R intensive

AM session II

May 14th, 2024

Goal

- Gain hands-on experience on **getting to know** and **tidying up** the data, to make the data ready for analysis. We will cover
  - Reading/writing the data
  - A quick check on the data
  - Tidy up the data using tidyverse (dplyr) functions
  - Advanced data wrangling

Content

I. Read the data
   a. read_csv
II. A quick check on the data
   a. janitor::clean_names
   b. summary, skim, count
III. Data wrangling I (data organization)
   a. arrange, filter, mutate, select, group_by, summarize
IV. Data wrangling II (medals, athletes)
   a. if_else, case_when, group_by and summarize, save
V. Trivia time!

120 years of Olympic history: athletes and results.

- This is a historical dataset on the modern Olympic Games, including all the Games from Athens 1896 to Rio 2016.
- This includes basic bio data (age, sex, height, weight) of athletes and medal results.

- https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results
- Original source: www.sports-reference.com

Data (athlete_events.csv)

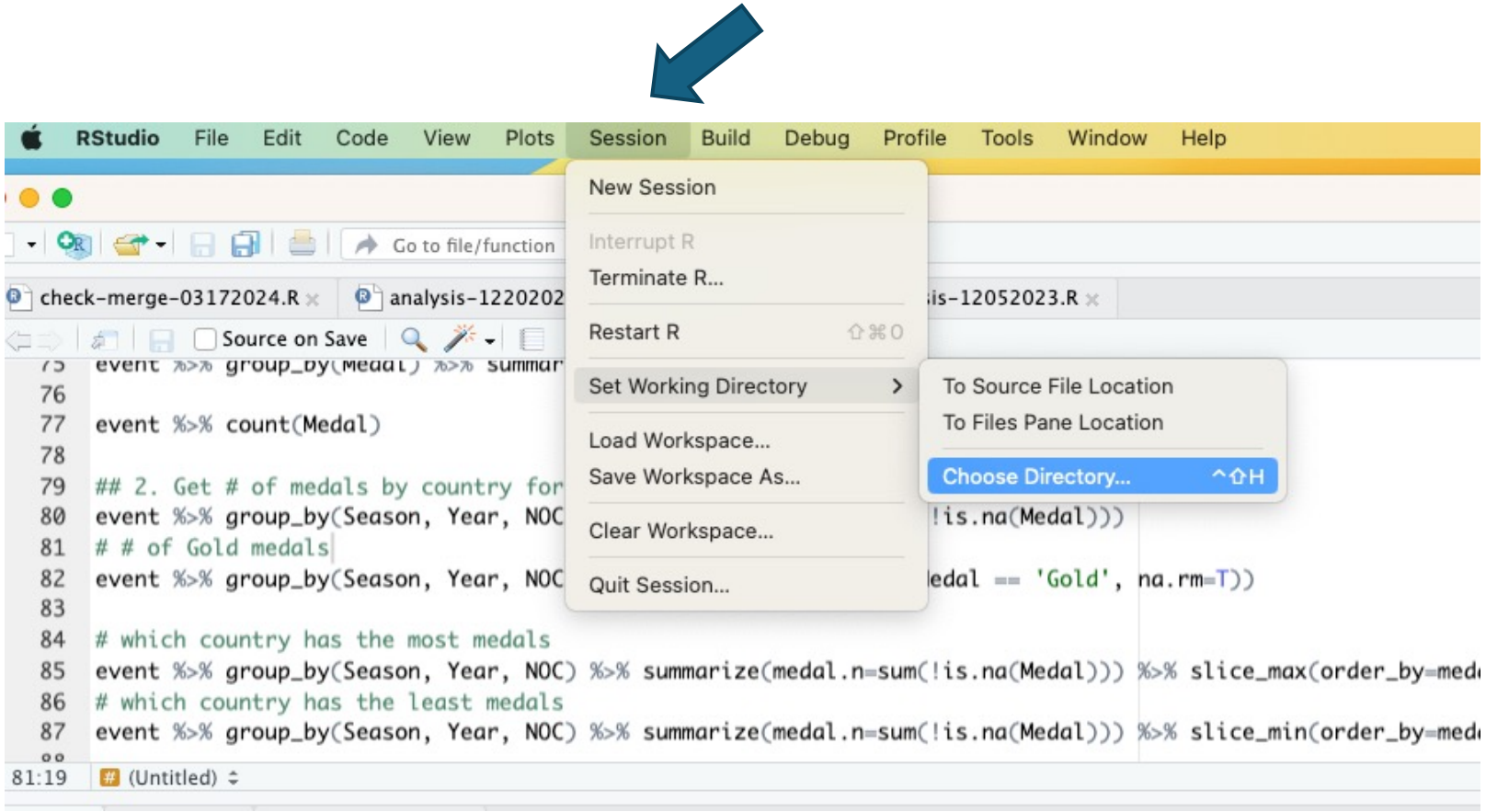
271,116 rows and 15 columns.
Each row corresponds to an individual athlete competing in an individual Olympic event.

