# NYC Marathon

## 1. Introduction

Millions of people run marathons world-wide each year. Marathons help raise money for charities, connect runners around the world, and help people to exercise and lead healthy lives while inspiring others to do the same. Our group chose to analyze the NYC Marathon Data because this marathon is a unique event which brings together extraordinary, driven people from all over the globe, from all age groups, at different levels of physical ability to push their bodies to the limit. **We were curious to find out who these people were, where they came from but also how these variables influence their performance. Additionally, we were interested in understanding how large is the participation of woman relative to men and what is the trend. Finally, we explored how to summarize the whole race in a graph**. The team consisted of Arthur Herbout, Antonia Lovjer, Andrea Navarrete and Andres Potapczynski.

Andrea took on the challenge of scraping the data from the NYC Road Runners website, and cleaning it using an R script which she shared with the rest of the group. The five data sets consisted of the NYC marathon data on race finishers for years 2015 to 2018. Andrea and Antonia continued to do a demographic analysis of the data, working towards understanding who were the people participating in the race. Arthur wanted to understand how the race was organized, and used variables such as "official_time" and "gender" to understand the meaning behind the "waves" and how they correlated with performance. Andres took a more in-depth approach to the data by clustering the runners according to their performance in order to analyze patterns amongst the different groups.

As a team, we worked on brainstorming ideas for the project, organizing the analysis, completing the exploratory data analysis and visualization, and creating the presentation. Design of the interactive Shiny app was completed as a group, with the development led by Andrea, and contributions by Arthur, Antonia and Andres.

## 2. Description of data

To answer research questions mentioned in the introduction we generated the data set by web scrapping the official site of the race (for all the details of this process look at section 2.2 below). It is worth mentioning that our data is made from only the competitors that finished the race. In 2018, our data set contains 52,669 runners. In 2017, we had 49,238 runners. In 2016, we had 50,486 runners and finally in 2015, we had 48,742 runners. This section is divided as follows. First we mention the relevant variables that we included in the data (as well as their actual R code names). Then we dive into the source of the information. Finally, we explain (or rather summarize) the intricate details of web scrapping the data.

### 2.1 Relevant variables for the analysis

Following is a list of the main variables used for the analysis (we omit adding the details for variables that we did not use in any of our analysis).

- `gender`: the gender of the athlete encoded as "M" for male and "F" for female.
- `age`: the age of the athlete.
- `official_time`: the time it took the athlete to finish the race.
- `gun_time`: the actual time when the athlete got to the finish line (remember that some athlete started at different times).
- `name`: the name of the athlete (we used this to keep track of how people improve or worsen their performance if they appear in other years).

- `city`: the city where the athlete comes from.
- `state`: the parse state abbreviation (only for the US) where the athlete comes from.
- `stat_name`: as above but the complete name of the state.
- `country`: the country where the athlete comes from.
- `lat`: the latitude given by the country where the athlete comes from.
- `long`: similar to above but now the longitude.
- `split_<x>k` the time when the athlete got to the $x$th split where $x \in \{5, 10, 15, 20, 25, 30, 35, 40\}$.
- `type`: the inferred group category. "R" is for runners, "H" for handicap and "W" for wheelchair.
- `team`: the name of the team if the athlete belongs to one. For example, professional athletes might be sponsored say by *NIKE*, other athletes represent charities associations like *Team for Kids* and finally some others might belong to an amateur group like *North Brooklyn Runners*. Nonetheless the majority does not report belonging to a group and are added as *NA*.

## 2.2 Source

Our data set comes from the *TCS New York City Marathon Results* which is hosted by the New York City Road Runners. We scrapped the data from their official website link. Accessing the previous link renders the following view:

[][./pics/webpage.png .png]

Thus, as it can be seen above the information that the website provides falls into two categories.

- **Demographics**: Age, gender and country / city
- **Performance metrics**: Official time, pace per mile and the time per 5 kilometer split

The main difficulty with this data is that it is embedded in a web page. Even though it is easily accessible it is hard to download locally! Thus, we had to figure out how to web scrape it.

## 2.3 Web Scrapping

Web scrapping was more time consuming than we thought. Even though we used the really well-develop package of `BeautifulSoup` it still required us to overcome two main obstacles (1) understanding the web scrapping process and (2) to come up with the template that our program should use in order to find the information.

To get a complete understanding of the process, we made intensive use of different resources online: either YouTube videos or other Q&A websites such as `StackOverflow`. A particular source that we found useful was link. Moreover, we became acquainted with the myriad of details that were not evident when we started the process. First, we had to distribute our work in a computer engine on the cloud. The difficulty is that web scrapping is a really slow process. For example, we have to run a long `for-loop` were we have to make a different connection for each of the over 50 K participants of the marathon (this times the number of year that we downloaded). Moreover, we cannot make a constant connection to the web site since that could potentially be considered as an attack. Thus, we had to replicate the pace that a person takes to access the website. Additionally, we had to set-up the computer engine and let it do the work (which took approximately 3 days)

In term of coming up with a template for our program to run. We had to learn (actually before starting D3) how to read all the html elements of a web page and embed that knowledge into a script for `BeautifulSoup` to perform its magic. Furthermore, this is a sensitive procedure. Altering any location of an element in the web page renders the script useless. Thus, on the one hand, we had to make several "robustness checks" for our script before letting it run (because we risk losing days of work). On the other hand, we had to develop for each year a new template! Every year the layout was altered in some particular way, and so we had to adapt for those changes. However, we were still able to download the same information every year (which made the template creating process slightly more exciting).

At the end, we were quite happy that we were able to download the data. Mostly because we were doing an analysis that excited us but also because, due to the difficulty of the process, we knew that not many people had done this analysis before.

# 3. Analysis of data quality

Due to the web scrapping procedure we were afraid that the data would have a bunch of errors, but actually it did not. We divided our analysis for data quality into four sections. First, we analyze the general missing values patters. Second, we delve into the one source of missing values: *messy locations*. Then, we analyze the other source of missing values: *inconsistent time formats*. Finally, we comment on the nature of outliers in this data.

## 3.1 Missing values and its patterns

[][Add visna plot from raw data to show what are the missing percentages and patterns - I believe Joyce expects this when treating missing values]

## 3.2 Messy Locations

Sometimes the geographical information displayed in the webpage related to the city of the runner, sometimes to her state and some others times to the county she was born (definitely our web scrapping program was not so smart as to make those types of swaps). Then we had to make a decision on what to do about this: either to spend quite some time on mapping manually all the cases or do some kind of parsing and leave aside the cases without a match.

A priori we were unsure what to do, we had mixed feelings about not using this information, which was a key variable for our analysis. Hence to make a decision first we assessed how bad our problem was. For this we ran our parsing procedure to see how many cases lack a match. Our parsing procedure consisted on two steps: first, we would try to parse the states and for the remaining we would pass our country mapping (which was tractable). Fortunately, we got quite lucky and the procedure almost mapped completely off-the-shelve. The reason is that the majority of the inputs that had a state name in the geographical location where from the USA and also because the R function `state.abb` was quite robust and it allowed us to get the state abbreviations. Therefore, we left the rest of the cases as NA and move on into adding the latitude and longitude of the place.

## 3.3 Inconsistent Time Formats

We have never had a good experience when working with a time series data sets and this was, unfortunately, not the exception. We made extensive use of the available tools to go from formats like "01:00:23" to an actual meaningful time stamp (on this, it appears as if rather than spreading the best practices of presenting time variables in a data set, the world is rather creating more sophisticated tools to go over every eventuality).

The previous discussed problem was multiplied by all the columns that contained time data which was quite a few: the finished time, the time per 5 k split, the official time and many more. Thus after running our parsing function on each column we filtered out any observation that had at least one missing value (that is, not a parsable time) in any of the main variables such as `gun_time`, `official_time`, and each of the splits `split_<x>k`.

### 3.4 Outliers

Outliers in this context come either from poor performing runners, excellent performing runners and other type of competitions categories (wheelchair and handicap). Thus, the notion of *outlier* here stems from mixing many different cohorts of people (professional athletes, standard competitors, wheelchair and handicap) in the data rather than by having some plausible error in a certain recording of a variable. Although there were some people that finished the marathon in around 7 hours. But we are uncertain if this is a mistake or maybe they burnt out and finished the race as they could.

Hence, after separating the data into different groups, now there where not evident outliers (in the sense of points going past the whiskers in the box plots). For example, the top 100 runners all stayed very tied. Whereas the latter group had similar running times but there was more variance. [][Think if it is worth adding the comparison of these two boxplots]

## 4. Main analysis

This section contains the main results of our exploratory data analysis. Each of the subsections below addresses one of the research questions that we posed in the introduction.

### 4.1 Demographic

** Introduction **

Millions of people run marathons world-wide each year. Marathons help raise money for charities, connect runners around the world, and help people to exercise and lead healthy lives while inspiring others to do the same. Our group chose to analyze the the NYC Marathon Data because this marathon is a unique event which brings together extraordinary, driven people from all over the globe, from all age groups, at different levels of physical ability to push their bodies to the limit. We were curious to find out who these people were, where they came from, and how the outcome of the race could be visualized and understood. The team consisted of Arthur Herbout, Antonia Lovjer, Andrea Navarrete and Andres Potapczynski.

Andrea took on the challenge of scraping the data from the NYC Road Runners website, and cleaning it using an R script which she shard with the rest of the group. The five data sets consisted of the NYC marathon data on race finishers for years 2015 to 2018. Andrea and Antonia continued to do a demographic analysis of the data, working towards understanding who were the people participating in the race. Arthur wanted to understand how the race was organized, and used variables such as “official_time” and “gender” to understand the meaning behind the “waves” and how they correlated with performance. Andres took a more in depth approach to the data by clustering the runners according to their performance in order to analyze patterns amongst the different groups.

As a team, we worked on brainstorming ideas for the project, organizing the analysis, completing the exploratory data analysis and visualization, and creating the presentation. Design of the interactive Shiny app was completed as a group, with the development led by Andrea, and contributions by Arthur, Antonia and Andres.

** Exploratory Data Analysis and Visualization **

** Age Distributions **

Performance in athletics is often impacted by the age of the athletes. For example, most of the participants in the Olympics and in international sports competitions tend to be people under the age of 40. Since Marathon running is a test of endurance, it would be interesting to see if similar trends exist in the participants of the NYC Marathon. We can plot density curves for age of the finishers in each year for which we have data (2015-2018).

From a first glance, we notice that the distribution of the data appears to have some sort of a normal shape, with some flattening out at the apex of the curve, and a clear mode. The median age is between 43 in 2015 and 40 in 2018. The youngest participant is 18, since the rules of the race most likely forbid minors from participating, and the oldest is 87.

