OLYMPICS.VISUALIZATION

Aisharriya Dasgupta

2024-09-24

```
setwd("C:\\Users\\aisharriya\\Desktop\\SULAXAN SIR")
a <- read.csv("athlete_events.csv")
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(tidyr)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
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
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.3.3
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.3.3
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(treemap)
```

```
## Warning: package 'treemap' was built under R version 4.3.3
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(gganimate)
```

```
## Warning: package 'gganimate' was built under R version 4.3.3
```

```
library(dplyr)
library(gifski)
```

```
## Warning: package 'gifski' was built under R version 4.3.3
```

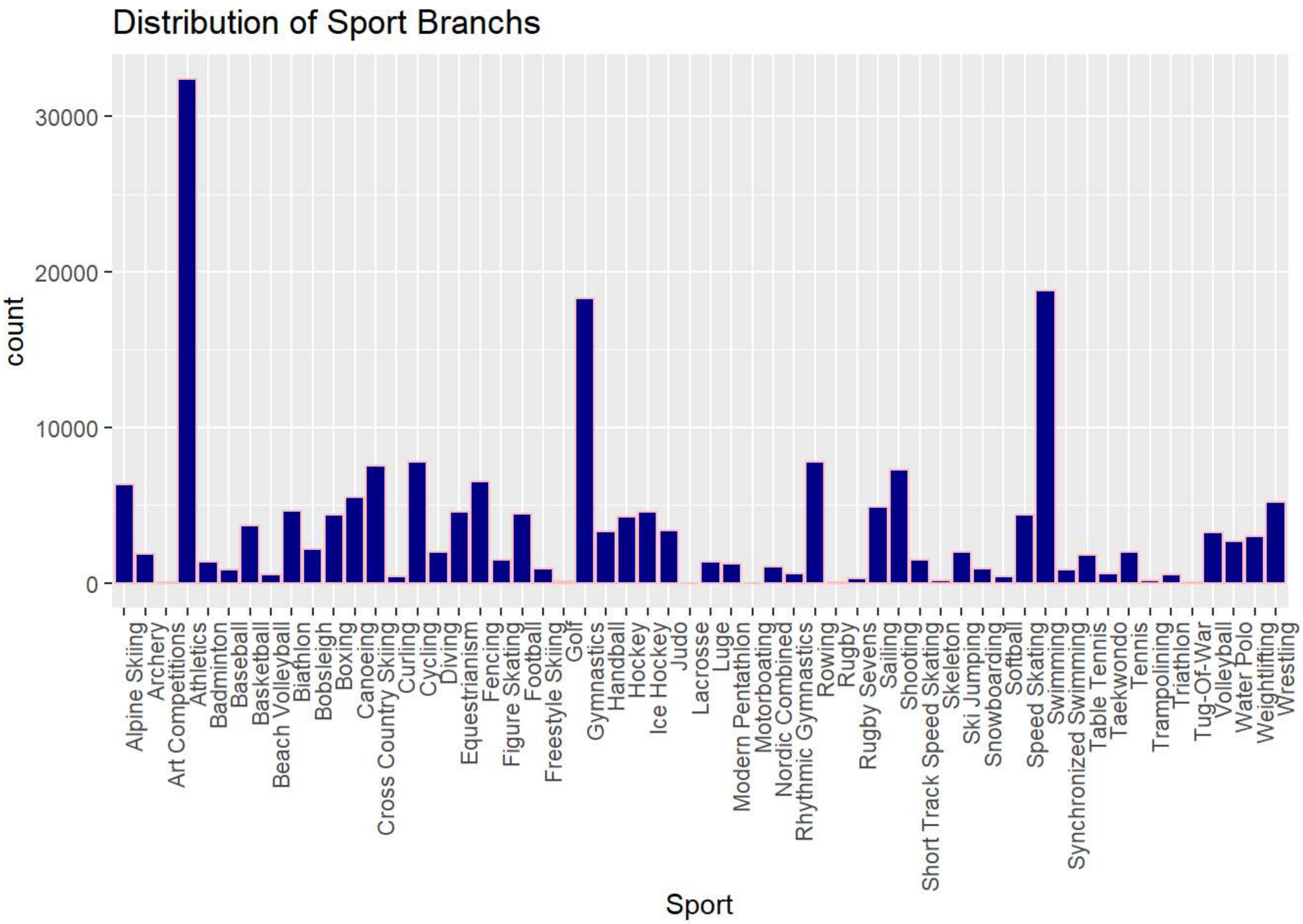
```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.3.3
```

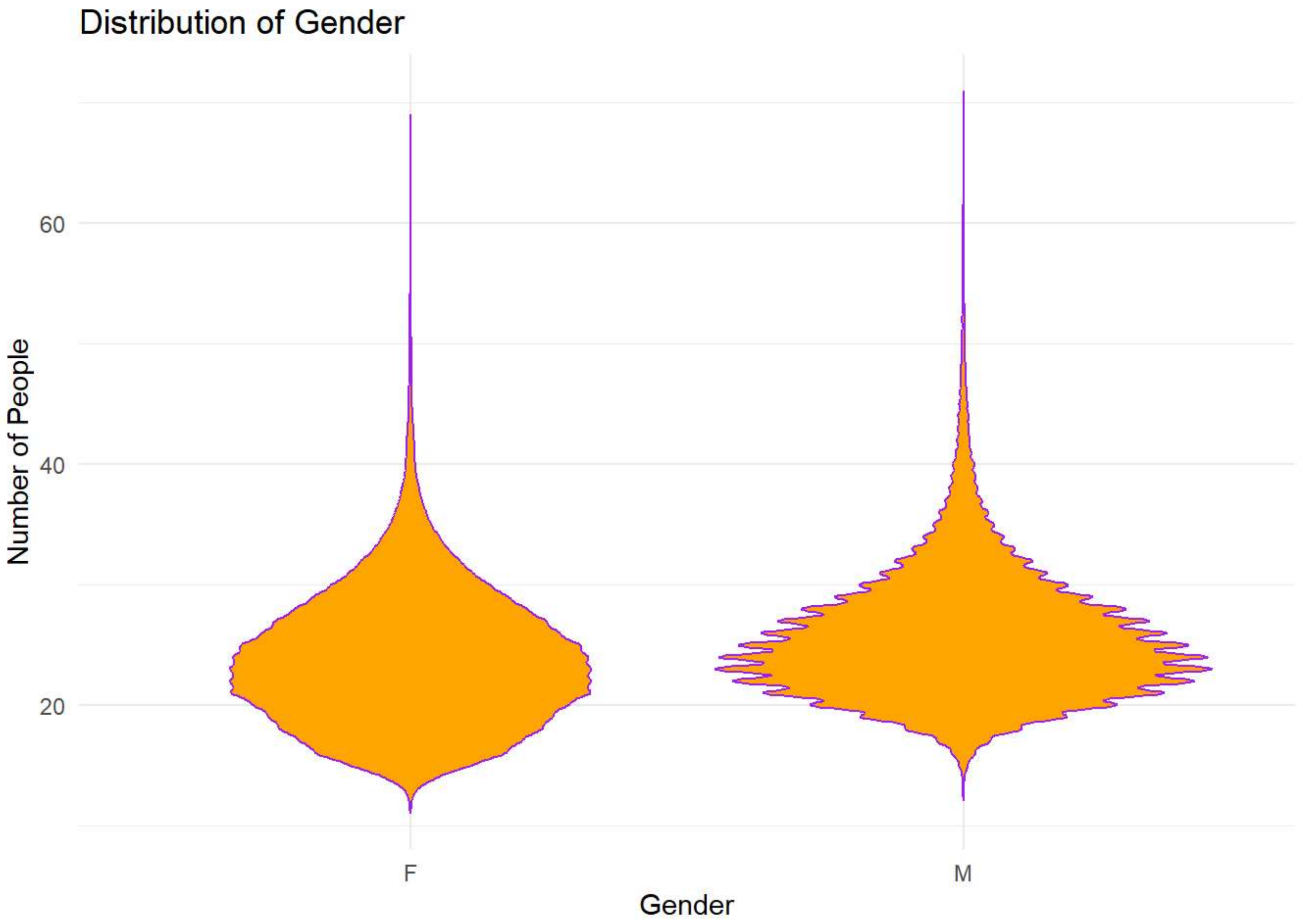
```
## Loading required package: RColorBrewer
```

```
na <- subset(a, !is.na(Height+Weight+Age))
na$Medal <- as.character(na$Medal)
na$Medal[is.na(na$Medal)] <- "No Medal"
```

```
 # 1. BAR PLOT
#Distribution of sport branches

p<-ggplot(na, aes(x = `Sport`))+
  geom_bar(color="pink", fill="darkblue")+
  ggtitle("Distribution of Sport Branchs") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
p
```

