, SDN, CHN, SLN, SFN, COL, FLO, LA~
## $ HomeScore    <int> 6, 6, 4, 3, 5, 6, 16, 6, 2, 3, 7, 11, 1, 3, 4, 2, 4, 3, 0~
## $ VisitorScore <int> 3, 0, 8, 5, 4, 3, 5, 11, 5, 5, 1, 5, 11, 5, 6, 1, 3, 6, 3~
```

There are 12,145 rows and 5 columns. `Home` is a categorical/character variable representing the home team name/state. `HomeScore` is a numeric variable representing the home teams score .

Question 5

```
baseballData[1,]
```

```
##       Date Home Visitor HomeScore VisitorScore
## 1 20100405  ANA     MIN         6            3
```

The row represents case, the unit of analysis is games so this row is one game with the date, home and visitor team names/states, and their respective scores.

```
baseballData[,2:3] %>% head()
```

```
##   Home Visitor
## 1  ANA     MIN
## 2  CHA     CLE
## 3  KCA     DET
## 4  OAK     SEA
## 5  TEX     TOR
## 6  ARI     SDN
```

The two columns represent variables for all of the cases in our data set (we are only showing the first 6 values because we used the `head` function). The first column is showing the name/state of the home team the second column is showing the name/state of the visitor team.

Question 6

Let's create a data set of our top ten hockey goal scorers.

```
# First, create the variables
name <-
  c(
    "Wayne Gretzky",
    "Gordie Howe",
    "Jaromir Jagr",
    "Brett Hull",
    "Marcel Dionne",
    "Phil Esposito" ,
    "Mike Gartner",
    "Alex Ovechkin",
    "Mark Messier" ,
    "Steve Yzerman")

goals <- c(894, 801, 766, 741, 731, 717, 708, 700, 694, 692)

year_started <- c(1979, 1946, 1990, 1986, 1971, 1963, 1979, 2005, 1979, 1983)

# Now let's actually build the data frame and view it!
df <- tibble(
  name = name,
  goals = goals,
  year_started = year_started)

glimpse(df)
```

```
## Rows: 10
## Columns: 3
## $ name         <chr> "Wayne Gretzky", "Gordie Howe", "Jaromir Jagr", "Brett Hu~
## $ goals        <dbl> 894, 801, 766, 741, 731, 717, 708, 700, 694, 692
## $ year_started <dbl> 1979, 1946, 1990, 1986, 1971, 1963, 1979, 2005, 1979, 1983
```

Chapter 2

Question 1

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
```

```
## x tibble::as_data_frame() masks igraph::as_data_frame(), dplyr::as_data_frame()
## x purrr::compose()         masks igraph::compose()
## x tidyr::crossing()        masks igraph::crossing()
## x dplyr::filter()          masks stats::filter()
## x igraph::groups()         masks dplyr::groups()
## x dplyr::lag()             masks stats::lag()
## x purrr::simplify()        masks igraph::simplify()
```

Now, let's add some data directly from the internet.

```
olympics<- read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2021/202
```

```
## Rows: 271116 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): name, sex, team, noc, games, season, city, sport, event, medal
## dbl  (5): id, age, height, weight, year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(olympics)
```

```
## Rows: 271,116
## Columns: 15
## $ id     <dbl> 1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, ~
## $ name   <chr> "A Dijiang", "A Lamusi", "Gunnar Nielsen Aaby", "Edgar Lindenau~
## $ sex    <chr> "M", "M", "M", "M", "F", "F", "F", "F", "F", "F", "M", "M", "M"~
## $ age    <dbl> 24, 23, 24, 34, 21, 21, 25, 25, 27, 27, 31, 31, 31, 31, 33, 33,~
## $ height <dbl> 180, 170, NA, NA, 185, 185, 185, 185, 185, 185, 188, 188, 188, ~
## $ weight <dbl> 80, 60, NA, NA, 82, 82, 82, 82, 82, 82, 75, 75, 75, 75, 75, 75,~
## $ team   <chr> "China", "China", "Denmark", "Denmark/Sweden", "Netherlands", "~
## $ noc    <chr> "CHN", "CHN", "DEN", "DEN", "NED", "NED", "NED", "NED", "NED", ~
## $ games  <chr> "1992 Summer", "2012 Summer", "1920 Summer", "1900 Summer", "19~
## $ year   <dbl> 1992, 2012, 1920, 1900, 1988, 1988, 1992, 1992, 1994, 1994, 199~
## $ season <chr> "Summer", "Summer", "Summer", "Summer", "Winter", "Winter", "Wi~
## $ city   <chr> "Barcelona", "London", "Antwerpen", "Paris", "Calgary", "Calgar~
## $ sport  <chr> "Basketball", "Judo", "Football", "Tug-Of-War", "Speed Skating"~
## $ event  <chr> "Basketball Men's Basketball", "Judo Men's Extra-Lightweight", ~
## $ medal  <chr> NA, NA, NA, "Gold", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

Now, we are going to look directly at the `medal` column.