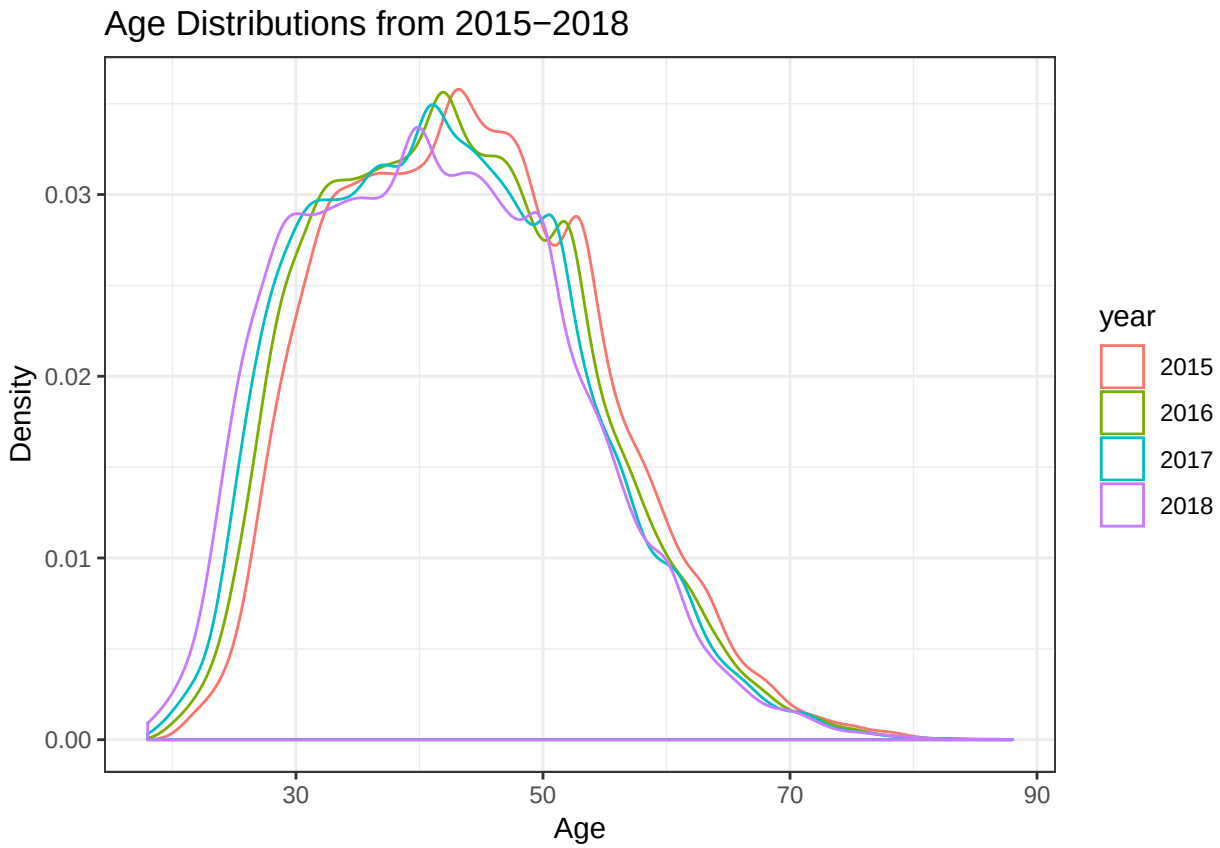
Overall, there are more younger people finishing the race in comparison to older people, with a large portion of the participants between the age of 30 and 50.

```
names(marathon)
```

```
##  [1] "bib"                    "gender"
##  [3] "age"                    "gun_place"
##  [5] "gun_time"               "name"
##  [7] "official_time"          "pace_per_mile"
##  [9] "percentile_age-graded" "city"
## [11] "state"                  "place_age-graded"
## [13] "place_age-graded_of"    "place_age-group"
## [15] "place_age-group_of"     "place_gender"
## [17] "place_gender_of"        "place_overall"
## [19] "place_overall_of"       "splint_10k"
## [21] "splint_15k"             "splint_20k"
## [23] "splint_25k"             "splint_30k"
## [25] "splint_35k"             "splint_40k"
## [27] "splint_5k"              "splint_half"
## [29] "team"                   "time_age-graded"
## [31] "type"                   "year"
## [33] "state_name"             "lat"
## [35] "long"                   "country"
```

```
marathon$year <- factor(marathon$year)

# names(marathon)[names(marathon) == 'year'] <- 'Year'

ggplot(marathon, aes(x=age, color=year)) +
  geom_density() +
  ggtitle("Age Distributions from 2015-2018") +
  ylab("Density") +
  xlab("Age") +
  theme_bw()
```

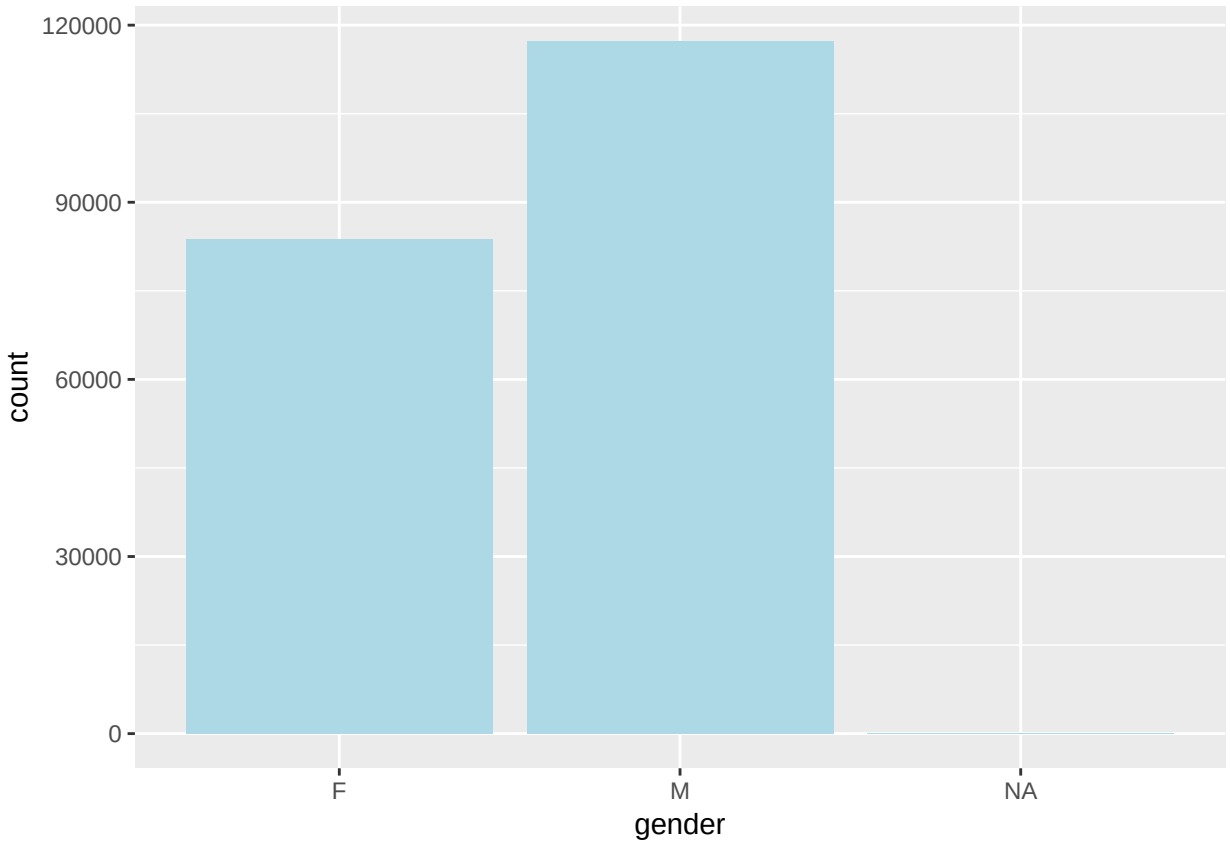
```
## Warning: Removed 76 rows containing non-finite values (stat_density).
```




** Gender Proportions **

A simple bar chart of the number of men and women amongst the participants will give us a basic understanding of who is participating in the race. From the graph we see that there are more men in the race in comparison to women. We can also see that we have a small number of NAs present in the data as well. This could be caused by an error in scraping the data, or the data being missing at collection.

```
ggplot(marathon, aes(x = gender)) +
  geom_bar(stat='count', fill='lightblue')
```




** Age Distribution by Gender **

The next step would be to examine the age distributions across the different genders. The data collection process only allowed for the representation of two genders: Male and Female, and so we will restrict our analysis to these two genders.

