DSC 385 - Project 1 Report

Name and EID

Marina Witherell
EID: mw44788

Setup

```
## Loading all the data files
## DO NOT CHANGE THIS CODE
championships <- read_tsv("WCA_export_championships.tsv.bz2")
competitions <- read_tsv("WCA_export_Competitions.tsv.bz2")
continents <- read_tsv("WCA_export_Continents.tsv.bz2")
countries <- read_tsv("WCA_export_Countries.tsv.bz2")
eligible_country_iso2s_for_championship <- read_tsv("WCA_export_eligible_country_iso2s_for_c
events <- read_tsv("WCA_export_Events.tsv.bz2")
formats <- read_tsv("WCA_export_Formats.tsv.bz2")
persons <- read_tsv("WCA_export_Persons.tsv.bz2")
ranksaverage <- read_tsv("WCA_export_RanksAverage_333.tsv.bz2")
rankssingle <- read_tsv("WCA_export_RanksSingle_333.tsv.bz2")
results <- read_tsv("WCA_export_Results_333.tsv.bz2")
roundtypes <- read_tsv("WCA_export_RoundTypes.tsv.bz2")
scrambles <- read_tsv("WCA_export_Scrambles.tsv.bz2")
```

Required Questions

Active Speed Cubers

How many active (3x3x3) speedcubers are there registered with the WCA?

For this question an *active speeedcuber* is defined as any person registered in the WCA who has competed in at least two competitions in the years 2022–2024.

```
# Make a list of valid years
active_years <- c("2022", "2023", "2024")

# Filter competitions by valid years and count the people who have competed at least twice
left_join(results, competitions, by= c("competitionId"="id")) |>
  filter(str_detect(competitionId, paste(active_years, collapse = "|"))) |>
  count(personId) |>
  count(n>1)
```

```
# A tibble: 2 x 2
  `n > 1`     n
  <lgl>   <int>
1 FALSE   31705
2 TRUE    50440
```

Answer: There are 50,440 active (3x3x3) speedcubers registered with the WCA.

World Records

This question has two parts:

1. According to the data, who holds the world record single best solve? On what date was this record set?

```
# Join results, competitions, and rankssingle to get the person who holds the record
left_join(results, competitions, by= c("competitionId"="id")) |>
  inner_join(rankssingle, by="personId") |>
  filter(best.x>0) |>
  slice_min(order_by = best.x) |>
  pull(personName)
```

```
[1] "Max Park"
```

Answer: Max Park

On what date was this record set?

```r
# Join results, competitions, and rankssingle to get the date the record was set
record_date <- left_join(results, competitions, by= c("competitionId"="id")) |>
  inner_join(rankssingle, by="personId") |>
  mutate(date = make_date(year=year,month=month,day=day)) |>
  filter(best.x>0) |>
  slice_min(order_by = best.x) |>
  pull(date)
record_date
```

```
[1] "2023-06-11"
```

Answer: June 11, 2023

2. According to the data, who previously held the world record best single solve?

```r
# Join results, competitions, and rankssingle and filter records set before the current reco
left_join(results, competitions, by= c("competitionId"="id")) |>
  inner_join(rankssingle, by="personId") |>
  mutate(date = make_date(year=year,month=month,day=day)) |>
  filter(date < record_date) |>
  filter(best.x>0) |>
  slice_min(order_by = best.x) |>
  pull(personName)
```

```
[1] "Yusheng Du ( )"
```

Answer: Yusheng Du

On what date was this previous record set?

```r
# Join results, competitions, and rankssingle and filter records set before the current reco
left_join(results, competitions, by= c("competitionId"="id")) |>
  inner_join(rankssingle, by="personId") |>
  mutate(date = make_date(year=year,month=month,day=day)) |>
  filter(date < record_date) |>
  filter(best.x>0) |>
  slice_min(order_by = best.x) |>
  pull(date)
```

```
[1] "2018-11-24"
```

Answer: November 24, 2018

NOTE: For these questions, consider all speedcubers (not just active ones) and define "best" as the fastest time for a single solve (not for an average).

