Technical Write Up

Andy Ki, Shu Amano

2018/12/18

Contents

Document Options

Load Packages

```
require(mosaic)
require(tidyverse)
require(lubridate)
require(rvest)
require(lme4)
```

Data

Load Datasets

```
regions <- read_csv("Data/NOC_Region.csv")
athletes <- read_csv("Data/Olympics_Athletes.csv")
countrycode <- read_csv("Data/ISOCountryCode.csv")
temphistory <- readxl::read_xlsx("Data/Historical_Temp_Data.xlsx", sheet = 2)
olymweat <- readxl::read_xlsx("Data/Olympics_Weather_Data.xlsx")
inout <- Class.csv <- read_csv("Data/indooroutdoorClass.csv")
```

Wrangling elevation data

```
# Create function to extract elevation data from wikipedia
wikiTable <- function(source) {
  read_html(source)%>%
    html_nodes("table.wikitable") %>%
    html_table(fill=T)%>%
    magrittr::extract2(1)
}

# Read in wikipedia table for average elevation per country
elevation <- wikiTable("https://en.wikipedia.org/wiki/List_of_countries_by_average_elevation#cite_note-

# Clean up format of elevation value
elelist <- strsplit(elevation$Elevation, "m")
vec <- c()
for (i in 1:length(elelist)){
  vec[i] <- elelist[[i]][1]
}
elevation$Elevation <- vec
elevation <- elevation %>%
  mutate(Elevation = parse_number(Elevation))

# Change format of name of countries
pos <- which(elevation$Country %in% c("Trinidad and Tobago","United Kingdom","United States"))
elevation[pos,1] <- c("Trinidad","UK", "USA")
```

Combine countrycode and temphistory data

```
# Join countrycode and temphistory
countrytemp <- left_join(temphistory, countrycode, by = c("ISO_3DIGIT" = "alpha-3")) %>%
  select(-c(16:24), -`ISO_3DIGIT`) %>%

# Rename each month column so that wrangling is easier later on
  plyr::rename(c("Jan_Temp" = "1", "Feb_temp" = "2",
                 "Mar_temp" = "3", "Apr_Temp" = "4",
                 "May_temp" = "5", "Jun_Temp" = "6",
                 "July_Temp" = "7", "Aug_Temp" = "8",
                 "Sept_temp" = "9", "Oct_temp" = "10",
                 "Nov_Temp" = "11", "Dec_temp" = "12"))
```

Joined country code and temperature history by ISO, so that each average monthly temperature was linked to a country name instead of the ISO country code. We reformatted some of the the names used in the the created dataset to match the country names used in other datasets. An example would be reformatting "United States of America" to USA. We called this new dataset "countrytemp".

Clean up countrytemp

```
# Rename some countries to standardize country name format
countrytemp <- countrytemp %>%
  mutate(name = ifelse(name == "Bolivia (Plurinational State of)", "Bolivia", name),
```

```
         name = ifelse(name == "Czechia", "Czech Republic", name),
         name = ifelse(name == "Korea (Republic of)", "South Korea", name),
         name = ifelse(name == "Russian Federation", "Russia", name),
         name = ifelse(name == "Trinidad and Tobago", "Trinidad", name),
         name = ifelse(name == "United Kingdom of Great Britain and Northern Ireland", "UK", name),
         name = ifelse(name == "United States of America", "USA", name),
         name = ifelse(name == "Venezuela (Bolivarian Republic of)", "Venezuela", name),
         name = ifelse(name == "Viet Nam", "Vietnam", name))
```

Cleaned up the names of countries in countrytemp.

Clean up dates of olymweat

```
# Create new variables for starting/ending months and days
olymweat2 <- olymweat %>%
  rename("StartDate" = `Start Date`,
         "EndDate" = `End Date`) %>%
  mutate(StartDate = as.character(StartDate),
         EndDate = as.character(EndDate),
         StartMonth = strsplit(StartDate, split = "-")[[1]][2],
         EndMonth = strsplit(EndDate, split = "-")[[1]][2],
         StartDay = strsplit(StartDate, split = "-")[[1]][3],
         EndDay = strsplit(EndDate, split = "-")[[1]][3]) %>%

# Get rid of original StartDate and EndDate columns
  select(-StartDate, -EndDate)
```

Next, we made changes to the dates contained in the olymweather's dataset. We created new variables for the starting month, ending month, starting day, ending day of the each olympics game. We called this "olymweather2".