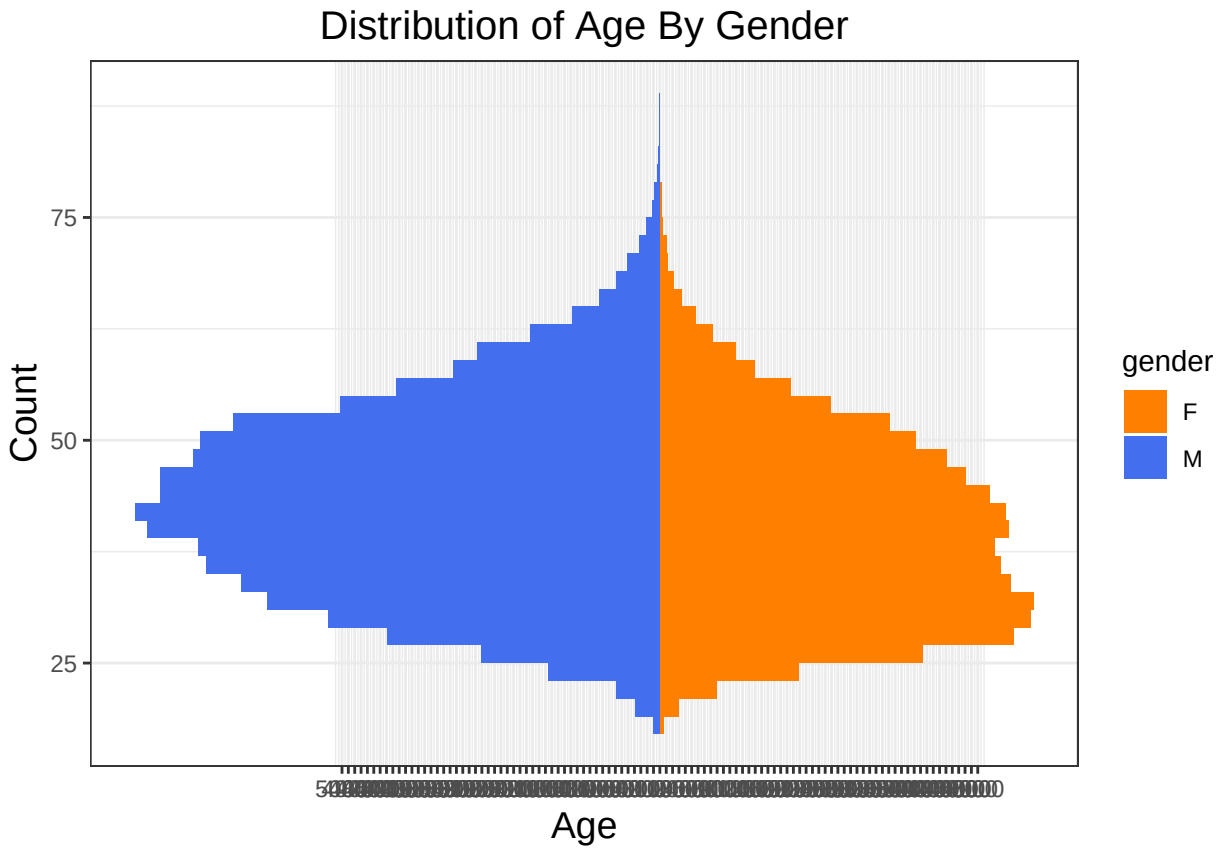
It is interesting to see that the distributions seem very similar across the two genders, with unimodal, approximately normal shapes. It is interesting as well to see that there are more younger women amongst the finishers than there are younger men. Overall, the men seem to be a few years older on average in comparison to the women.

```
# names(marathon)[names(marathon) == 'gender'] <- 'Gender'

# marathon$age_factor <- as.factor(marathon$age)
marathon$age_factor <- marathon$age

ggplot(marathon, aes(x = age_factor, fill = gender)) +
  geom_bar(data = subset(marathon, gender == "F"), binwidth =2) +
  geom_bar(data = subset(marathon, gender == "M"), binwidth =2, aes(y=..count..*(-1))) +
  scale_y_continuous(breaks = seq(-5000, 5000, 100),
                     labels=abs(seq(-5000, 5000, 100))) +
  scale_fill_manual(values = c("darkorange1",
    "royalblue2")) +
  coord_flip() +
  theme_bw() +
  ggtitle('Distribution of Age By Gender') +
  xlab("Count") +
  ylab("Age") +
```

```
  theme(plot.title = element_text(hjust = 0.5, size = 15) ) +
  theme(axis.title=element_text(size=14))
```

```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```

```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```

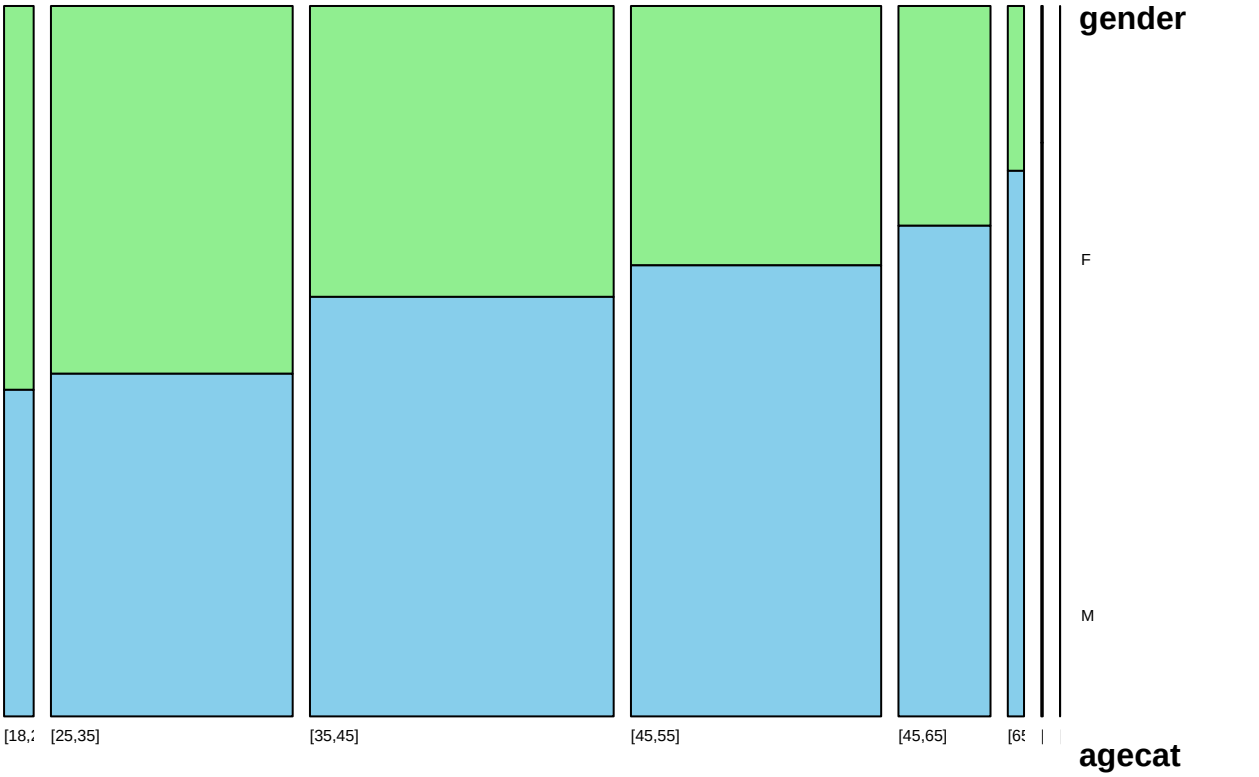



We can also examine this data in the form of a mosaic plot. Is is more clear from this plot that there are more men in the race than there are women. One of the benefits of a mosaic plot is that relative proportions between different categories are easier to determine, but one of the downsides is that distributions cannot be determined.

```
marathon$agecat <- c(0)
# create age categories for 10 year age groups (7 years for the first group)
marathon$agecat[marathon$age > 17 & marathon$age <= 25] <- "[18,25]"
marathon$agecat[marathon$age > 25 & marathon$age <= 35] <- "[25,35]"
marathon$agecat[marathon$age > 35 & marathon$age <= 45] <- "[35,45]"
marathon$agecat[marathon$age > 45 & marathon$age <= 55] <- "[45,55]"
marathon$agecat[marathon$age > 55 & marathon$age <= 65] <- "[45,65]"
marathon$agecat[marathon$age > 65 & marathon$age <= 75] <- "[65,75]"
marathon$agecat[marathon$age > 75 & marathon$age <= 85] <- "[75,85]"
marathon$agecat[marathon$age > 85 & marathon$age <= 95] <- "[85,95]"
```

```
cols <- c("lightgreen", "skyblue")
vcd::doubledecker(gender ~ agecat,
                  data = marathon,
```

```
                spacing_args = list(start =0.3, rate = 1.5),
                gp = gpar(fill=cols),
                labeling_args=list(
                boxes = FALSE,
                 offset_labels = c(1, 0, -1, 0),
                           gp_labels=(gpar(fontsize=6))))
```

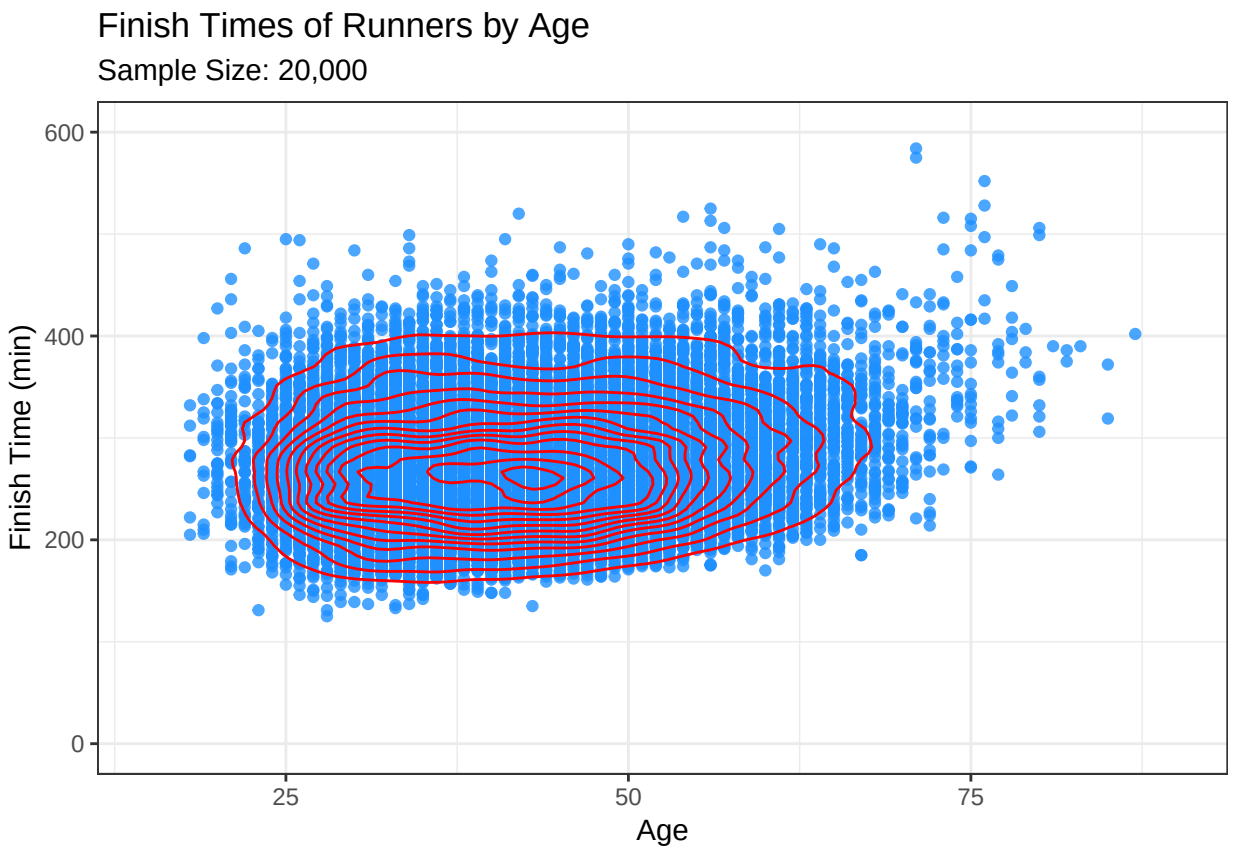



### Finish Times by Age

Something that would be interesting to examine would be the finish times by age of the runners. Here we will look only at the actual "runners" of the race, excluding hand cyclists and wheelchair racers, as their times would be significantly faster than the runners. We can try to display the data in several formats, the first will be a scatterplot where we take a sample of the data and plot age on the x-axis and finish time in minutes on the y-axis. We can see that there are a lot of runners aged 35 to 50 and running time between 4 and 5 hours.