Regional Rankings

This question has two parts:

1. Amongst all speedcubers, who is the top ranked male speedcuber (for best single solve) in Australia?

```
# Combine rankssingle and persons and filter appropriately
left_join(rankssingle, persons, by= c("personId"="id")) |>
  filter(gender == "m") |>
  filter(countryId == "Australia") |>
  slice_min(order_by = best) |>
  pull(name)
```

```
[1] "Jode Brewster"
```

Answer: Jode Brewster

2. Amongst all speedcubers, who is the top ranked female speedcuber (for best single solve time) in Europe?

NOTE: Europe is identified under the `continentId` column of the `countries` table.

```
# Combine rankssingle, persons, and countries and filter appropriately
left_join(rankssingle, persons, by= c("personId"="id")) |>
  left_join(countries, by=c("countryId"="id")) |>
  filter(gender == "f") |>
  filter(continentId == "_Europe") |>
  slice_min(order_by = best) |>
  pull(name.x)
```

```
[1] "Magdalena Pabisz"
```

Answer: Magdalena Pabisz

Time Until Sub-5

Having a time below 5 seconds is considered an elite achievement and most speedcubers have to complete a large number of solves before they can obtain a sub-5 second solve.

1. For the current top 10 speedcubers in the world (as recorded in the `RanksSingle` table), on average, how many solves did they have to do before achieving a sub-5 second solve?

NOTE: Each round of a competition has 5 solves that should be considered separately when counting the number of solves.

```
# Get the ids
personId10 <- rankssingle |>
  arrange(best) |>
  slice(1:10) |>
  pull(personId)

# make a helper function to count the number of solves
count_solves <- function(data) {
  num_solves = 0
  for (i in 1:nrow(data)){
    for (j in c("value1", "value2", "value3", "value4", "value5")){
      if (data[i,j] > 500){
        num_solves <- num_solves+1
      }
      else{
        if(data[i,j]>0){
          return(num_solves)
        }
      }
    }
  }
}

# Combine results, competitions, roundtypes, and rankssingle and sort by date
new_df <- left_join(results, competitions, by= c("competitionId"="id")) |>
  left_join(roundtypes, by= c("roundTypeId"="id")) |>
  left_join(rankssingle, by="personId") |>
  mutate(date = make_date(year=year,month=month,day=day)) |>
  arrange(date, rank)

# Find the number of solves needed per person
solves = list()
```

```
for(person in personId10) {
  indiv_df <- filter(new_df, personId == person)
  num_solves <- count_solves(indiv_df)
  solves <- append(solves, num_solves)
}

# Find the average number of solves
average_solves <- mean(unlist(solves))
average_solves
```

```
[1] 502
```

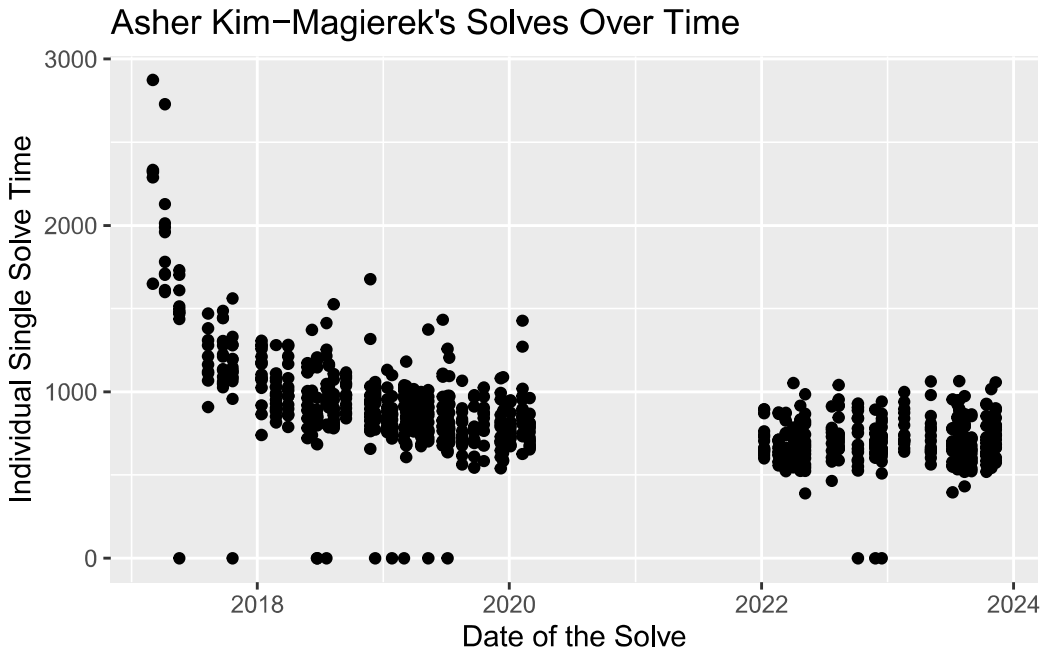
Answer: On average, the top 10 speedcubers needed to do 502 solves before achieving a sub-5 second solve.

