Cumulative analysis on the history of Olympics using R

Dataset Description

Name of the Dataset

120 years of Olympic history: athletes and results

Description of the fields

basic bio data on athletes and medal results from Athens 1896 to Rio 2016. Each row is an athlete-event. The ID column can be used to uniquely identify athletes, since some athletes have the same name.

athlete_events.csv contains 271116 rows and 15 columns. Each row corresponds to an individual athlete competing in an individual Olympic event (athlete-events). The columns are:

ID - Unique number for each athlete Name - Athlete's name Sex - M or F Age - Integer Height - In centimeters Weight - In kilograms Team - Team name NOC - National Olympic Committee 3-letter code Games - Year and season Year - Integer Season - Summer or Winter City - Host city Sport - Sport Event - Event Medal - Gold, Silver, Bronze, or NA

file noc regions: NOC (National Olympic Committee 3 letter code) Country name (matches with regions in map_data("world")) Notes

weblink:

https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results?select=athlete_events.csv

https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results?select=noc_regions.csv

---

REQUIRED LIBRARIES

```
library("gganimate")
```

```
## Loading required package: ggplot2
```

```
## No renderer backend detected. gganimate will default to writing frames to separate files
## Consider installing:
## - the `gifski` package for gif output
## - the `av` package for video output
## and restarting the R session
```

```
library("data.table")
library("knitr")
library("gridExtra")
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::combine()   masks gridExtra::combine()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library("plotly")
```

```
##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##     last_plot
## The following object is masked from 'package:stats':
##
##     filter
## The following object is masked from 'package:graphics':
##
##     layout
```

Load the datasets # LOAD ATHLETES EVENTS DATA

```
dataOlympics <- read_csv("athleteEvents.csv", col_types = cols(
                   ID = col_character(),
                   Name = col_character(),
                   Sex = col_factor(levels = c("M","F")),
                   Age =  col_integer(),
                   Height = col_double(),
                   Weight = col_double(),
                   Team = col_character(),
                   NOC = col_character(),
                   Games = col_character(),
                   Year = col_integer(),
                   Season = col_factor(levels = c("Summer","Winter")),
                   City = col_character(),
                   Sport = col_character(),
                   Event = col_character(),
                   Medal = col_factor(levels = c("Gold","Silver","Bronze"))
                 )
)
```

```
glimpse(dataOlympics)
```

```
## Rows: 271,116
```

```
## Columns: 15
## $ ID     <chr> "1", "2", "3", "4", "5", "5", "5", "5", "5", "5", "6", "6", "6"~
## $ Name   <chr> "A Dijiang", "A Lamusi", "Gunnar Nielsen Aaby", "Edgar Lindenau~
## $ Sex    <fct> M, M, M, M, F, F, F, F, F, F, M, M, M, M, M, M, M, M, M, M, ~
## $ Age    <int> 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   <int> 1992, 2012, 1920, 1900, 1988, 1988, 1992, 1992, 1994, 1994, 199~
## $ Season <fct> Summer, Summer, Summer, Summer, Winter, Winter, Winter, Winter,~
## $ 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  <fct> NA, NA, NA, Gold, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

There are few missing values, changes are not required and will remove records with missing values in visualization for that category.

```
head(dataOlympics)
```

```
## # A tibble: 6 x 15
##   ID    Name    Sex     Age Height Weight Team   NOC   Games   Year Season City
##   <chr> <chr>   <fct> <int>  <dbl>  <dbl> <chr>  <chr> <chr>  <int> <fct>  <chr>
## 1 1     A Diji~ M        24    180     80 China  CHN   1992 ~  1992 Summer Barc~
## 2 2     A Lamu~ M        23    170     60 China  CHN   2012 ~  2012 Summer Lond~
## 3 3     Gunnar~ M        24     NA     NA Denma~ DEN   1920 ~  1920 Summer Antw~
## 4 4     Edgar ~ M        34     NA     NA Denma~ DEN   1900 ~  1900 Summer Paris
## 5 5     Christ~ F        21    185     82 Nethe~ NED   1988 ~  1988 Winter Calg~
## 6 5     Christ~ F        21    185     82 Nethe~ NED   1988 ~  1988 Winter Calg~
## # ... with 3 more variables: Sport <chr>, Event <chr>, Medal <fct>
```

We find 271116 observations, with 15 variables in this first data set. Now you will load a second data set that contains the NOCs (National Olympic Committees) information.

```
# LOAD DATA MATCHING NOCs (NATIONAL OLYMPIC COMMITTEE) WITH COUNTRIES
NOCs <- read_csv("nocRegions.csv", col_types = cols(
                  NOC = col_character(),
                  region = col_character()
                ))
glimpse(NOCs)
```

```
## Rows: 230
## Columns: 3
## $ NOC    <chr> "AFG", "AHO", "ALB", "ALG", "AND", "ANG", "ANT", "ANZ", "ARG", ~
## $ region <chr> "Afghanistan", "Curacao", "Albania", "Algeria", "Andorra", "Ang~
## $ notes  <chr> NA, "Netherlands Antilles", NA, NA, NA, NA, "Antigua and Barbud~
```

```
head(NOCs)
```

```
## # A tibble: 6 x 3
##   NOC   region      notes
##   <chr> <chr>       <chr>
## 1 AFG   Afghanistan <NA>
## 2 AHO   Curacao     Netherlands Antilles
## 3 ALB   Albania     <NA>
```

```
## 4 ALG   Algeria     <NA>
## 5 AND   Andorra     <NA>
## 6 ANG   Angola      <NA>
```

General Analysis

Has the number of athletes, countries, and events increased or decreased throughout time?

We can answer this question with the data that we have in the first data set. we can create a visualization to identify how the participation of nations and athletes has changed, as well as events.

```
# NUMBER OF NATIONS, ATHLETES AND AND EVENTS, WITHOUT ART COMPETITIONS
numbers <- dataOlympics %>%
  group_by(Year, Season) %>%
  summarize(Nations = length(unique(NOC)), Athletes = length(unique(ID)), Events = length(unique(Event)
  )
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