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijiang	M	24	180	80	China	CHN	1992 Summ	1992	Summer	Barcelona	Basketball	Basketball M	NA
2	A Lamusi	M	23	170	60	China	CHN	20					udo Men's E	NA
3	Gunnar Niels	M	24	NA	NA	Denmark	DEN	19					ootball Mer	NA
4	Edgar Linden	M	34	NA	NA	Denmark/Sw	DEN	19					ug-Of-War	Gold
5	Christine Jac	F	21	185	82	Netherlands	NED	19					peed Skatin	NA
5	Christine Jac	F	21	185	82	Netherlands	NED	19					peed Skatin	NA
5	Christine Jac	F	25	185	82	Netherlands	NED	19					peed Skatin	NA
5	Christine Jac	F	25	185	82	Netherlands	NED	19					peed Skatin	NA
5	Christine Jac	F	27	185	82	Netherlands	NED	19					peed Skatin	NA
5	Christine Jac	F	27	185	82	Netherlands	NED	19					peed Skatin	NA
6	Per Knut Aal	M	31	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	31	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	31	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	31	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	33	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	33	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	33	188	75	United State	USA	19					ross Countr	NA
6	Per Knut Aal	M	33	188	75	United State	USA	19					ross Countr	NA
7	John Aalberg	M	31	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	31	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	31	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	31	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	33	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	33	183	72	United State	USA	19					ross Countr	NA
7	John Aalberg	M	33	183	72	United State	USA	1994 Winter	1994	Winter	Lillehammer	Cross Countr	Cross Countr	NA

1. **ID** - Unique number for each athlete
2. **Name** - Athlete's name
3. **Sex** - M or F
4. **Age** - Integer
5. **Height** - In centimeters
6. **Weight** - In kilograms
7. **Team** - Team name
8. **NOC** - National Olympic Committee 3-letter code
9. **Games** - Year and season
10. **Year** - Integer
11. **Season** - Summer or Winter
12. **City** - Host city
13. **Sport** - Sport
14. **Event** - Event
15. **Medal** - Gold, Silver, Bronze, or NA

I. Read data

```
# set working directory
setwd('/Users/moon/Dropbox (EinsteinMed)/R workshop/2024')
```



```
# read athlete_event.csv file
events_data <- read_csv('archive/athlete_events.csv')
```

II. A quick check on the data

```
# glimpse the data
glimpse(events_data)
```

```
> glimpse(events_data)
Rows: 271,116
Columns: 15
$ ID      <dbl> 1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7
$ Name    <chr> "A Dijiang", "A Lamusi", "Gunnar Nielsen Aaby", "Edgar Lindenau Aab
$ Sex     <chr> "M", "M", "M", "M", "F", "F", "F", "F", "F", "F", "M", "M", "M", "M
$ Age     <dbl> 24, 23, 24, 34, 21, 21, 25, 25, 27, 27, 31, 31, 31, 31, 33, 33, 33,
$ Height  <dbl> 180, 170, NA, NA, 185, 185, 185, 185, 185, 185, 188, 188, 188, 188,
$ Weight  <dbl> 80.0, 60.0, NA, NA, 82.0, 82.0, 82.0, 82.0, 82.0, 82.0, 75.0, 75.0,
$ Team    <chr> "China", "China", "Denmark", "Denmark/Sweden", "Netherlands", "Neth
$ NOC     <chr> "CHN", "CHN", "DEN", "DEN", "NED", "NED", "NED", "NED", "NED", "NED
$ Games   <chr> "1992 Summer", "2012 Summer", "1920 Summer", "1900 Summer", "1988 W
$ Year    <dbl> 1992, 2012, 1920, 1900, 1988, 1988, 1992, 1992, 1994, 1994, 1992, 1
$ Season  <chr> "Summer", "Summer", "Summer", "Summer", "Winter", "Winter", "Winter
$ City    <chr> "Barcelona", "London", "Antwerpen", "Paris", "Calgary", "Calgary",
$ Sport   <chr> "Basketball", "Judo", "Football", "Tug-Of-War", "Speed Skating", "S
$ Event   <chr> "Basketball Men's Basketball", "Judo Men's Extra-Lightweight", "Foo
$ Medal   <chr> NA, NA, NA, "Gold", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
```