```
table(olympics$medal)
```

```
##
## Bronze   Gold Silver
##  13295  13372  13116
```

Let's filter out for only Gold Medalist.

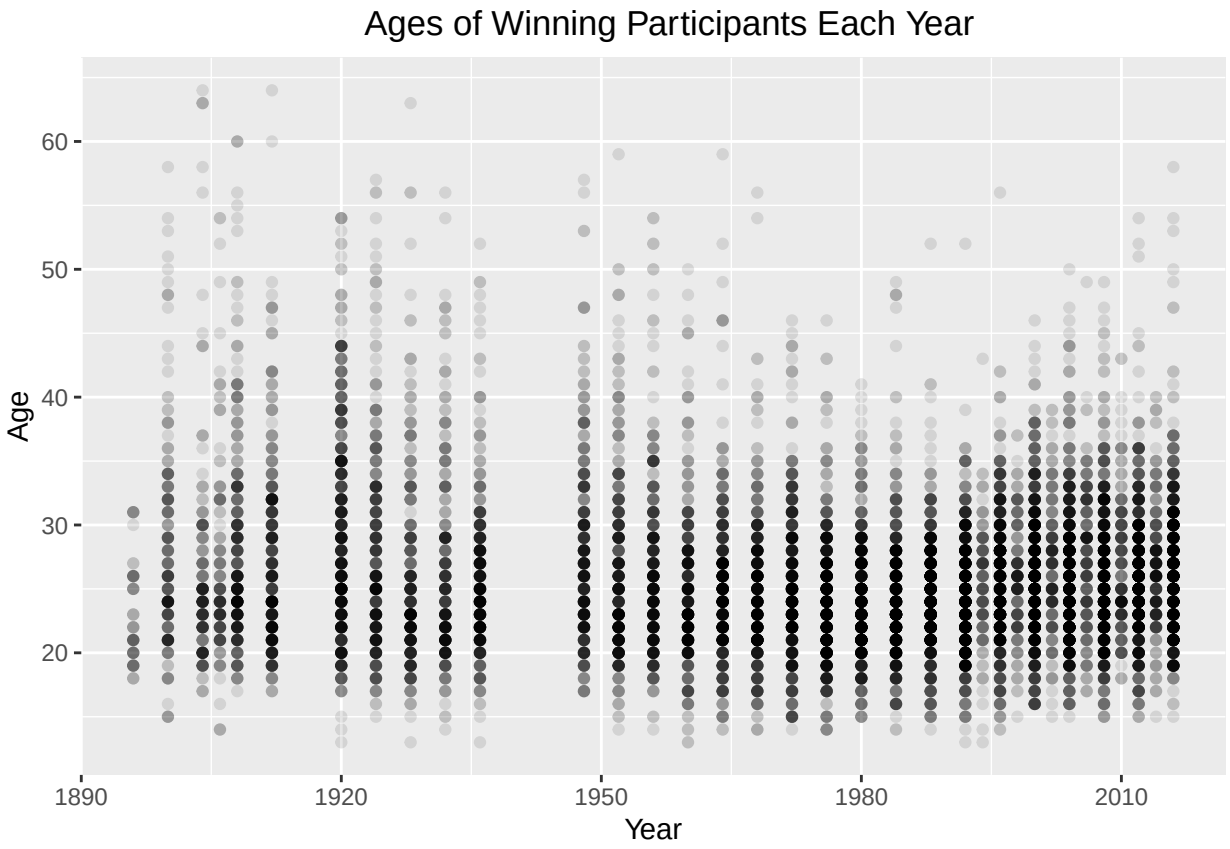
```r
gold_medalists <- olympics %>%
  filter(medal == "Gold")