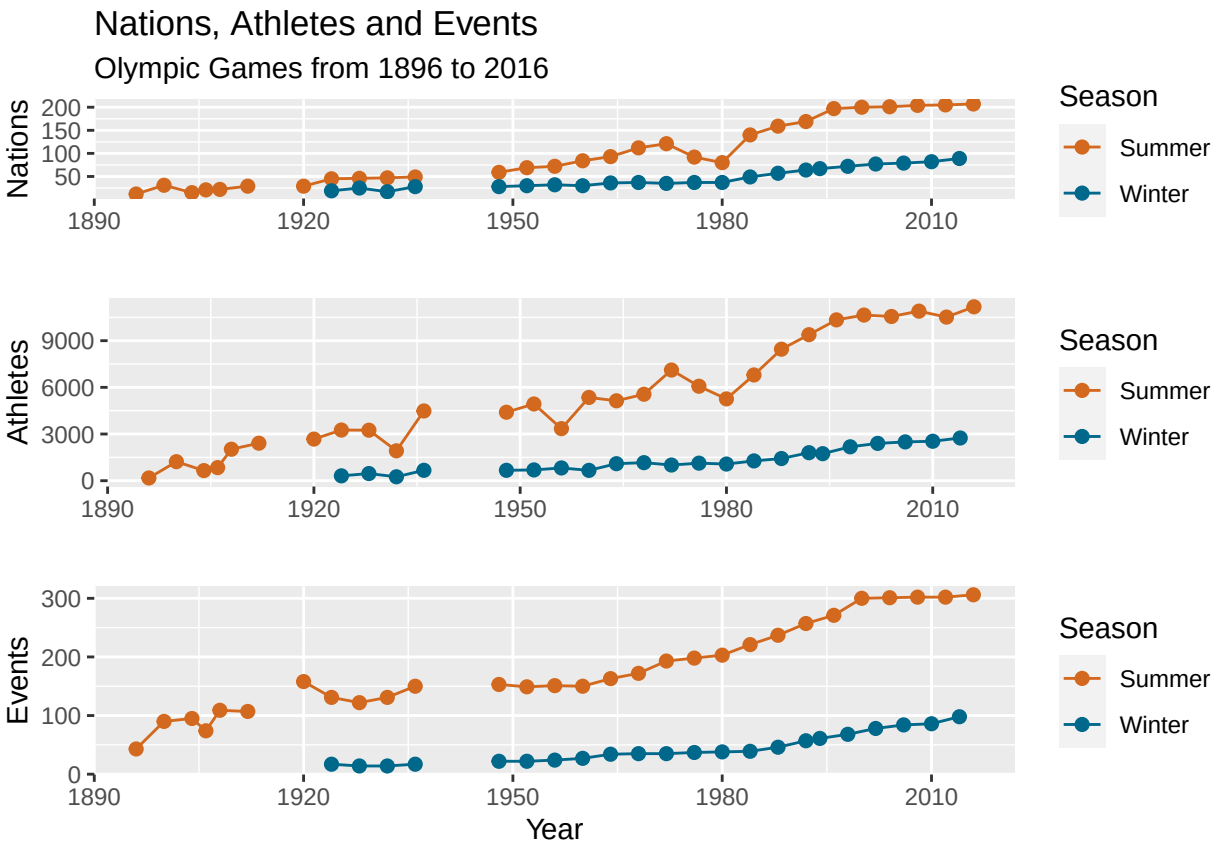
```
numbers <- numbers %>%
  mutate(gap= if(Year<1920) 1 else if(Year>=1920 & Year<=1936) 2 else 3)
```

```
## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used
```

```
## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used

## Warning in if (Year < 1920) 1 else if (Year >= 1920 & Year <= 1936) 2 else 3:
## the condition has length > 1 and only the first element will be used

## Warning in if (Year >= 1920 & Year <= 1936) 2 else 3: the condition has length >
## 1 and only the first element will be used
```

```
plotNations <- ggplot(numbers, aes(x=Year, y=Nations, group=interaction(Season,gap), color=Season)) +
  geom_point(size=2) +
  geom_line() +
  scale_color_manual(values=c("chocolate","deepskyblue4")) +
  labs(x = " ", y = "Nations",
       title="Nations, Athletes and Events",
```

```
       subtitle = "Olympic Games from 1896 to 2016")

plotAthletes <- ggplot(numbers, aes(x=Year, y=Athletes, group=interaction(Season,gap), color=Season)) +
  geom_point(size=2) +
  geom_line() +
  scale_color_manual(values=c("chocolate","deepskyblue4")) +
  xlab("")

plotEvents <- ggplot(numbers, aes(x=Year, y=Events, group=interaction(Season,gap), color=Season)) +
  geom_point(size=2) +
  geom_line() +
  scale_color_manual(values=c("chocolate","deepskyblue4"))

grid.arrange( plotNations, plotAthletes, plotEvents, ncol=1)
```



we can see how it has been increasing over time.

Medal winners

Which nations took home the most medals?

we visualize the proportion of gold, silver, and bronze medals that each nation has accumulated. Let's say we want to get the top 30.

```
# THE TOTAL NUMBER OF MEDALS GIVEN TO EACH TEAM
medalCounts <- dataOlympics %>% filter(!is.na(Medal))%>%
```

```
  group_by(NOC, Medal, Event, Games) %>%
  summarize(isMedal=1)
```

```
## `summarise()` has grouped output by 'NOC', 'Medal', 'Event'. You can override using the `.groups` ar
```

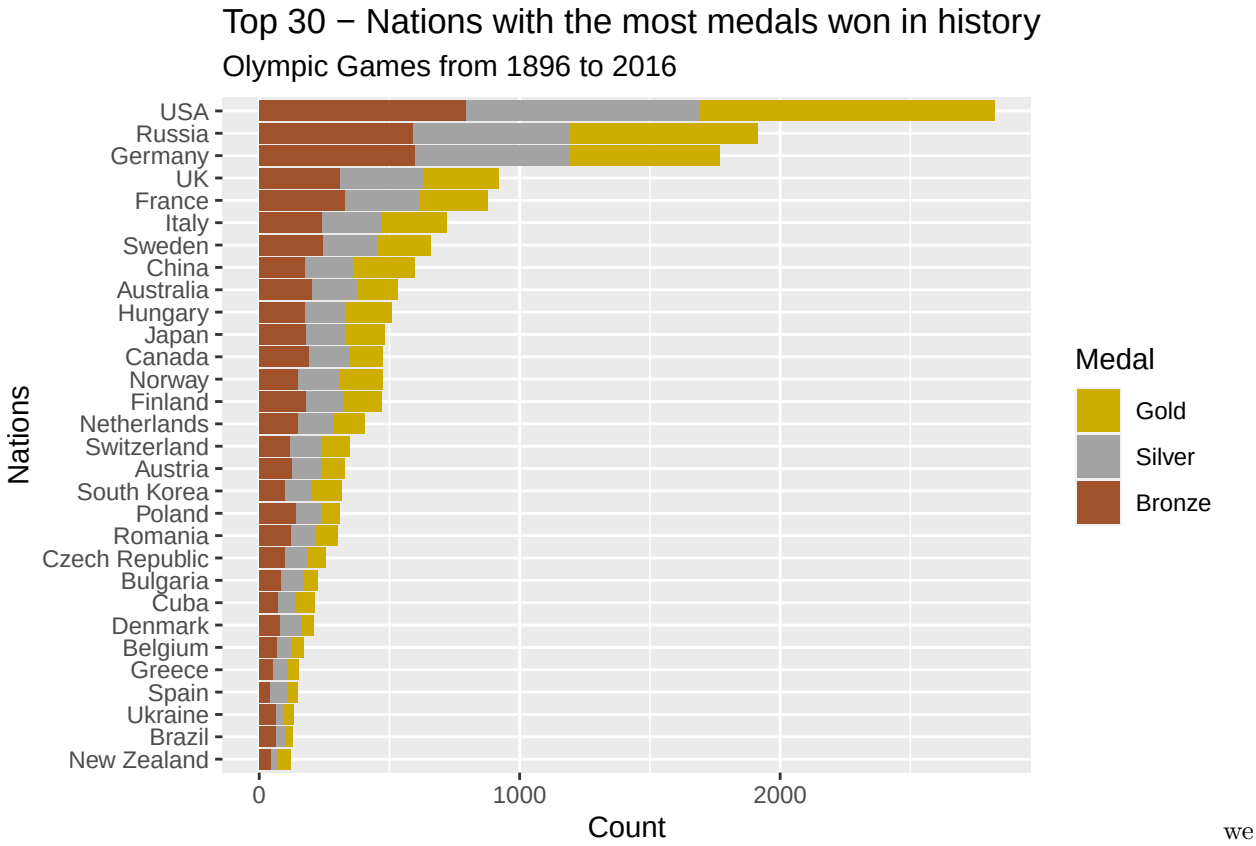
```
medalCounts <-  medalCounts %>%
  group_by(NOC, Medal) %>%
  summarize(Count= sum(isMedal))
```

```
## `summarise()` has grouped output by 'NOC'. You can override using the `.groups` argument.
```

```
medalCounts <- left_join(medalCounts, NOCs, by= "NOC" )

medalCounts <- medalCounts %>%
  mutate (Team = region)

medalCounts <- medalCounts %>% select( Medal, Team, Count)
```

```
## Adding missing grouping variables: `NOC`
```

```
# ORDERING TEAM BY TOTAL MEDAL COUNT

levelsTeam <- medalCounts %>%
  group_by(Team) %>%
  summarize(Total=sum(Count)) %>%
  arrange(desc(Total)) %>%
  select(Team) %>%
  slice(30:1)

medalCounts$Team <- factor(medalCounts$Team, levels=levelsTeam$Team)

medalCounts <- medalCounts %>% filter(Team != "NA")

# PLOT MEDAL COUNTS

ggplot(medalCounts, aes(x=Team, y=Count, fill=Medal)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values=c("gold3","gray64","sienna")) +
  labs(x = "Nations", y = "Count",
       title="Top 30 - Nations with the most medals won in history",
       subtitle = "Olympic Games from 1896 to 2016")
```



we can see that the USA, Russia, and Germany are the countries that lead the top.

