Tingji-AssignmentE

```
library(tidyverse)
library(knitr)
```

Part 1

```
wdbd_raw <- read.csv("WDBD.csv")

#Remove metadata rows at the bottom
wdbd_clean <- wdbd_raw[1:2976,] %>%
  select(-Series.Code,-Country.Code) #Series name and code are one to one
```

After briefly read the csv file, I recognized the rows after 2976 are explanation for units, so only keep 2976 rows. And the series names and codes are one to one, so remove the series.codes col. And for the same reason, remove country code col.

```
#Reshape the data set
#pivot_longer
wdbd_longer <- wdbd_clean %>%
  pivot_longer(cols = 3:9, names_to = "Year", values_to = "Value")%>%
  mutate(Year = str_extract(Year, "\\d{4}")) %>%         # 提取4位数字
  mutate(Year = as.integer(Year))                        #Change Year to 4 digit year number

#pivot_wider
wdbd_wider <- wdbd_longer%>%
  mutate(Value = na_if(Value,".."))%>%
  mutate(Value = as.numeric(Value)) %>%
  pivot_wider(names_from = Series.Name,
              values_from = Value)

# summary(wdbd_wider)
wdbd_tidy <- wdbd_wider %>% select(-(38:43),-47)
```

Use pivot_longer() to gather year columns into a single Year column, Use pivot_wider() if you want to restructure Series IDs and Series Names. Also, remove 7 total empty cols with all NAs(Unusual)

```
summary(wdbd_tidy$`Death rate, crude (per 1,000 people)`)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
 0.841   6.170   7.385   7.719   9.233  17.134      62 
```

Death rate with Min = 0.841 which is extremely small, but it make sense in Qatar with very few population.

```
summary(wdbd_tidy$`Adjusted net savings, including particulate emission damage (current US$)`)
```

```
      Min.     1st Qu.      Median        Mean     3rd Qu.        Max.        NA's 
```

```
-1.325e+10  2.429e+09  2.379e+10  1.362e+11  9.641e+10  2.870e+12        218
```

Adjusted net savings with e+10 values(extremely high), which is also make sense for country's saving

```r
head(wdbd_tidy)
```

```
# A tibble: 6 × 43
  Country.Name  Year Access to clean fuels and technolo…¹ Access to clean fuel…²
  <chr>        <int>                                <dbl>                  <dbl>
1 Afghanistan   2018                                 14.5                   31.4
2 Afghanistan   2019                                 15.6                   32.6
3 Afghanistan   2020                                 16.4                   33.8
4 Afghanistan   2021                                 17.4                   34.9
5 Afghanistan   2022                                 18.5                   36.1
6 Afghanistan   2023                                 NA                     NA  
# ℹ abbreviated names:
#   ¹`Access to clean fuels and technologies for cooking, rural (% of rural population)`,
#   ²`Access to clean fuels and technologies for cooking (% of population)`
# ℹ 39 more variables:
#   `Access to clean fuels and technologies for cooking, urban (% of urban population)` <dbl>,
#   `Access to electricity (% of population)` <dbl>,
#   `Access to electricity, rural (% of rural population)` <dbl>, …
```

Part 2

```r
movie_raw = read.csv("movies.csv")
movie_clean <- movie_raw %>%
  separate_wider_delim(
    cols = genres,
    delim = "|",
    names_sep = "_",
    too_few = "align_start"
  )%>%
  select(-genres_5,-genres_6,-genres_7,-genres_4)

movies_tidy <- movie_clean %>%
  mutate(
    year = str_extract(title, "\\(\\d{4}\\)"),
    year = as.integer(str_remove_all(year, "[()]")),
    title = str_trim(str_remove(title, "\\(\\d{4}\\)"))
  ) %>%
  relocate(year, .after = title)

# movies_longer <- movie_clean %>%
#   pivot_longer(
#     cols = starts_with("genres_"),
#     names_to = "genre_num",
#     values_to = "genre",
#     values_drop_na = TRUE
#   ) %>%
#   select(-genre_num)%>%
#   distinct(movieId, title, genre) %>%
#   mutate(value = 1)
```

```
#
# movies_dummy <- movies_longer %>%
#   pivot_wider(
#     names_from = genre,
#     values_from = value,
#     values_fill = 0
#   )
#
# movies_dummy <- movie_clean %>%
#   select(movieId, title) %>%
#   left_join(movies_dummy, by = c("movieId", "title")) %>%
#   mutate(across(where(is.numeric), ~replace_na(., 0)))  # 把空的 dummy 填 0
#
# #get date from ( ) from ChatGPT
# movies_tidy <- movies_dummy %>%
#   mutate(
#     year = str_extract(title, "\\(\\d{4}\\)"),
#     year = as.integer(str_remove_all(year, "[()]")),
#     title = str_trim(str_remove(title, "\\(\\d{4}\\)"))
#   ) %>%
#   relocate(year, .after = title)
```

```
links = read.csv("links.csv")
ratings = read.csv("ratings.csv")
tags = read.csv("tags.csv")

tags_summary <- tags %>%
  group_by(movieId) %>%
  summarise(
    all_tags = str_c(unique(tag), collapse = ", ")
  )
links <- links %>%
  filter(movieId %in% movies_tidy$movieId)

avg_ratings <- ratings%>%
  group_by(movieId)%>%
  summarise(
    avg_ratings = mean(rating,na.rm = TRUE),
    num_ratings = n()
  ) %>%
  filter(movieId %in% movies_tidy$movieId)

movies_tidy <- movies_tidy%>%
  left_join(tags_summary,by = "movieId")%>%
  left_join(links,by = "movieId")%>%
  left_join(avg_ratings,by = "movieId")
head(movies_tidy)
```

```
# A tibble: 6 × 11
  movieId title            year genres_1 genres_2 genres_3 all_tags imdbId tmdbId
    <int> <chr>           <int> <chr>    <chr>    <chr>    <chr>     <int>  <int>
1  182337 Cinétracts       1968 (no gen… <NA>     <NA>     antholo… 2.08e5 287929
2  195495 Familia          2005 Drama    <NA>     <NA>     addicti… 4.26e5  42052
```

```
3    3078 Liberty Heigh…  1999 Drama    <NA>     <NA>     Hebrew,… 1.66e5  27141
4  134704 Comedy Centra…  2011 Comedy   <NA>     <NA>     The Com… 1.99e6 296192
5  219976 47 Hours to L…  2019 Horror   Thriller <NA>     <NA>     7.84e6 615017
6  205715 Reis            2017 (no gen… <NA>     <NA>     <NA>     5.99e6 421682
# ℹ 2 more variables: avg_ratings <dbl>, num_ratings <int>
```