2. For **one** of the top 10 speedcubers make a scatterplot of their individual single solve times vs. the date of the solve, with date on the x-axis and solve time on the y-axis.

```
# Helper function to get the solve times
get_solves <- function(data) {
  my_solves = data.frame(
    solveDate = double(0),
    solveTime = numeric(0)
  )
  for (i in 1:nrow(data)){
    for (j in c("value1", "value2", "value3", "value4", "value5")){
        val <- data[i,j] |>
          slice(1) |>
          pull()
        new_solve <- data.frame(solveDate = data$date[i], solveTime = val)
        my_solves <- rbind(my_solves, new_solve)
    }
  }
  return(my_solves)
}

# Plot the solves of one person
new_df |>
  filter(personId == "2017KIMM01") |>
  get_solves() |>
  ggplot(aes(x = solveDate, y = solveTime)) +
```

```
  geom_point() +
  labs(
    title = "Asher Kim-Magierek's Solves Over Time",
    x = "Date of the Solve",
    y = "Individual Single Solve Time"
  )
```



Up-and-Coming Speed Cubers

Which speed cubers **not** in the top 10,000 (worldwide for single best time) should we keep an eye on for the near future?

The idea here is to identify "up-and-coming" speedcubers who are not yet achieving elite times. Come up with a list of **five** speedcubers (provide their names and WCA IDs) that you have identified as "up-and-coming". There is no one way to answer this question and the goal is to provide an analysis of the data that justifies the selection of your five names.

```
# Make a dataframe of active speed cubers per the question 1 standards
active_df <- new_df |>
  filter(str_detect(competitionId, paste(active_years, collapse = "|"))) |>
  count(personId) |>
  filter(n>1) |>
```

```
  arrange(desc(n))

# Make a list of people that are active and in the top 10,500, but not the top 10,000
promising_ppl <- inner_join(rankssingle, active_df, by="personId") |>
  slice(10001:10500) |>
  pull(personId)

# Helper function to fit a linear model to each persons average time by date
get_slope <- function(data) {
  model <- lm(average ~ date, data = data)
  slope <- coef(model)[2]
  return(as.integer(slope))
}

# Make a data frame that contains a person's ID and their average improvement over time
up_and_coming = data.frame(
    improvement = double(0),
    personId = character(0)
  )

# Fit a linear model of improvement for each person that seems promising (determined above)
for(person in promising_ppl) {
  indiv_df <- new_df |>
    filter(personId == person & year >2019)
  slope<- get_slope(indiv_df)
  new_stats <- data.frame(improvement = slope, personId = person)
  up_and_coming <- rbind(up_and_coming, new_stats)
}

# Find the top 5 promising people with the most improvement
up_and_coming |>
  arrange(improvement) |>
  slice(1:5)|>
  inner_join(new_df, by="personId") |>
  select(personName, personId) |>
  unique()
```

```
                  personName   personId
1          Hans Dominic Gamboa 2018GAMB02
6                  Conan Chang 2023CHAC08
10              Gaspard Machut 2021MACH05
15               Itamar Sigeti 2022SIGE01
```

`24 Vadhir Roberto Lopez Ortega 2022ORTE06`

Answer: The 5 people I determined to be up-and-coming speed cubers are Hans Dominic Gamboa (2018GAMB02), Conan Chang (2023CHAC08), Gaspard Machut (2021MACH05), Itamar Sigeti (2022SIGE01), and Vadhir Roberto Lopez Ortega (2022ORTE06).

Reasoning: I chose to find people that were not in the top 10,000, but still highly ranked (the next 500 ranked). Additionally, the people had to be active speedcuber competitors (per the specifications of Question 1). Therefore, these are the next highest ranked people that are currently competing. Then, using a linear model, I fit their average times in each competition by their date, in order to see if they were improving (achieving lower solve times) over time. I did specify this to be the improvement in the last five years so that it found those improving the most rapidly the most recently. Finally, I sorted the people by their improvement, with the most negative slope representing the highest improvement over time. From those, I chose the top five competitors to deem up-and-coming.

Discussion

Reflection: This project was challenging compared to the homeworks. The most challenging part for me was utilizing piping, since that is what we have mostly used in this class. However, I don't have much experience with it and ended up using some helper functions as well as just assigning variables because it made more sense in my head, especially when the `for` loops came into play. I definitely learned more about piping and types of joins while doing this project, which is very helpful information.

Acknowledgements: I referenced Stack Overflow and GeeksforGeeks frequently for reminders on basic dataframe operations, as well as piping operations. This was very helpful, especially since I've been working in Python the most frequently recently and sometimes need a syntax reminder.