DSC520 Final Project

Nathanael Ochoa

March 2, 2024

Analyzing Previous Olympic Games Data

I have chosen to analyze past Summer Olympic Games data and use this to predict the outcomes of the upcoming Paris 2024 Summer Olympics. This is not necessarily a problem but data scientists are always looking for clues in past data that may help them make accurate predictions on events that are yet to occur. Some gamblers out there may want to use this information when placing their bets! I am not a betting man nor do I hold any strong feelings towards gambling but I thought this would be a fun research topic.

I altered and combined the 3 different data sets that I found (Olympic Sports and Medals, 1896-2014, Olympics Althlete Events Analysis, and Tokyo 2020 Olympic Summer Games) and plotted the top 10 countries with the most medals from 1896-2020. After doing so it was not difficult to notice the USA's dominance in the international competition so I thought it was best to compare the 2nd and 3rd place countries, Australia and Russia, respectively. Due to obvious reasons Russia did not partake in the last Olympics so I chose the next country down the line, China. I thought it would be interesting to compare how each nation has done in recent years and estimate which country will do better later this year.

Take note that the data I'm using consists of athletes and their medals won. So if an athlete competed but did not receive a medal then they are not included. Also numbers may seem off due to the fact that if a multiple athlete/team event is won by country x it is reported as a single medal for that country. I was thrown off by this at first since I was double checking some of my data and I realized that my data has athlete medal data, so my counts are technically higher than what you'd find by using Google alone. Another way to look at it is that my data is counting physical medals handed out. Please take this into account as you read through my analysis.

Data importing and cleaning

I used the readr library and its read_csv() function to import each csv file that I will be using. I used the dplyr library for its ability to manipulate data frames. I also used the str_split_fixed() function that belongs to the stringr library.

```
# Packages to be used
library(dplyr)
library(ggplot2)
library(readr)
library(stringr)
```

Summer 1896-2012 data

I imported the csv file into a temporary data frame and then split the name column into its own data frame. I then rearranged the names into the order I wanted and capitalized every name for each athlete. I added

this new column to the original data frame and then selected the 8 columns I want to analyze. I also renamed the columns and changed the formats for the sex column.

```
# Summer 1896-2012 data, temporary data frame
import1896_2012 <-
  read_csv("C:/Users/natha/Desktop/DSC520_Assignments/Summer_Olympic_Data/Summer 1896-2012/summer.csv")

# Splits the athlete name column into multiple, stores it into another temp data frame
name_df <- str_split_fixed(import1896_2012$Athlete, ", ", 2)

# Rename columns of name_df
colnames(name_df) <- c("last", "first")

# Convert to data frame so we can use $ format to access columns
name_df <- data.frame(name_df)

# Combine columns and add result as a new column in original temp df
# toupper() used to capitalize first names
import1896_2012$athleteName <- paste(toupper(name_df$first), name_df$last)

# Select important columns and save into a new df:
summer_1896_2012 <- import1896_2012 %>% select(athleteName, Gender, Country,
                                               Discipline, Event, Medal, Year,
                                               City)

# Change 'Men' to 'M' and 'Women' to 'F'
summer_1896_2012["Gender"][summer_1896_2012["Gender"] == "Men"] <- "M"
summer_1896_2012["Gender"][summer_1896_2012["Gender"] == "Women"] <- "F"

# Rename columns: rename(newName = currentName)
summer_1896_2012 <- summer_1896_2012 %>% rename(sex = Gender, nationality = Country,
                            sport = Discipline, event = Event, medal = Medal,
                            year = Year, hostCity = City)

# Dataset is now prepped and ready to go
head(summer_1896_2012, 5)
```

```
## # A tibble: 5 x 8
##   athleteName       sex   nationality sport    event        medal  year hostCity
##   <chr>             <chr> <chr>       <chr>    <chr>        <chr> <dbl> <chr>
## 1 ALFRED HAJOS      M     HUN         Swimming 100M Freest~ Gold   1896 Athens
## 2 OTTO HERSCHMANN   M     AUT         Swimming 100M Freest~ Silv~  1896 Athens
## 3 DIMITRIOS DRIVAS  M     GRE         Swimming 100M Freest~ Bron~  1896 Athens
## 4 IOANNIS MALOKINIS M     GRE         Swimming 100M Freest~ Gold   1896 Athens
## 5 SPIRIDON CHASAPIS M     GRE         Swimming 100M Freest~ Silv~  1896 Athens
```

Summer 2016 data

This dataset was easier to work with since there was no need to split any columns. I selected only the relevant data, 2016 Games and medal winners. I also selected the columns that I needed which was the same 8 that I selected in the previous dataset. I renamed the columns and then moved onto the next csv file.

```r
# Summer 2016 data, temporary data frame
import2016 <-
  read_csv("C:/Users/natha/Desktop/DSC520_Assignments/Summer_Olympic_Data/newSummer16/1896-2016-data.csv

# Dataset is larger than needed, filter for only 2016 games and only for medal
# winners. Then use select to keep important columns only
summer_2016 <- import2016 %>% filter(Games == "2016 Summer", !is.na(Medal)) %>%
  select(Name, Sex, NOC, Sport, Event, Medal, Year, City)

# Capitalize athlete names
summer_2016$Name <- toupper(summer_2016$Name)

# Rename columns: rename(newName = currentName)
summer_2016 <- summer_2016 %>% rename(athleteName = Name, sex = Sex,
                                      nationality = NOC, sport = Sport,
                                      event = Event, medal = Medal, year = Year,
                                      hostCity = City)

# this dataset is now prepped and ready to go
head(summer_2016, 5)
```

```
## # A tibble: 5 x 8
##   athleteName                 sex   nationality sport event medal  year hostCity
##   <chr>                       <chr> <chr>       <chr> <chr> <chr> <dbl> <chr>
## 1 GIOVANNI ABAGNALE           M     ITA         Rowi~ Rowi~ Bron~  2016 Rio de ~
## 2 PATIMAT ABAKAROVA           F     AZE         Taek~ Taek~ Bron~  2016 Rio de ~
## 3 LUC ABALO                   M     FRA         Hand~ Hand~ Silv~  2016 Rio de ~
## 4 SAEID MORAD ABDEVALI        M     IRI         Wres~ Wres~ Bron~  2016 Rio de ~
## 5 DENIS MIKHAYLOVICH ABLYAZIN M     RUS         Gymn~ Gymn~ Silv~  2016 Rio de ~
```

