Olympic Data

```
# Set the CRAN mirror
options(repos = c(CRAN = "https://cran.r-project.org"))
```

```
olympic_results <- read.csv ("C:/Users/aleen/OneDrive/Desktop/Data Analytics/Olympic Data/olympi
c_results.csv")
olympic_athletes <- read.csv ("C:/Users/aleen/OneDrive/Desktop/Data Analytics/Olympic Data/olymp
ic_athletes.csv")
olympic_medals <- read.csv ("C:/Users/aleen/OneDrive/Desktop/Data Analytics/Olympic Data/olympic
_medals.csv")
```

Installing all the necessary packages

```
install.packages("tidyverse")
```

```
## Installing package into 'C:/Users/aleen/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\aleen\AppData\Local\Temp\Rtmp6bGNOk\downloaded_packages
```

```
library("tidyverse")
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
install.packages("readr")
```

```
## Warning: package 'readr' is in use and will not be installed
```

```
library("readr")
install.packages("here")
```

```
## Installing package into 'C:/Users/aleen/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'here' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\aleen\AppData\Local\Temp\Rtmp6bGNOk\downloaded_packages
```

```
library("here")
```

```
## here() starts at C:/Users/aleen/OneDrive/Desktop/Data Analytics/Olympic Data
```

```
install.packages("skimr")
```

```
## Installing package into 'C:/Users/aleen/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'skimr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\aleen\AppData\Local\Temp\Rtmp6bGNOk\downloaded_packages
```

```
library("skimr")
install.packages("janitor")
```

```
## Installing package into 'C:/Users/aleen/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'janitor' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\aleen\AppData\Local\Temp\Rtmp6bGNOk\downloaded_packages
```

```
library("janitor")
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

We have a giant data-set with the results from Olympic Summer & Winter games, 1986-2022. This was sourced from Kaggle at https://www.kaggle.com/datasets/piterfm/olympic-games-medals-19862018 (https://www.kaggle.com/datasets/piterfm/olympic-games-medals-19862018)

We needed to filter the data to only include the various events in men's and woman's gymnastics.

```
filtered_olympic_results <- olympic_results[grepl("gymnastics", olympic_results$discipline_titl
e, ignore.case = TRUE), ]
```

The data-set Olympic_results had a column called first_game which included the Location and the year of the athletes first Olympic games. We needed to extract just the year for that column and create a new column that only displayed the years.

```
olympic_athletes$first_olympic_year <- substr(olympic_athletes$first_game, nchar(olympic_athlete
s$first_game) - 3, nchar(olympic_athletes$first_game))
```

The new column first_olympic_year then needed to be changed into an integer so that we can do some simple calculations

```
olympic_athletes$first_olympic_year <- as.integer(olympic_athletes$first_olympic_year)
olympic_athletes$athlete_year_birth <- as.integer(olympic_athletes$athlete_year_birth)
olympic_athletes$first_olympic_year <- as.integer(olympic_athletes$first_olympic_year)
```

Now we can subtract the first_olympic_year column from the athlete_birth_year to get the age that they first entered in the Olympics. The outcome was then placed into a new column called age_at_competition

```
olympic_athletes$age_at_competition <- olympic_athletes$first_olympic_year - olympic_athletes$at
hlete_year_birth
```

We wanted to see from the the olympic_medal data-set the column of the dicipline_title filtered to include only data that included the word "gymnastics"

```
filtered_olympic_medals <- subset(olympic_medals, grepl("gymnastics", discipline_title, ignore.c
ase = TRUE))
```

Connecting the columns called athlete_full_name columns from the two data-sets named filtered_olympic_results and olympic_athletes

```
merged_data <- merge(filtered_olympic_results, olympic_athletes, by.x = "athlete_full_name", by.
y = "athlete_full_name")
```

For ease of typing and clarification of description I renamed a few of the newly created data-sets

```
gymnastic_medals <- filtered_olympic_medals
gymnastic_results <- filtered_olympic_results
gymnastic_data <- merged_data
rm(filtered_olympic_medals)
rm(filtered_olympic_results)
rm(merged_data)
```

I created a new column to display if the row was referring to a male or female athlete. I displayed the results in a new column called male_female

```
gymnastic_data$male_female <- ifelse(grepl("women", gymnastic_data$event_title, ignore.case = TR
UE), "female", "male")
```

I needed to install dplyr

```
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library("dplyr")
```

The column of age_at_competition needed to be changed to an integer

```
gymnastic_data$age_at_competition <- as.integer(gymnastic_data$age_at_competition)
```

I created a copy of my data-set gymnastic_data so that I could do some experimentation with filters

```
gymnastic_data_copy <- gymnastic_data
```

I changed the name of the column age_at_competition to age for ease of typing

```
 names(gymnastic_data)[names(gymnastic_data) == "age_at_competition"] <- "age"
```

Doing some basic filtering to exclude any incorrect data due to age of gymnast. I excluded any gymnast that was younger than 10 and older than 45

```
gymnastic_data <- gymnastic_data[gymnastic_data$age >= 10, ]
gymnastic_data <- gymnastic_data[gymnastic_data$age <= 45, ]
```

The athlete name needed to be changed to all lower case then the final desired affect of proper case.

```
gymnastic_data$athlete_full_name <- tolower(gymnastic_data$athlete_full_name)
```

Needed to get the tools package for toTitleCase to work

```
library(tools, lib.loc = "C:/Program Files/R/R-4.3.1/library")
```

Changing the name for Title Case

```
gymnastic_data$athlete_full_name <- toTitleCase(gymnastic_data$athlete_full_name)
```

I noticed that there were some rows that did not contain any data so I removed them from the data-set

```
gymnastic_data <- gymnastic_data[!is.na(gymnastic_data$athlete_full_name), ]
```

The column gymnastic_bio contained commas which caused the data to not import/export correctly so I removed that column

```
gymnastic_data$bio <- NULL
```

loading the package “stringr”

```
library(stringr)
```

Counting up the medals

```
gymnastic_data <- gymnastic_data %>%
 mutate(gold = str_count(medal_type, "GOLD"))

gymnastic_data <- gymnastic_data %>%
 mutate(silver = str_count(medal_type, "SILVER"))

gymnastic_data <- gymnastic_data %>%
 mutate(bronze = str_count(medal_type, "BRONZE"))
```

Total Medals

saving file locally so I can work in tableau to create an awesome chart.

```
write.csv(gymnastic_data, "gymnastic_data.csv")
```