glimpse(gold_medalists)
```

```
## Rows: 13,372
## Columns: 15
## $ id     <dbl> 4, 17, 17, 17, 20, 20, 20, 20, 21, 40, 42, 56, 72, 73, 73, 76, ~
## $ name   <chr> "Edgar Lindenau Aabye", "Paavo Johannes Aaltonen", "Paavo Johan~
## $ sex    <chr> "M", "M", "M", "M", "M", "M", "M", "M", "F", "M", "M", "M", "M"~
## $ age    <dbl> 34, 28, 28, 28, 20, 30, 30, 34, 27, 31, 25, 21, 28, 23, 27, 22,~
## $ height <dbl> NA, 175, 175, 175, 176, 176, 176, 176, 163, NA, NA, NA, 180, 18~
## $ weight <dbl> NA, 64, 64, 64, 85, 85, 85, 85, NA, NA, NA, NA, 83, 86, 86, 82,~
## $ team   <chr> "Denmark/Sweden", "Finland", "Finland", "Finland", "Norway", "N~
## $ noc    <chr> "DEN", "FIN", "FIN", "FIN", "NOR", "NOR", "NOR", "NOR", "NOR", ~
## $ games  <chr> "1900 Summer", "1948 Summer", "1948 Summer", "1948 Summer", "19~
## $ year   <dbl> 1900, 1948, 1948, 1948, 1992, 2002, 2002, 2006, 2008, 1960, 191~
## $ season <chr> "Summer", "Summer", "Summer", "Summer", "Winter", "Winter", "Wi~
## $ city   <chr> "Paris", "London", "London", "London", "Albertville", "Salt Lak~
## $ sport  <chr> "Tug-Of-War", "Gymnastics", "Gymnastics", "Gymnastics", "Alpine~
## $ event  <chr> "Tug-Of-War Men's Tug-Of-War", "Gymnastics Men's Team All-Aroun~
## $ medal  <chr> "Gold", "Gold", "Gold", "Gold", "Gold", "Gold", "Gold", "Gold",~
```

We have filtered out only gold medalist, there are 13,372 rows (cases). ### Question 2

```r
library(ggplot2)
ggplot(data = gold_medalists, mapping = aes(x = year, y = age)) +
  geom_point(alpha = 0.1) + labs(title =
                                   "Ages of Winning Participants Each Year",
                                 y = "Age", x = "Year") +
  theme(plot.title = element_text(hjust = 0.53))
```