Which nations took home the most medals? — Animated Plot

an attractive way to visualize what has been previously obtained is with an animated plot, which can show in detail over the years the medals obtained by each nation, let's say that for this you only need the top 10.

```
# NUMBER OF MEDALS GIVEN TO EACH TEAM

medalCounts <- dataOlympics %>% filter(!is.na(Medal))%>%
  group_by(NOC, Medal, Event, Games, Year) %>%
  summarize(isMedal=1)
```

```
## `summarise()` has grouped output by 'NOC', 'Medal', 'Event', 'Games'. You can override using the `.g
```

```
medalCounts <-  medalCounts %>%
  group_by(NOC, Medal, Year) %>%
  summarize(Count= sum(isMedal))
```

```
## `summarise()` has grouped output by 'NOC', 'Medal'. You can override using the `.groups` argument.
```

```
medalCounts <- left_join(medalCounts, NOCs, by= "NOC" )

medalCounts <- medalCounts %>%
  mutate (Team = region)
medalCounts <- medalCounts %>% select( Medal, Team, Count, Year)
```

```
## Adding missing grouping variables: `NOC`
```

```
# ORDERING TEAM BY TOTAL MEDAL COUNT
levelsTeam <- medalCounts %>%
  group_by(Team) %>%
  summarize(Total=sum(Count)) %>%
  arrange(desc(Total)) %>%
  select(Team) %>%
  slice(10:1)

medalCounts$Team <- factor(medalCounts$Team, levels=levelsTeam$Team)

medalCounts <- medalCounts %>% filter(Team != "NA")

# ANIMATED PLOT MEDAL COUNT

plotMedalsAnim<- ggplot(medalCounts, aes(x=Team, y=Count, fill=Medal)) +
  labs(x = "Nations", y = "Count",
       title='Top 10 - Comparison over time, nations with the most medals',
       subtitle = 'Olympic Games from 1896 to 2016 - Year: {frame_time}')  +
  transition_time(Year)+
  geom_col() +
  coord_flip() +
  scale_fill_manual(values=c("gold3","gray64","sienna"))

# animate(plotMedalsAnim,fps=2,renderer = ffmpeg_renderer(format = "webm"))
```

we loaded the gganimate library for this.

View the map to see which countries won the most medals

Another way to visualize how many accumulated medals each nation has could be through a map. This will also provide a visual overview of the entire globe in a general.

```
# MAP NATIONS WITH MOST MEDALS WON
medalCounts <- dataOlympics %>% filter(!is.na(Medal))%>%
  group_by(NOC, Medal, Event, Games) %>%
  summarize(isMedal=1)
```

```
## `summarise()` has grouped output by 'NOC', 'Medal', 'Event'. You can override using the `.groups` arg
```

```
medalCounts <-  medalCounts %>%
  group_by(NOC, Medal) %>%
  summarize(Count= sum(isMedal))
```

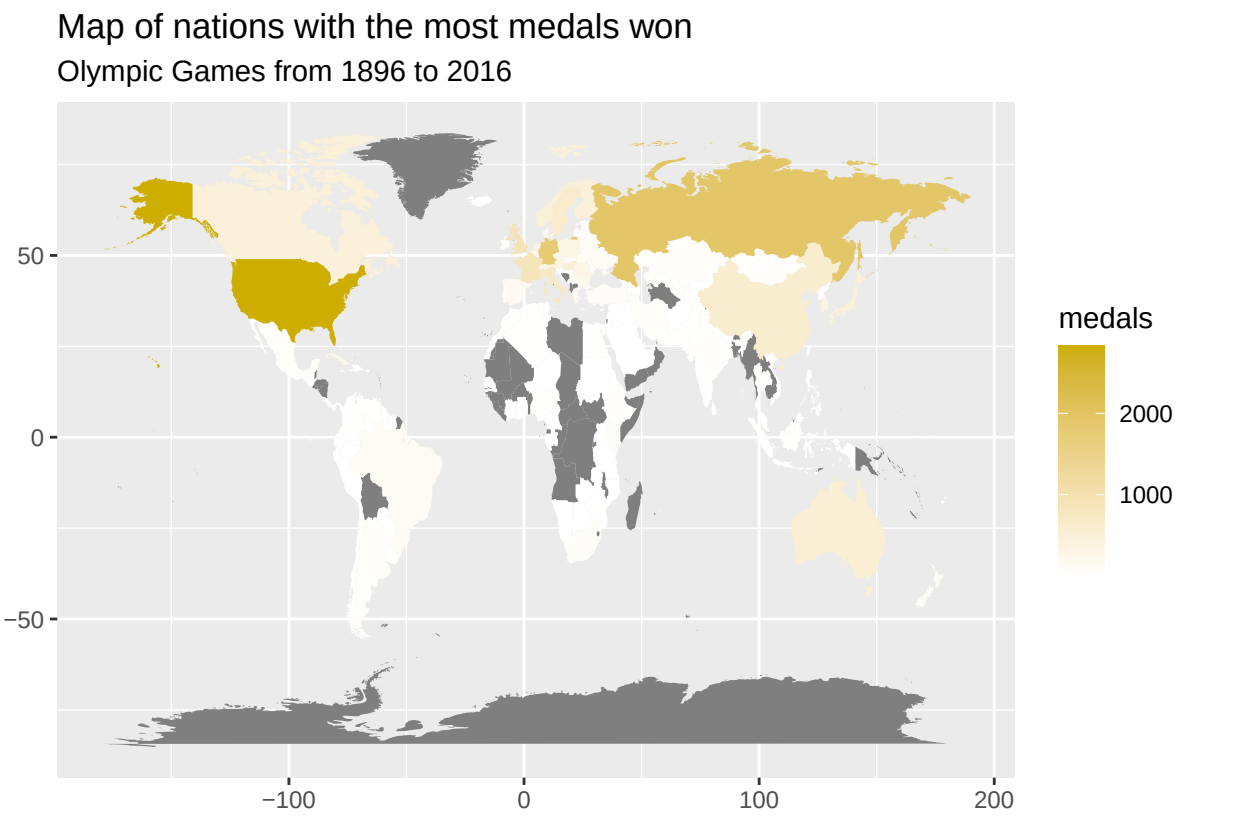
```
## `summarise()` has grouped output by 'NOC'. You can override using the `.groups` argument.
```

```
medalCounts <- left_join(medalCounts, NOCs, by= "NOC" ) %>%
  select(region, NOC, Medal, Count)

medalCounts <- medalCounts %>%
  group_by(region) %>%
  summarize(Total=sum(Count))

data_regions <- medalCounts %>%
  left_join(NOCs,by="region") %>%
  filter(!is.na(region))
```

```
earth <- map_data("world")

earth <- left_join(earth, data_regions, by="region")

# PLOT MAP

plotMapMedals <- ggplot(earth, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = Total, label= region)) +
  labs(x = "", y = "",
       title="Map of nations with the most medals won",
       subtitle = "Olympic Games from 1896 to 2016") +
  guides(fill=guide_colourbar(title="medals")) +
  scale_fill_gradient(low="white",high="gold3")
```

```
## Warning: Ignoring unknown aesthetics: label
```

```
plotMapMedals
```




```
ggplotly(plotMapMedals)
```

we get a map , where we can see the countries that until the Rio 2016 games have not obtained a single medal, such as Honduras, Bolivia, and Albanie, for example.