Summer 2020 data

I followed similar steps as before, selected needed columns and renamed them to keep the format the same. The dataset did not include the year or city so I added that which was not a problem at all. I shortened the input in the medal column from 'x medal' to just 'x'. I renamed the columns and also fixed some errors in the data entry.

```r
# Summer 2020 data, temporary data frame
import_2020_medals <-
  read_csv("C:/Users/natha/Desktop/DSC520_Assignments/Summer_Olympic_Data/Summer 2020/medals.csv")

# Select the needed columns and save to a new data frame
summer_2020 <- import_2020_medals %>% select(athlete_name, athlete_sex,
                                             country_code, discipline, event,
                                             medal_type)
# Year and city were not available in the data set so I will be adding both of
# those columns

# Adding said columns
summer_2020 <- transform(summer_2020, year = "2020", hostCity = "Tokyo")

# Now drop the word 'medal' in medal column
```

```
summer_2020["medal_type"][summer_2020["medal_type"] == "Gold Medal"] <- "Gold"
summer_2020["medal_type"][summer_2020["medal_type"] == "Silver Medal"] <- "Silver"
summer_2020["medal_type"][summer_2020["medal_type"] == "Bronze Medal"] <- "Bronze"

# Change W to F since this is the format I have been using
summer_2020["athlete_sex"][summer_2020["athlete_sex"] == "W"] <- "F"

# Rename columns
summer_2020 <- summer_2020 %>% rename(athleteName = athlete_name, sex = athlete_sex,
                                      nationality = country_code, sport = discipline,
                                      medal = medal_type)
# Fix errors in data entry: "Epee"
summer_2020["event"][summer_2020["event"] ==
                       "Women's &Eacute;p&eacute;e Individual"] <- "Women's Epee Individual"
summer_2020["event"][summer_2020["event"] ==
                       "Men's &Eacute;p&eacute;e Individual"] <- "Men's Epee Individual"
summer_2020["event"][summer_2020["event"] ==
                       "Women's &Eacute;p&eacute;e Team"] <- "Women's Epee Team"
summer_2020["event"][summer_2020["event"] ==
                       "Men's &Eacute;p&eacute;e Team"] <- "Men's Epee Team"

# this dataset is now prepped and ready to go
head(summer_2020, 5)
```

```
##           athleteName sex nationality   sport      event  medal year hostCity
## 1         KIM Je Deok   X         KOR Archery Mixed Team   Gold 2020    Tokyo
## 2              AN San   X         KOR Archery Mixed Team   Gold 2020    Tokyo
## 3 SCHLOESSER Gabriela   X         NED Archery Mixed Team Silver 2020    Tokyo
## 4        WIJLER Steve   X         NED Archery Mixed Team Silver 2020    Tokyo
## 5        ALVAREZ Luis   X         MEX Archery Mixed Team Bronze 2020    Tokyo
```

Summer 1896-2020 data combination

I used the rbind() function to combine the 3 datasets and now everything is all together in one file/data frame.

```
# rbind() to vertically combine the 3 separate data frames
summer_1896_2020 <- rbind(summer_1896_2012, summer_2016, summer_2020)

# Show the beginning
head(summer_1896_2020, 3)
```

```
## # A tibble: 3 x 8
##   athleteName      sex   nationality sport    event         medal year  hostCity
##   <chr>            <chr> <chr>       <chr>    <chr>         <chr> <chr> <chr>
## 1 ALFRED HAJOS     M     HUN         Swimming 100M Freesty~ Gold  1896  Athens
## 2 OTTO HERSCHMANN  M     AUT         Swimming 100M Freesty~ Silv~ 1896  Athens
## 3 DIMITRIOS DRIVAS M     GRE         Swimming 100M Freesty~ Bron~ 1896  Athens
```