```
## Warning: Removed 148 rows containing missing values (geom_point).
```



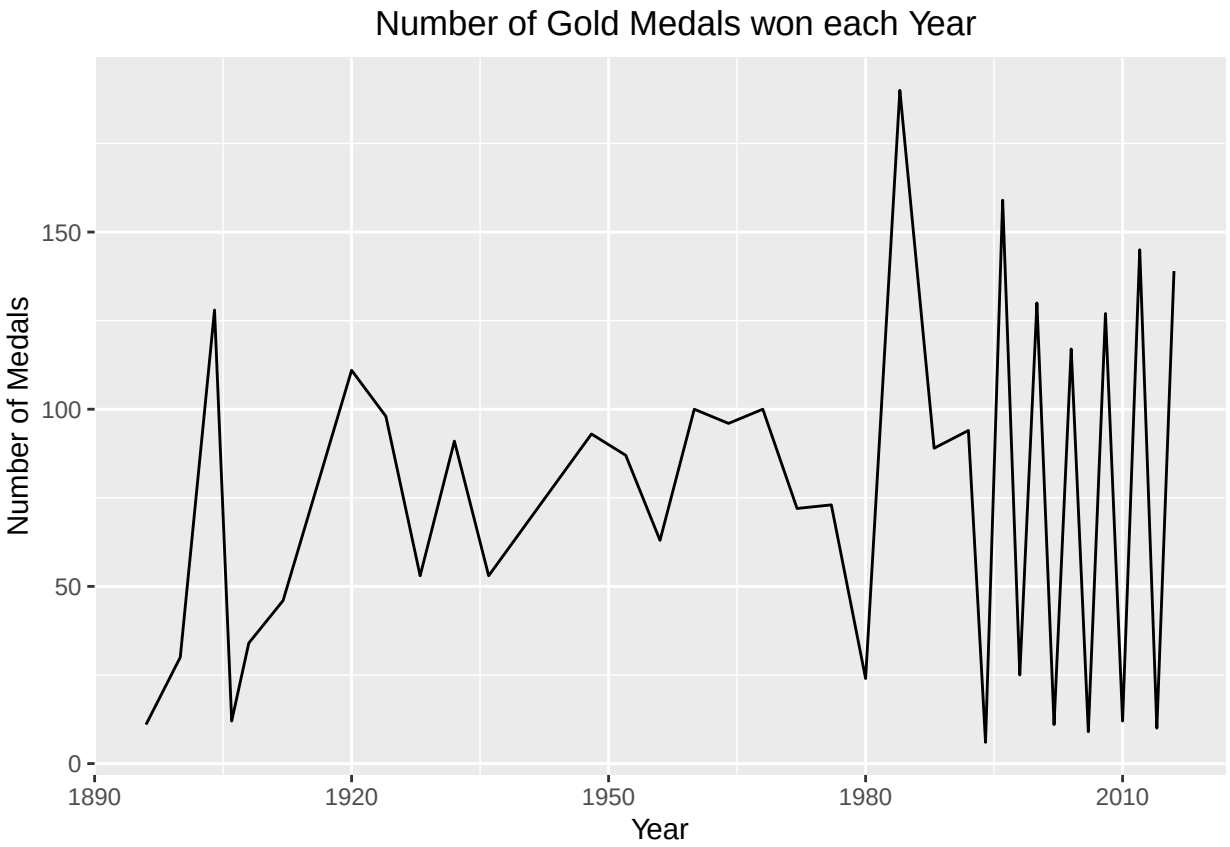
The most appropriate graph is a scatter plot because we have two numeric variables with a multitude of cases.

The age of participants has clustered around 20-40 years.

I used the alpha functionality to help view where the graph clusters, but did not use the `jitter` functionality because it made it more confusing on the actual ages of the participants.

Question 3

```
us_medals <- gold_medalists %>%
  filter(noc == "USA") %>%
  group_by(year) %>%
  summarise(num_medals = n())

ggplot(data = us_medals, mapping = aes(x = year, y = num_medals)) + geom_line() +
  labs(title = "Number of Gold Medals won each Year", y = "Number of Medals",
       x = "Year") + theme(plot.title = element_text(hjust = 0.53))
```



What was the country's most successful year? 1984

```
# Note to self: first write what variable you're looking for, use brackets to to begin sub-setting by t
us_medals$year[us_medals$num_medals == max(us_medals$num_medals)]
```

```
## [1] 1984
```

As a bonus, can you guess why the line is so wiggly (technical term) towards the end? Since the graph points are what matter when analyzing the rate of change, when we see these wiggly lines it seems like there is a larger change between the points (maybe seasons winter & summer) during these years. For instance, the top points of the line show that we were excelling (e.g. great at sports in the summer), and the low points show we were doing pretty terrible (e.g. terrible at winter sports).

Question 4