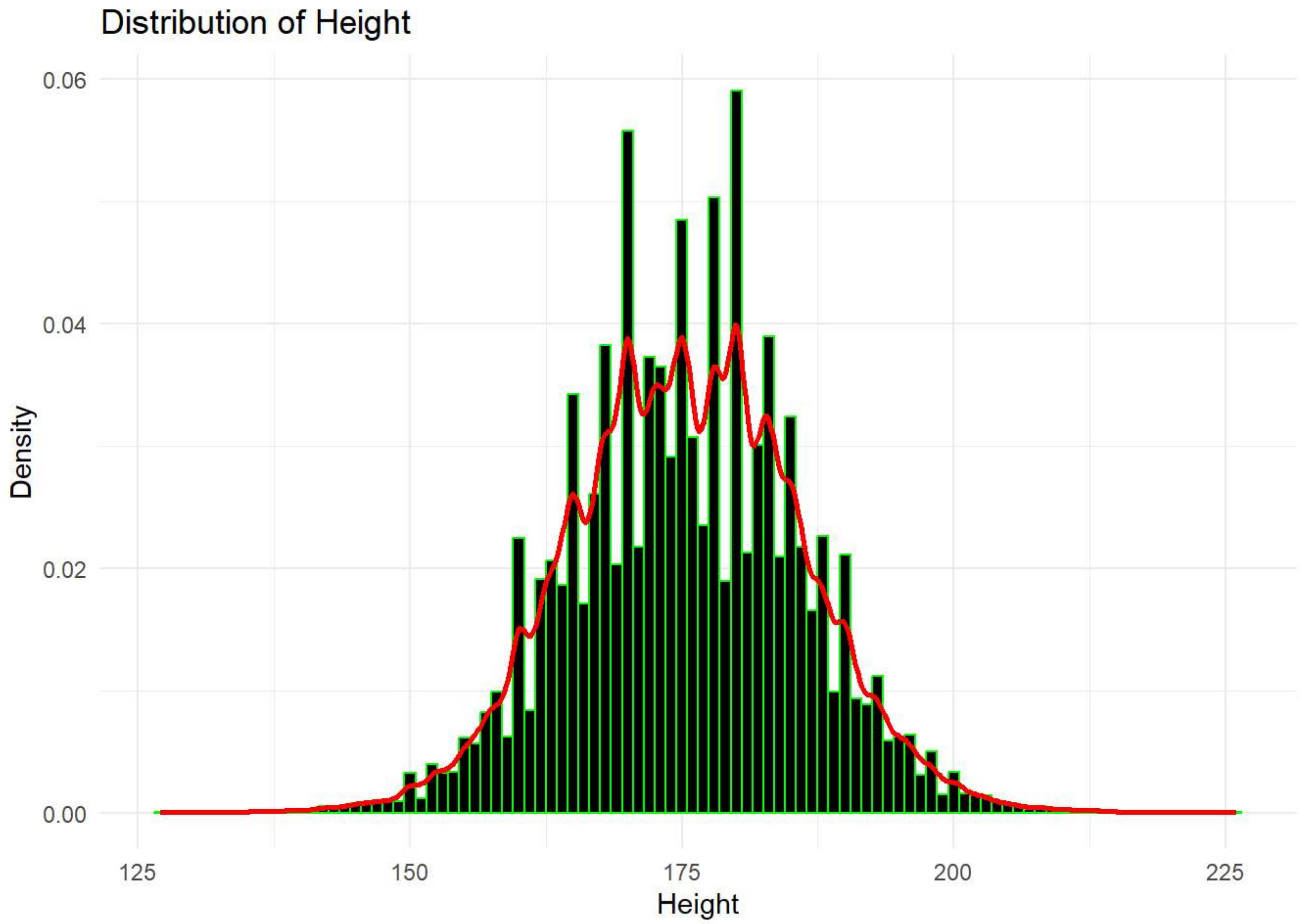


```
# 2.VIOLIN PLOT
#Distribution of sex
ggplot(na, aes(x=Sex, y=Age, fill=Sex)) +
  geom_violin(fill="orange", color="purple") +
  labs(x="Gender",
       y="Number of People",
       title="Distribution of Gender") +
  theme_minimal()
```

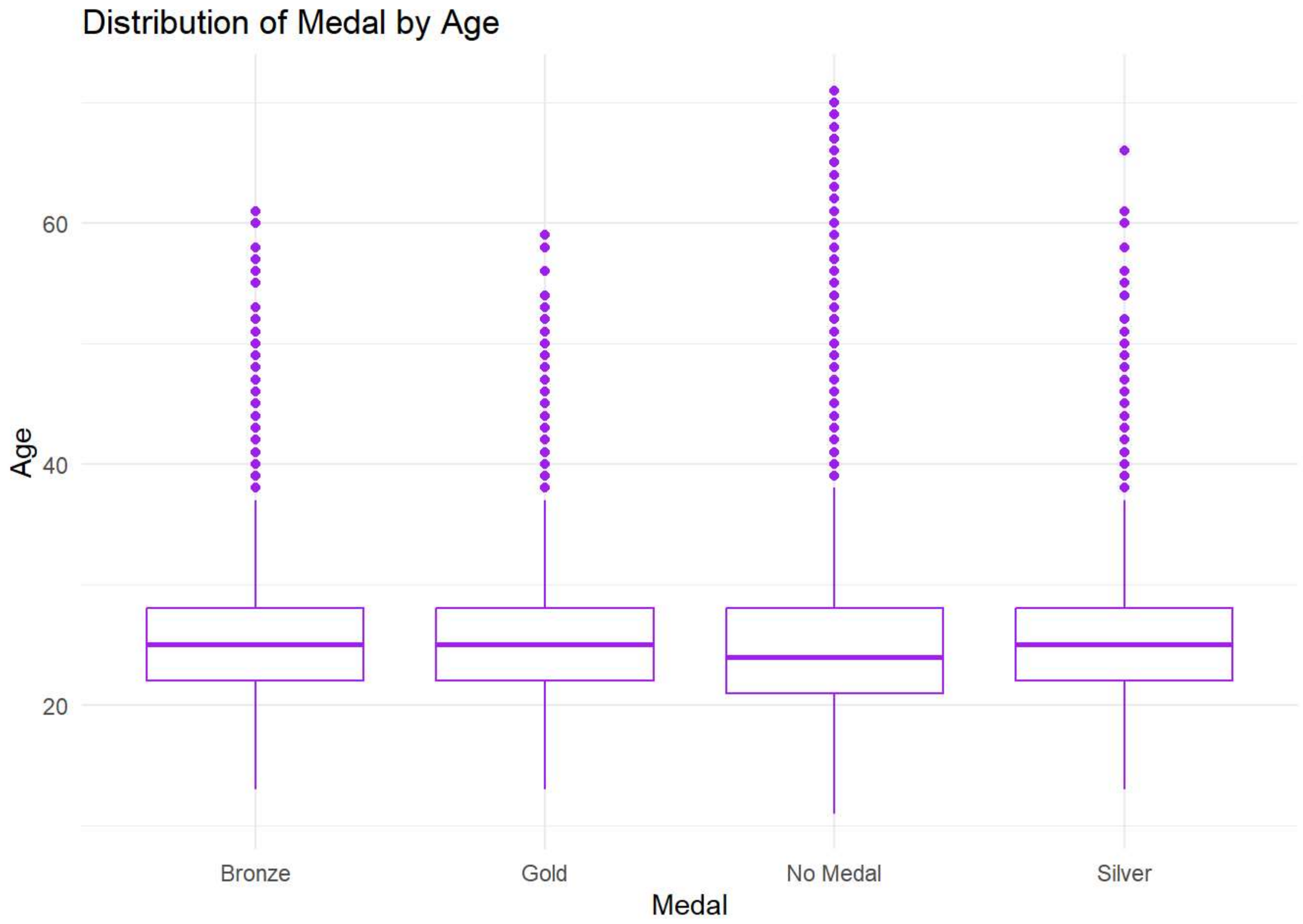


```
# 3.KDENSITY PLOT
#Distribution of heights
ggplot(na, aes(x=Height)) +
  geom_histogram(aes(y=after_stat(density)), binwidth=1, fill="black", color="green") +
  geom_density(color="red", size=1) +
  labs(x="Height",
       y="Density",
       title="Distribution of Height") +
  theme_minimal()
```

