DATA2020 Final Project Data Preprocessing

Team: Final Project Group 5

Members: Jinyu Wang, Letian Yu, Xiaoyan Liu, Yijia Xue, Ziao Zhang

Author: Letian

Last Update Date: April 28 2024

Introduction

This document is a report on the data preprocessing for the DATA2020 final project.

Data Description

The data used in this project is the Future of Families and Child Wellbeing Study Public Data. We aim to predict the child's overall success at year 15 using early year data. The dataset contains information on the child's family, health, and education from year 1 to year 6. The dataset is in Stata format and contains 6 waves of data.

Data Preprocessing

```r
library(haven)
library(dplyr)
library(purrr)
```

Data Loading

We selected interested columns in wave 1 to wave 6 from the original dataset. The selected columns are as follows:

```r
# Define the columns and descriptions
# as named lists for each wave
wave1_cols <- c(cf1edu = "Father baseline education (father report, then mother report)",
    cm1edu = "Mother baseline education (own report)",
    cf1hhinc = "Household income (with imputed values)",
    f1j8 = "About how much did you earn?",
    m1i2b = "About how much did you earn?")

wave2_cols <- c(f2b13 = "Does child walk or crawl yet?",
    f2b32 = "In general, how is your child's health?",
    f2d1a = "Does mother have any contact with child?",
    m2d2 = "Does father have any contact with child?",
    m2b43a = "On a scale of 1-(least like) to 5-(most like) - Child tends to be shy",
    m2b10 = "Since child was born, how many times has he/she stayed overnight in hospital?")
```

```r
wave3_cols <- c(cf3marm = "Is father married to child's mother at year three?",
    cf3kids = "Number of children under 18 in household",
    cf3md_case_lib = "Father meets depression criteria (liberal) at three-year (CIDI)",
    cf3hhinc = "Household income (with imputed values)",
    f3c3c = "Days/week: father tell child she loves him/her?",
    m3c3c = "Days/week: mother tell child she loves him/her?")

wave4_cols <- c(cf4cohm = "Constructed - Father living with child's mother at five-year",
    t4d7 = "number of kids present with child",
    p4l63 = "Child is interested in many and different things",
    p4l59 = "Child threatens people", p4d1c = "Last 12M, couldn't afford to eat balanced meals",
    m4d1b = "You can trust father to take good care of child",
    f4b8d = "Does person/agency give money/voucher/scholarship to help pay for program?")

wave5_cols <- c(k5d1f = "Amount of time on a weekday you play computer games on the computer or TV",
    k5d1g = "Amount of time on a weekday you watch TV and movies",
    k5h1 = "Condition of health in general",
    k5d1e = "Amount of time on a weekday you chat with friends on the computer",
    k5f1f = "Hurt an animal on purpose")

wave6_cols <- c(p6b1 = "PCG's description of youth's health",
    k6b5b = "Kids in this school work hard (?)",