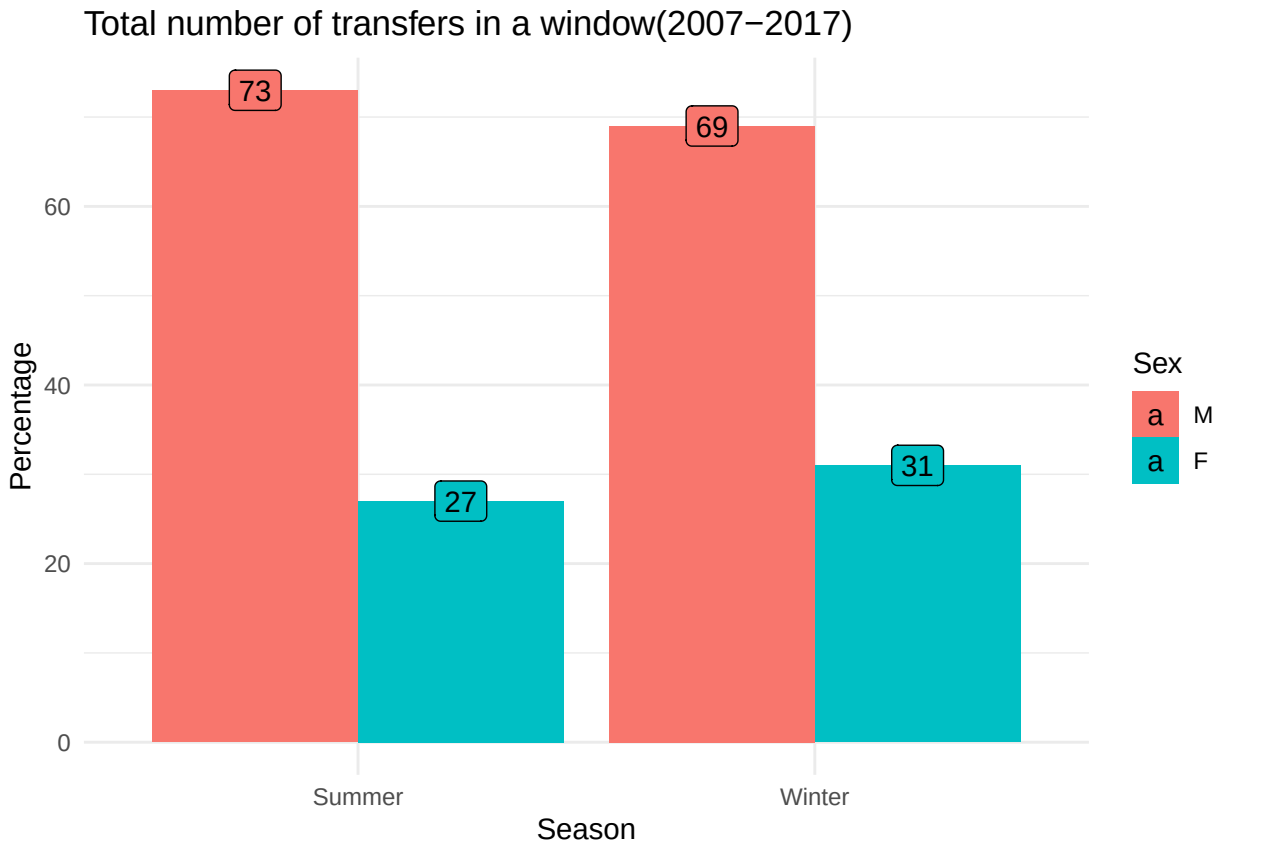
Analysis By Sex

Participation of male and female athletes over time

```
df <- dataOlympics %>%
group_by(Season, Sex) %>%
summarise(Count = n()) %>%
mutate(Percentage = round(Count*100 / sum(Count)))
```

```
## `summarise()` has grouped output by 'Season'. You can override using the `.groups` argument.
```

```
df %>%
ggplot(aes(x=Season, y=Percentage, fill = Sex)) + geom_bar(stat='identity',position=position_dodge()) +
ggtitle("Total number of transfers in a window(2007-2017)") +
geom_label(label=df$Percentage, position = position_dodge(0.9)) +
theme_minimal()
```




```
3
```

```
## [1] 3
```

During the winter and the summer the percentage and women have remained the same.

By trend

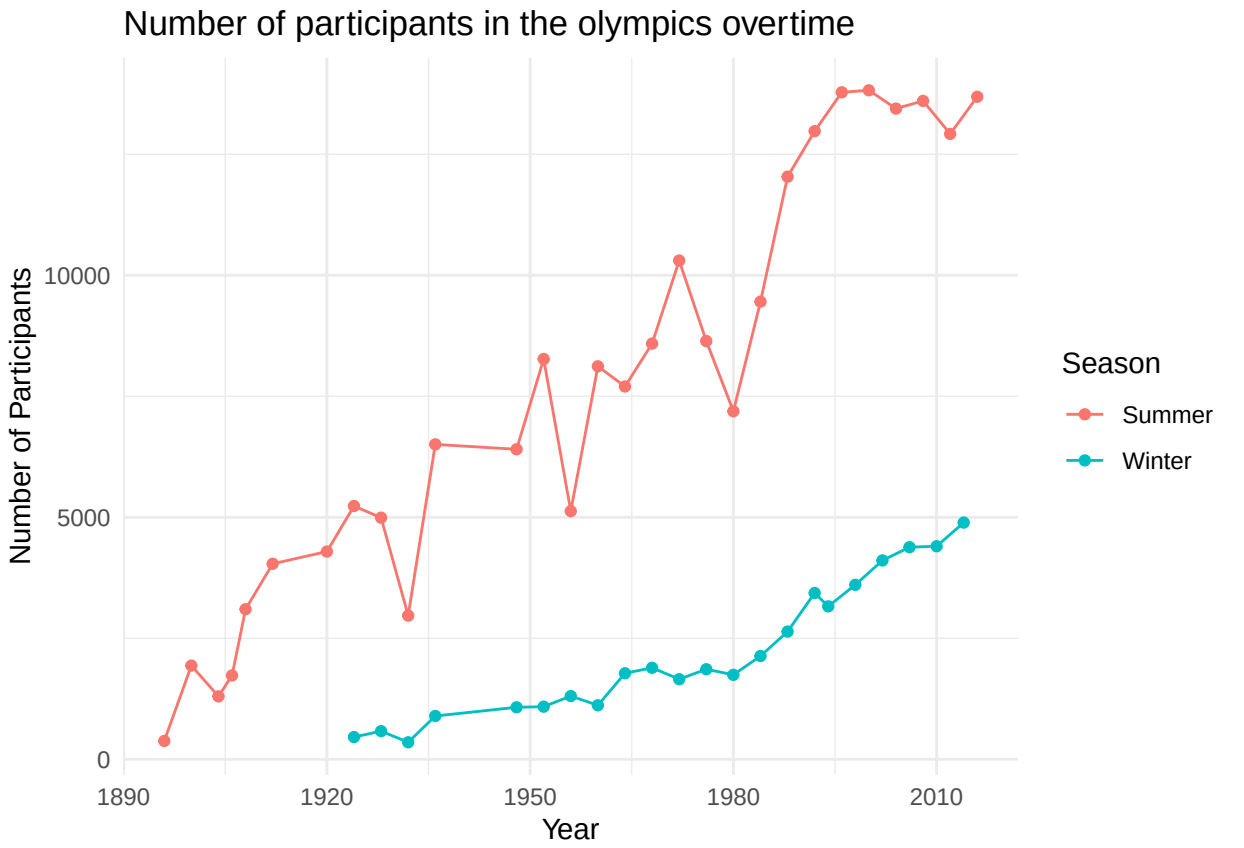
Overall trend

As the olympics progressed through the ages the number of people participating in it would have increased, which meant that the number of men and women participating in the olympics grew.

```
dataOlympics %>%
group_by(Year, Season) %>%
summarise(NumberOfParticipants = n()) %>%
```

```
ggplot(aes(x = Year, y = NumberOfParticipants, group = Season)) +
geom_line(aes(color = Season)) +
geom_point(aes(color = Season)) +
labs(x = "Year", y = "Number of Participants", title = "Number of participants in the olympics overtime
theme_minimal()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```



The number of participants in the olympics have grown overtime. It is also obvious that the number of participants in the summer olympics are more than that of the winter olympics.

Trend of sex ratio

```
groupMale <- dataOlympics %>%
filter(Sex == "M") %>%
group_by(Year, Season) %>%
summarise(Number_Of_Men = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

```
groupFemale <- dataOlympics %>%
filter(Sex == "F") %>%
group_by(Year, Season) %>%
summarise(Number_Of_Women = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

