London Marathon Proposal

Anant, Nikunj, Badal, Maitri

Introduction

The dataset from the London Marathon, made available by (2022) as part of the LondonMarathon R package, provides a glimpse into various elements of this renowned race. It consists of two datasets obtained from Wikipedia as of November 1,2022:

Winners.csv : This dataset outlines the champions of multiple race categories, featuring their names, nationalities, finishing time and the year of the event.

london_marathon.csv : This dataset encompasses overall statistics such as the count of applicants, accepted entrants, starters and finishers, along with the total amount raised for charity and the official charity associated with each year.

The dataset facilitates the examination of trends in winning performances, participation patterns and the influence of the marathon on charitable donations. This analysis seeks to reveal significant insights and address particular research queries through statistical techniques and modeling.

The data was retrieved from the (2024), with additional information on its origin available in (2023) entry, “Scraping London Marathon data with {rvest}”.

Data Preprocessing

The Category, Athlete, and Nationality feature in Winners dataset are then factorized for further analysis.

The data is preprocessed to handle NA values in Raised column in London_marathon dataset by replacing NA values with 0.

Numerical Summaries

Column Name	Min	1st Quartile	Median	3rd Quartile	Max	Mean
Time.Seconds	5187	6550	7675	8414	16143	7608
Year	1981	1992	2002	2012	2022	2002

- The Years of data span from 1981 to 2022 in winners whereas, there is data from 1981 to 2020 on london marathons.
- The highest time to finish a marathon is 16143 seconds which is a outlier.

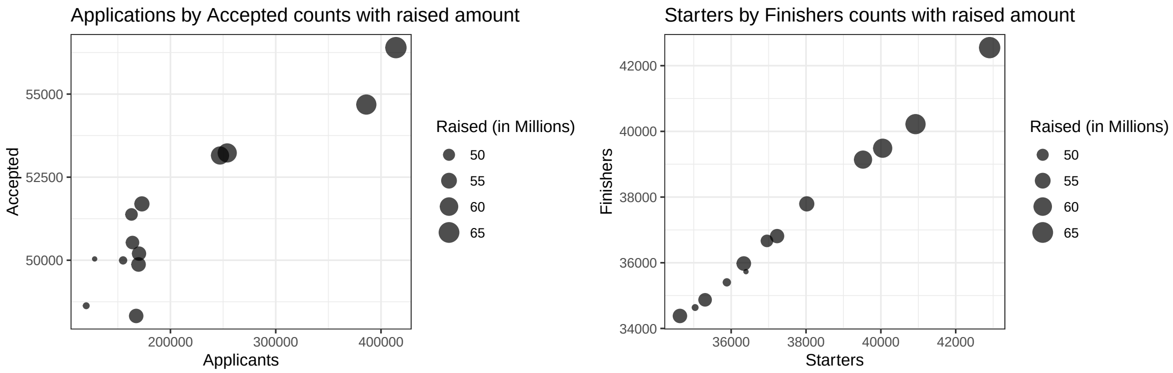
Column Name	Min	1st Quartile	Median	3rd Quartile	Max	Mean
Accepted	77	33057	43057	49903	56398	39269
Starters	77	24488	31369	35671	42906	28886
Finishers	61	23252	30584	35326	42549	28145

- In one of the marathons only 77 applicants were accepted and started the marathon.

Date	Year	Applicants	Accepted	Starters	Finishers	Raised
2020-10-04	2020	457861	77	77	61	0

By subsetting the dataframe, the marathon was conducted in 2020 and there were 4.5 lakhs of applicants but only 77 were accepted.

Visualization



(a) Applicants and Accepted individuals

(b) Starters and Finishers of marathon

Figure 1: Plots showing the relationship of Applicants - Accepted and Finishers - Starters where there is amount of charity is Raised