    k6b20a = "Grade in English or language arts",
    k6b20b = "Grade in Math", k6b20c = "Grade in History or social studies",
    k6b20d = "Grade in Science", k6b21d = "Trouble getting along with other students",
    k6b22a = "Spend time on athletic or sports teams",
    k6b22b = "Spend time on group performance activities")

# Function to load and select columns
# from dataset
load_and_select_columns <- function(dataset_path,
    cols_dict) {
    df <- read_dta(dataset_path)
    df_selected <- select(df, idnum, one_of(names(cols_dict)))
    return(df_selected)
}

# Read in the data for all waves
df_wave1 <- load_and_select_columns("../data/FF_wave1_2020v2.dta",
    wave1_cols)
df_wave2 <- load_and_select_columns("../data/FF_wave2_2020v2.dta",
    wave2_cols)
df_wave3 <- load_and_select_columns("../data/FF_wave3_2020v2.dta",
    wave3_cols)
df_wave4 <- load_and_select_columns("../data/FF_wave4_2020v2.dta",
    wave4_cols)
df_wave5 <- load_and_select_columns("../data/FF_wave5_2020v2.dta",
    wave5_cols)
df_wave6 <- load_and_select_columns("../data/FF_wave6_2020v2.dta",
    wave6_cols)

# Merge the data frames by 'idnum'
df_merged <- reduce(list(df_wave1, df_wave2,
    df_wave3, df_wave4, df_wave5, df_wave6),
    ~inner_join(.x, .y, by = "idnum"))

# Convert all columns to character
# strings
df_merged <- df_merged %>%
    mutate(across(everything(), as.character))

# Write the merged data frame to a .dta
# file write_dta(df_merged,
# '../data/ff_selected_cols.dta')
df_merged
```

```
## # A tibble: 4,898 x 39
##    idnum cf1edu cm1edu cf1hhinc f1j8  m1i2b f2b13 f2b32 f2d1a m2d2  m2b43a m2b10
##    <chr> <chr>  <chr>  <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr>  <chr>
##  1 0001  3      3      22500    32000 10    -6    -6    -5    1     -6     -6
##  2 0002  1      1      -9       -9    260   -6    1     2     1     -6     3
##  3 0003  3      3      62500    20000 6     -6    2     2     1     -6     -6
##  4 0004  2      2      30000    1800  400   -6    1     1     1     -6     -6
##  5 0005  -3     2      -9       -9    8     -9    -9    -9    -9    -9     -9
##  6 0006  2      2      -9       -9    6.69~ -9    -9    -9    -6    -6     1
##  7 0007  1      1      36490    -1    7     -9    -9    -9    1     -6     -6
##  8 0008  2      1      21063    6.5   5.15~ -6    1     1     1     -6     -6
##  9 0009  1      3      30000    10    7     -6    -6    -5    1     -6     -6
## 10 0010  2      3      -9       -9    -1    -9    -9    -9    -9    -9     -9
## # i 4,888 more rows
## # i 27 more variables: cf3marm <chr>, cf3kids <chr>, cf3md_case_lib <chr>,
## #   cf3hhinc <chr>, f3c3c <chr>, m3c3c <chr>, cf4cohm <chr>, t4d7 <chr>,
## #   p4l63 <chr>, p4l59 <chr>, p4d1c <chr>, m4d1b <chr>, f4b8d <chr>,
## #   k5d1f <chr>, k5d1g <chr>, k5h1 <chr>, k5d1e <chr>, k5f1f <chr>, p6b1 <chr>,
## #   k6b5b <chr>, k6b20a <chr>, k6b20b <chr>, k6b20c <chr>, k6b20d <chr>,
## #   k6b21d <chr>, k6b22a <chr>, k6b22b <chr>
```