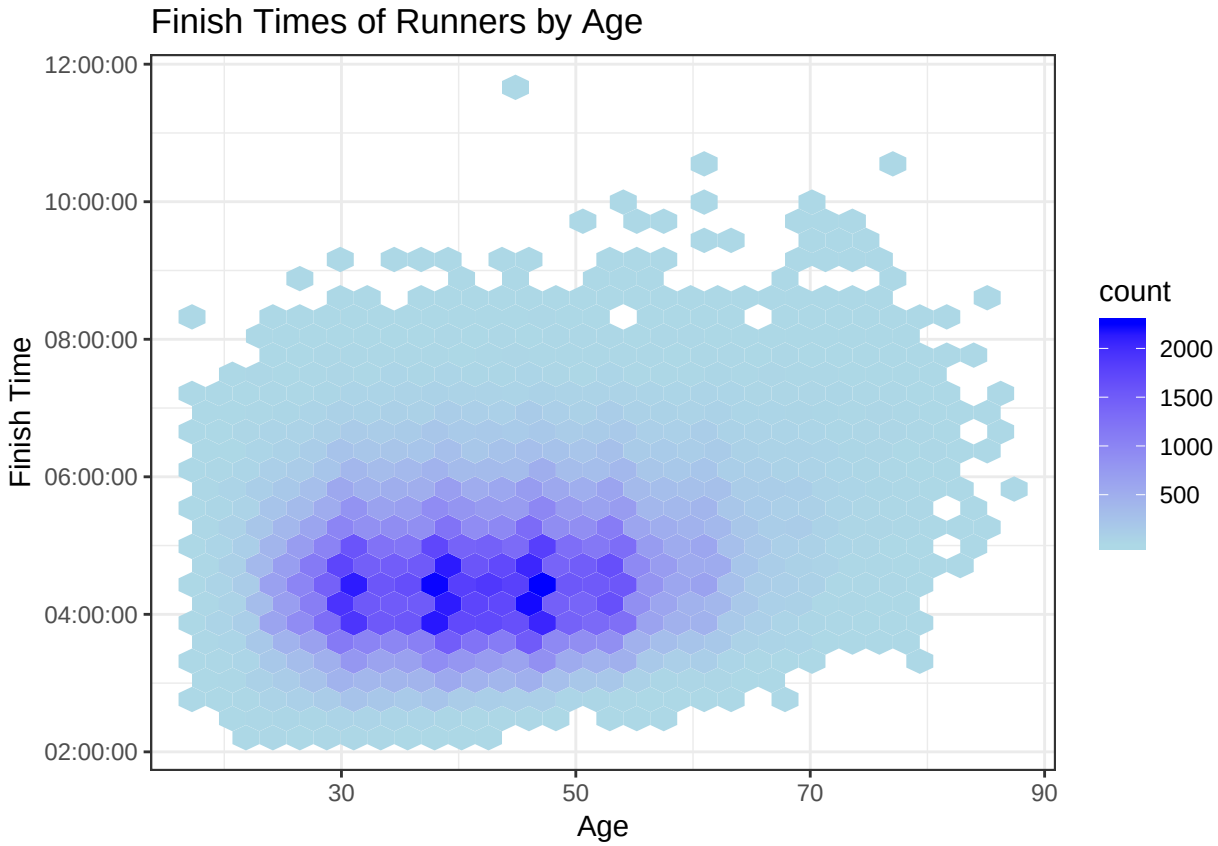
```
marathon <- marathon %>%
 mutate(time_min = hour(official_time)* 60 + minute(official_time))

runners <- marathon %>%
  filter(type == "R")

data_sample <- sample_n(runners, 20000)

ggplot(data_sample, aes(x = age, y = time_min)) +
  geom_point(color = 'dodgerblue1', alpha = 0.8) +
  geom_density2d(color = "red") +
  xlim(15,90) +
  ylim(0,600) +
  theme_bw() +
  ylab("Finish Time in Minutes") +
```

```
  xlab("Age") +
  labs(x="Age",
      y="Finish Time (min)",
      title="Finish Times of Runners by Age",
      subtitle="Sample Size: 20,000")
```




Another way to visualize this information is by creating a heat map with the same axis as in the previous plot. Here we will use the entire data set, and have the darker colors represent the value pairs where there are more data points present. We see that the highest concentration of runners can be found between he ages of 30 and 50, and between 4 and 5 hours. Another observation is that the older people get, the fewer fast runners there are. This is evident by the fact that there are no runners in the bottom right hand corner. What is also interesting is that all of the fastest runners cover a fairly wide age range from 20 to 42.

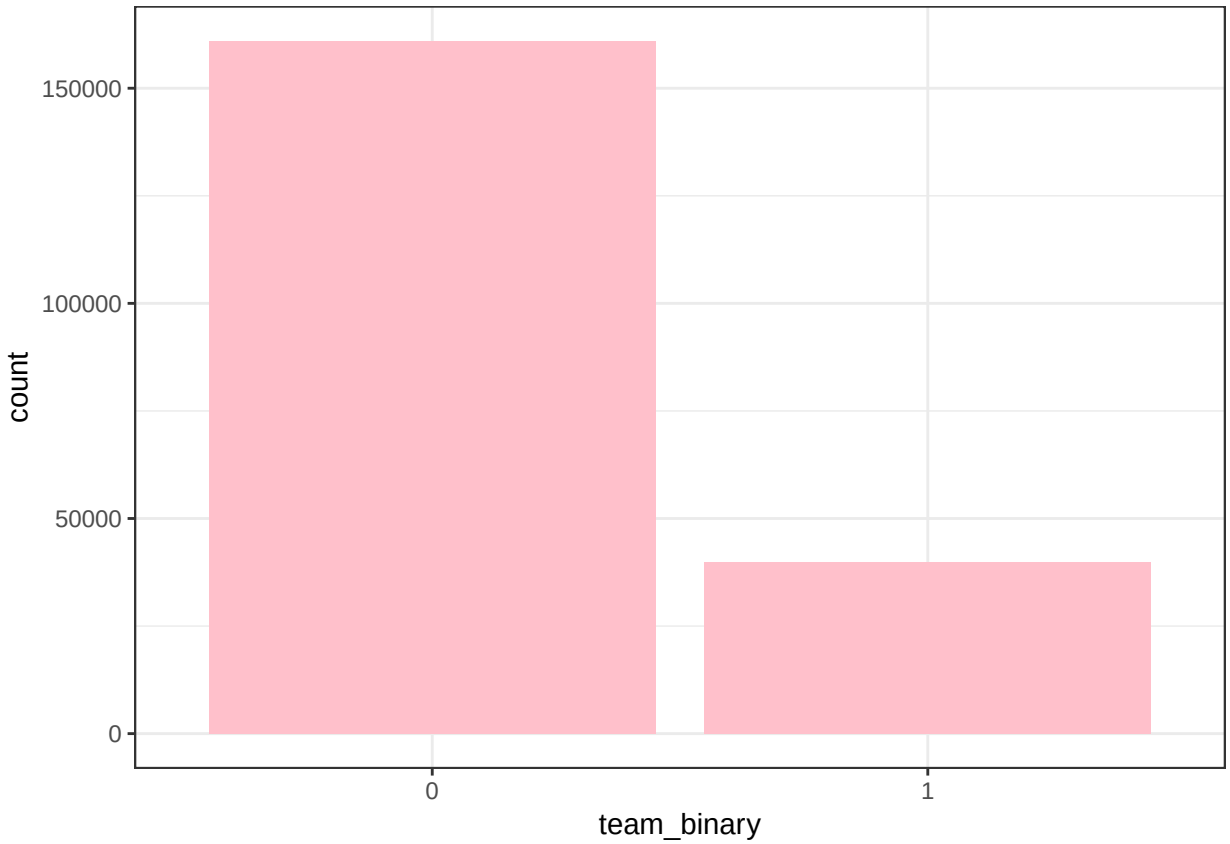
```
ggplot(runners, aes(x=age, y=official_time)) +
  scale_fill_gradient(low = "lightblue", high = "blue") +
  geom_hex() +
  ylab("Finish Time in Minutes") +
  theme_bw() +
  xlab("Age") +
  labs(x="Age",
      y="Finish Time",
      title="Finish Times of Runners by Age")
```




** Analysis of Teams **

All together there are around 651 different teams that entered the NYC Marathon Race in the running category. Many of these teams are groups of volunteers that joined their company team, or a local charity team. They are also include amateur running clubs as well as professional teams that are sponsored by large athletic equipment firms such as Nike and Adidas. If we look at all of the runners, we see that 39815 of the runners were part of a running team, while 160998 we not, which is over 80% of the runners. Therefore, it's only about 1/5 of the finishers that were part of an organized team. In the graph below, 0 indicates that the non-team runners, and 1 indicates the runners on teams.