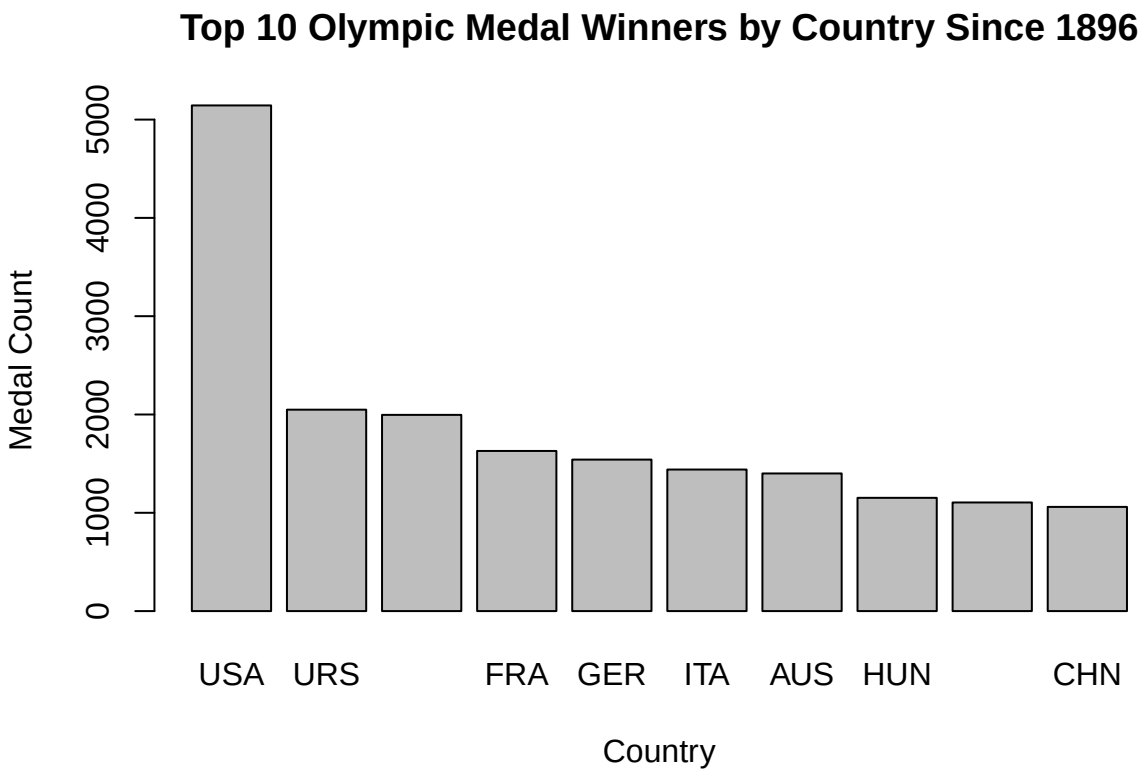
```
# Show the end, which is now 2020
tail(summer_1896_2020, 3)
```

```
## # A tibble: 3 x 8
##   athleteName   sex   nationality sport      event medal  year  hostCity
##   <chr>         <chr> <chr>       <chr>      <chr> <chr>  <chr> <chr>
## 1 VARGA Denes   M     HUN         Water Polo Men   Bronze 2020  Tokyo
## 2 VOGEL Soma    M     HUN         Water Polo Men   Bronze 2020  Tokyo
## 3 ZALANKI Gergo M     HUN         Water Polo Men   Bronze 2020  Tokyo
```

Plotting

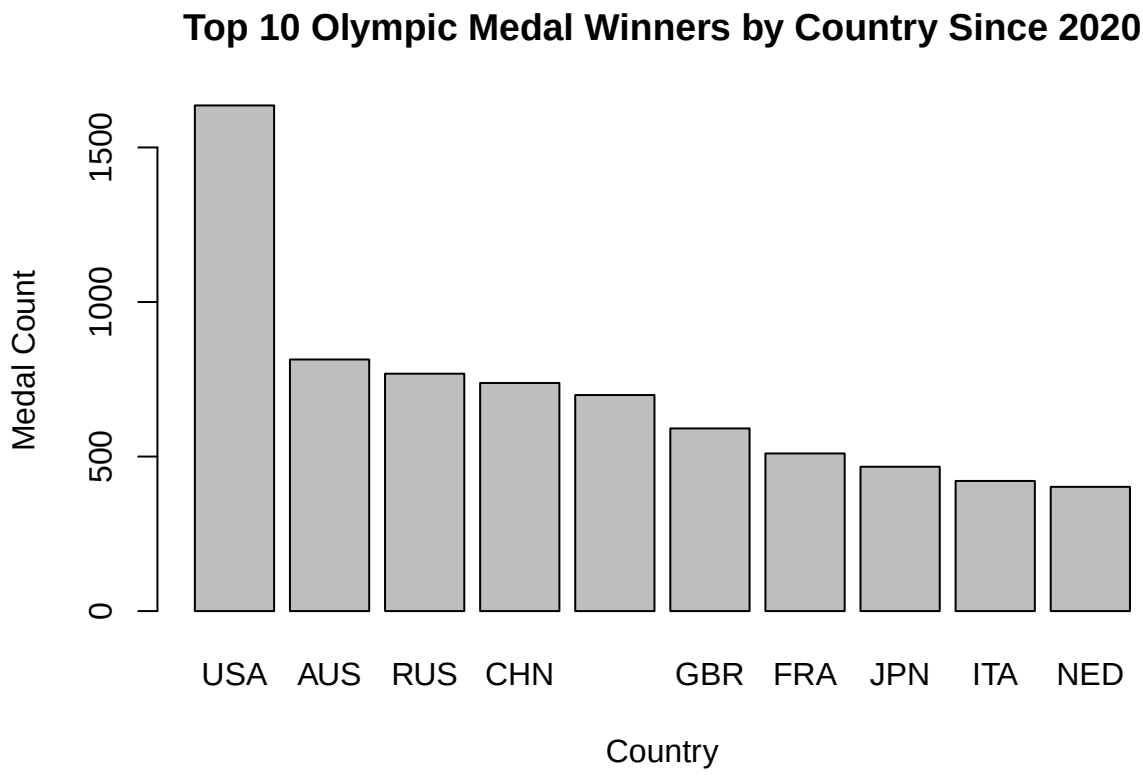
Below is a bar chart of the top 10 countries with the most medals since 1896. This is disregarding medal type.

```r
# Group by country
country_grp <- summer_1896_2020 %>% group_by(nationality)

# Count medals by country disregarding specific medals
country_medal_count <- country_grp %>% summarize(count = n())

# Order by
ordered_country_medal_count <- country_medal_count %>%
  arrange(desc(country_medal_count$count))

# Creating a data frame of the top 10
top_ten <- head(ordered_country_medal_count, 10)

# Plotting 'top_ten'
barplot(top_ten$count, names.arg = top_ten$nationality, xlab = "Country",
        ylab = "Medal Count",
        main = "Top 10 Olympic Medal Winners by Country Since 1896")
```