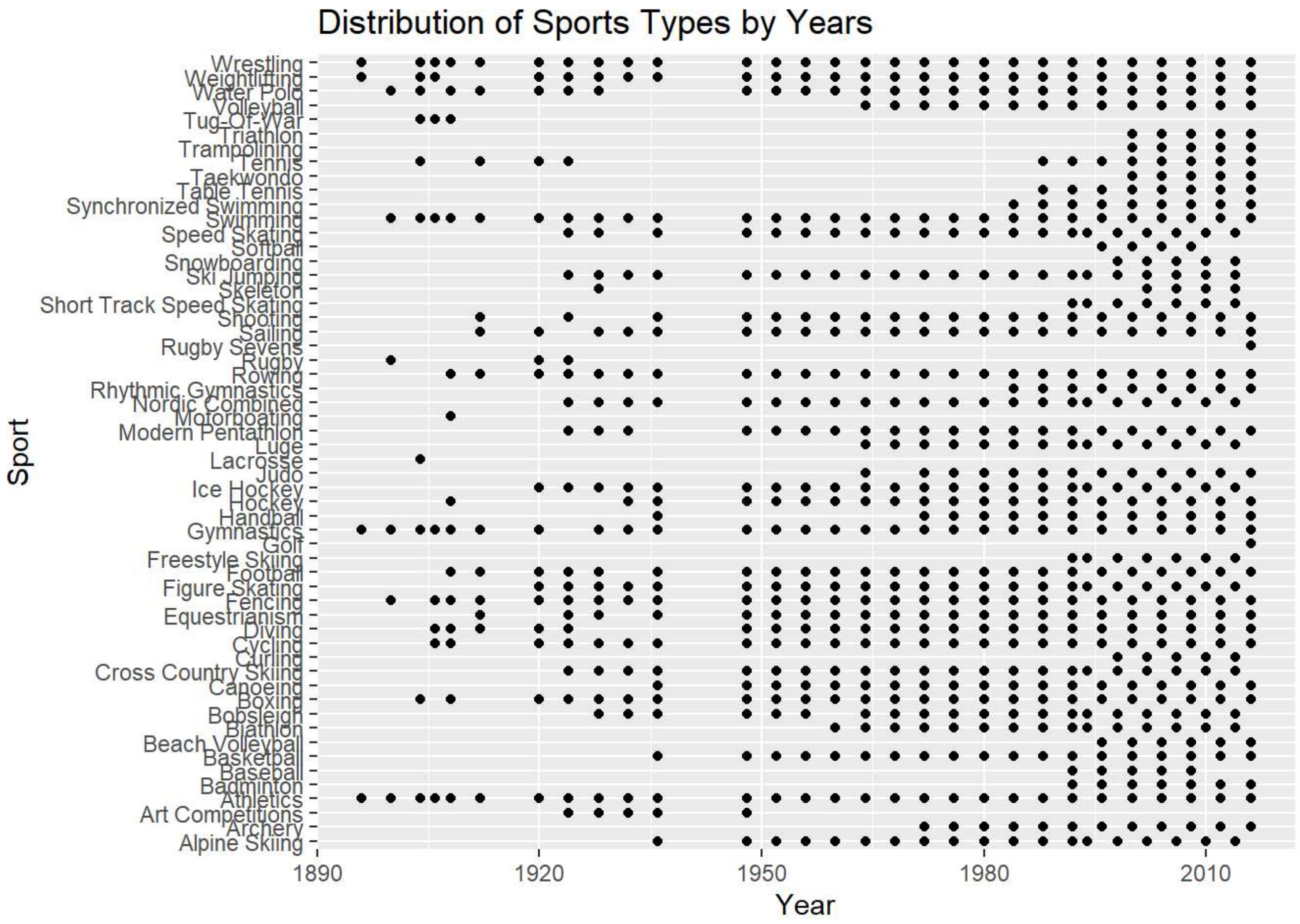
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# 4. BOXPLOT
#Distribution of medal and ages
ggplot(na, aes(x=Medal, y=Age)) +
  geom_boxplot(color="purple") +
  labs(x="Medal",
       y="Age",
       title="Distribution of Medal by Age") +
  theme_minimal()
```



```
# 5. POINT GRAPH
#Distribution of olympic games over years
ggplot(data = na) +
  aes(x = Year) +
  aes(y = `Sport`) +
  geom_point() +
  scale_color_manual(values = c("red", "yellow")) +
  labs(col = "") +
  labs(title = "Distribution of Sports Types by Years")
```

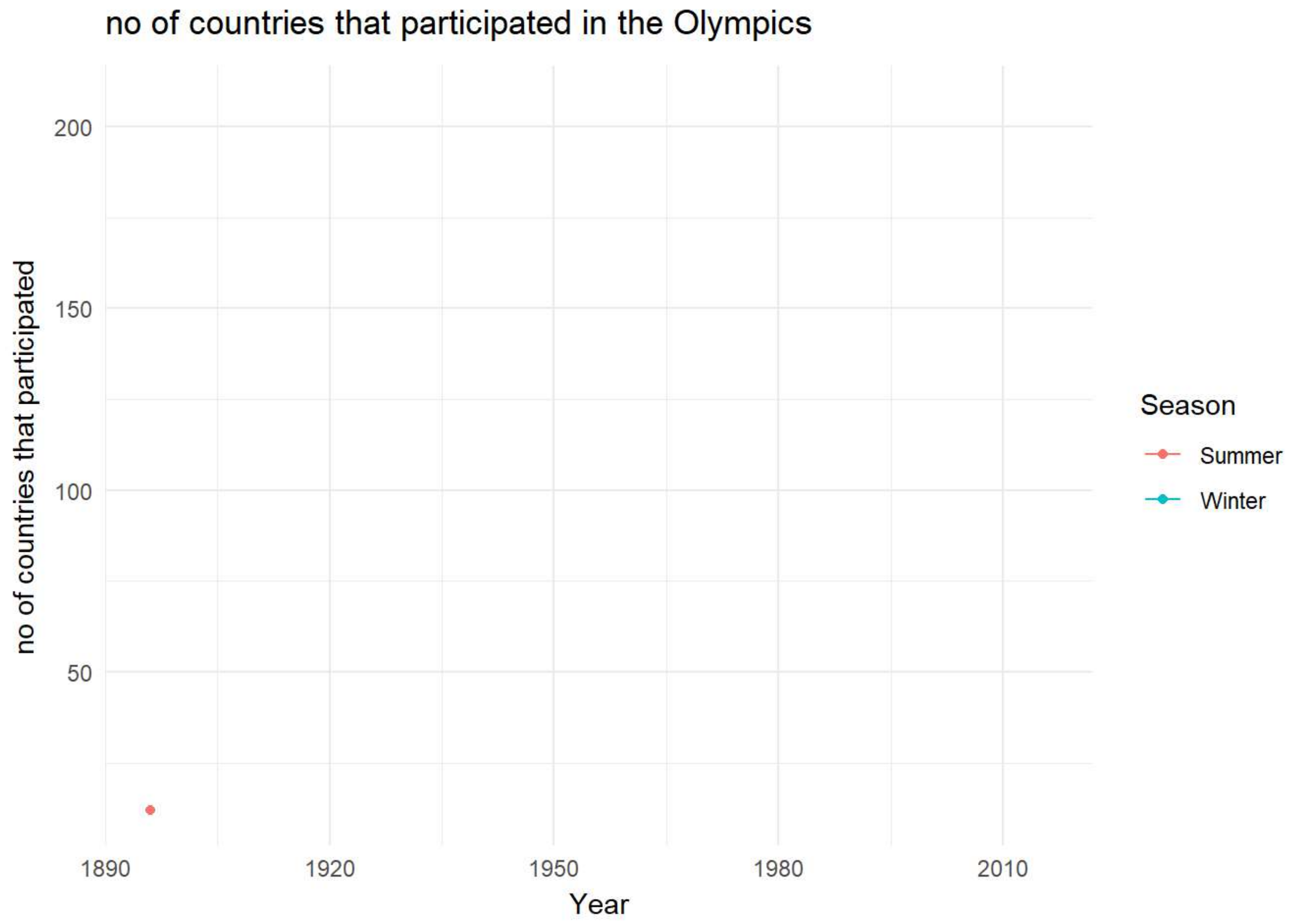


```
# 6. LINE GRAPH
#Distribution of olympics participants over season
s<-a %>%
  group_by(Year, Season) %>%
  summarise(NoOfCountries = length(unique(NOC))) %>%
  ggplot(aes(x = Year, y = NoOfCountries, group = Season)) +
  geom_line(aes(color = Season)) +
  geom_point(aes(color = Season)) +
  labs(x = "Year", y = "no of countries that participated", title = "no of countries that partic
ipated in the Olympics") +
  theme_minimal()+
  transition_reveal(Year)
```

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