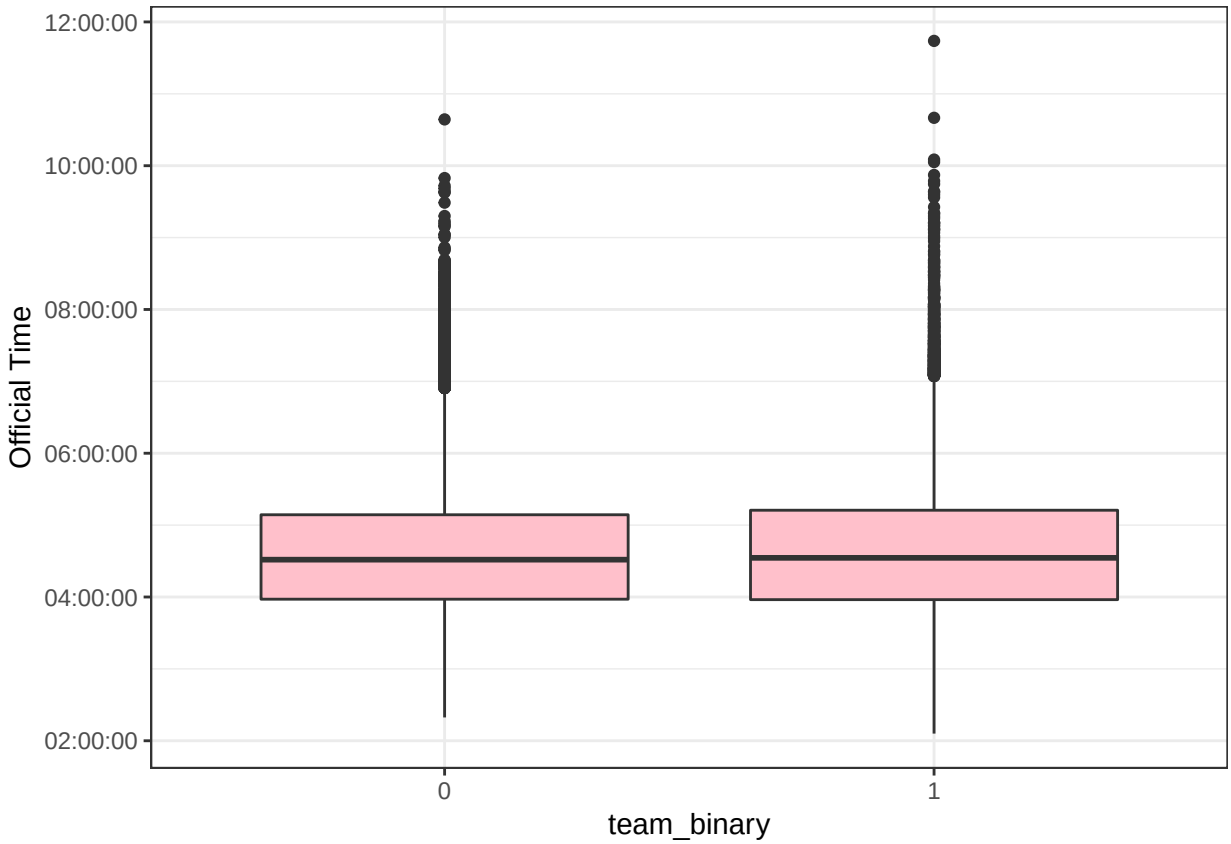
```
runners$team_binary <- ifelse(is.na(runners$team), 0, 1)
runners$team_binary <- as.factor(runners$team_binary)

teams <- runners %>%
  filter(n() > 5)

ggplot(teams, aes(x = team_binary)) +
  geom_bar(fill="pink") +
  theme_bw()
```




Next we look at box plots for the the official finishing times for team runners vs. non-team runners. Surprisingly the distributions look almost identical. The medians are very close to one another for the two groups, as are each of the other quantiles.

We can see that the outliers are slightly more extreme for the team runners in both directions. The fastest runner is part of a team, and the slowest runner is also part of a team.

```
ggplot(runners, aes(x = team_binary, y = official_time)) +
  geom_boxplot(fill= "pink") +
  theme_bw() +
  ylab("Official Time")
```

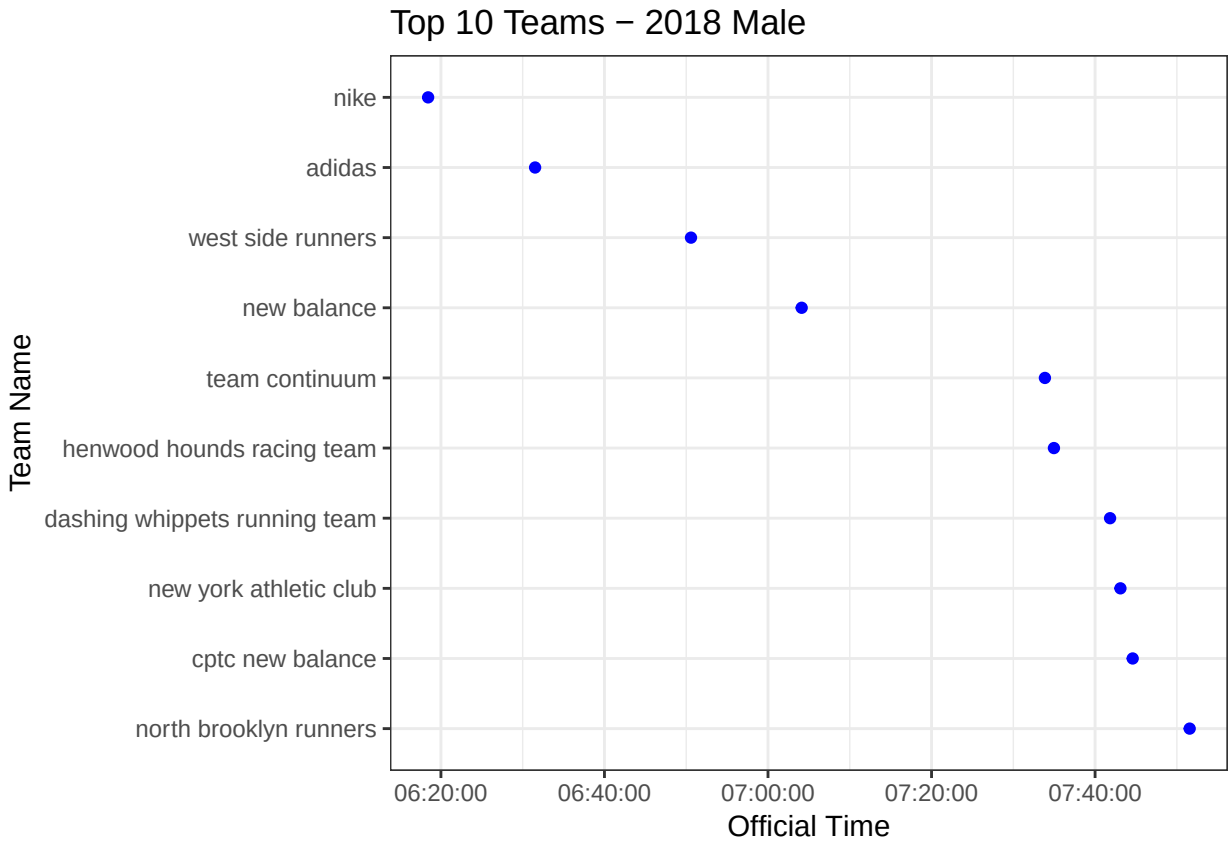



Now we can look within the teams to identify the best teams for women and for men. To do this we looked at teams that contained more than 5 members, and ranked the team members according to their performance. Then we took the sum of the top 3 runners per team, and again ranked the teams.

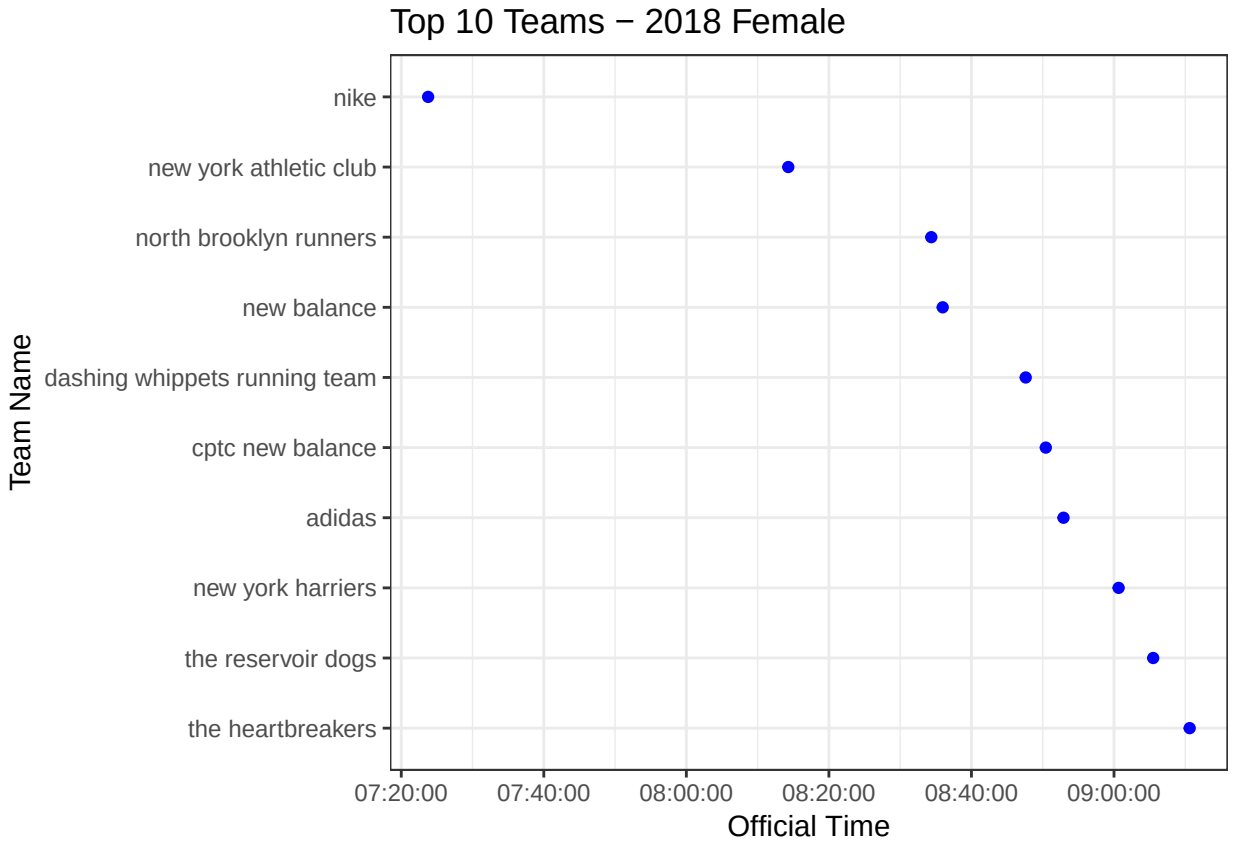
It is not surprising that many of the team names repeat for both genders. Additionally we can see what many of the names would be familiar to people, such as Nike, Adidas, New Balance and the New York Athletic Club

```
runners <- runners %>%
  mutate(team = tolower(team)) %>%
  filter(year == 2018)

best_teams_M_2018 <- runners %>%
  filter(!is.na(team)) %>%
  filter(gender == "M") %>%
  group_by(team) %>%
  filter(n() > 5) %>%
  arrange(official_time)%>%
  mutate(rank = rank(official_time)) %>%
  select(team, official_time, rank) %>%
  arrange(team, rank) %>%
  filter(rank <= 3) %>%
  dplyr::summarize(sum_top_time = sum(official_time)) %>%
  arrange(sum_top_time)

best_teams_M <- best_teams_M_2018[1:10, ]

ggplot(best_teams_M, aes(x=sum_top_time, y=reorder(team, -sum_top_time))) +
  geom_point(color='blue') +
```

```
  ggtitle("Top 10 Teams - 2018 Male") +
  theme_bw() +
  ylab("Team Name") +
  xlab("Official Time")
```




```
best_teams_F <- runners %>%
  filter(!is.na(team)) %>%
  filter(gender == "F") %>%
  group_by(team) %>%
  filter(n() > 5) %>%
  arrange(official_time)%>%
  mutate(rank = rank(official_time)) %>%
  select(team, official_time, rank) %>%
  arrange(team, rank) %>%
  filter(rank <= 3) %>%
  dplyr::summarize(sum_top_time = sum(official_time)) %>%
  arrange(sum_top_time)

best_teams_F <- best_teams_F[1:10, ]

ggplot(best_teams_F, aes(x=sum_top_time, y=reorder(team, -sum_top_time))) +
  geom_point(color='blue') +
  ggtitle("Top 10 Teams - 2018 Female") +
  theme_bw() +
  ylab("Team Name") +
  xlab("Official Time")
```




** Conclusion **

One of the challenges that we faced was that wheelchair racers and hand cyclists participated in such few numbers that they were ignored for the most part since there was so little data.