Data Cleaning

In this section, we will clean the data by replacing negative values with NA and calculating the missing rate of relevant wave 6 columns for each row. We will then filter out rows with target columns missing. Finally, we will construct the target Y variable (our self-defined measurement of teenager success) based on the selected columns.

```
construct_y <- function(data) {
    # Convert y candidate values

    # Health
    data$y_good_health <- as.numeric(str_extract(data$p6b1,
        "^(-?\\d+)"))
    data$y_good_health[data$y_good_health <=
        2] <- 1
    data$y_good_health[is.na(data$y_good_health) |
        data$y_good_health > 2] <- 0

    # Attitude
    data$y_work_hard <- as.numeric(str_extract(data$k6b5b,
        "^(-?\\d+)"))
    data$y_work_hard[data$y_work_hard <=
        2] <- 1
    data$y_work_hard[is.na(data$y_work_hard) |
        data$y_work_hard > 2] <- 0

    # Academic
    academic_cols <- c("k6b20a", "k6b20b",
        "k6b20c", "k6b20d")
    for (col in academic_cols) {
        data[[col]] <- as.numeric(str_extract(data[[col]],
            "^(-?\\d+)"))
        data[[col]] <- ifelse(data[[col]] <=
            1, 4, ifelse(data[[col]] <= 2,
            3, ifelse(data[[col]] <= 3, 2,
                0)))
    }
    data$y_gpa_good <- rowMeans(data[, academic_cols],
        na.rm = TRUE)
    data$y_gpa_good <- ifelse(data$y_gpa_good >=
        3.6, 1, 0)

    # Social
    data$y_social_good <- as.numeric(str_extract(data$k6b21d,
        "^(-?\\d+)"))
    data$y_social_good[data$y_social_good <=
        1] <- 1
    data$y_social_good[is.na(data$y_social_good) |
        data$y_social_good > 1] <- 0

    # Group Performance
    data$y_art <- as.numeric(str_extract(data$k6b22b,
        "^(-?\\d+)"))
    data$y_art <- ifelse(data$y_art >= 3,
        1, 0)

    # Athlete
    data$y_sport <- as.numeric(str_extract(data$k6b22a,
        "^(-?\\d+)"))
    data$y_sport <- ifelse(data$y_sport >=
        3, 1, 0)

    # Composite score
    data$y_score <- rowSums(data[, c("y_good_health",
        "y_work_hard", "y_gpa_good", "y_social_good",
        "y_art", "y_sport")], na.rm = TRUE)
    data$y_binary <- ifelse(data$y_score >=
        3, 1, 0)

    return(data)
}
```

```
# Convert to R data frame
df <- as_tibble(df_merged)
```

```r
df[, names(wave6_cols)] <- lapply(df[, names(wave6_cols)],
    as.character)

# Replace values with negative signs at
# the beginning with NA
negative_value_pattern <- "^-\\d+.*$"
df[, names(wave6_cols)] <- lapply(df[, names(wave6_cols)],
    function(col) {
        ifelse(grepl(negative_value_pattern,
            col), NA, col)
    })

# Recalculate the number of missing
# values per row across the specific
# columns
df$y_missing_rate <- rowSums(is.na(df[, names(wave6_cols)]))/length(wave6_cols)

# Filter rows
df_filtered <- df %>%
    filter(y_missing_rate <= 0.1)

# Construct Y variables
data_processed <- construct_y(df_filtered)

# Select X and Y columns
X_cols <- c(names(wave1_cols), names(wave2_cols),
    names(wave3_cols), names(wave4_cols),
    names(wave5_cols))
Y_cols <- c("y_missing_rate", "y_score",
    "y_binary")
data_processed <- select(data_processed,
    one_of(c(X_cols, Y_cols)))

# Write the processed data to a .dta
# file write_dta(data_processed,
# '../data/ff_data_preprocessed.dta')
data_processed
```

```
## # A tibble: 3,113 x 32
##    cf1edu cm1edu cf1hhinc f1j8  m1i2b       f2b13 f2b32 f2d1a m2d2  m2b43a m2b10
##    <chr>  <chr>  <chr>    <chr> <chr>       <chr> <chr> <chr> <chr> <chr>  <chr>
##  1 3      3      22500    32000 10          -6    -6    -5    1     -6     -6
##  2 1      1      -9       -9    260         -6    1     2     1     -6     3
##  3 3      3      62500    20000 6           -6    2     2     1     -6     -6
##  4 2      2      30000    1800  400         -6    1     1     1     -6     -6
##  5 2      2      -9       -9    6.69999980~ -9    -9    -9    -6    -6     1
##  6 2      1      21063    6.5   5.15000009~ -6    1     1     1     -6     -6
##  7 1      3      30000    10    7           -6    -6    -5    1     -6     -6
##  8 2      3      -9       -9    -1          -9    -9    -9    -9    -9     -9
##  9 2      3      22500    300   150         -9    -9    -9    1     -6     1
## 10 2      3      3750     200   6.15000009~ -6    3     1     2     -6     -6
## # i 3,103 more rows
## # i 21 more variables: cf3marm <chr>, cf3kids <chr>, cf3md_case_lib <chr>,
## #   cf3hhinc <chr>, f3c3c <chr>, m3c3c <chr>, cf4cohm <chr>, t4d7 <chr>,
## #   p4l63 <chr>, p4l59 <chr>, p4d1c <chr>, m4d1b <chr>, f4b8d <chr>,
## #   k5d1f <chr>, k5d1g <chr>, k5h1 <chr>, k5d1e <chr>, k5f1f <chr>,
## #   y_missing_rate <dbl>, y_score <dbl>, y_binary <dbl>
```