```
s
```

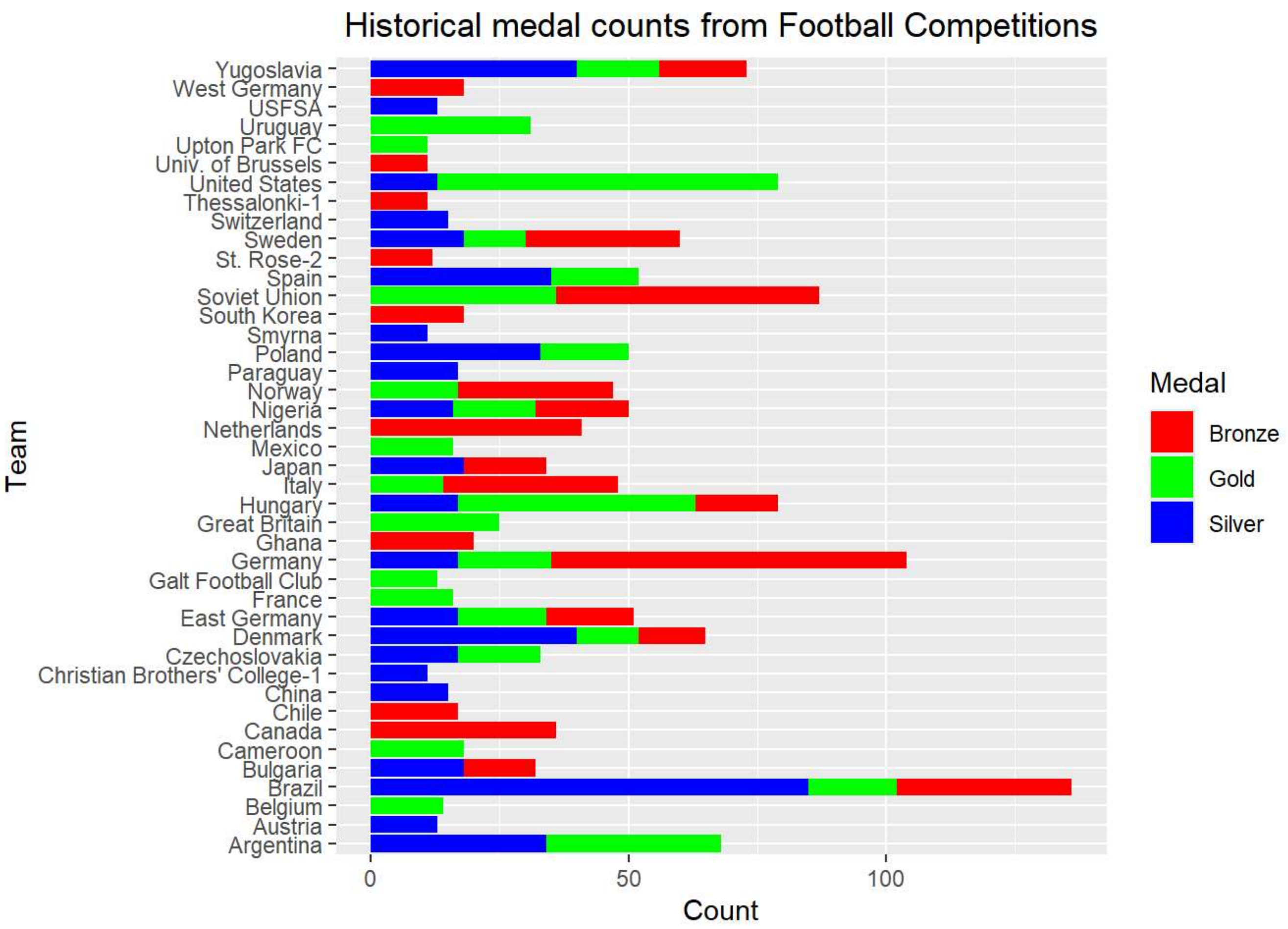
```
## `geom_line()`: Each group consists of only one observation.
## ℹ Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## ℹ Do you need to adjust the group aesthetic?
```



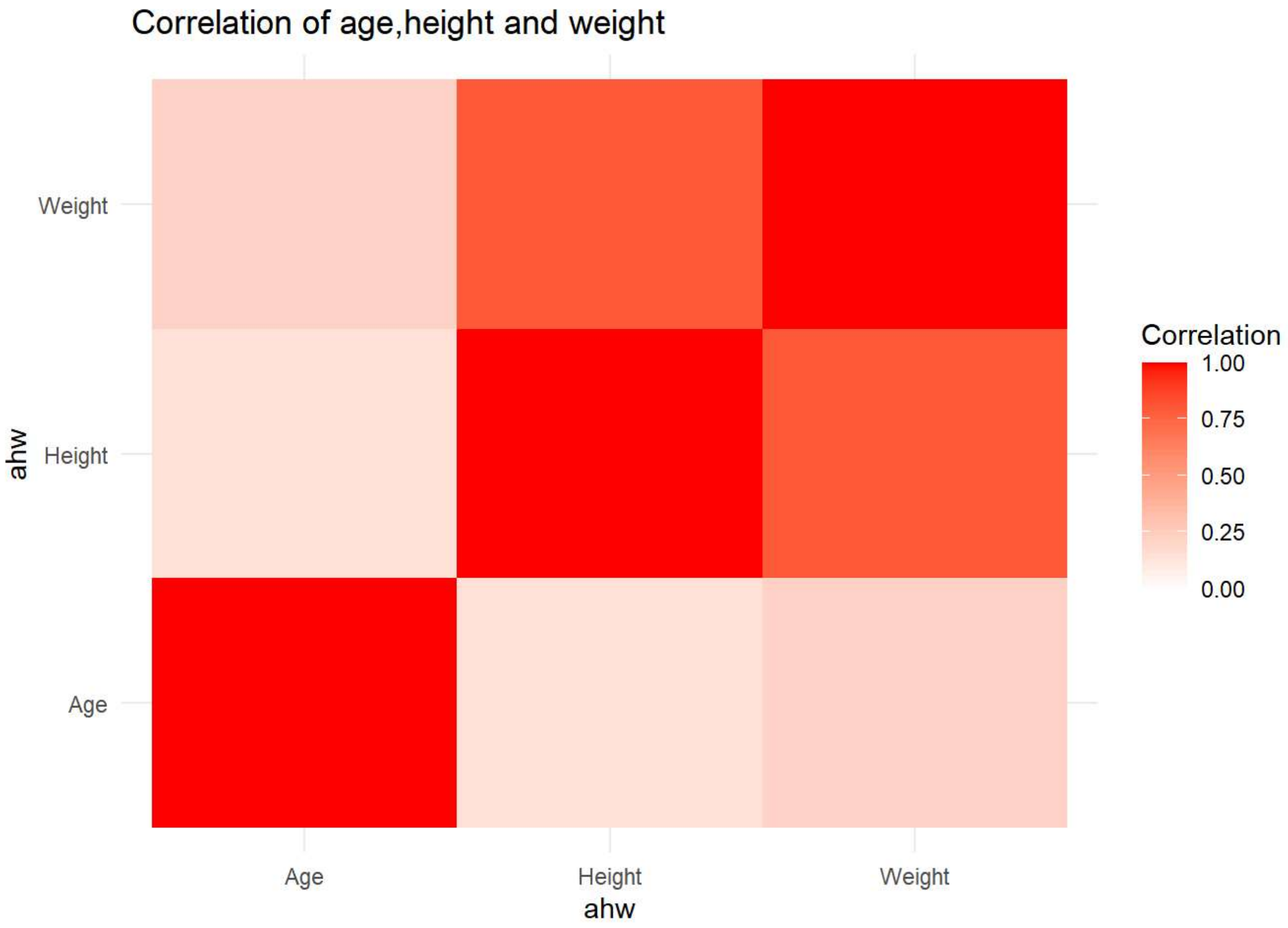
```
#7. COMBINED BAR PLOT
#Distribution of medal counts and types over countries
ftbl <- a %>%
  filter(Sport == "Football") %>%
  select(Name, Sex, Age, Team, NOC, Year, City, Event, Medal)

# Count Events, Nations, and Football competitions each year
counts_ftbl <- ftbl %>% filter(Team != "Unknown") %>%
  group_by(Year) %>%
  summarize(
    Events = length(unique(Event)),
    Nations = length(unique(Team)),
    Footballs = length(unique(Name))
  )