## 4.2 Running Variables

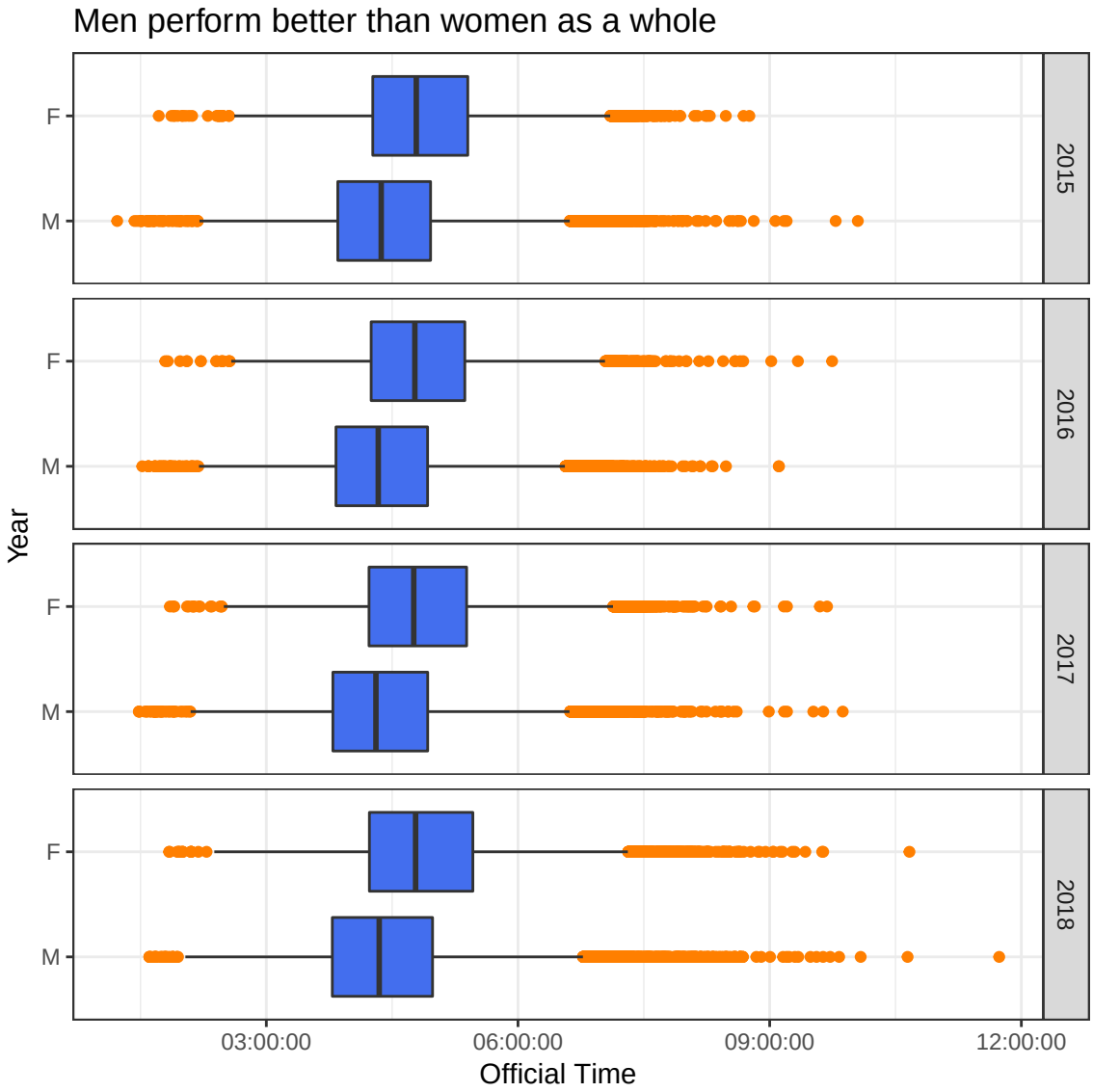
Is gender a significant indicator of performance? In all the following sections, we will only consider the runners, not other categories. Indeed, for instance, the wheelchair athletes would be considered as outliers since they race way faster than runners. Only considering runners will help us draw conclusions.

When analyzing a sport event, a natural question instantaneously arises: does gender affect performance, and to what extend?

Only take into account the runners, not the wheelchairs

Let us first draw a boxplot of the performance based on gender for the 4 year span we are analyzing.

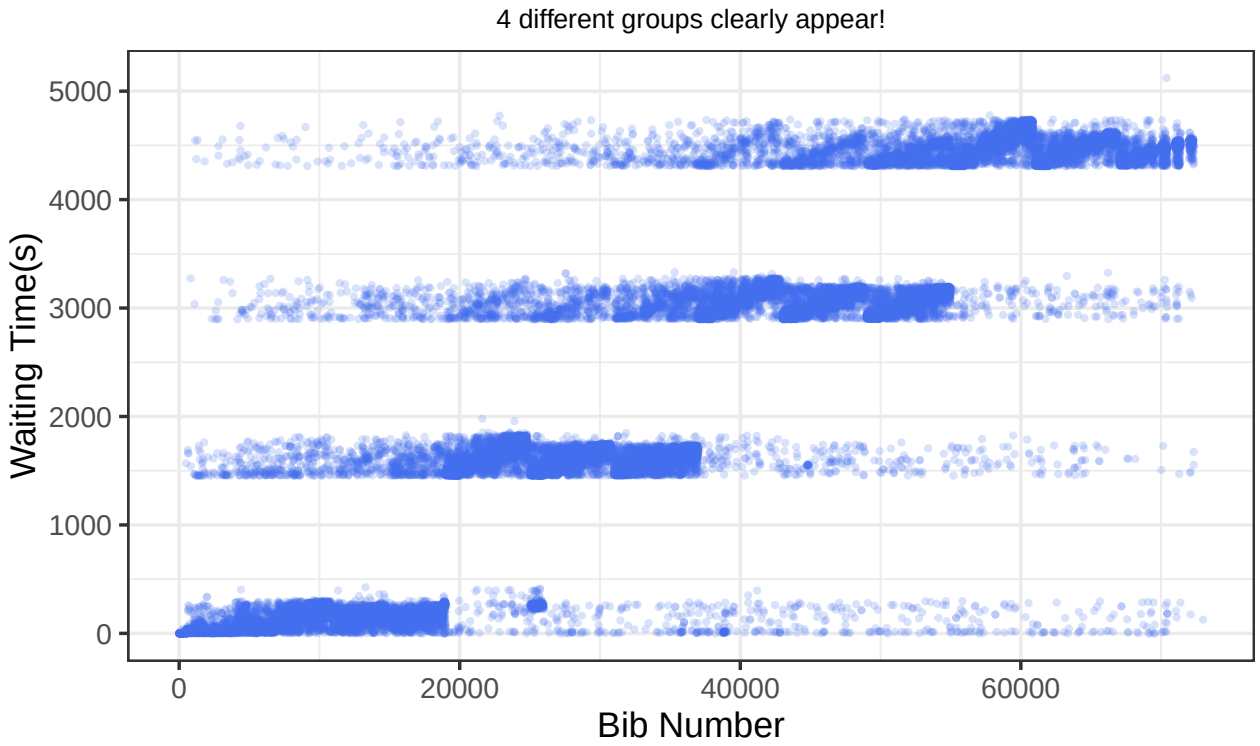
```
marathon %>% filter(gender != 'NA') %>%
ggplot(aes(x = factor(gender, levels = unique(gender)) , y = official_time)) +
  geom_boxplot(fill = "royalblue2", outlier.colour = "darkorange1") +
  facet_grid(year~.) +
  coord_flip() +
  ylab("Official Time") +
  xlab("Year") +
  ggtitle("Men perform better than women as a whole") +
  theme(plot.title = element_text(hjust = 0.5, size=15)) +
  theme_bw()
```




We can clearly see that men consistently perform better than women in the marathon. For those four years, the women's first quartile is roughly the men's median. Therefore, given this plot, there is a clear correlation between gender and performance.

However, this simple plot does not take into account the runner's physical preparation. Let us now dive into the details of the race's organization.