Handle Missing X Values

In this section, we will handle missing values in the X variables.

```
x_processed <- data_processed

# Using mutate across to apply the
# operations to all X_cols
x_processed <- x_processed %>%
    mutate(across(all_of(X_cols), ~as.numeric(str_extract(.,
        "^(-?\\d+)"))))

# Write the processed X data to a .dta
# file write_dta(x_processed,
# '../data/ff_data_x_preprocessed.dta')
x_processed
```

```
## # A tibble: 3,113 x 32
##    cf1edu cm1edu cf1hhinc  f1j8 m1i2b f2b13 f2b32 f2d1a  m2d2 m2b43a m2b10
##     <dbl>  <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
##  1      3      3    22500 32000    10    -6    -6    -5     1     -6    -6
##  2      1      1       -9    -9   260    -6     1     2     1     -6     3
##  3      3      3    62500 20000     6    -6     2     2     1     -6    -6
##  4      2      2    30000  1800   400    -6     1     1     1     -6    -6
##  5      2      2       -9    -9     6    -9    -9    -9    -6     -6     1
##  6      2      1    21063     6     5    -6     1     1     1     -6    -6
##  7      1      3    30000    10     7    -6    -6    -5     1     -6    -6
##  8      2      3       -9    -9    -1    -9    -9    -9    -9     -9    -9
##  9      2      3    22500   300   150    -9    -9    -9     1     -6     1
## 10      2      3     3750   200     6    -6     3     1     2     -6    -6
## # i 3,103 more rows
## # i 21 more variables: cf3marm <dbl>, cf3kids <dbl>, cf3md_case_lib <dbl>,
## #   cf3hhinc <dbl>, f3c3c <dbl>, m3c3c <dbl>, cf4cohm <dbl>, t4d7 <dbl>,
## #   p4l63 <dbl>, p4l59 <dbl>, p4d1c <dbl>, m4d1b <dbl>, f4b8d <dbl>,
## #   k5d1f <dbl>, k5d1g <dbl>, k5h1 <dbl>, k5d1e <dbl>, k5f1f <dbl>,
## #   y_missing_rate <dbl>, y_score <dbl>, y_binary <dbl>
```

Select Interested Columns after EDA

```
X_selected <- c('cf1edu', 'cm1edu', 'f2b32', 'm2d2', 'm2b43a', 'cf3marm', 'cf3kids',
                'cf3md_case_lib', 'cf4cohm', 't4d7', 'cf1hhinc', 'k5d1f', 'k5f1f')
Y_selected <- c('y_binary')

# Select columns and export to Stata format
df_selected <- data_processed %>%
  select(all_of(c(X_selected, Y_selected)))
# write_dta(df_selected, '../data/ff_data_preprocessed_v1.dta')

df_selected_x_processed <- x_processed %>%
  select(all_of(c(X_selected, Y_selected)))
```

```
# write_dta(df_selected_x_processed, '../data/ff_data_x_preprocessed_v1.dta')

df_selected_x_processed
```

```
## # A tibble: 3,113 x 14
##    cf1edu cm1edu f2b32  m2d2 m2b43a cf3marm cf3kids cf3md_case_lib cf4cohm  t4d7
##     <dbl>  <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl>          <dbl>   <dbl> <dbl>
##  1      3      3    -6     1     -6       0       0              0       0    -9
##  2      1      1     1     1     -6      -9      -9             -9       0    -9
##  3      3      3     2     1     -6       0       2              0       0    21
##  4      2      2     1     1     -6       0       2              0       0    -9
##  5      2      2    -9    -6     -6      -9      -9             -9      -9    -9
##  6      2      1     1     1     -6       1       3              0       0    -9
##  7      1      3    -6     1     -6       0       1              1       0    -9
##  8      2      3    -9    -9     -9      -9      -9             -9      -9    20
##  9      2      3    -9     1     -6       1       2              1       0    22
## 10      2      3     3     2     -6       0       2              1       0    21
## # i 3,103 more rows
## # i 4 more variables: cf1hhinc <dbl>, k5d1f <dbl>, k5f1f <dbl>, y_binary <dbl>
```

Summary

In this report, we have preprocessed the Future of Families and Child Wellbeing Study Public Data. We have selected interested columns from wave 1 to wave 6, cleaned the data by replacing negative values with NA, calculated the missing rate of relevant wave 6 columns for each row, filtered out rows with target columns missing, and constructed the target Y variable based on the selected columns. We have also handled missing values in the X variables. The preprocessed data is ready for further analysis.