Assignment E: Tidying and Transforming Data in R

Madeleine Livaudais

Part 1: World Development Bank Data Dataset (WDBD.csv)

1. We will read in the data and separate the bottom information part from the data itself. This data has a lot of character columns. To tidy the data we first need to make a year column and convert the data into a value column based on the year for each country.

```
#Read in the bank data
bank_data_raw <- read.csv("WDBD.csv")

#Slice off so just have the data
bank_data <- bank_data_raw %>%
  slice(-(2977:3079))

#Slice off the meta data from the bottom into it's own data frame
bank_data_info <- bank_data_raw %>%
  slice(2982:3079)
```

2. Let's tidy the metadata information data frame. First we have to get each row to be one row rather than the two rows it was read in as.

```
#put the header lines each as a vector and combine into one
headers1 <- as.character(bank_data_info[1, ])
headers2 <- as.character(bank_data_info[2,1:7])
new_headers <- c(headers1, headers2, c("n1", "n2", "n3", "n4"))

#remove the header rows
bank_data_info <- bank_data_info[-(1:2), ]

# create a new empty data frame and assign the headers
empty_matrix <- matrix(vector(), nrow = 48, ncol = 22)
```

```
bank_info_clean <- data.frame(empty_matrix)
colnames(bank_info_clean) <- new_headers

# loop to get the info that is all in two lines into one line each.
j <- 1
for (i in seq(1, 95, by = 2)) {
  part1 <- as.character(bank_data_info[i, ])
  part2 <- as.character(bank_data_info[i+1, ])
  comb_rows <- c(part1, part2)

  bank_info_clean[j, ] <- comb_rows
  j <- j+1
}

bank_info_clean <- bank_info_clean %>%
  #remove empty columns
  select(-n2, -n3, -n4) %>%
  #replace empty cells with NA's
  mutate(across(everything(), ~na_if(., ""))) %>%
  #Combine columns that were off sync
  mutate(`License URL` = coalesce(`License URL`, n1)) %>%
  #remove unnecessary columns
  select(-n1) %>%
  #rename to match data
  rename(`Series Code` = Code, `Series Name` = `Indicator Name`) %>%
  relocate(`Unit of measure`, .after = `License Type`)
```

3. Now let's tidy the actual bank data. Get all the years, which are columns, as a single column with their values as a second column.

```
# AI helped with some of this code

bank_data <- bank_data %>%
  #rename the year columns to just be the year
  rename_with(
    ~str_extract(.x, "\\d{4}"),
    .cols = starts_with("X")
  ) %>%
  #rename the first columns to have a space instead of a .
  rename_with(
    ~gsub("\\.", " ", .x),
    .cols = 1:4
  ) %>%
  #change the ..'s in the data to NA's
  mutate(across(everything(), ~na_if(., ".."))) %>%
  #Convert the year columns into a single year column and a value column
  pivot_longer(
    cols = c(`2018`:`2024`),
    names_to = "Year",
    values_to = "Value"
  ) %>%
  #rearrange columns
  relocate(`Series Code`, .after = `Country Code`)
```

```
#pivot data wider so series names are the column headers (variables) and each year observatio
bank_data_wide <- bank_data %>%
  pivot_wider(
    id_cols = c("Country Name", "Country Code", "Year"),
    names_from = "Series Name",
    values_from = c("Value")
  )

#convert columns with numbers to numeric
bank_data_wide <- bank_data_wide %>%
  mutate(across(-c(`Country Code`, `Country Name`), as.numeric))

# remove any columns where the entire column is NA's
bank_data_wide <- bank_data_wide %>%
  #if all entries are NA, remove column
  select(where(~ !all(is.na(.)))) %>%
  #if less than 5% of the cells have data, remove column
  select(where(~mean(!is.na(.)) > 0.05))


write_csv(bank_data_wide, "bank_data_CLEAN.csv")
```