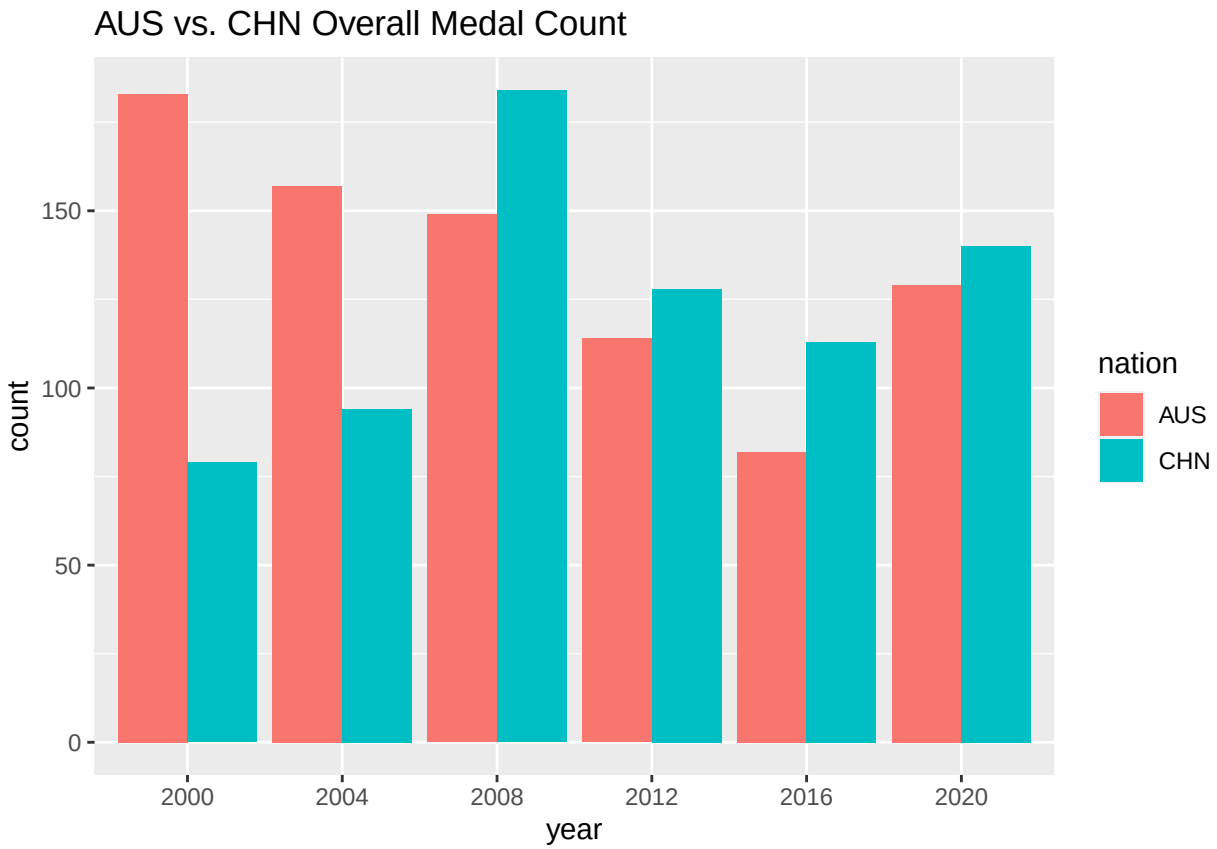
I then ran some similar code but limited the data from 2000 and on. I came up with the following bar chart, which led my decision to put Australia and China up against one another.

```
# Group by country
country_grp_20 <- summer_1896_2020 %>% filter(year >= 2000) %>% group_by(nationality)

# Count medals by country disregarding specific medals
country_medal_count_20 <- country_grp_20 %>% summarize(count = n())

# Order by
ordered_country_medal_count_20 <- country_medal_count_20 %>%
  arrange(desc(country_medal_count_20$count))

# Creating a data frame of the top 10
top_ten_20 <- head(ordered_country_medal_count_20, 10)

# Plotting
barplot(top_ten_20$count, names.arg = top_ten_20$nationality, xlab = "Country",
        ylab = "Medal Count",
        main = "Top 10 Olympic Medal Winners by Country Since 2020")
```



I created a new data frame that compares the overall medal count between the nations and plotted them side by side using the ggplot2 package.

```
# AUS overall medal count by year since 2000
aus_medals_overall <- summer_1896_2020 %>%
  filter(nationality == "AUS", year >= 2000) %>% group_by(year) %>%
  summarize(count = n())
aus_medals_overall <- cbind(aus_medals_overall, nation = c("AUS"))

# CHN overall medal count by year since 2000
chn_medals_overall <- summer_1896_2020 %>%
  filter(nationality == "CHN", year >= 2000) %>% group_by(year) %>%
  summarize(count = n())
chn_medals_overall <- cbind(chn_medals_overall, nation = c("CHN"))

# Combine data frames:
first_second <- rbind(aus_medals_overall, chn_medals_overall)

# Plot comparing the countries
ggplot(first_second, aes(year, count, fill = nation)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = "AUS vs. CHN Overall Medal Count")
```




```
# Getting AUS total
aus_total <- aus_medals_overall %>% summarize(totalAUS = sum(count))
aus_total
```

```
##   totalAUS
## 1      814
```