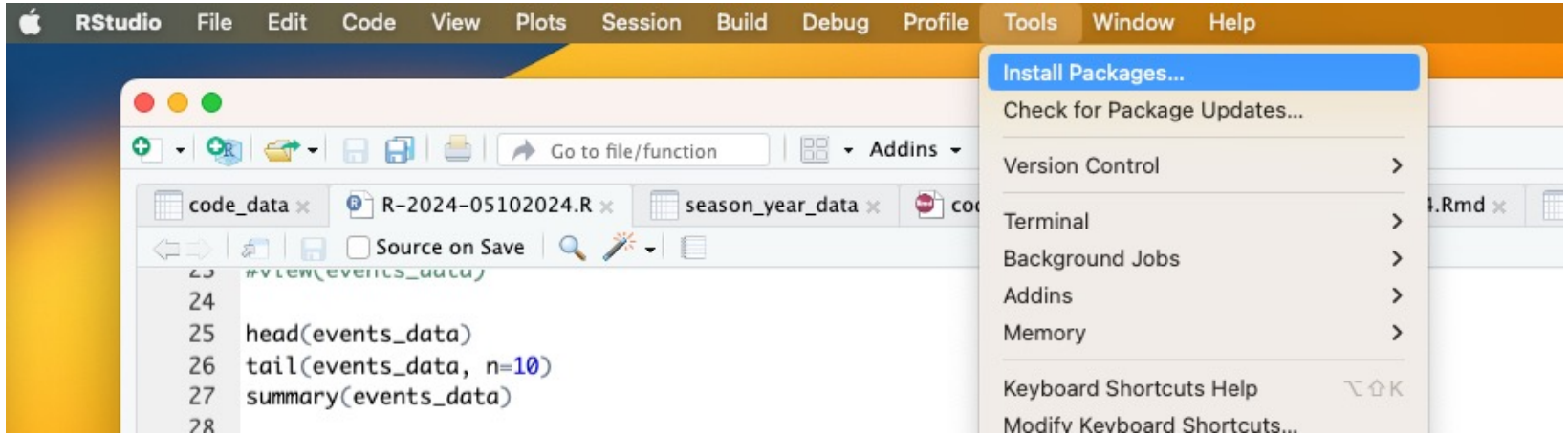
Check numerical variables

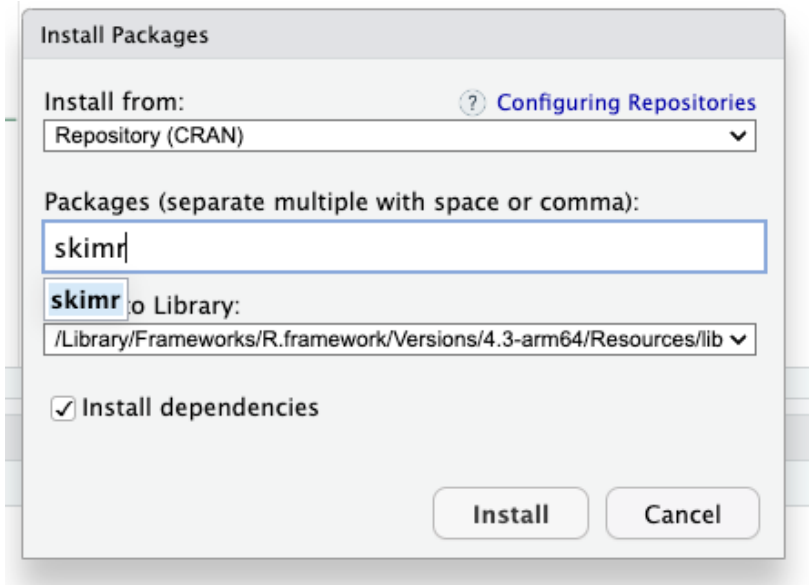
```
> summary(events_data)
       ID             Name               Sex                 Age              Height          Weight          Team              NOC          
 Min.   :     1   Length:271116      Length:271116      Min.   :10.00   Min.   :127.0   Min.   : 25.0   Length:271116      Length:271116     
 1st Qu.: 34643   Class :character   Class :character   1st Qu.:21.00   1st Qu.:168.0   1st Qu.: 60.0   Class :character   Class :character  
 Median : 68205   Mode  :character   Mode  :character   Median :24.00   Median :175.0   Median : 70.0   Mode  :character   Mode  :character  
 Mean   : 68249                                        Mean   :25.56   Mean   :175.3   Mean   : 70.7                                        
 3rd Qu.:102097                                        3rd Qu.:28.00   3rd Qu.:183.0   3rd Qu.: 79.0                                        
 Max.   :135571                                        Max.   :97.00   Max.   :226.0   Max.   :214.0                                        
                                                       NA's   :9474    NA's   :60171   NA's   :62875                                        
    Games                Year         Season              City              Sport              Event              Medal          
 Length:271116      Min.   :1896   Length:271116      Length:271116      Length:271116      Length:271116      Length:271116     
 Class :character   1st Qu.:1960   Class :character   Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Median :1988   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
                    Mean   :1978                                                                                                   
                    3rd Qu.:2002                                                                                                   
                    Max.   :2016                                                                                                   
```