Combine athletes and region

```
athletes2 <- athletes %>%
  left_join(regions, by = c('NOC' = 'NOC')) %>%
  select(-notes, -NOC, -Games)
```

We then joined the athletes dataset with regions dataset by NOC so that each athlete had a corresponding regions that they were from. These regions were country. We called this dataset "athletes2".

Combine olymweat2 and athletes and filter out indoor sports

```
# Join athletes2 and olymweat2 dataset
olympics1 <- left_join(athletes2, olymweat2, by = c("Year" = "Year", "Season" = "Season")) %>%
  select(-City.y) %>%
  rename(City = City.x) %>%

# Filter out Olympics that do not follow modern format
  filter(!(Year %in% c(1896,1900,1904,1906,1908,1912,1920,1924,1928) & Season == "Summer")) %>%
  mutate(CityTemp1 = TempMon1,
```

```
         CityTemp2 = TempMon2) %>%

# Join inout dataset that contains information about indoor/outdoor sports
  left_join(inout, by = c("Sport" = "Sport")) %>%

# Filter out indoor sports
  filter(isindoor == 0) %>%

# Select out columns without further use
  select(-ID, -Sex, -Age, -Height, -Weight, -Team, -Event, -TempMon1, -TempMon2)
```

We then joined countrytemp and athletes2 by Year and Season, then joined it further with inout by Sport. The created dataset had each Year, Season, Host City, Sport, Medal, region (Country), Altitude of Host City, Starting Month, Ending Month, Starting Day, Ending Day, Number of Days, Host City Temperature of Month1 and Host City Temperature of Month2 and isIndoor for every single athlete that competed in the olympics in the past.We then filtered out athletes from older Olympics (1896,1900,1904,1906,1908,1912,1920,1924,1928) because they were hosted over longer periods (50+ days), which is different from more modern Olympics which are hosted over around 20 days. We also filtered out athletes competing in indoor sports since we thought indoor sports aren't affected by temperature or elevation difference. We called this dataset "olympics1".

Combine olympics1 with countrytemp

```
olympics2 <- left_join(olympics1, countrytemp, by = c("region" = "name"))

# create vector of the starting and ending months of each olympic
startmon <- as.character(as.integer(olympics1$StartMonth))
endmon <- as.character(as.integer(olympics1$EndMonth))

# Loop through olympics2 to gather the temperature of months that corresponds to the olympic months for
# Store this data in histtemp1 and histtemp2
histtemp1 <- c()
histtemp2 <- c()
for (i in 1:nrow(olympics2)){
  histtemp1[i] <- olympics2[[i,startmon[i]]]
  histtemp2[i] <- olympics2[[i,endmon[i]]]
}

# Create new column corresponding to historic temperature of home countries in each Olympics
olympics2$histtemp1 <- histtemp1
olympics2$histtemp2 <- histtemp2

# Select out columns without further use
olympics2 <- olympics2 %>%
  select(-`1`,-`2`,-`3`,-`4`,-`5`,-`6`,-`7`,-`8`,-`9`,-`10`,-`11`,-`12`, -Annual_temp)
```

We next joined olympics1 with countrytemp. We then filtered out all the historic average monthly temperature of the athlete's home country if they did not correspond to the Olympic months. So for example, if an athlete competed in an Olympic held from August 23rd to September 7th, then the dataset would contain historic average temperature for August and September, in addition to all the information contained in olympics1. We called this dataset "olympics2".

Calculate difference between host city temperature and home country temperature

```
# Calculate the ratio of days in the first and second month
olympics3 <- olympics2 %>%
  mutate(NumDayMon1 = ifelse(as.integer(EndDay) > `Number of Days`,
                             `Number of Days`,
                             `Number of Days` - as.integer(EndDay)),
         NumDayMon2 = `Number of Days` - NumDayMon1,
         RatioMon1 = NumDayMon1/`Number of Days`,
         RatioMon2 = 1 - RatioMon1,
         CityTemp2 = ifelse(is.na(CityTemp2), 0, CityTemp2)) %>%

# Calculate the total difference
  mutate(tempdiff = (CityTemp1 - histtemp1)*RatioMon1 + (CityTemp2 - histtemp2)*RatioMon2) %>%

# Select out columns without further use
  select(-CityTemp1, -CityTemp2, -histtemp1, -histtemp2, -NumDayMon1,
         -NumDayMon2, -RatioMon1, -RatioMon2, -`Number of Days`, -StartMonth,
         -EndMonth, -StartDay, -EndDay, -isindoor)
```

We then used olympics2 to calculate the temperature difference between the host city and home country. If the Olympics were held in just one month, the calculation was simply subtracting the home country temperature from host city temperature. If the Olympics were held across two months, the calculation was a bit more complex. We first calculated the temperature difference for both months and added the weighted sum. The weights were calculated by the ratio of days in each months. We called this dataset "olympics3".