```
group <- groupMale %>%
left_join(groupFemale) %>%
```

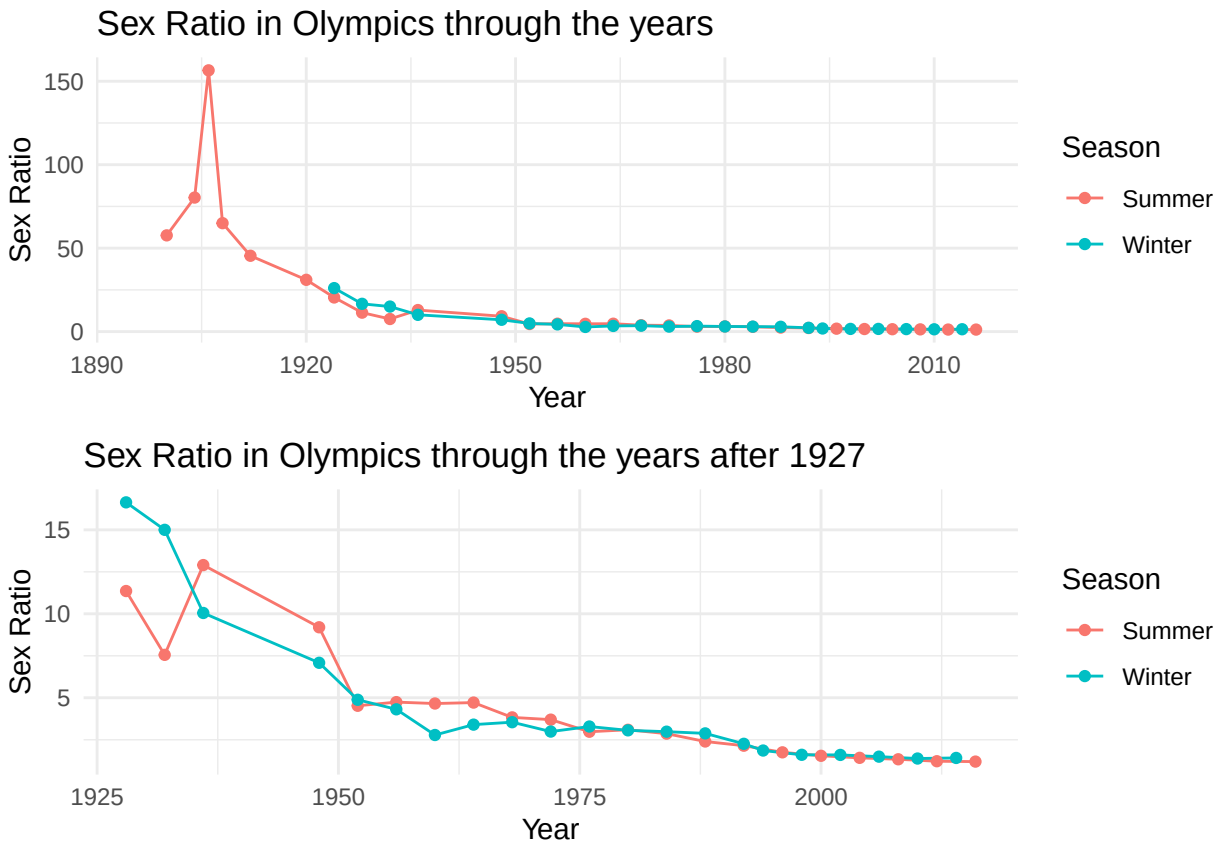
```
mutate(Sex_Ratio = Number_Of_Men/Number_Of_Women)
```

```
## Joining, by = c("Year", "Season")
```

```
p1 <- group %>%
ggplot(aes(x = Year, y= Sex_Ratio, group = Season)) +
geom_line(aes(color = Season)) +
geom_point(aes(color = Season)) +
  labs(x = "Year", y = "Sex Ratio", title = "Sex Ratio in Olympics through the years") +
theme_minimal()
p2 <- group %>%
filter(Year>1927) %>%
ggplot(aes(x = Year, y= Sex_Ratio, group = Season)) +
geom_line(aes(color = Season)) +
geom_point(aes(color = Season)) +
labs(x = "Year", y = "Sex Ratio", title = "Sex Ratio in Olympics through the years after 1927") +
theme_minimal()
cowplot::plot_grid(p1,p2, ncol = 1,
align = 'h', axis = 'l')
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



When the olympics started no women participated in the Olympics. In 1900 women started participating in the olympics. As years passed the sex ratio i.e. the ratio of men to women became smaller. After 2000 the ratio started to move towards 1, which means that the olympics now are more diverse than they used to be, which is great.

Analysis by Age

Age is something might have changed from the olymipics started. Hypothesis: The participants during the1900 of the olympics had a median age greater than the median age of the participants during the 2000's.

Age density plots

```
dataOlympics$Age[is.na(dataOlympics$Age)] <- median(dataOlympics$Age, na.rm = T)

cat("The median age of the athletes in the modern olympics is", median(dataOlympics$Age))
```

```
## The median age of the athletes in the modern olympics is 24
```

```
cat("The median age of the male athletes in the modern olympics is", median(dataOlympics$Age[dataOlympi
```

```
## The median age of the male athletes in the modern olympics is 25
```

```
cat("The median age of the female athletes in the modern olympics is", median(dataOlympics$Age[dataOlym
```

```
## The median age of the female athletes in the modern olympics is 23
```

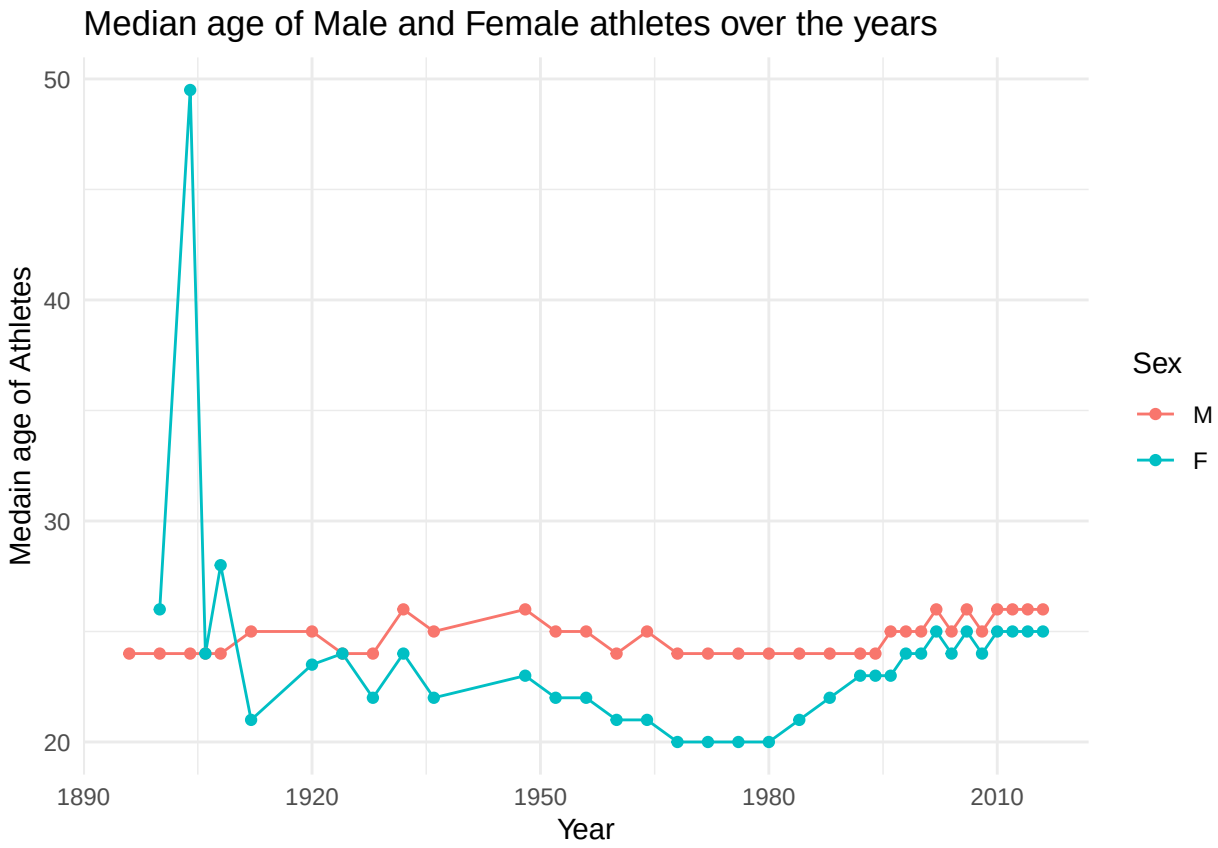
```
# Filling the missing ages with median values.

p1 <- dataOlympics %>%
      ggplot(aes(x = Age)) +
      geom_density(color = "black", fill = "tomato") +
      labs(x = "Age", title = "Distribution of Age") +
      theme_minimal()

p2 <- dataOlympics %>%
      ggplot(aes(x=Age, fill=Sex)) +
      geom_density(alpha=0.4) +
      labs(x = "Age", title = "Distribution of Age by Sex") +
      theme_minimal()

cowplot::plot_grid(p1,p2, ncol = 1,
          align = 'h', axis = 'l')
```

***

Age of athletes over the years.

```
dataOlympics %>%
  group_by(Year, Sex) %>%
  summarise(Median_Age = median(Age)) %>%
  ggplot(aes(x = Year, y = Median_Age, Group = Sex)) +
  geom_line(aes(color = Sex)) +
  geom_point(aes(color = Sex)) +
  labs( x = "Year", y = "Medain age of Athletes", title = "Median age of Male and Female athletes over
  theme_minimal()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```



The median age of men and women participating in the olymipics has increased a bit after the 1980's.

#Analysis by Team: Teams here refer to the countries and the different atletic clubs that have participated in the olympics over the years.