Check categorical variables (text)

```
> events_data %>% count(Sex)
# A tibble: 2 x 2
  Sex        n
  <chr>  <int>
1 F      74522
2 M     196594
> events_data %>% distinct(Sex)
# A tibble: 2 x 1
  Sex
  <chr>
1 M
2 F
```

Install skimr package for a quick skim through





```
# To install skimr package
install.packages('skimr')
# To load skimr package
library(skimr)
# To skim through the data
skim(events_data)
```

```
> skim(events_data)
── Data Summary ────────────────────────
                           Values
Name                       events_data
Number of rows             271116
Number of columns          15
_______________________
Column type frequency:
  character                10
  numeric                  5
________________________
Group variables            None

── Variable type: character ──────────────────────────────────────────────────────────────────────
   skim_variable n_missing complete_rate min max empty n_unique whitespace
 1 Name                  0         1       2 108     0   134731          0
 2 Sex                   0         1       1   1     0        2          0
 3 Team                  0         1       2  47     0     1184          0
 4 NOC                   0         1       3   3     0      230          0
 5 Games                 0         1      11  11     0       51          0
 6 Season                0         1       6   6     0        2          0
 7 City                  0         1       4  22     0       42          0
 8 Sport                 0         1       4  25     0       66          0
 9 Event                 0         1      15  85     0      765          0
10 Medal            231333         0.147   4   6     0        3          0

── Variable type: numeric ────────────────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate    mean       sd   p0   p25   p50     p75   p100 hist
1 ID                    0         1     68249.  39022.      1 34643 68205 102097. 135571 ▇▇▇▇▇
2 Age                9474         0.965    25.6     6.39   10    21    24      28     97 ▇▅▁▁▁
3 Height            60171         0.778   175.     10.5   127   168   175     183    226 ▁▃▇▃▁
4 Weight            62875         0.768    70.7    14.3    25    60    70      79    214 ▁▇▂▁▁
5 Year                  0         1      1978.     29.9  1896  1960  1988    2002   2016 ▁▂▃▅▇
```

Clean up variable names

names(events_data)

events_data <- events_data %>% **janitor::clean_names()**

janitor::clean_names()

Make variable names to be unique, consisting only of letters (a,b,...), numbers (0,1,..), and _ (underscore).

By default, change upper case into lower case.

```
# to save the data into a csv file
write_csv(events_data, file='olympic.csv')
```

Practice

1. Read noc_regions.csv file
2. Clean up variable names
3. Save it as 'olympic_noc_code.csv'
4. A quick check on the data.
    1. How many data points (# of rows) in the data?
    2. How many variables (# of columns) in the data?

More check on the data

- Which Olympic games (when to when) do the data include?

Practice

1. Examine the sport variable
    a. How many records per sport?
    b. How many sports in the data?
2. How many records per sex?

You want to apply multiple functions (operations)

f3(f2(f1(x)))

Through piping (narrative)

x %>% f1() %>% f2() %>% f3()

Now, you got more-or-less familiar with the data!

Let’s handle/wrangle the data to our taste using tidyverse functions.

- do excel jobs in R (filtering, arranging, selecting variables, new variable, summary stat) and more!