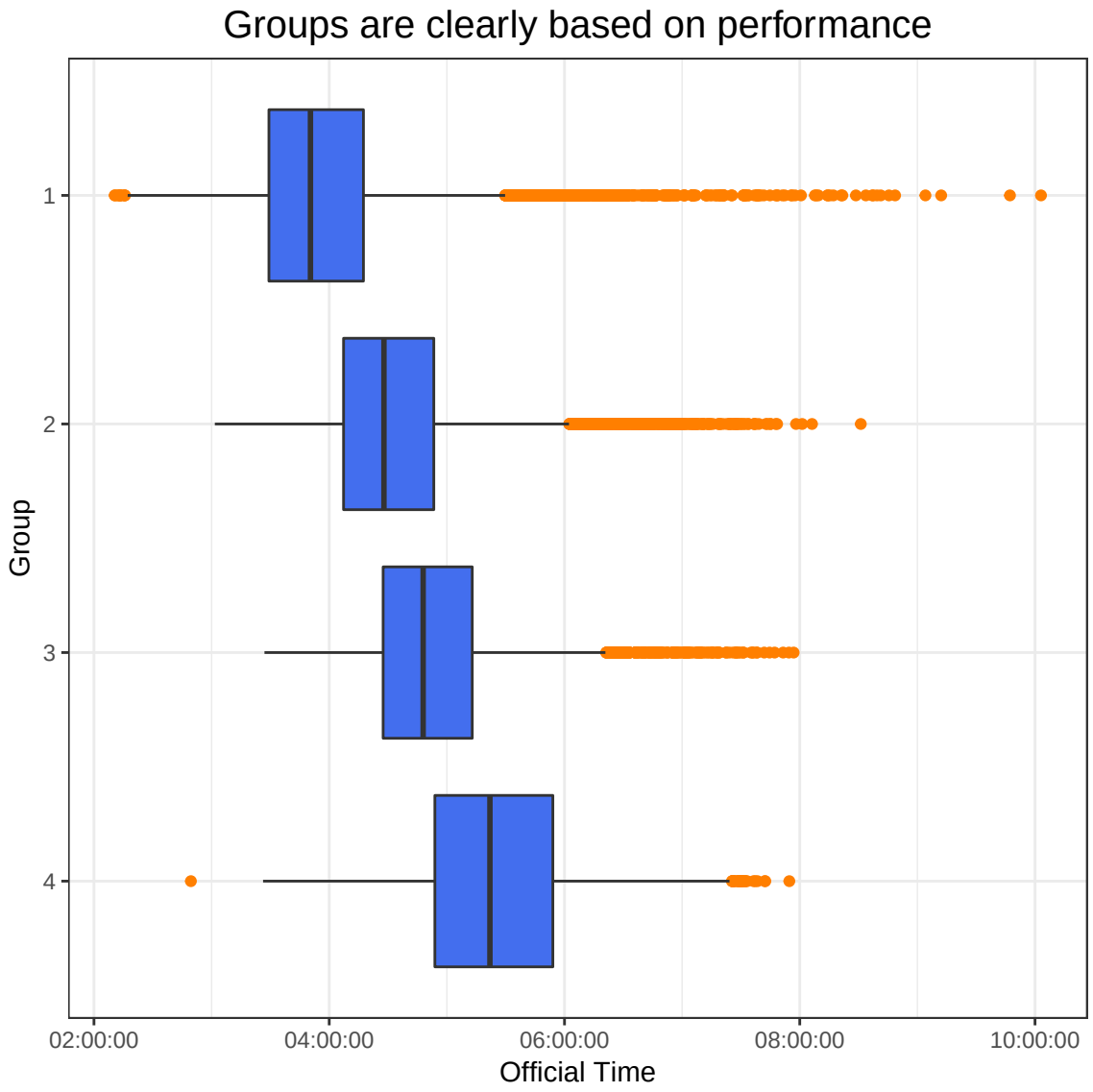
```
ggplot(r15, mapping = aes(x = bib, y = diff)) +
  geom_point(alpha=1/5, col= "royalblue2") +
  xlab("Bib Number") +
  theme_bw(20) +
  ylab("Waiting Time(s)") +
  theme(plot.title = element_text(hjust = 0.5, size=15))+
  ggtitle("4 different groups clearly appear!")
```




This first very simple plot gives us some information about the organization. First, there are clearly 4 different starts: * as expected, the first one starts at 0. It is the one that coincide with the gun start, * A second start occurs 25 minutes after the gun start, * A third start occurs 50 minutes after the gun start, * The fourth start occurs 1 hour 10 minutes after the gun start.

It is now clear that four different groups can be identified. But what are they based on?

```
ggplot(r15, aes(x = factor(wave, levels = c('4','3','2','1')) , y = official_time)) +
  geom_boxplot(fill = "royalblue2", outlier.colour = "darkorange1") +
  coord_flip() +
  ylab("Official Time") +
  xlab("Group") +
  theme_bw() +
  ggtitle("Groups are clearly based on performance")+
  theme(plot.title = element_text(hjust = 0.5, size=15))
```

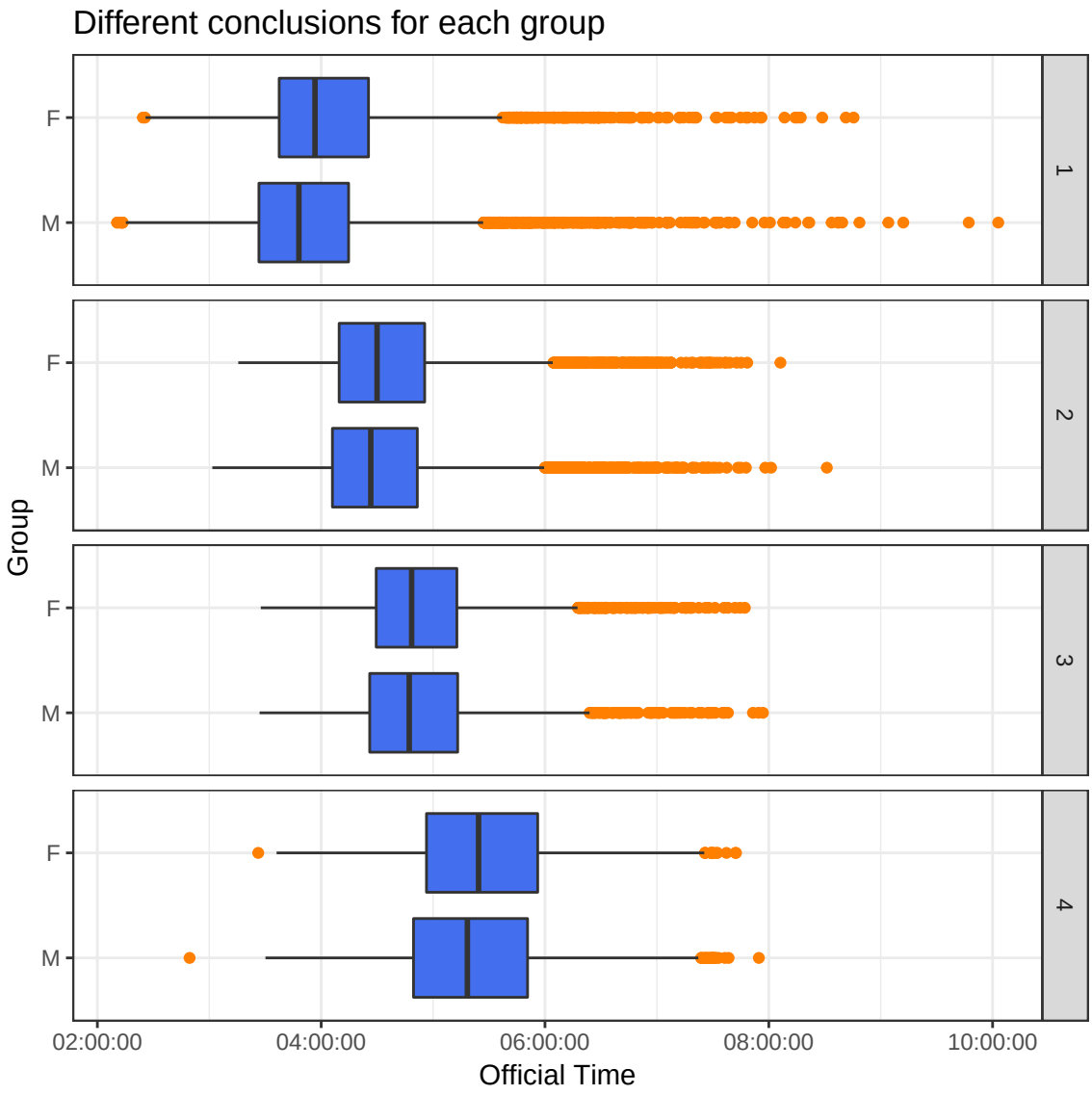



As we could have imagined, those groups are based on performance, the first one being the best one. In order to be allowed to run in a given group, participants have to prove that they already did a good enough time in an official race in the recent past. An obvious flow of this method is that rookie runners cannot be placed in their real group since they have no record. We can see one of those in the fourth group: the outlier that run it in under three hours. Moreover, if a runner is out of shape but has a good record, (s)he still can be in the first group and still perform badly.

Now that we know how the groups are structured, we can look into them to see if there is a clear pattern acknowledging the fact that men perform better in marathon than women.

```
r15 %>% filter(gender != 'NA') %>%
ggplot(aes(x = factor(gender, levels = unique(gender)) , y = official_time)) +
  geom_boxplot(fill = "royalblue2", outlier.colour = "darkorange1") +
  facet_grid(wave~.) +
  coord_flip() +
  ylab("Official Time") +
  xlab("Group") +
  ggtitle("Different conclusions for each group") +
  theme(plot.title = element_text(hjust = 0.5, size=15)) +
```

```
theme_bw()
```




The results are quite striking and diverse: * `first group`: men significantly do better than women. * `second group`: men and women perform similarly. * `third group`: men and women perform similarly. * `fourth group`: men significantly do better than women.

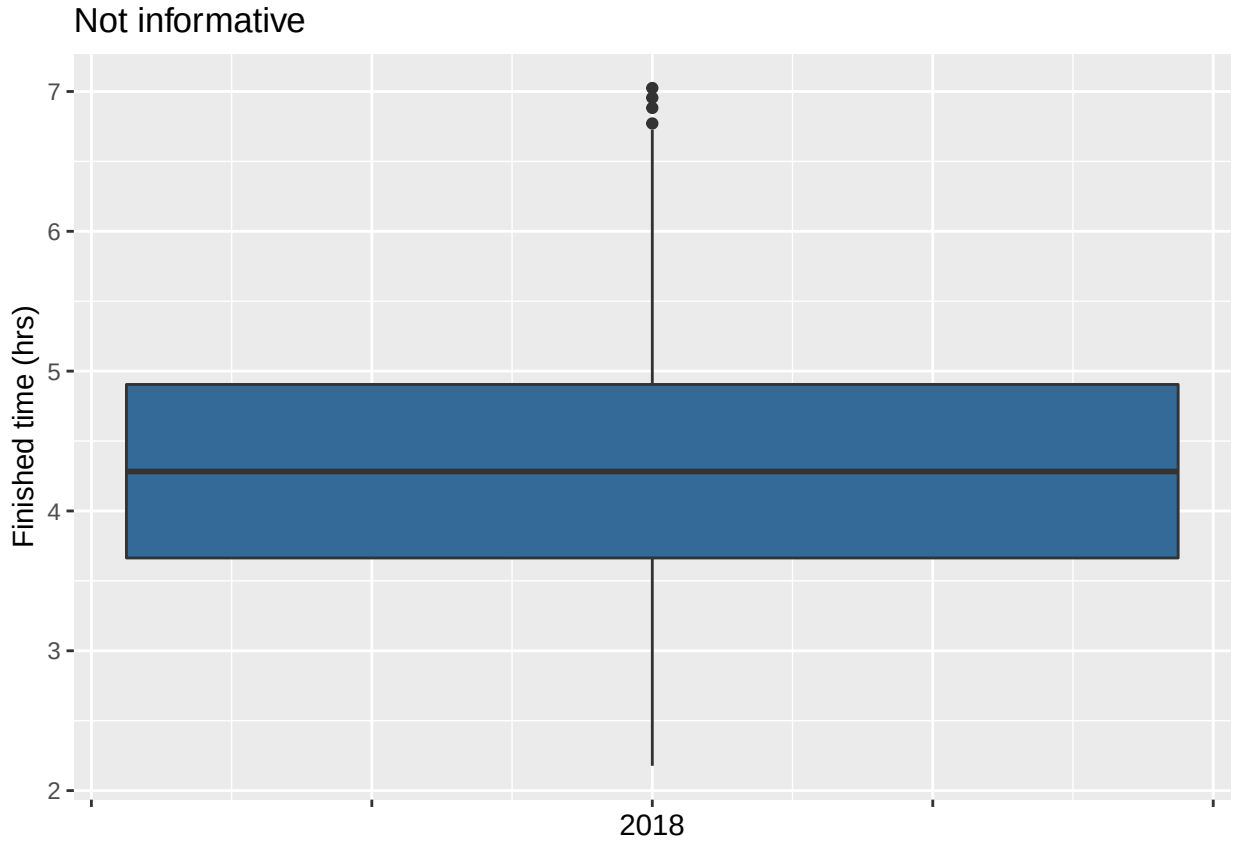
How can we explain those two completely different behaviors of our data? It seems that for the runners in the intermediary groups, gender does not count when it comes to the race performance. Those runners have some prior experience and/or have trained regularly to achieve those results. It lets us think that **training can erase physiological differences**. The first group case is quite easy to interpret. **For athletes, training means pushing the body to its limits**. Therefore, it is not so surprising that this group exhibit a strong performance dependence on gender, since we find that dependence in almost all athletic disciplines.