# count number of medals awarded to each Team
medal_counts_ftbl <- ftbl %>% filter(!is.na(Medal))%>%
  group_by(Team, Medal) %>%
  summarize(Count=length(Medal))
```

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

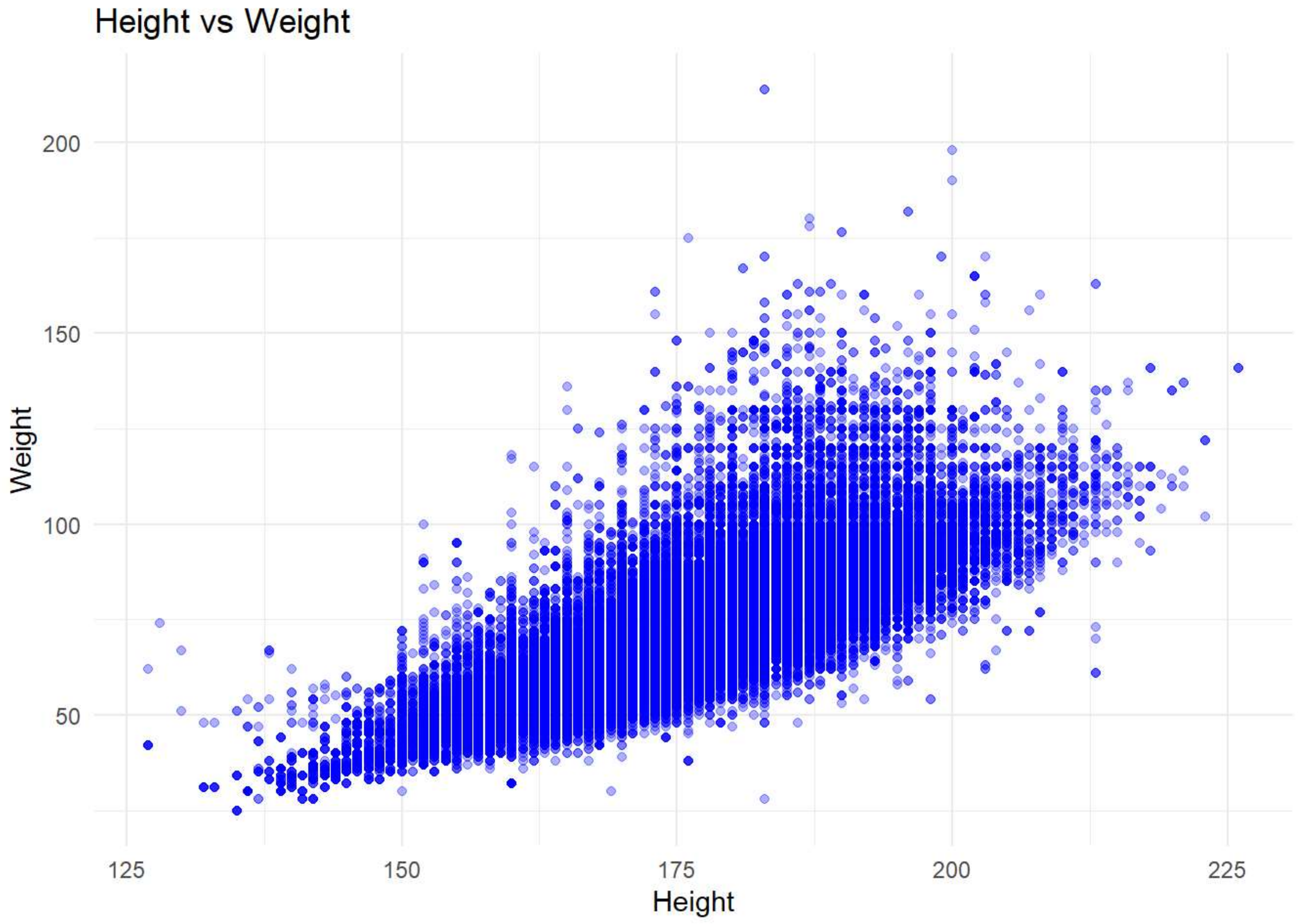
```r
#plot
ggplot(medal_counts_ftbl, aes(x=Team, y=Count, fill=Medal)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values=c("red","green","blue")) +
  ggtitle("Historical medal counts from Football Competitions") +
  theme(plot.title = element_text(hjust = 0.5))
```



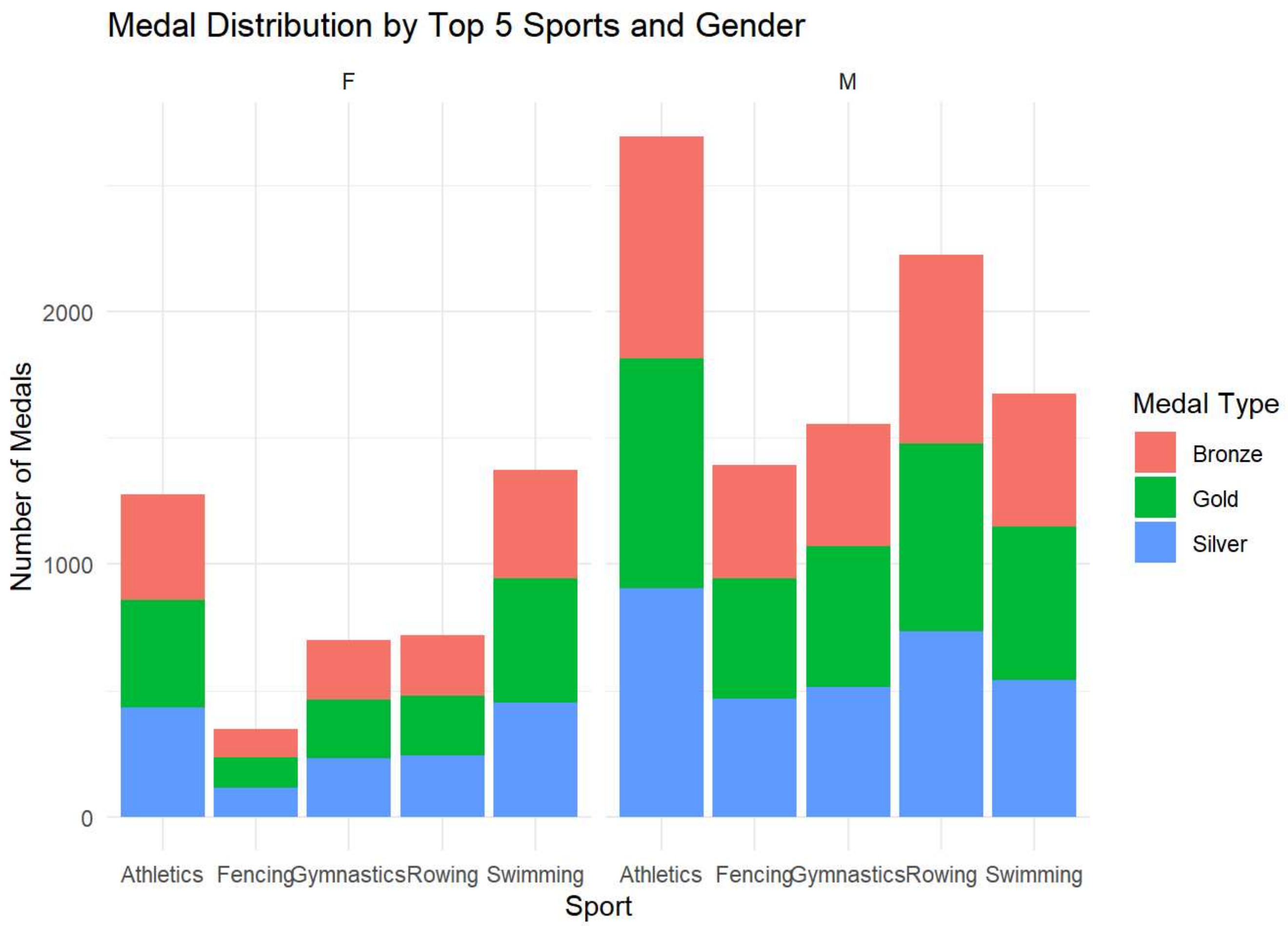
```
# 8.HEATMAP
# Subset the relevant columns and remove rows with missing values
na <- na.omit(na[, c("Age", "Height", "Weight")])