```
cat("The total number of teams that have paricipated in the olympics are", length(unique(dataOlympics$T
```

```
## The total number of teams that have paricipated in the olympics are 1184
```

```
atheletes <- dataOlympics %>%
              left_join(NOCs, by = "NOC")
```

We have now joined athletes dataset with the regions dataset. I would like to analyze the dataset based on the National Olympics Committee rather than the teams.

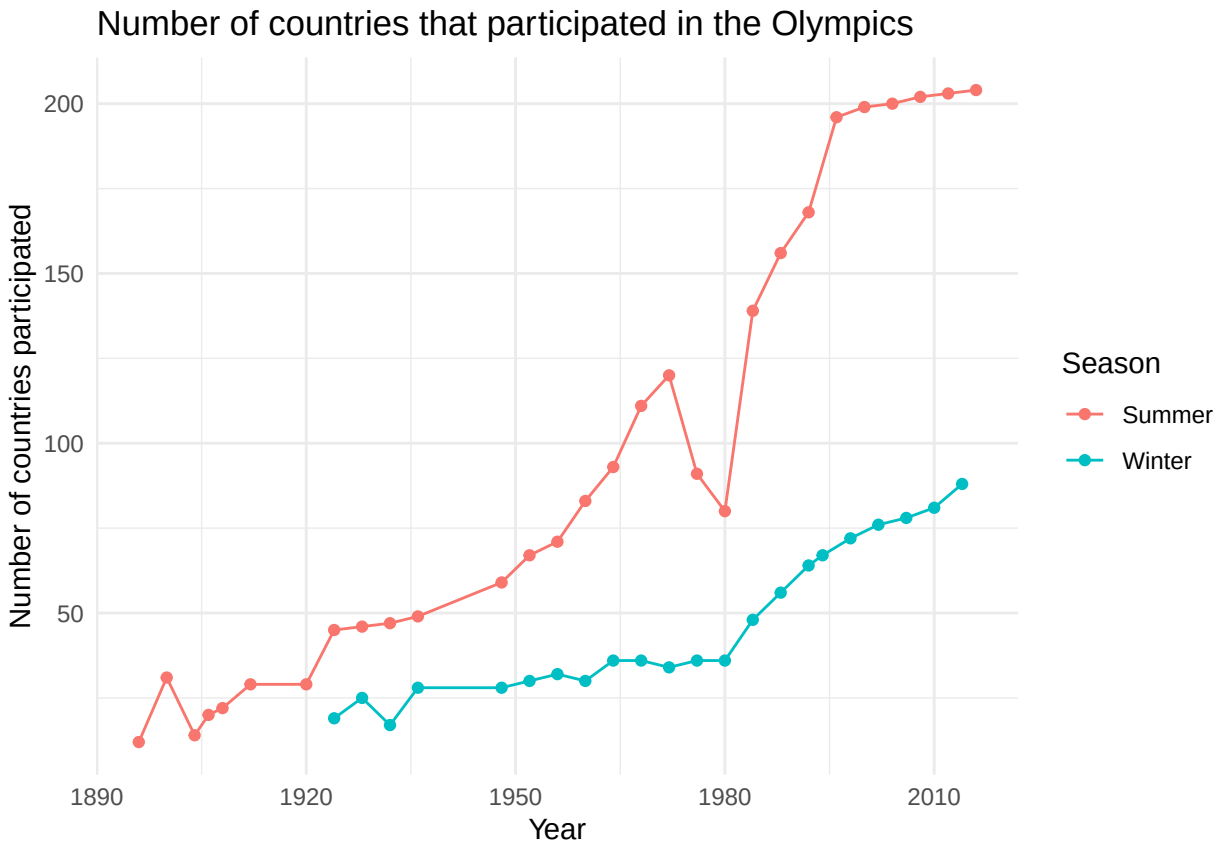
```
  cat("The total number of National Olympics Committees that have paricipated in the olympics are", leng
```

```
## The total number of National Olympics Committees that have paricipated in the olympics are 207
```

#The 1976 and 1980 Olympic Boycott.

```
atheletes %>%
  group_by(Year, Season) %>%
  summarise(NoOfCountries = length(unique(region))) %>%
  ggplot(aes(x = Year, y = NoOfCountries, group = Season)) +
  geom_line(aes(color = Season)) +
  geom_point(aes(color = Season)) +
  labs(x = "Year", y = "Number of countries participated", title = "Number of countries that participat
  theme_minimal()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```



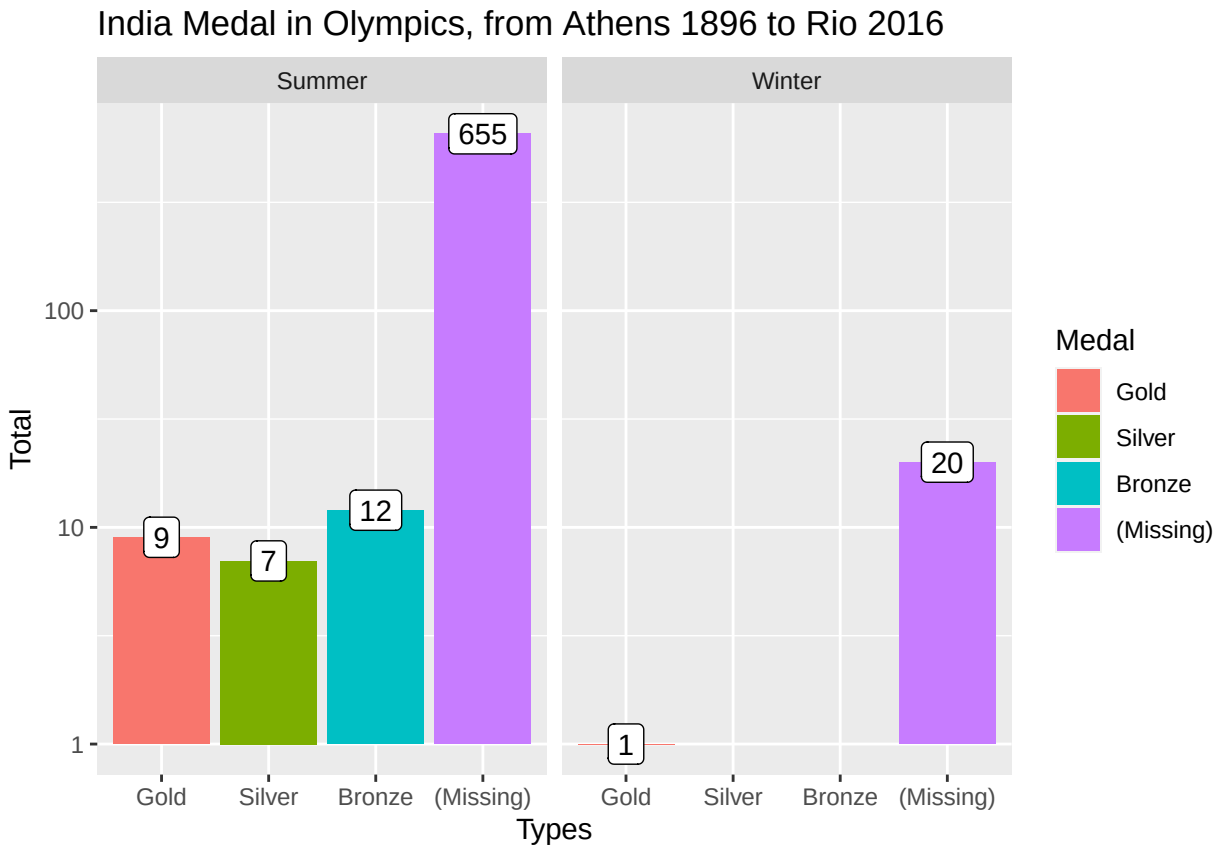
The number of countries that participated in the olymics have seen a steady increase over time. But, in 1976 and 1980 the number has seen a sharp decrease.Due to boycott.