```
#create dataset with gymnastics and 100 meter dash
two_events <- gold_medalists %>%
  filter(
    event == "Gymnastics Men's Individual All-Around"  |
      event == "Gymnastics Women's Individual All-Around" |
      event == "Athletics Women's 100 metres"   |
      event == "Athletics Men's 100 metres"
  )
```

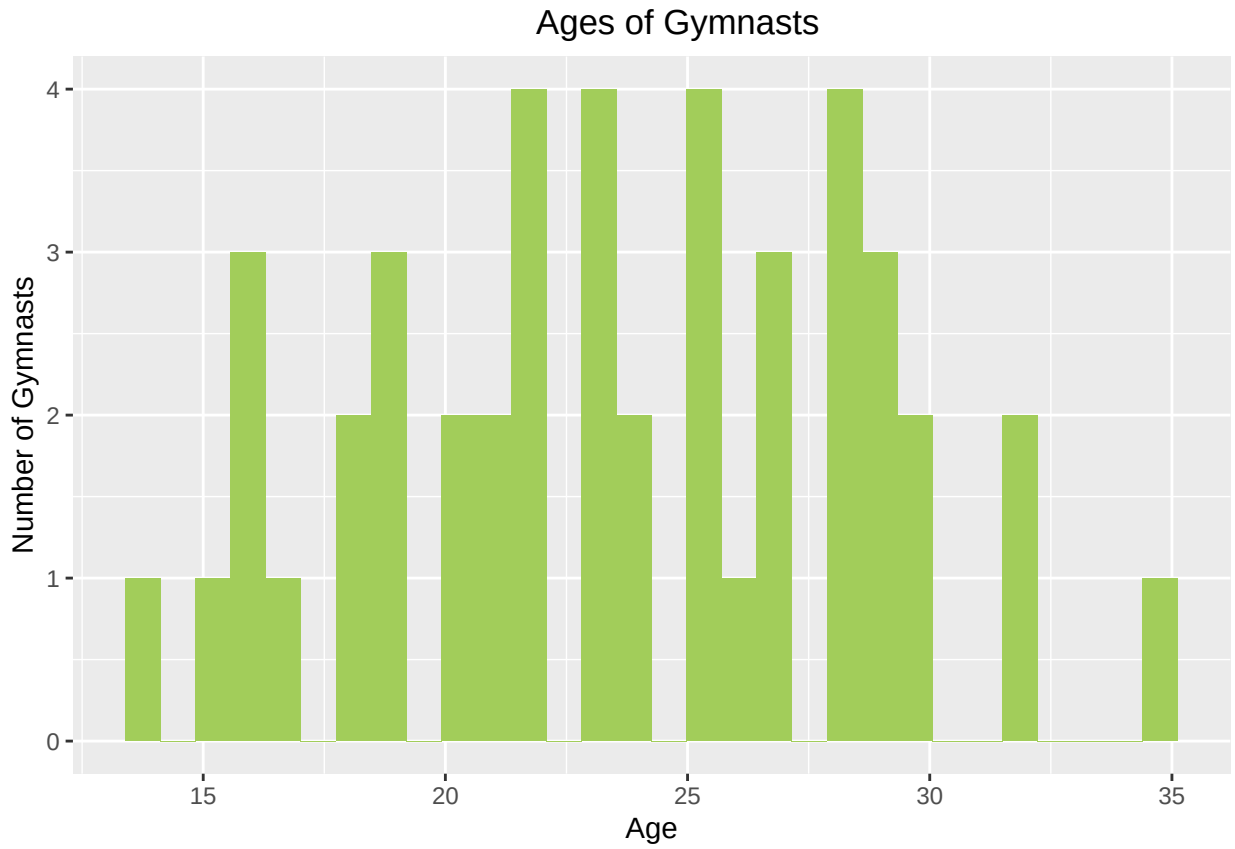
Now, filter out just gymnastics from the two events

```
gymnastics <- two_events %>%
  filter(
    event == "Gymnastics Men's Individual All-Around" |
      event == "Gymnastics Women's Individual All-Around")
```

Now, let's make a histogram using the gymnast and their ages.

```
ggplot(data = gymnastics, mapping = aes(x = age)) +
  geom_histogram(fill = "darkolivegreen3" ) +
  labs(title = "Ages of Gymnasts", x = "Age", y = "Number of Gymnasts") +
  theme(plot.title = element_text(hjust = 0.53))
```

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