# Calculate the correlation matrix
cor_data <- cor(na)

# Melt the correlation matrix into long format
melted_cor <- melt(cor_data)



ggplot(melted_cor, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low="blue", high="red", mid="white",
                       midpoint=0, limit=c(0,1), space="Lab",
                       name="Correlation") +theme_minimal() +
  labs(title=" Correlation of age,height and weight",x="ahw",
       y="ahw")
```



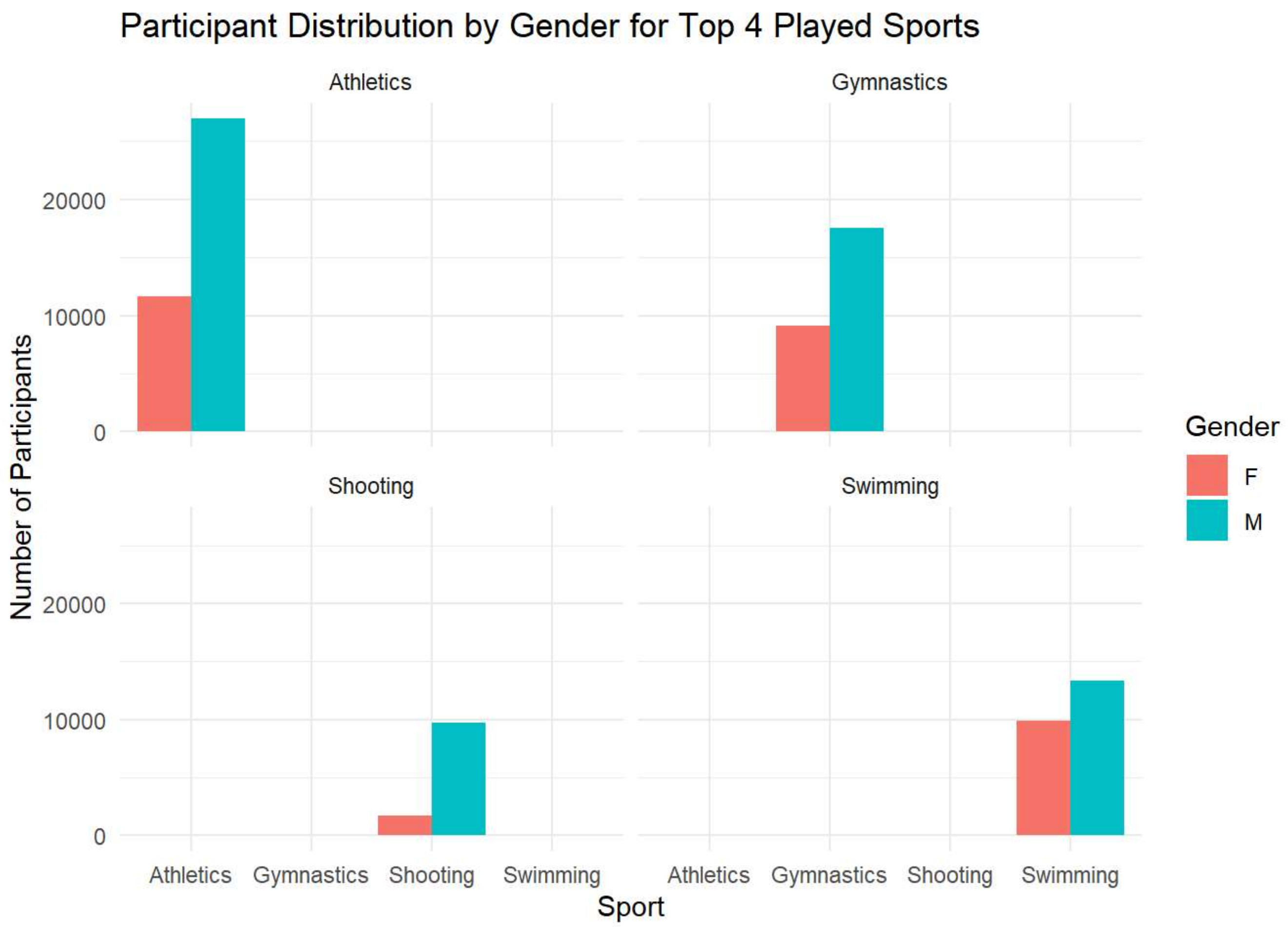
```
# 9.SCATTERPLOT
ab<-ggplot(na, aes(x=Height, y=Weight)) +
  geom_point(alpha=0.3, color="blue") +
  labs(x="Height", y="Weight", title="Height vs Weight") +
  theme_minimal()
ab
```



```
#10.OVERLAY GRAPH 1

medal_data <- a %>%
  filter(!is.na(Medal))

top_sports <- medal_data %>%
  group_by(Sport) %>%
  summarise(Medal_Count = n()) %>%
  arrange(desc(Medal_Count)) %>%
  slice(1:5)

medal_data_top_5 <- medal_data %>%
  filter(Sport %in% top_sports$Sport)

ggplot(medal_data_top_5, aes(x = Sport, fill = Medal)) +
  geom_bar() +
  facet_grid(.~Sex) +  # Facet by sex on rows
  labs(title = "Medal Distribution by Top 5 Sports and Gender",
       x = "Sport",
       y = "Number of Medals",
       fill = "Medal Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "top") +
  theme_minimal()
```



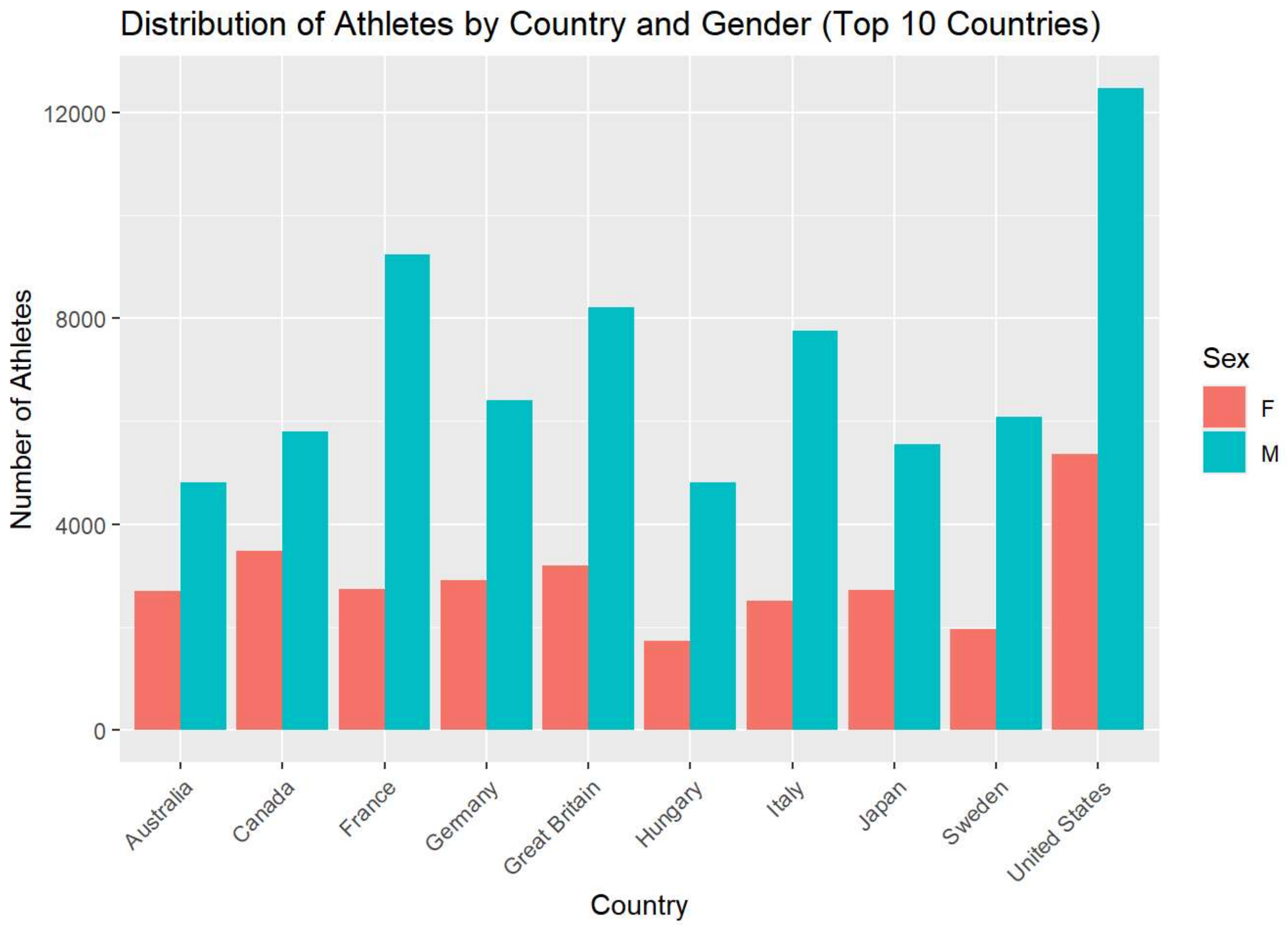
```
#11. OVERLAY GRAPH 2
# Count participants by sport
top_sports <- a %>%
  group_by(Sport) %>%
  summarise(Participant_Count = n()) %>%
  arrange(desc(Participant_Count)) %>%
  slice(1:4)  # Select top 4 sports