Now for the units!

```
# create a new data frame to get units
bank_data_units <- bank_data %>%
  select(`Series Code`, `Series Name`)


bank_data_units <- bank_data_units %>%
  # separate the units out from the series name column
  mutate(Units = str_extract(`Series Name`, "\\(([^)]+)\\)$|\\(([^)]+)\\) \\(([^)]+)\\)$")) %
  #eliminate duplicate rows
  unique() %>%
  #remove the parentheses from the Units column
  mutate(Units = str_replace(Units, "\\(", "")) %>%
  mutate(Units = str_replace(Units, "\\)", "")) %>%
  #Clean up the original Series Name by removing the extracted unit strings
  mutate(`Series Name` = str_replace(
    `Series Name`,
    "\\s*\\([^)]*\\) *\\([^)]*\\)$|\\s*\\([^)]*\\)$",
    ""
  ))


bank_data_units <- bank_data_units %>%
  #combine metadata units column to fill NA's from series name units
  left_join(bank_info_clean %>%
              select(`Series Code`, `Unit of measure`),
              by = "Series Code") %>%
  mutate(Units = coalesce(Units, `Unit of measure`)) %>%
  select(-`Unit of measure`)


write_csv(bank_data_units, "bank_units_CLEAN.csv")
```

Part 2: Movies Data Dataset (movies.csv)

```
#read in the data set
movie_data_raw <- read.csv("movies.csv")
movie_data <- movie_data_raw
```

Now, lets tidy the movies data. When I look at the raw data, I see that the tear the movie was made is attached in the same column as the movie title. That will need to be separated. All of the genres the movie falls into are listed in a single cell. These need to be separated into one line each.

```
#Separate the year from the movie title and clean up
#Used AI to help fix my code to work
movie_data <- movie_data %>%
```

```r
  mutate(
    year = str_extract(title, "\\((\\d{4})\\)"),
    year = str_remove_all(year, "[()]"),
    title = str_remove_all(title, "\\s\\(\\d{4}\\)")
  ) %>%
  relocate(year, .after = movieId)

#Clean up movie titles so they're all the same format
#Used AI to help fix my code to work
movie_data <- movie_data |>
  mutate(
    # First, fix commas inside parentheses
    title = sub(
      pattern = "(\\(([^,]+), ([^,)]+)\\))",
      replacement = "(\\3 \\2)",
      x = title
    ),
    # Second, fix the general comma-and-word-at-end pattern
    title = sub(
      pattern = "^(.*), ([^,)]+)(?:\\s*)((?:\\(.*\\))?)$",
      replacement = "\\2 \\1 \\3",
      x = title
    ),
    # Third, clean up any extra whitespace
    title = str_squish(title)
  )

#Separate the genres out from being together
movie_data <- movie_data %>%
  separate_wider_delim(genres, delim = "|", names_sep = "_", too_few = "align_start" )

#Replaces no genre with NA
movie_data <- movie_data %>%
  mutate(`genres_1` = na_if(`genres_1`, "(no genres listed)"))

#Remove Genre columns if data is less than 5% data in the column
movie_data <- movie_data %>%
  select(-genres_7, -genres_6, -genres_5, -genres_4)


#Rename columns
movie_data <-  movie_data %>%
  rename(`Movie Id` = movieId,
          Year = year,
          Title = title,
          `Genre 1` = genres_1,
          `Genre 2` = genres_2,
          `Genre 3` = genres_3
        )


head(movie_data)
```

```
# A tibble: 6 x 6
  `Movie Id` Year  Title                             `Genre 1` `Genre 2` `Genre 3`
       <int> <chr> <chr>                             <chr>     <chr>     <chr>
1     182337 1968  Cinétracts                        <NA>      <NA>      <NA>
2     195495 2005  Familia                           Drama     <NA>      <NA>
3       3078 1999  Liberty Heights                   Drama     <NA>      <NA>
4     134704 2011  Comedy Central Roast of Charli~ Comedy    <NA>      <NA>
5     219976 2019  47 Hours to Live                  Horror    Thriller  <NA>
6     205715 2017  Reis                              <NA>      <NA>      <NA>
```

```
write_csv(movie_data, "movie_data_CLEAN.csv")
```