India in OLympics

Filter India's Athletes

```
dataOlympics %>%
filter(Team == 'India') %>%
distinct(Season, Year, Event, Medal) %>%
group_by(Season, Medal = fct_explicit_na(Medal)) %>%
summarize(Total = length(Medal)) %>%
ggplot(aes(Medal, Total)) +
geom_bar(stat = "identity", aes(fill=Medal)) +
ylab("Total") +
xlab("Types") +
ggtitle("India Medal in Olympics, from Athens 1896 to Rio 2016") +
geom_label(aes(label=Total)) +
facet_grid(~Season) +
scale_y_continuous(trans = "log10")
```

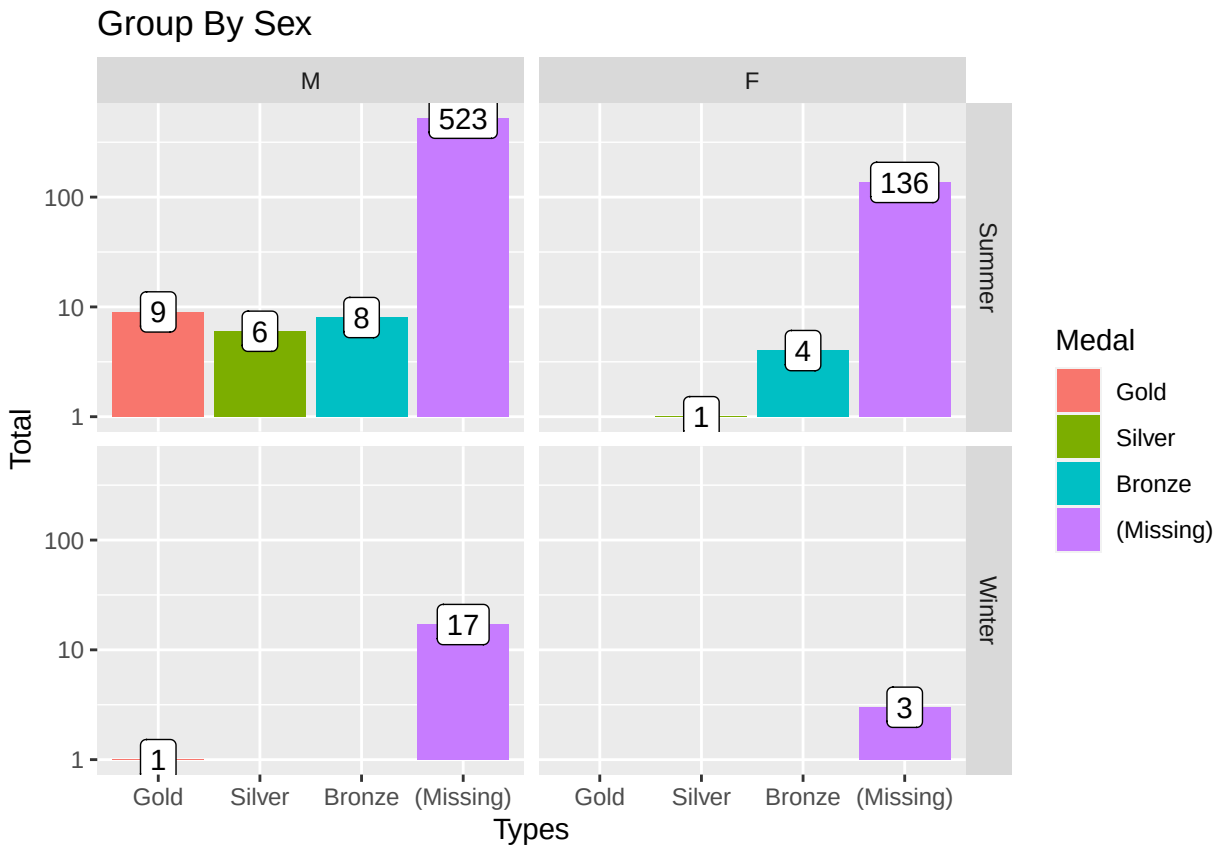
```
## `summarise()` has grouped output by 'Season'. You can override using the `.groups` argument.
```



Men x Women participation

```
dataOlympics %>%
filter(Team == 'India') %>%
distinct(Season, Year, Event, Sex, Medal) %>%
group_by(Season, Sex, Medal = fct_explicit_na(Medal)) %>%
summarize(Total = length(Medal)) %>%
ggplot(aes(Medal, Total)) +
geom_bar(stat = "identity", aes(fill=Medal)) +
ylab("Total") +
xlab("Types") +
ggtitle("Group By Sex") +
geom_label(aes(label=Total)) +
facet_grid(Season ~ Sex) +
scale_y_continuous(trans = "log10")
```

```
## `summarise()` has grouped output by 'Season', 'Sex'. You can override using the `.groups` argument.
```