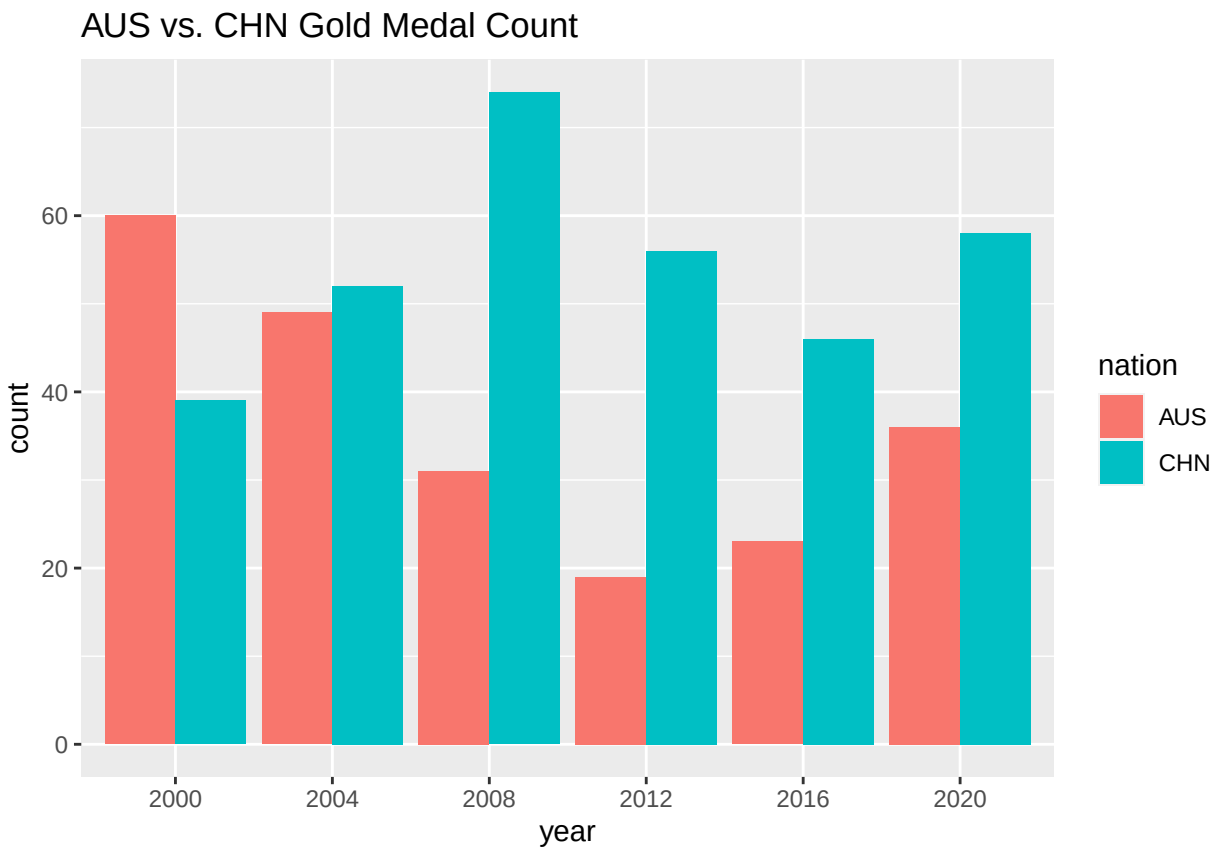
```
# Getting CHN total
chn_total <- chn_medals_overall %>% summarize(totalCHN = sum(count))
chn_total
```

```
##   totalCHN
## 1      738
```

I repeated this process but this time broke the data down by medal type. I also calculated the total of each medal type. I'll be using this later.

```
# Break up into data frames by AUS medals
aus_medals_gold <- summer_1896_2020 %>%
  filter(nationality == "AUS", year >= 2000, medal == "Gold") %>%
  group_by(year) %>% summarize(count = n())
aus_medals_gold <- cbind(aus_medals_gold, nation = c("AUS"))

aus_medals_silver <- summer_1896_2020 %>%
```

```
  filter(nationality == "AUS", year >= 2000, medal == "Silver") %>%
  group_by(year) %>% summarize(count = n())
aus_medals_silver <- cbind(aus_medals_silver, nation = c("AUS"))

aus_medals_bronze <- summer_1896_2020 %>%
  filter(nationality == "AUS", year >= 2000, medal == "Bronze") %>%
  group_by(year) %>% summarize(count = n())
aus_medals_bronze <- cbind(aus_medals_bronze, nation = c("AUS"))

# Break up into data frames by CHN medals
chn_medals_gold <- summer_1896_2020 %>%
  filter(nationality == "CHN", year >= 2000, medal == "Gold") %>%
  group_by(year) %>% summarize(count = n())
chn_medals_gold <- cbind(chn_medals_gold, nation = c("CHN"))

chn_medals_silver <- summer_1896_2020 %>%
  filter(nationality == "CHN", year >= 2000, medal == "Silver") %>%
  group_by(year) %>% summarize(count = n())
chn_medals_silver <- cbind(chn_medals_silver, nation = c("CHN"))

chn_medals_bronze <- summer_1896_2020 %>%
  filter(nationality == "CHN", year >= 2000, medal == "Bronze") %>%
  group_by(year) %>% summarize(count = n())
chn_medals_bronze <- cbind(chn_medals_bronze, nation = c("CHN"))

# Combine data frames:
first_second_gold <- rbind(aus_medals_gold, chn_medals_gold)
first_second_silver <- rbind(aus_medals_silver, chn_medals_silver)
first_second_bronze <- rbind(aus_medals_bronze, chn_medals_bronze)

# Plot comparing the countries
ggplot(first_second_gold, aes(year, count, fill = nation)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = "AUS vs. CHN Gold Medal Count")
```