# Filter the original data for these top 4 sports
top_sports_data <- a %>%
  filter(Sport %in% top_sports$Sport)

# Create the faceted bar plot
ggplot(top_sports_data, aes(x = Sport, fill = Sex)) +
  geom_bar(position = "dodge") +
  facet_wrap(~Sport) +  # Facet by sport
  labs(title = "Participant Distribution by Gender for Top 4 Played Sports",
       x = "Sport",
       y = "Number of Participants",
       fill = "Gender") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "top") +
  theme_minimal()
```



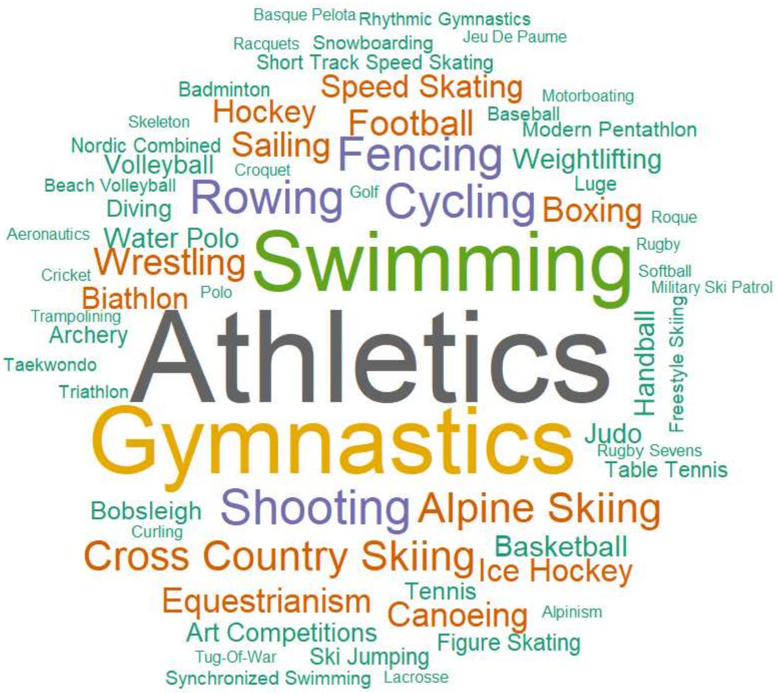
```
#12. DODGE
top_countries <- a %>%
  group_by(Team) %>%
  summarise(Athlete_Count = n()) %>%
  top_n(10, Athlete_Count)

country_gender_count <- a %>%
  filter(Team %in% top_countries$Team) %>%
  group_by(Team, Sex) %>%
  summarise(Athlete_Count = n())
```

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

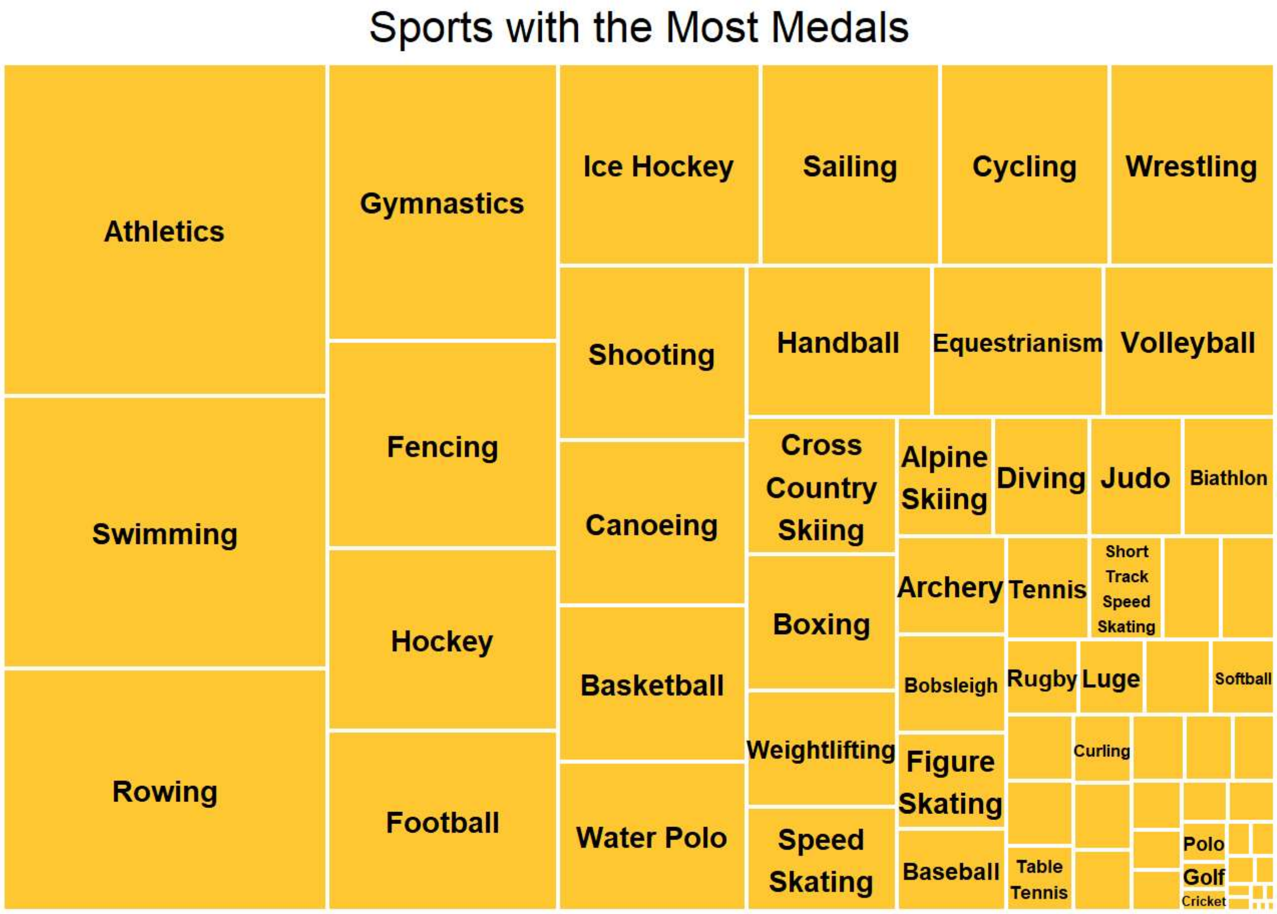
```
ggplot(country_gender_count, aes(x = Team, y = Athlete_Count, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of Athletes by Country and Gender (Top 10 Countries)",
       x = "Country", y = "Number of Athletes") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