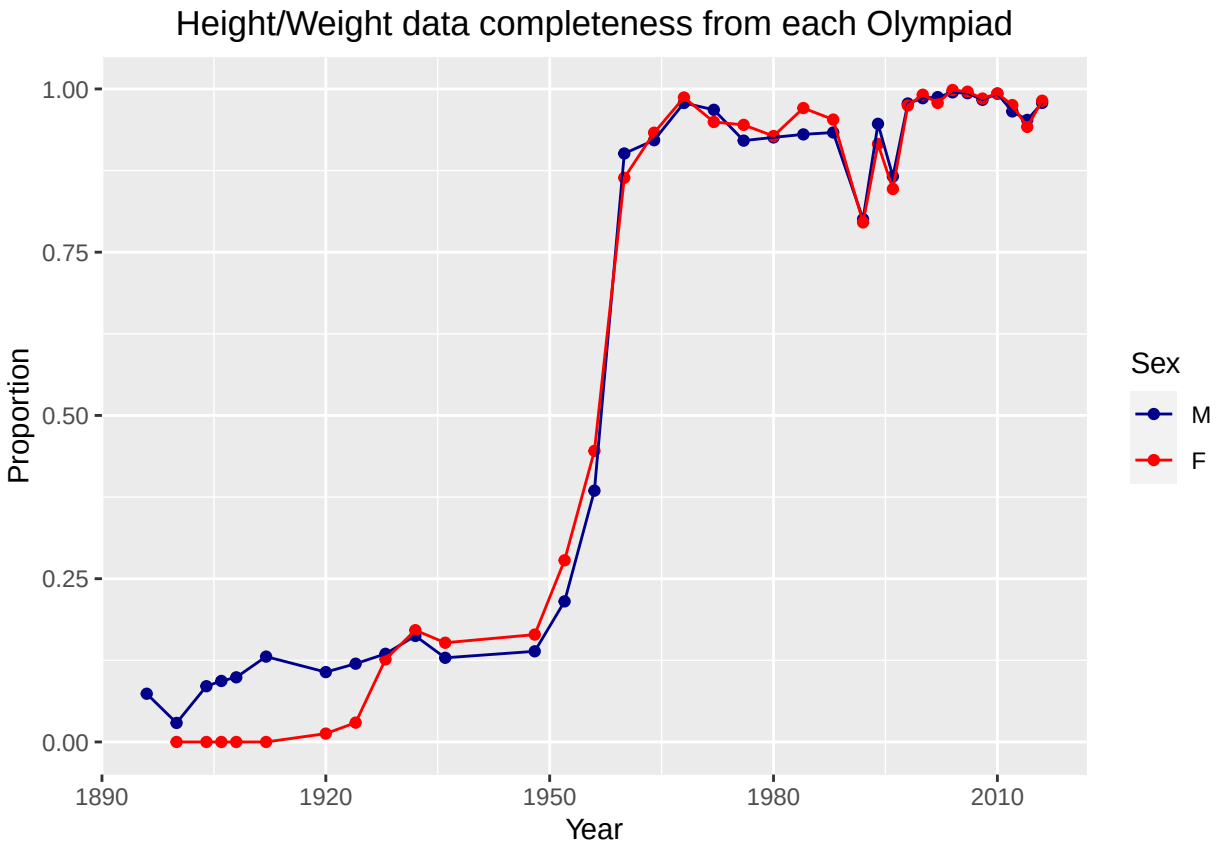


Height and weight of athletes

Data completeness

```
# Check data availability
dataOlympics %>% group_by(Year, Sex) %>%
  summarize(Present = length(unique(ID[which(!is.na(Height) & !is.na(Weight))])),
            Total = length(unique(ID))) %>%
  mutate(Proportion = Present/Total) %>%
  ggplot(aes(x=Year, y=Proportion, group=Sex, color=Sex)) +
  geom_point() +
  geom_line() +
  scale_color_manual(values=c("darkblue","red"))  +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(title="Height/Weight data completeness from each Olympiad")
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```



There was a dramatic increase in data completeness starting in 1960, reaching 86% for women and 90% for men. For all of the Games after this point, data completeness remained above 85% except for 1992, where completeness dips down to 80% for unclear reasons.

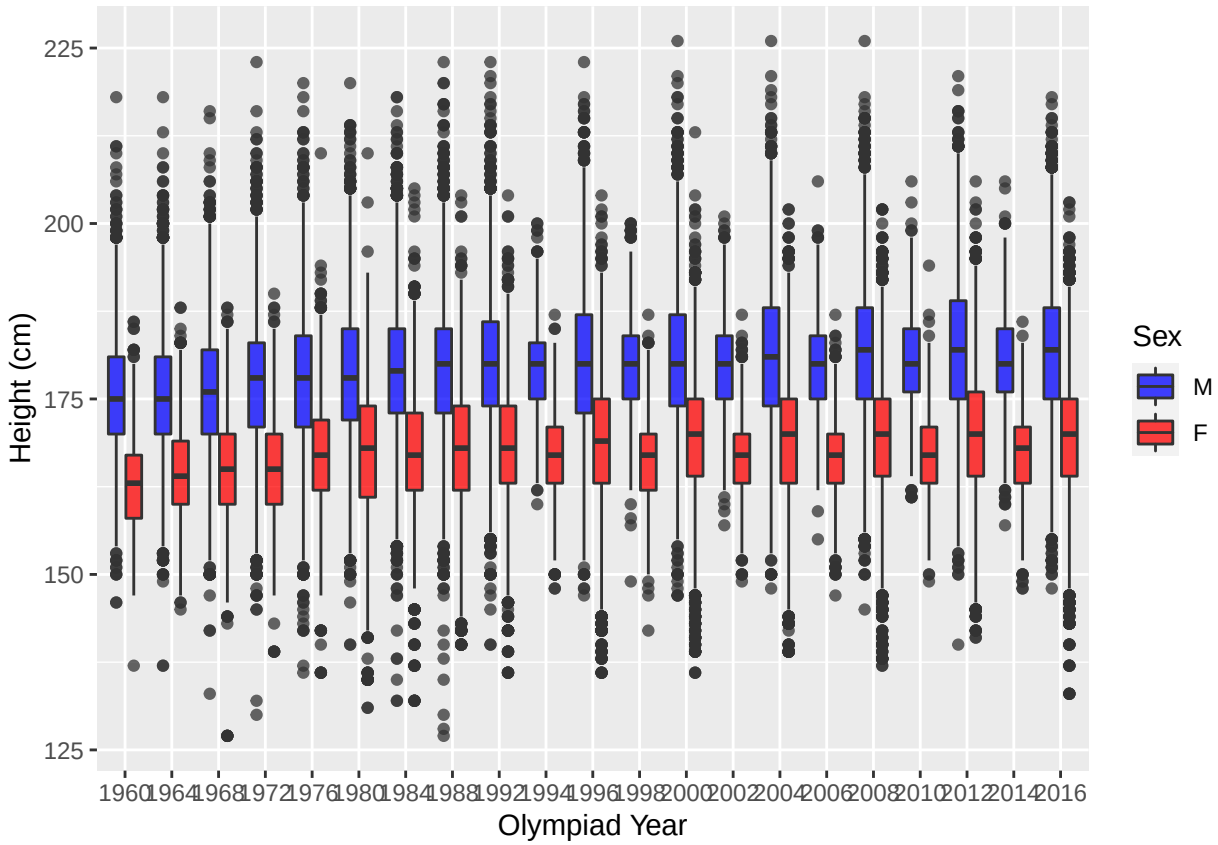
In light of this, We limited the remainder of this data exploration to Games from 1960 onward, which includes a total of 15 Olympiads spread over a 56 year period.

```
# Remove missing Height/Weight data and limit to years from 1960 onward
dataOlympics <- dataOlympics%>% filter(!is.na(Height), !is.na(Weight), Year > 1959)
```

The next two plots show trends in the heights and weights of Olympic athletes over time, with the data grouped by sex.

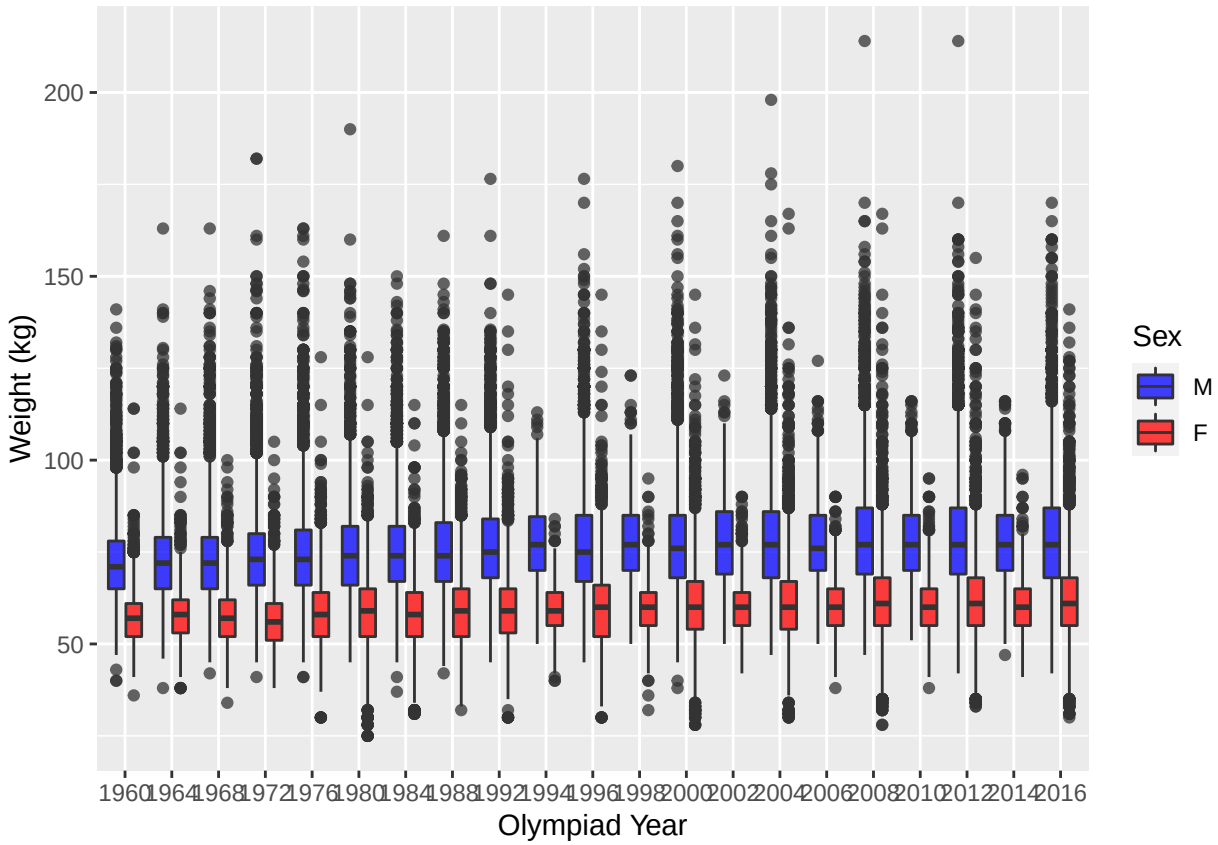
Athlete height over time

```
dataOlympics%>% ggplot(aes(x=as.factor(Year), y=Height, fill=Sex)) +
  geom_boxplot(alpha=0.75) +
  xlab("Olympiad Year") + ylab("Height (cm)") +
  scale_fill_manual(values=c("blue","red"))
```



Athlete weight over time

```
dataOlympics %>% ggplot(aes(x=as.factor(Year), y=Weight, fill=Sex)) +
  geom_boxplot(alpha=0.75) +
  xlab("Olympiad Year") + ylab("Weight (kg)") +
  scale_fill_manual(values=c("blue","red"))
```



These plots show that for both men and women, height and weight has increased gradually over the history of the Games. However, these plots could be hiding important variation since different body types are favored in different events.

In most sports this means taller and heavier, but in a few sports such as gymnastics, athletes have become smaller.