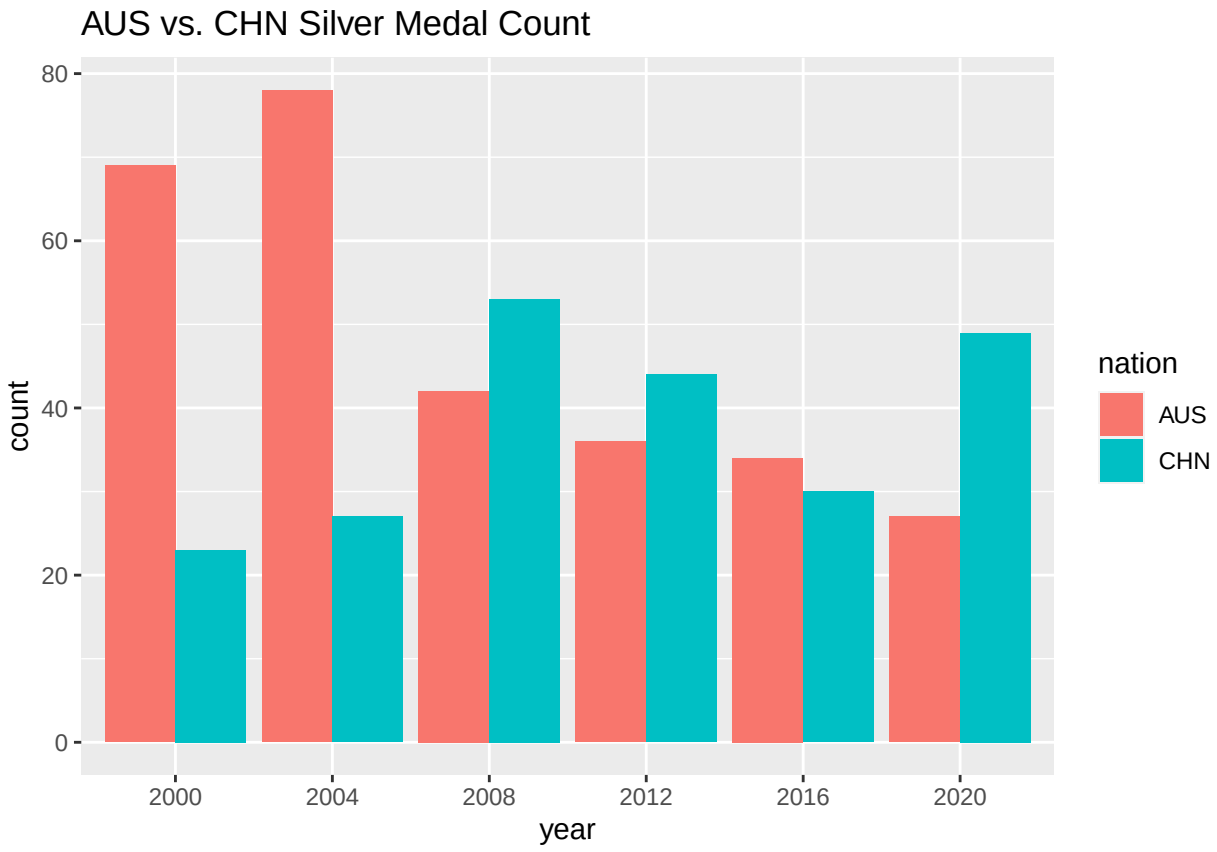

```
# Getting AUS gold total
aus_total_gold <- aus_medals_gold %>% summarize(totalAUS = sum(count))
aus_total_gold
```

```
##   totalAUS
## 1      218
```

```
# Getting CHN gold total
chn_total_gold <- chn_medals_gold %>% summarize(totalCHN = sum(count))
chn_total_gold
```

```
##   totalCHN
## 1      325
```

```
ggplot(first_second_silver, aes(year, count, fill = nation)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = "AUS vs. CHN Silver Medal Count")
```