Combine elevation data with olympics3 and compute elevation difference

```
# Join olympics3 and elevation dataset
olympics4 <- olympics3 %>%
  left_join(elevation, by = c("region" = "Country")) %>%

# Calculate average elevation difference
  mutate(elevdiff = Altitude - Elevation) %>%

# Select out columns without further use
  select(-Elevation, -Altitude)
```

We next joined olympics3 and the elevation dataset and computed the elevation difference for each athletes. We called this dataset "olympics4".

Clean up medal data

```
# Quantify each medals as follows: Gold = 3, Silver = 2, Bronze = 1, None = 0
olympics5 <- olympics4 %>%
  mutate(Medal = ifelse(is.na(Medal), "0", Medal),
         Medal = ifelse(Medal == "Bronze", "1", Medal),
         Medal = ifelse(Medal == "Silver", "2", Medal),
         Medal = ifelse(Medal == "Gold", "3", Medal),
         Medal = as.integer(Medal))
```

We then quantified each medals using the following point system: NA = 0, Bronze = 1, Silver = 2 and Gold = 3. We used this point system because looking at sports literature, this was the most commonly used. For example "https://www.topendsports.com/events/summer/medal-tally/rankings.htm". We called the dataset after the quantification "olympics5".

Create dataset that contains total medals for each olympics

```
# Group by Year and Season and take the sum of Medal
totalmedals <- olympics5 %>%
  group_by(Year,Season) %>%
  summarise(TotMed = sum(Medal))
```

Using olympics5, we calculated the total medal points awarded in each Olympics. To do this, we first grouped by Year and Season, then took the sum of all medal points. An important point to note is that the medal points only include medals awarded in outdoor sports. We called this dataset "totalmedals".

Calculate percentage of medals won by each country

```
# Calculate the sum of all medals won by each country in each olympics
olympics6 <- olympics5 %>%
  group_by(Year, Season, region) %>%
  summarise(medalswon = sum(Medal)) %>%

# Calculate the proportion of total medals that each country won in each olympics
  left_join(totalmedals, by = c("Year" = "Year", "Season" = "Season")) %>%
  mutate(medratio = medalswon / TotMed)
```

Next, we grouped by Year, Season and Region in olympics5 and calculated the sum of medals. We leftjoined the totalmedals dataset onto the result and calculated the proportion of medals (medal ratio) won by each country in each Olympics. We called this dataset "olympics6". In essence, olympics6 has medal ratios for every country for every olympics.

Get rid of distinctions between each athletes and focus on country level performance

```
# Get rid of all individual distinctions between athletes and focus on country
olympics7 <- olympics5 %>%
  select(Year, Season, region, tempdiff, elevdiff)

# There are multiple athletes from the same countries in every olympics, creating duplicates
# Find which rows has duplicates in olympics7
duplicated <- olympics7 %>%
  duplicated()

# Get rid of duplicates
olympics8 <- olympics7[!duplicated,]
```

Here, we return to olympics5. We created olympics7 by selecting only Region, Year, Season, Temperature Difference and Elevation Difference. We also created a dataset called "duplicate" since olympics7 has a lot of duplicate rows that needs to be taken out. We called the dataset created after taking out duplicate values

olympics8. In essence, olympics8 had the temperature difference and elevation difference for all countries for every Olympics.

Combined join medalratio data onto temperature/elevation data

```
# Join dataset that contains medalratio info onto olympics8
olympics9 <- olympics6 %>%
  left_join(olympics8, by = c("Year" = "Year", "Season" = "Season", "region" = "region")) %>%

# Select out columns that does not have further use
  select(-medalswon, -TotMed)
```

After leftjoining olympics8 onto olympics6, we were able to create the final dataset, olympics9.