## 4.3 Summarizing the whole race in a graph

We wanted to understand how we could summarize the race in a single graph. This was a challenge because we were unsure of what to graph to choose: should it be a histogram a box plot. Moreover, what variable

should we used. The `official_time` but what about the performance between the different splits? Maybe some athlete had a strong start but finished poorly. Our first take to this problem was to generate a box plot of the running time.

```
gbox <- ggplot(data = df_embed) +
  geom_boxplot(aes(x = year, y = sec_8 / 3600, fill = year)) +
  ggtitle('Not informative') +
  ylab('Finished time (hrs)') +
  theme(legend.position="none",
        axis.text.x = element_blank()) +
  xlab('2018')
gbox
```



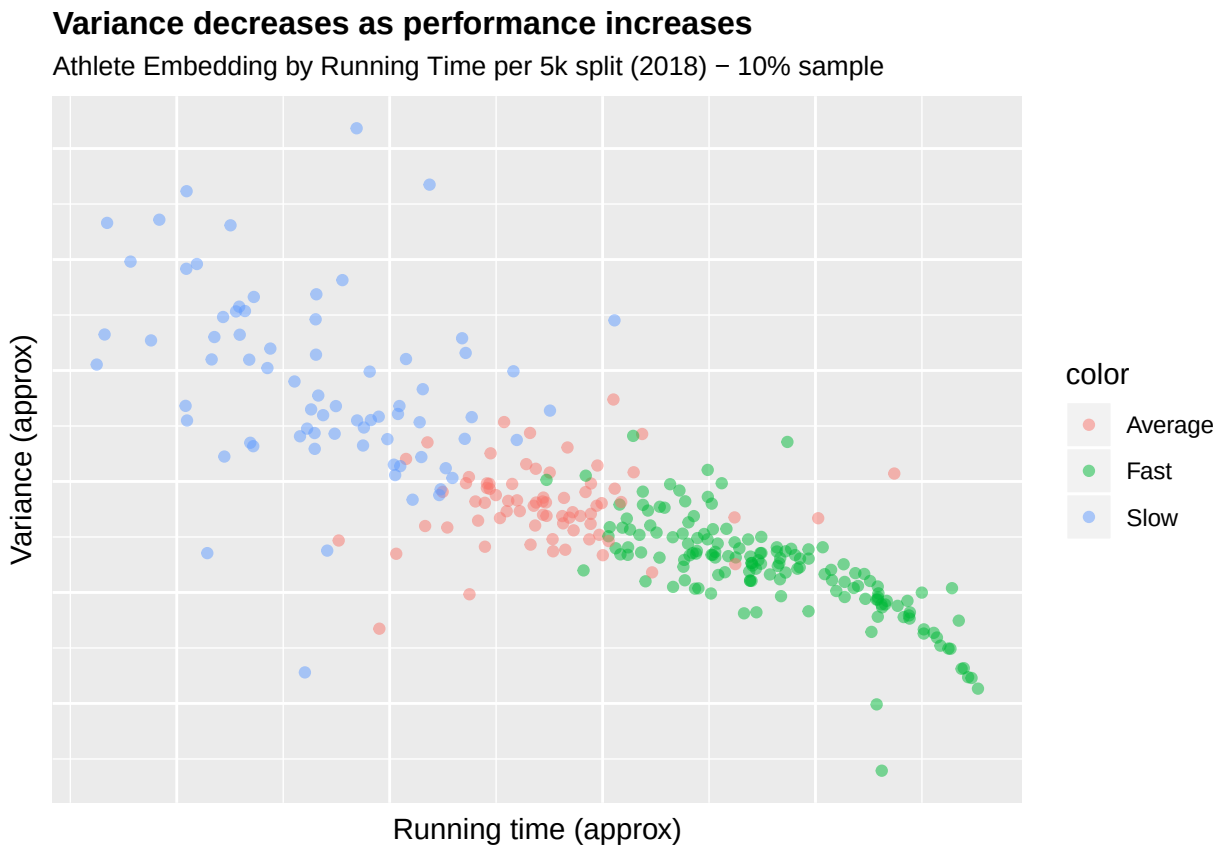

as we see above, the box plot is not as informative as we would like. It tells us that the finishing time was over 4 hours, where the mass concentrates (from slightly below 4 hours to 5 hours) and that the max and min. But we were not satisfied. Then, based on one of our ML homeworks, we saw that there was a technique that could go from distances between the observations to a two dimensional embedding. As an illustration, this technique could take the distances from all the cities within the US and recover the whole map

[][Add the distance matrix and the map recovered... maybe]

where we see how New York and Boston are bunched together were as Seattle and Miami are thrown to two different corners. Thus, we thought that maybe this two dimensional embedding would be a useful summary of our data. The only missing ingredient was to define what the distance between different runners meant.

We found that the most useful summary resulted when we defined the distance between runners as the difference between each of its 5 k splits. The resulting embedding that we got for 2018 and only 10 % of the data is

```
df_embed$color = 'Fast'
df_embed$color[df_embed$sec_8 > quantile(df_embed$sec_8, 0.5)] = 'Average'
df_embed$color[df_embed$sec_8 > quantile(df_embed$sec_8, 0.75)] = 'Slow'
g <- ggplot(data = df_embed, mapping = aes(x = x, y = y)) +
  geom_point(aes(color=color), alpha = 0.5) +
  xlab('Running time (approx)') +
  ylab('Variance (approx)') +
  labs(title = 'Variance decreases as performance increases',
       subtitle = 'Athlete Embedding by Running Time per 5k split (2018) - 10% sample') +
  theme(plot.title = element_text(size = 12, face = "bold")) +
  theme(plot.subtitle = element_text(size = 10),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.y=element_blank())
g
```




This is really cool! Before going into the details of the graph it is worth mentioning a couple of points. **The x and y axis are not the actual variables displayed in the labels but rather an approximation or interpretation** (it is like in PCA, where the two principal components are not a single variable but rather a combination of many that might render a certain interpretation). Thus, as seen from the graph **the x-axis displays the running time** where the time increases as you move to the left; the first green dot to the right was the fastest runner. It is worth pointing out that I colored the data points to show that this interpretation holds, but the embedding did what it thought best. Also, fast (green) means that your overall running time was below the median, average (red) means that you were above the median and below the 75% percentile and finally slow (blue) are the remaining points.

In terms of **the y-axis**, the interpretation is that this coordinate **captures the variance** in running time per 5 k split. Since points that had dissimilar performance on each split get separated, then the more dissimilar you are on each 5 k split to the athletes that had a similar running time, the further away you get pulled from them. As an illustration, take the green dot that is closest to the x-axis. That athlete had a great overall running time but the reason that it got thrown away from the elite "pack" is that it did not maintain a constant pace in all the splits as the rest of the top performers. What happened was that this athlete decreases severely its speed at the last 5 k split and that is why it got separated.

Given the introduction to this new type of graph, the main insights that we got are the following. Look how **the performance differences decelerate as we move to the right**. The graph resembles a logarithm curve that has been rotated on the x-axis. Based on this curvature, we see that a change for the green dots move you further above than in the red dots. **Thus, for the faster runners small perturbations in their running times sets them further apart than for slower runners**. This make a lot of intuitive sense since a difference in minutes is more detrimental and propagates a higher overall performance for the top athletes that for the rest. The second insight is that **the race was tight**. Look how close are the first points are. It looks as if they were on top of each other. Finally, we see something that we expected. **As performance decreases variance increases**. The green dots are close together because they maintain a constant pace and their finishing time was similar. In contrast, the slow runners kept changing pace on each split and therefore the cloud of points is more spread around that.

In summary, **we find the previous graph a powerful summary of the race. It tells you how tight the race was but also how much variance is there for the different running groups!** Just for completeness we add the mathematical formulation. The problem that the embedding solves is

$$\min_{x} \sum_{i,j} \left( \|x_i - x_j\|_2 - D_{ij} \right)^2$$

where $D_{ij}$ stands for the difference in running times on each 5k split for runner $i$ and runner $j$ and the $x$s constitute the two dimensional embedding. Hence, we see how to minimize the previous objective function we have to make the difference in representation of each runner $x_i$ and $x_j$ match their actual distance $D_{ij}$.

# 5. Executive Summary

[][Write the conclusion – need to have all the graphs from the previous section]

# 6. Interactive component

[][Add the link to the interactive component] [][Explain how to use the visualization. What to move, what to look for, etc]

# 7. Conclusion

[][Discuss limitations, lessons learned and future directions]

- What problems did we face...
- Lessons learned...
- What do we want to further do with the data? Maybe add more years, maybe parse the state/countries better, maybe do team analysis...

## Questions

1. Is gender a significant indicator of performance? (Arthur)
2. How could we summarize the whole race in a graph? (Andres)
3. How has the gender ratio changed over time? (Andrea & Antonia)
4. How do performance change by location? (Andrea & Antonia)