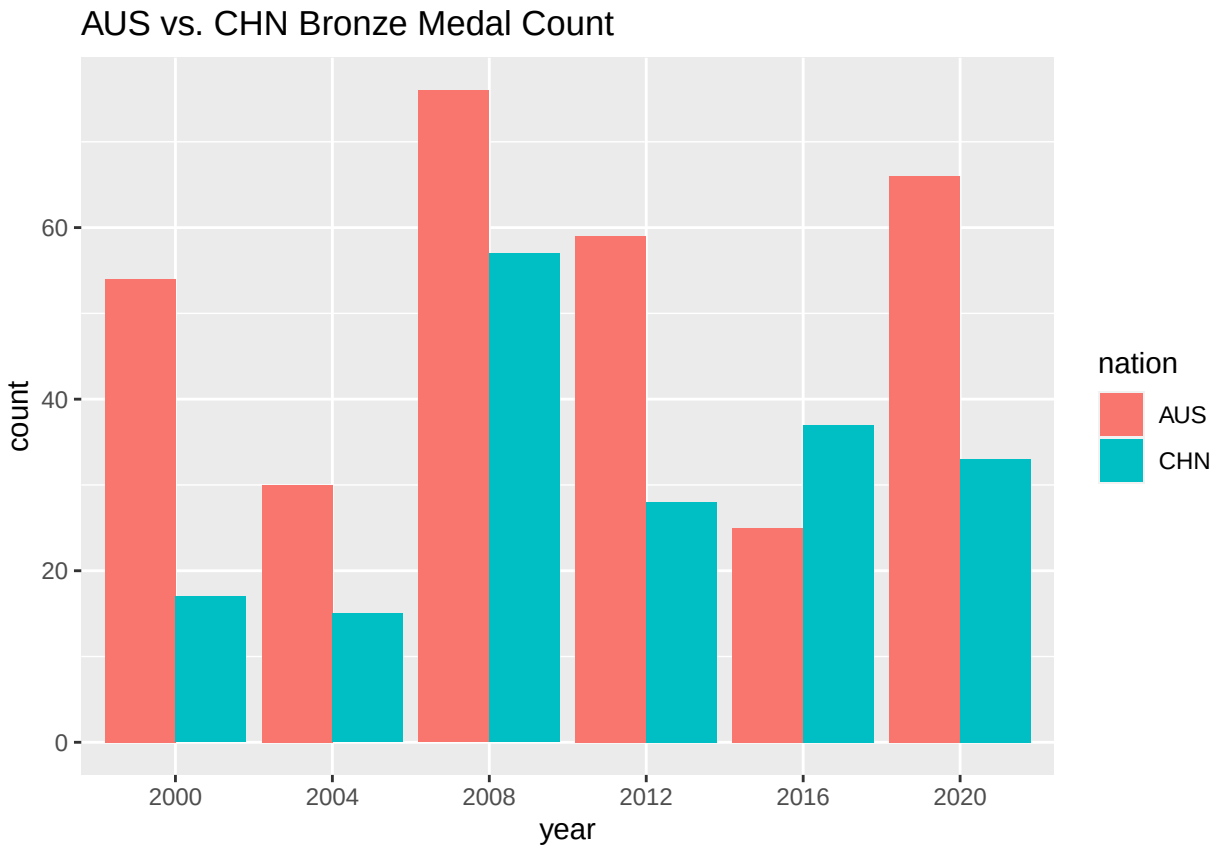

```
# Getting AUS silver total
aus_total_silver <- aus_medals_silver %>% summarize(totalAUS = sum(count))
aus_total_silver
```

```
##   totalAUS
## 1      286
```

```
# Getting CHN silver total
chn_total_silver <- chn_medals_silver %>% summarize(totalCHN = sum(count))
chn_total_silver
```

```
##   totalCHN
## 1      226
```

```
ggplot(first_second_bronze, aes(year, count, fill = nation)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = "AUS vs. CHN Bronze Medal Count")
```




```
# Getting AUS bronze total
aus_total_bronze <- aus_medals_bronze %>% summarize(totalAUS = sum(count))
aus_total_bronze
```

```
##   totalAUS
## 1      310
```

```
# Getting CHN bronze total
chn_total_bronze <- chn_medals_bronze %>% summarize(totalCHN = sum(count))
chn_total_bronze
```

```
##   totalCHN
## 1      187
```

The first plot displaying the total medals throughout the years makes it seem like Australia has China beat. Is that really the case?

```
# Percentage calculation AUS
result_aus_gold <- (aus_total_gold / aus_total) * 100
result_aus_silver <- (aus_total_silver / aus_total) * 100
result_aus_bronze <- (aus_total_bronze / aus_total) * 100

# Rename columns: rename(newName = currentName)
result_aus_gold <- result_aus_gold %>% rename(percentAUS_Gold = totalAUS)
```

```r
result_aus_silver <- result_aus_silver %>% rename(percentAUS_Silver = totalAUS)
result_aus_bronze <- result_aus_bronze %>% rename(percentAUS_Bronze = totalAUS)

# Percentage calculation CHN
result_chn_gold <- (chn_total_gold / chn_total) * 100
result_chn_silver <- (chn_total_silver / chn_total) * 100
result_chn_bronze <- (chn_total_bronze / chn_total) * 100

# Rename columns: rename(newName = currentName)
result_chn_gold <- result_chn_gold %>% rename(percentCHN_Gold = totalCHN)
result_chn_silver <- result_chn_silver %>% rename(percentCHN_Silver = totalCHN)
result_chn_bronze <- result_chn_bronze %>% rename(percentCHN_Bronze = totalCHN)

# AUS percentages
# percent Gold
result_aus_gold
```

```
##   percentAUS_Gold
## 1        26.78133
```

```r
# percent Silver
result_aus_silver
```

```
##   percentAUS_Silver
## 1          35.13514
```

```r
#percent Bronze
result_aus_bronze
```

```
##   percentAUS_Bronze
## 1          38.08354
```

```r
# CHN percentages
# percent Gold
result_chn_gold
```

```
##   percentCHN_Gold
## 1        44.03794
```

```r
# percent Silver
result_chn_silver
```

```
##   percentCHN_Silver
## 1          30.62331
```

```r
# percent Bronze
result_chn_bronze
```

```
##   percentCHN_Bronze
## 1          25.33875
```

Even though Australia has won 76 more medals than China since the 2000 Olympic Games, China has won more gold medals. China has less medals overall but when it comes down to 1st place finishes, China has more. China has 107 more to be exact. Australia has 60 more silver medals and 123 more bronze medals.