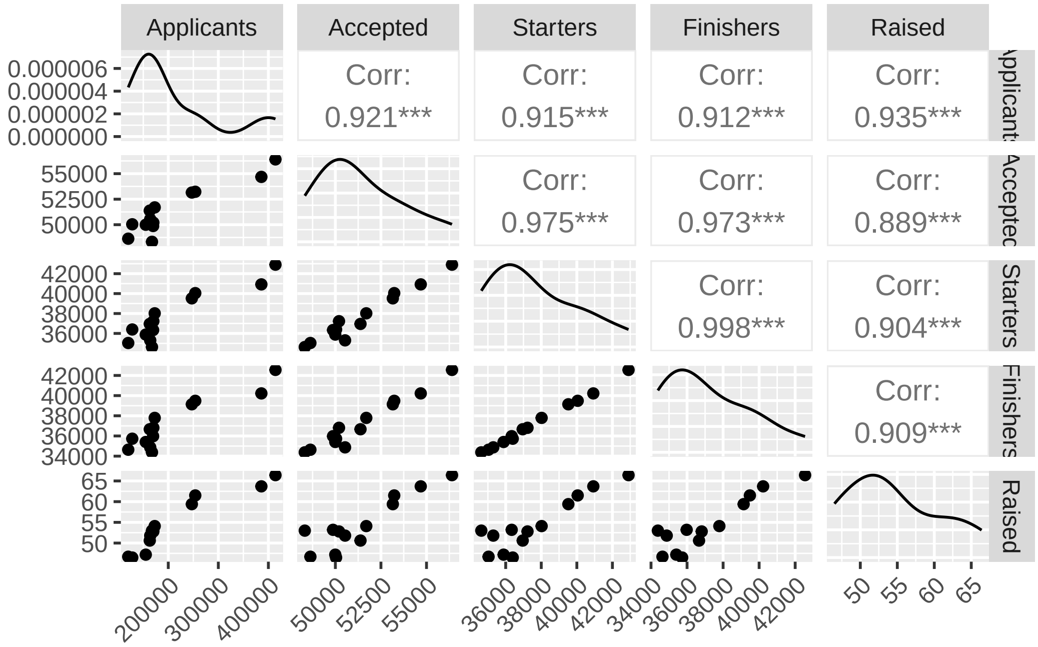


Figure 2: Pair plot between applicants, accepted participants, starters, finishers, and charity amount raised from the marathon

From the scattered bubble plot it is evident that the raised amount increases with the increase in applicants and accepted. Same goes with starters and finishers, there is linearity and the amount raised also increases.

From the above pair plot it is clear that there is high correlation between Applicants, Accepted, Starters, Finishers, and Raised.

Question: Can we predict the possible charity that can be raised in upcoming london marathons based on the count of Applicants, Accepted participants, starters, and finishers?

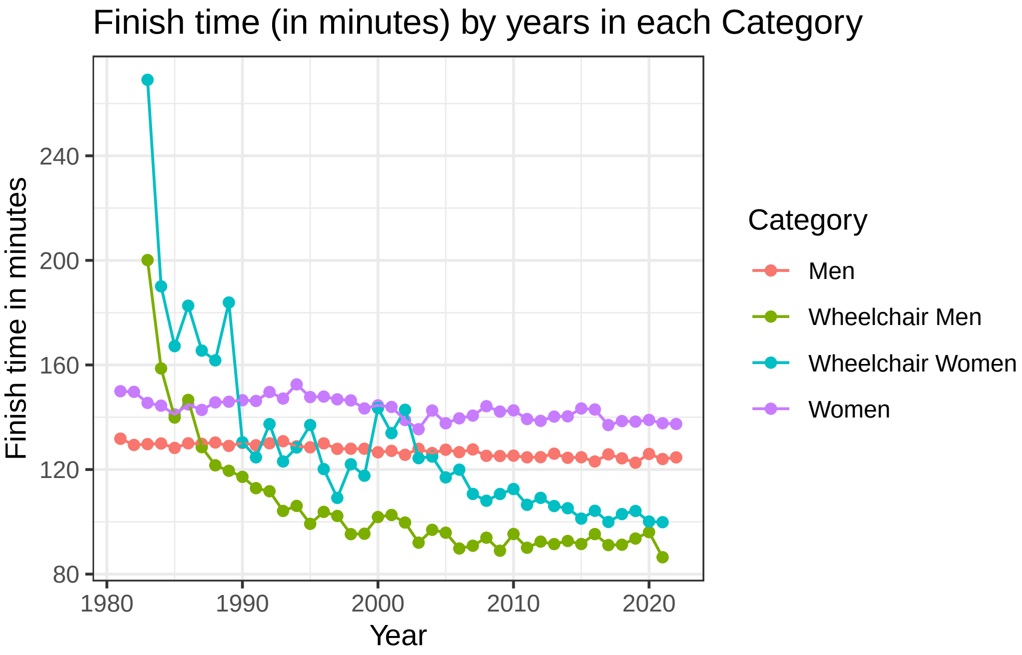


Figure 3: Plot shows the finishing time of each category in each year

Question: Wheelchair individuals have some correlation with time to finish the race? OR Does Category have relation with the finish time over the years?