```
mean(gymnastics$age)
```

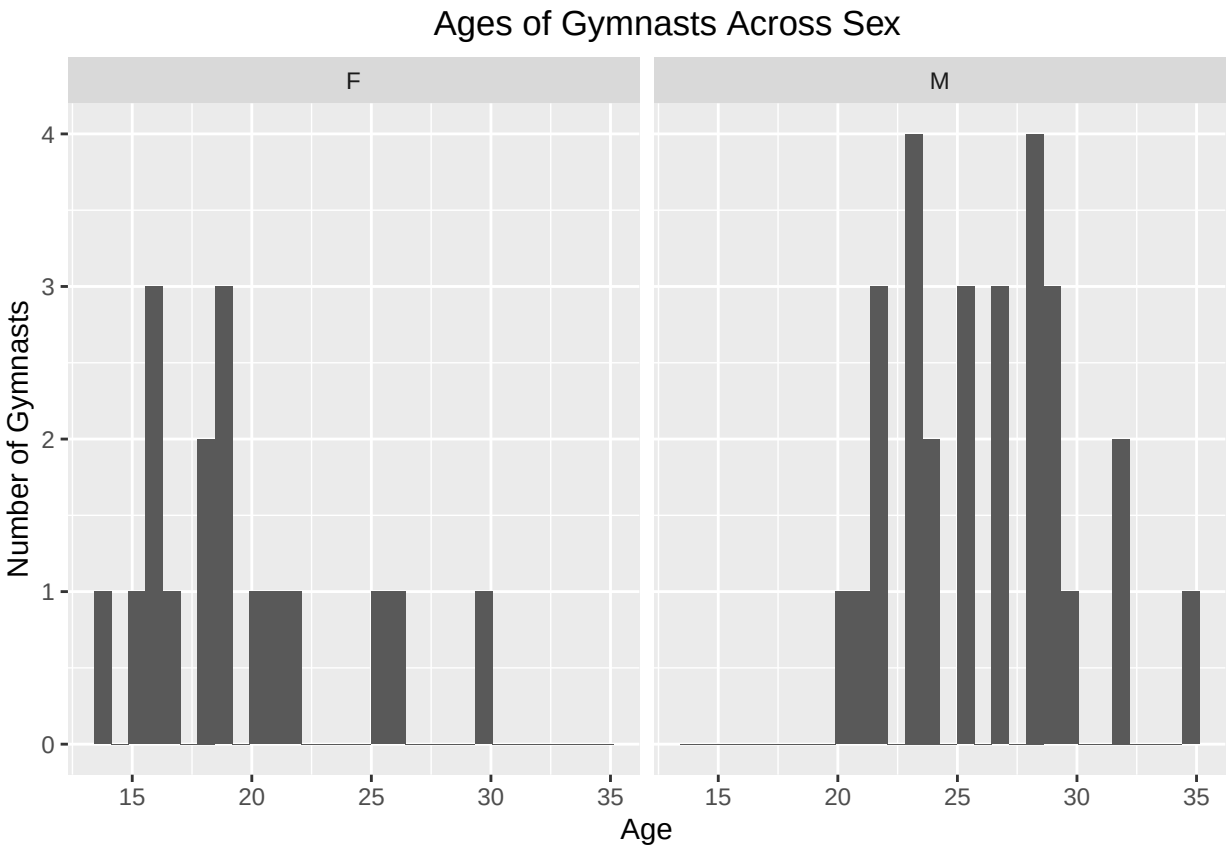
```
## [1] 23.6
```

Overall, the distribution looks pretty normal with the higher frequencies clustered around the mean (23.6) years.

Now, let's try to see the differences between female and male gymnasts' ages.