Going off of the AUS vs. CHN Gold Medal Count plot we can see that China's wins are for the most part higher than Australia's, but they still vary. Whereas Australia's seem to be on this upward trend. If I was betting with some friends I would say that China will end up with more gold medalists compared to Australia but the gap between would be much smaller. I think that if there was a 2024 section in the plot China's bar would be smaller than it was in 2020 and Australia's would be higher than it was in 2020 but still lower than China's.

I have a similar analysis when viewing the Silver Medal Count plot. China won almost twice as many silver medals as Australia but there's a pattern of slight decline. There's actually slight decline for both countries so my bet is that China doesn't win as many silver medals as they did in 2020 and Australia also wins less than what they won in 2020.

Australia has a bit of a pattern going when studying the Bronze Medal Count plot. I think they'll win more bronze medals than China but will walk away with less compared to the 2020 games.

Now let's revisit the original dataset and find out in which events does the USA win gold the most in. I'll filter by sex and event and create a top 10 dataset like before.

```
# Create a data frame with the required conditions
usa_gold <- summer_1896_2020 %>% filter(nationality == "USA", medal == "Gold") %>%
  group_by(sex, event) %>% summarize(count = n()) %>% arrange(desc(count))

# Getting the top 10
usa_gold_top <- head(usa_gold, 10)

# Display
usa_gold_top
```

```
## # A tibble: 10 x 3
## # Groups:   sex [2]
##    sex   event                     count
##    <chr> <chr>                     <int>
##  1 M     Basketball                  174
##  2 M     Eight With Coxswain (8+)    108
##  3 F     Basketball                   84
##  4 M     4X400M Relay                 72
##  5 F     Football                     68
##  6 M     4X200M Freestyle Relay       68
##  7 M     4X100M Medley Relay          67
##  8 F     4X100M Freestyle Relay       64
##  9 M     4X100M Relay                 63
## 10 M     Rugby                        46
```

Mind you that this including all of the data from 1896-2000. If we were to use just the data from year 2000 and on we'd get something like this:

```
# Create a data frame with the required conditions
usa_gold_2000 <- summer_1896_2020 %>%
  filter(nationality == "USA", medal == "Gold", year >= 2000) %>%
  group_by(sex, event) %>% summarize(count = n()) %>% arrange(desc(count))
```

```r
# Getting the top 10
usa_gold_top_2000 <- head(usa_gold_2000, 10)

# Display
usa_gold_top_2000
```

```
## # A tibble: 10 x 3
## # Groups:   sex [2]
##    sex   event                  count
##    <chr> <chr>                  <int>
##  1 F     Football                  52
##  2 F     Basketball                48
##  3 F     Women                     44
##  4 M     Basketball                36
##  5 F     Softball                  30
##  6 M     Baseball                  24
##  7 M     4X100M Medley Relay       23
##  8 F     4X400M Relay              20
##  9 F     4X200M Freestyle Relay    13
## 10 F     Water Polo                13
```

The only events that stayed in the top 10 are men's basketball, women's basketball, women's soccer (football), and men's 4x100 medley relay. Some of the events in the newer top 10 fall under the same sport umbrella so this information can still help when placing bets. I personally wouldn't bet on the US women's soccer team to do well at this year's Olympics since they did abysmal at the 2023 FIFA Women's World Cup but that's besides the point.

Implications

I'd personally be wary when placing bets on who's going to win what when it comes to this year's summer Olympics. Certain bets are pretty safe, for example the USA will most likely take home the most amount of medals. The US women's soccer (football) team has always been strong but in recent years they've terrible so I would bet on some other countries instead. Spain actually won last year's Women's World Cup so I'd definitely bet on them to do well.

I didn't take things to the athlete level but if someone were to do so you'd also need the events they competed in and didn't win a medal. That way you can calculate conversion rate and that will help with predicting how many events an athlete will win. Some athletes are just monsters and win pretty much everything. Michael Phelps is a good example of one of those athletes. I hope I've shown that using previous data and statistics can make something as trivial as betting very interesting and rewarding (if you know where to put those bets).

Limitations

I didn't get the chance to any complicated analysis methods. I tried using the glm() function as it was used in our week 10 assignment, when we calculated the model's accuracy for the binary-classifier-data.csv dataset. I was hoping I'd get to use data up until 2016 to estimate each countries medal count in 2020 and compare the estimates with the actual counts but I was getting errors so I had to change plans.

I also would've have liked to use a model to estimate an athlete's performance but I was limited due to the errors just mentioned and I had some hiccups when combining datasets. One of the data sets listed the athlete names in a format that I liked and another had the format: LASTNAME, firstname. I used the

comma as a delimiter and splot the names apart and rearranged them in the format: firstname LASTNAME. I also capitalized the athlete names that way my end goal format was: FIRSTNAME LASTNAME. The final dataset had the format: lastname firstname and included middle names and nicknames so there was no delimiter I could use. I had no choice but to capitalize it and leave it as is. Since some athlete names are "backwards" the same athlete would be counted as two different ones so I thought it was best to spare myself the headache and avoid analyzing athlete performances. Ideally I would have loved to but there was no easy way to go about that.

Conclusion

I think I've shown that past data is always useful. Someone who's ready to win big can use data like this to help make accurate predictions as to what may happen in the future. All it requires is someone that pays attention to the details, knows sports/athletes, and has *a lot* of time.

I definitely would like to compare this year's results between Australia and China and add them to the bar charts I created and compare them to my estimates. I hope I turn out to be right. Thankfully no cash was placed on the table.