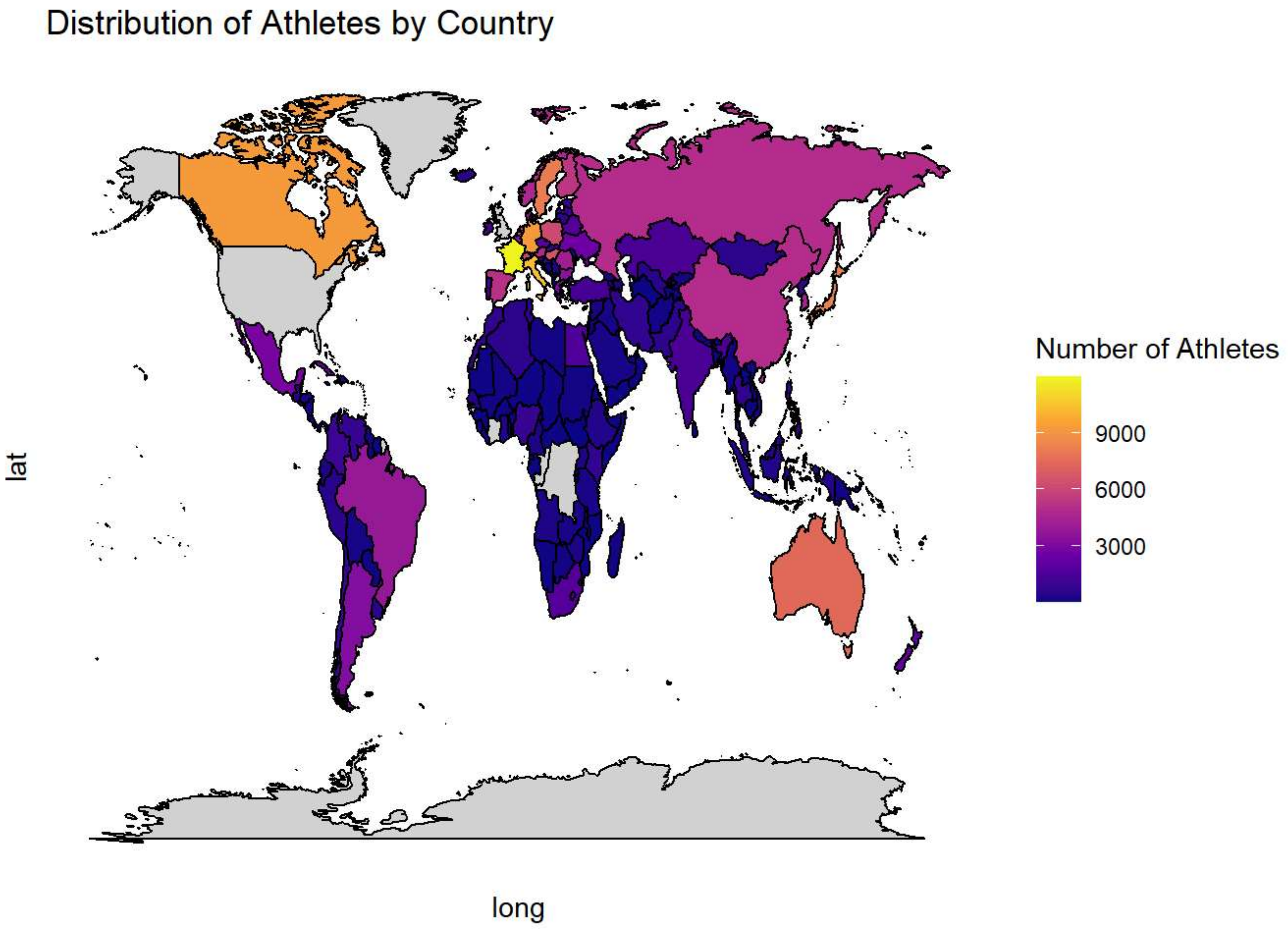
```
#13. WORDCLOUD For max played sports
sports_count <- a %>%
  group_by(Sport) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  arrange(desc(Count))

set.seed(123)
wordcloud(words = sports_count$Sport,
          freq = sports_count$Count,
          scale = c(4, 0.5),
          min.freq = 1,
          max.words = 100,
          random.order = FALSE,
          colors = brewer.pal(8, "Dark2"))
```



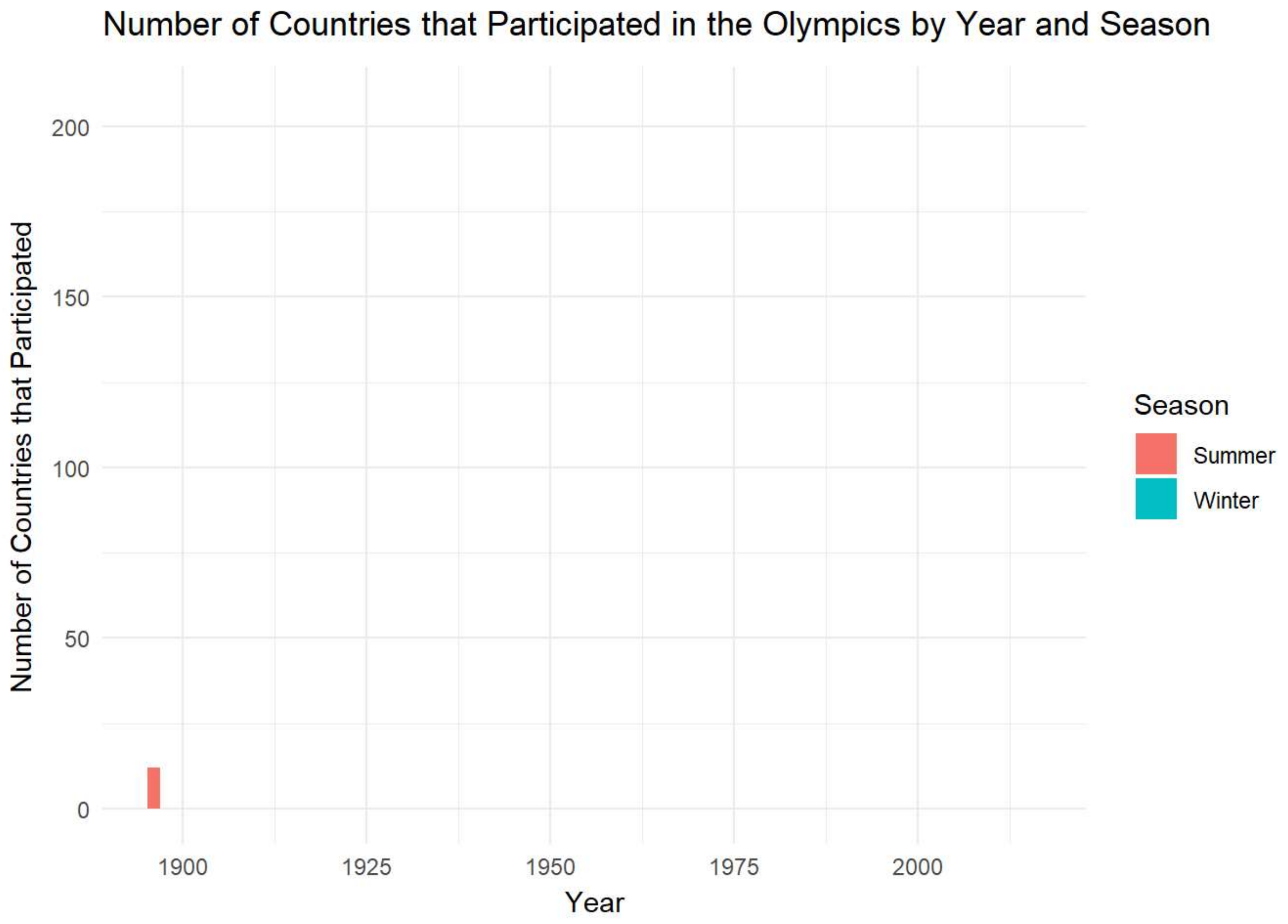
```
#.14 TREEMAP
# Count medals by sport
medal_count <- a %>%
  filter(Medal == "Gold") %>%
  group_by(Sport) %>%
  summarise(Medal = n(), .groups = 'drop') %>%
  arrange(desc(Medal))

#  Treemap of sports with the most medals

treemap(medal_count,
        index = "Sport",
        vSize = "Medal",
        title = "Sports with the Most Medals",
        palette = "#FFCC33",
        border.col = "white",
        fontsize.title = 16)
```



```
#15 MAP
athlete_by_country <- a %>%
  group_by(Team) %>%
  summarise(Athlete_Count = n())
world_map <- map_data("world")

world_athletes <- world_map %>%
  left_join(athlete_by_country, by = c("region" = "Team"))

ggplot(data = world_athletes, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = Athlete_Count), color = "black") +
  scale_fill_viridis_c(option = "plasma", na.value = "lightgray") +
  labs(title = "Distribution of Athletes by Country",
       fill = "Number of Athletes") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank())
```



```
#16 ANIMATED PLOT
# Ensure the 'Year' column is numeric
a$Year <- as.numeric(a$Year)

# Grouping and summarizing the data
olympics_data <- a %>%
  group_by(Year, Season) %>%
  summarise(NoOfCountries = length(unique(NOC)), .groups = "drop")

# Create the animated plot
animated_bar_plot <- olympics_data %>%
  ggplot(aes(x = Year, y = NoOfCountries, fill = Season)) +
  geom_bar(stat = "identity", position = "dodge") +  # Bar plot with dodge position to separate
seasons
  labs(x = "Year",
       y = "Number of Countries that Participated",
       title = "Number of Countries that Participated in the Olympics by Year and Season") +
  theme_minimal() +
  transition_states(Year, transition_length = 2, state_length = 1) +  # Animate through the year
s
  ease_aes('linear')  # Smooth transition

animated_bar_plot
```


Number of Countries that Participated in the Olympics by Year and Season
Number of Countries that Participated
Year
Season
Summer
Winter