```
ggplot(data = gymnastics, mapping = aes(x = age)) + geom_histogram() +
  facet_wrap(~ sex) + labs(title = "Ages of Gymnasts Across Sex",
                           y = "Number of Gymnasts", x = "Age") +
  theme(plot.title = element_text(hjust = 0.53))
```

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



Description: The males seem to be older than women.

Question 5

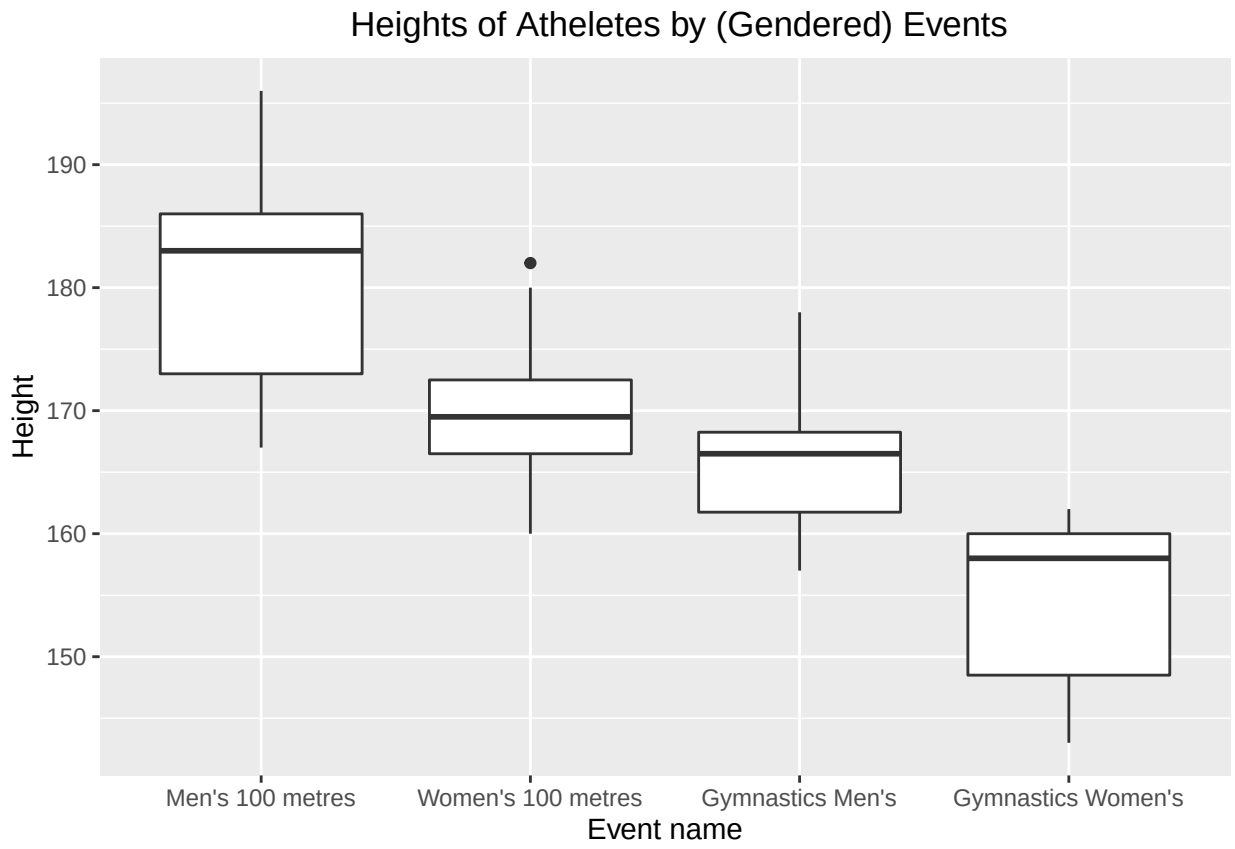
Now let's create boxplots looking at the two sports and their events in relation to the athelets heights. We're going to try to rename the events so that they fit better within the graph.

```
# Lets see what the events are called
unique(two_events$event)
```

```
## [1] "Athletics Men's 100 metres"
## [2] "Gymnastics Women's Individual All-Around"
## [3] "Gymnastics Men's Individual All-Around"
## [4] "Athletics Women's 100 metres"
```

```
# Let's create the boxplot and change event names to fit
ggplot(data = two_events, mapping = aes(x = event, y = height)) +
  geom_boxplot() + scale_x_discrete(labels = c("Men's 100 metres",
                                               "Women's 100 metres",
                                               "Gymnastics Men's",
                                               "Gymnastics Women's")) +
  labs(title = "Heights of Atheletes by (Gendered) Events", x = "Event name",
       y = "Height") + theme(plot.title = element_text(hjust = 0.53))
```