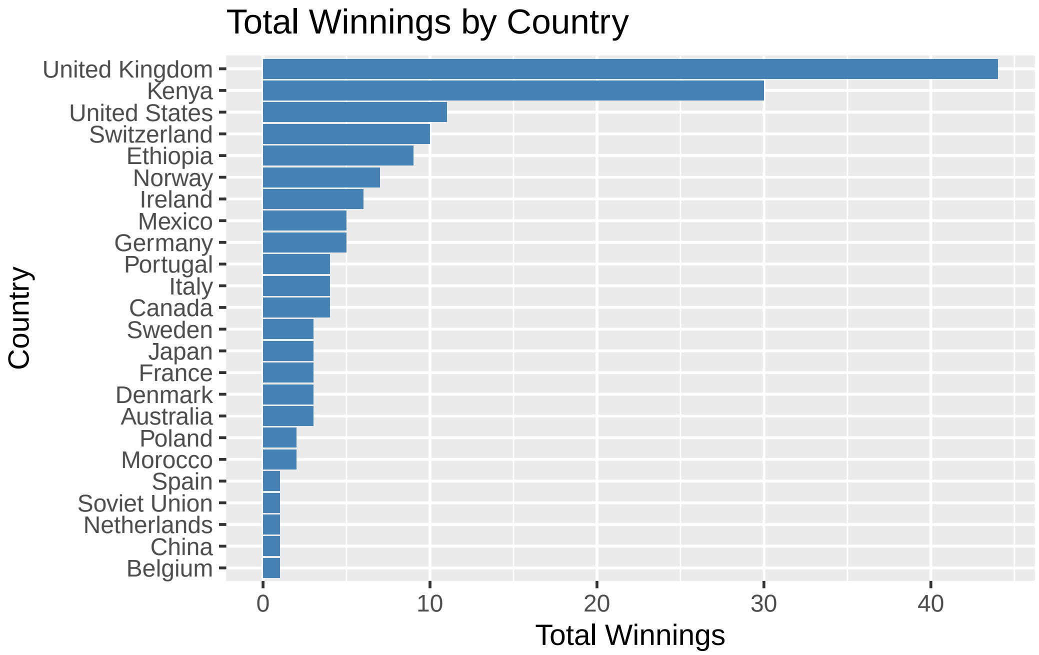


Figure 4: The horizontal barplot show the total winnings of the country in descending order

Question: Does country have significant relation with total winnings?

Bibliography

Community, Data Science Learning. 2024. "Tidy Tuesday: A Weekly Social Data Project." https://tidytues.day.

Rennie, Nicola. 2022. "London Marathon r Package: Dataset on London Marathon Winners." https://github.com/nrennie/LondonMarathon.

———. 2023. "Scraping London Marathon Data with rvest." March 16, 2023. https://nrennie.rbind.io/blog/web-scraping-rvest-london-marathon.

Appendix

Code

```
suppressWarnings(library(dplyr))
suppressWarnings(library(ggplot2))
suppressWarnings(library(lubridate))
suppressWarnings(library(GGally))
#libraries
library(lubridate)
library(dplyr)
library(ggplot2)
library(GGally)
# Loading data
tuesdata <- tidytuesdayR::tt_load('2023-04-25')
tuesdata <- tidytuesdayR::tt_load(2023, week = 17)

winners <- tuesdata$winners
london_marathon <- tuesdata$london_marathon

# Convert the time from hour:min:seconds to seconds
winners$Time.Seconds <- period_to_seconds(hms(winners$Time))

# Factoring the variables
winners$Category <- factor(winners$Category)
winners$Athlete <- factor(winners$Athlete)
winners$Nationality <- factor(winners$Nationality)

# Handling the NA in Raised
london_marathon$Raised[is.na(london_marathon$Raised)] = 0
london_marathon <- london_marathon[rowSums(is.na(london_marathon)) <= 2,]
summary(winners)
summary(london_marathon)
# Subset the dataframe for starters = 77
london_marathon[london_marathon$Starters == 77,]
# Option to print values without scientific notation
options(scipen = 999)
# Applicants vs Accepted participants by amount raised
london_marathon %>%
  filter(Raised > 0 ) %>%
  ggplot(aes(x=Applicants, y = Accepted, size = Raised)) +
```

```r
  geom_point(alpha = 0.7) +
  scale_size_continuous(name = "Raised (in Millions)")+
  labs(title="Applications by Accepted counts with raised amount") +
  theme_bw()

# Starters and finishers by amount raised
london_marathon %>%
  filter(Raised > 0 ) %>%
  ggplot(aes(x=Starters, y = Finishers, size = Raised)) +
  geom_point(alpha = 0.7) +
  scale_size_continuous(name = "Raised (in Millions)")+
  labs(title="Starters by Finishers counts with raised amount") +
  theme_bw()

# ggpairs plot
london_marathon[,c("Applicants","Accepted","Starters","Finishers","Raised")] %>%
  filter(Raised > 0) %>%
  ggpairs() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # chat gpt helped me
# Year vs Time by Category
winners %>%
  ggplot(aes(x = Year, y = Time.Seconds / 60, color = Category)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Finish time (in minutes) by years in each Category",
    x = "Year",
    y = "Finish time in minutes"
  ) +
  theme_bw()

winners_count <- data.frame(table(winners$Athlete))
names(winners_count) <- c("Athlete","Frequency")
#
winners_nationality <- unique(left_join(winners_count,
                                       winners[,c("Athlete","Nationality")],
                                       by="Athlete"))
grouped_nationality <- winners_nationality %>%
  group_by(Nationality) %>%
  summarise(Total_winnings = sum(Frequency))

grouped_nationality %>%
  ggplot(aes(x=reorder(Nationality,Total_winnings), y=Total_winnings)) +
  geom_bar(stat="identity",fill="steelblue") +
  labs(title="Total Winnings by Country", x = "Country", y = "Total Winnings")+
  coord_flip()
```