III. Data wrangling I (data organization)

We will try the following functions one by one.

arrange()

filter()

mutate()

select()

group_by()

summarize()

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

events_data %>% group_by(games)

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games
1	2008 Summer
2	2008 Summer

id	games
3	2012 Summer

id	games
1	2004 Summer
4	2004 Summer
5	2004 Summer

events_data %>% group_by(games) %>% mutate(n_row=n())

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games
1	2008 Summer
2	2008 Summer

id	games
3	2012 Summer

id	games
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games	n_row
1	2008 Summer	2
2	2008 Summer	2
3	2012 Summer	1
1	2004 Summer	3
4	2004 Summer	3
5	2004 Summer	3

events_data %>% group_by(games) %>% summarize(n_row=n())

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games
1	2008 Summer
2	2008 Summer

id	games
3	2012 Summer

id	games
1	2004 Summer
4	2004 Summer
5	2004 Summer

games	n_row
2008 Summer	2
2012 Summer	1
2004 Summer	3

Practice.

A. Create the data of two sports of your choice, then arrange by year

B. (a) Create the data of number of records and number of unique athletes by game, then (b) calculate the mean number of records and athletes.

C. We want to know the number of countries attended at each Winter Olympic game and summarize (min, max, median, SD) across years

Now, let’s try to go beyond excel jobs.

IV. Data wrangling II

1. **Which country has the most medals at each game?**

   Step 1. Create the number of medal variable

   Step 2. Summarize the number of medals (take the sum across athletes), grouped by country and game

   Step 3. Find the country with the most medals at each game (slice_max)

1-2. When did USA have the most medals?

events_data %>% group_by(games) %>% mutate(n_row=n())

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games
1	2008 Summer
2	2008 Summer

id	games
3	2012 Summer

id	games
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games	n_row
1	2008 Summer	2
2	2008 Summer	2
3	2012 Summer	1
1	2004 Summer	3
4	2004 Summer	3
5	2004 Summer	3

events_data **%>% group_by(games)** %>% summarize(n_row=n())

id	games
1	2008 Summer
2	2008 Summer
3	2012 Summer
1	2004 Summer
4	2004 Summer
5	2004 Summer

id	games
1	2008 Summer
2	2008 Summer

id	games
3	2012 Summer

id	games
1	2004 Summer
4	2004 Summer
5	2004 Summer

games	n_row
2008 Summer	2
2012 Summer	1
2004 Summer	3

events_data

id	games	noc
1	2008 Summer	USA
2	2008 Summer	CAN
3	2012 Summer	CAN
1	2004 Summer	USA
4	2004 Summer	USA
5	2004 Summer	CAN

%>% group_by(games, noc)

1	**2008 Summer**	**USA**
2	2008 Summer	CAN
3	2012 Summer	CAN
1	**2004 Summer**	**USA**
4	2004 Summer	USA
5	**2004 Summer**	**CAN**

%>% summarize(n_row=n())

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

Note that summarize() returns one row (for each grouping).

If there is one grouping variable, no strong reason to keep the grouping structure. By default, drop the grouping structure.

If there are multiple grouping variables, drop the last grouping (.group='drop.last') .

events_data %>% group_by(games, noc) %>% summarize(n_row=n(), **.group**=....)

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

By default

.group = 'drop.last'

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

.group = 'keep'

events_data %>% group_by(games, noc) %>% summarize(n_row=n(), **.group**=....)

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

.group = 'drop'

games	noc	n_row
2008 Summer	USA	1
2008 Summer	CAN	1
2012 Summer	CAN	1
2004 Summer	USA	2
2004 Summer	CAN	1

.group = 'rowwise'

Practice

A. **In terms of medal point system (3 for gold, 2 for silver, and 1 for bronze), which country has the highest points at each game?**

2. Create athletes' demographic data (age, sex, height, weight) at Olympic games. Note that some athletes participated in multiple events at a game.

Step 1. Select athlete-related variables you want to include.

Step 2. Group by games and id

Step 3. Remove duplicated data

Save

```
# Save in .csv file
write_csv(athlete_data2, file='athlete.csv’)

# Save on R object in .rds file
saveRDS(athlete_data2, file='athlete.rds')
readRDS('athlete.rds')

# save() can save multiple R objects
save(athlete_data2, file='athlete.RData')
load('athlete.RData', verbose=T)
```

V. Olympic Trivia! (Coding challenge)

A. **Who attended the most events in history?**
B. **Who got the most medals in history? (hint: slice_max to get the max)**