```
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
```



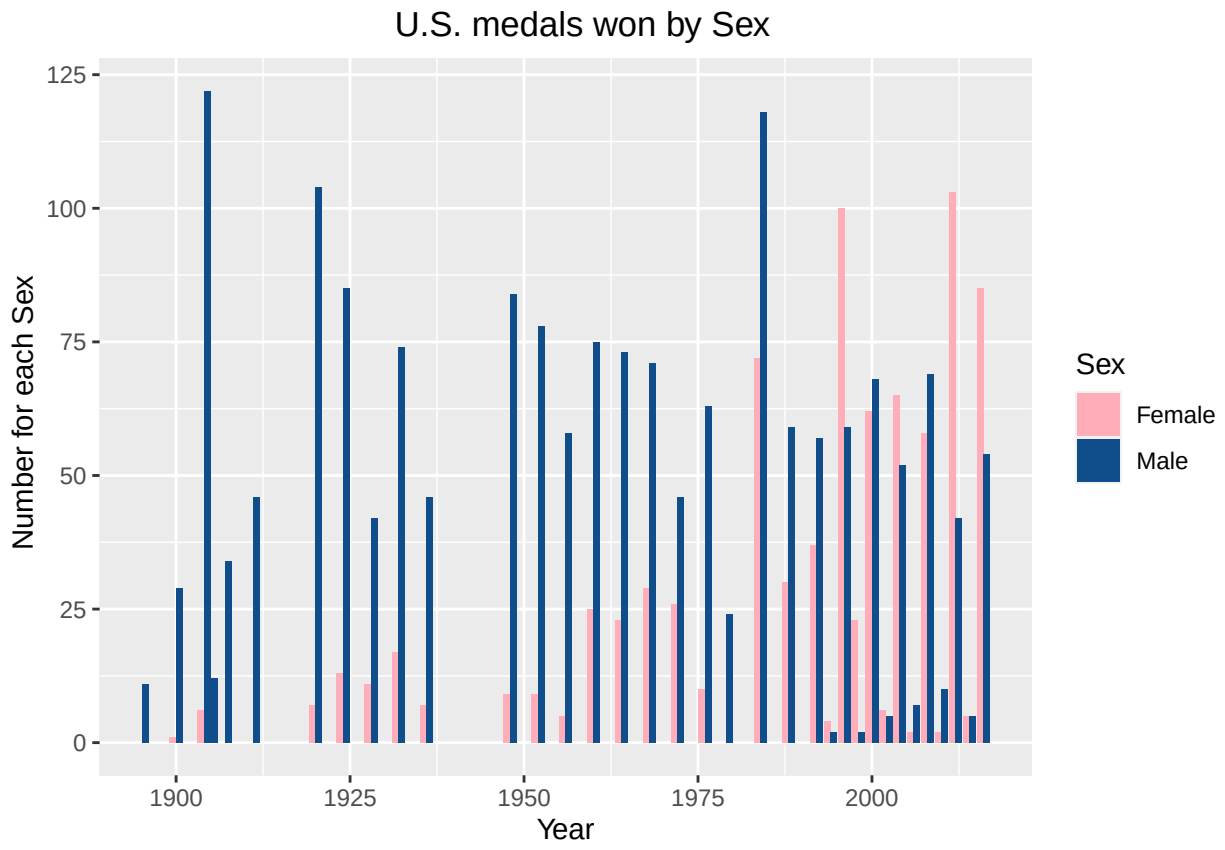
Description: In both events, women are shorter than men.

Question 6

Finally, let's explore the proportion of U.S. medals that were won by male and female athletes each year.

```
# Keep US cases for gold medalists
us_medalists <- gold_medalists %>%
  filter(noc == "USA")

# Making barplot
```

```
ggplot(data = us_medalists, mapping = aes(x = year, fill = sex)) +
  geom_bar(position = position_dodge(preserve = "single")) +
  labs(title = "U.S. medals won by Sex", x = "Year",
       y = " Number for each Sex") +
  scale_fill_manual(values = c("lightpink1", "dodgerblue4"),
                    name = "Sex", labels = c("Female","Male")) +
  theme(plot.title = element_text(hjust = 0.53))
```




```
# Let's see the years we're working with
sort(unique(us_medalists$year))
```

```
##  [1] 1896 1900 1904 1906 1908 1912 1920 1924 1928 1932 1936 1948 1952 1956 1960
## [16] 1964 1968 1972 1976 1980 1984 1988 1992 1994 1996 1998 2000 2002 2004 2006
## [31] 2008 2010 2012 2014 2016
```

Can you notice any patterns? I notice that for the first four years, there are not as many women who earned Gold medals as Men. However, after the 1980s, there were a significantly higher number of women who won medals.