From Open to Text Book Data: The Role of Initial Data Analysis in Longitudinal Data Cleaning

A Preprint

Dewi Amaliah
Department of Econometrics and Business Statistics
Monash University
Clayton, VIC 3800

Kate Hyde

Emi Tanaka *
Department of Econometrics and Business Statistics
Monash University
Clayton, VIC 3800
emi.tanaka@monash.edu

Nicholas Tierney

Dianne Cook †
Department of Econometrics and Business Statistics
Monash University
Clayton, VIC 3800
dicook@monash.edu

March 18, 2021

Abstract

1 Introduction

The idea of "Open Data" is that the data are freely accessible, modified, and shared by anyone for any purpose (Open Knowledge Foundation n.d.). Due too its openness, Open Data are sometimes utilized and incorporated as the example data in statistical text book or in research. Despite its traits, open data often have an issue with its quality in the sense that it is often not tidy and clean. Hence, it should be tidied and cleaned before further analysis to prevent inappropriate usage of statistical method that may lead to a misleading result (Huebner, Vach, and Cessie 2016).

Huebner, Vach, and Cessie (2016) also argued that data tidying and cleaning is the first part of Initial Data Analysis (IDA). The second and third steps are to explore the data properties and to document and report the process for the later formal analysis. This second step is the same with Chatfield's perspective on the goal of IDA (Chatfield 1985), which is to get the "feel" of the data. In other word, IDA could be regarded as a way to assess the data quality.

Dasu and Johnson (2003) mentioned that data cleaning and exploration, which is the IDA notion, is a difficult task and determine 80 percent of the data mining result. Despite its importance, this stage is often undervalued and neglected (Chatfield 1985). Wickham (2014) also stated that there has been few research of the good practice of data cleaning. Furthermore, the decisions made in this stage is often unreported in term

*emitanaka.org
†dicook.org

that the IDA are often performed in an unplanned and unstructured way and is only shared among restricted party (Huebner et al. 2020). In fact, documenting the result of IDA is utterly essential as the information preservation for the downstream statistical analyses and model building (Huebner, Vach, and Cessie 2016) as well as to avoid publication bias (Huebner et al. 2020). It is also important to keep the data cleaning process accountable and transparent.

This paper aims to demonstrate the steps of IDA, which is tidying, cleaning, and documenting the data. We used the longitudinal data from the National Longitudinal Survey of Youth 1979 (NLSY79) or is known as the NLSY79. Particularly, we are interested in doing the IDA to the wages and several demographic variables of the NLSY79 cohort whose highest grade are up to 12th grade and the high school dropouts.

We chose the NLSY79 data because this data is one of the instances of Open Data that has been playing an important role in research in various disciplines including but are not limited to economics, sociology, education, public policy, and public health for more than a quarter of the century (Pergamit et al. 2001). In addition, The National Longitudinal Survey is considered as survey with high retention rates and carefully designed making it suitable for a life course research (Pergamit et al. 2001) and (Cooksey 2017). According to Cooksey (2017) as of 2015, thousand of articles, and hundreds of book chapters and monographs has incorporated the NLSY79 data. Moreover, the NLSY79 is considered as the most widely used and most important cohort in the NLSY79 data sets (Pergamit et al. 2001). Singer and Willett (2003) used the wages and other variables of the NLSY79 data in their book in longitudinal data analysis. This is also being our motivation in incorporating the wages of high school cohort, specifically the high school dropouts, is that we want to make an open data to be suitable as an example data in text book or in research.

2 The NLSY79

The NLSY79 is a longitudinal survey held by the U.S Bureau of Labor Statistics that follows the lives of a sample of American youth and born between 1957-1964. The cohort originally included 12,686 respondents ages 14-22 when first interviewed in 1979. It was comprised of Blacks, Hispanics, economically disadvantaged non-Black non-Hispanics, and youth in the military. In 1984 and 1990, two sub-samples were dropped from the interview; the dropped subjects are the 1,079 members of the military sample and 1,643 members of the economically disadvantaged non-Black non-Hispanics respectively. Hence, 9,964 respondents remain in the eligible samples. The surveys were conducted annually from 1979 to 1994 and biennially thereafter. Data are now available from Round 1 (1979 survey year) to Round 28 (2018 survey year).

Although the main focus area of the NLSY is labor and employment, the NLSY also cover several other topics including education; training and achievement; household, geography and contextual variables; dating, marriage, cohabitation; sexual activity, pregnancy, and fertility; children; income, assets and program participation; health; attitudes and expectations; and crime and substance use.

There are two ways to conduct the interview of the NLSY79, which are face to face interview or by telephone. In recent survey years, more than 90 percent of respondents were interviewed by telephone (Cooksey 2017).

3 The NLSY79 Data Cleaning

3.1 Getting and Tidying the Data

The NLSY79 data are stored in a database and could be downloaded by variables. Several variables are available for download and could be browsed by index. For the wages data set, we only extracted these variables:

- Education, Training & Achievement Scores
  - Education -> Summary measures -> All schools -> By year -> Highest grade completed
    - Downloaded all of the 78 variables in Highest grade completed.
- Employment
  - Summary measures -> By job -> Hours worked and Hourly wages
    - Downloaded all of the 427 variables in Hours worked
    - Downloaded all of the 151 variables in Hourly wages

    Both hours worked and hourly wages are recorded by the job, up to five jobs for each id/subject.

Table 1: The Untidy Form of the NLSY79 Raw Data

CASEID_1979	HRP1_1979	HRP2_1979	HRP3_1979	HRP4_1979	HRP5_1979	HRP1_1980
1	328	NA	NA	NA	NA	NA
2	385	NA	NA	NA	NA	457
3	365	NA	275	NA	NA	397
4	NA	NA	NA	NA	NA	NA
5	310	375	NA	NA	NA	333
6	NA	NA	NA	250	NA	275

- Household, Geography & Contextual Variables
  - Context -> Summary measures -> Basic demographics
    * Downloaded year and month of birth, race, and sex variables.

    There are two versions of the year and month of birth, i.e. 1979 and 1981 data. We downloaded these two versions.

The downloaded data set came in a csv (NLSY79.csv) and dat (NLSY79.dat) format. We only used the .dat format. It also came along with these files:

- NLSY79.NLSY79: This is the tagset of variables that can be uploaded to the web site to recreate the data set.
- NLSY79.R: This is an R script provided automatically by the database for reading the data into R and convert the variables' name and its label into something more sensible. We utilized this code to do an initial data tidying. It produced two data set, `categories_qnames` (the observations are stored in categorical/interval values) and `new_data_qnames` (the observations are stored in integer form).

```
source(here::here("data-raw/NLSY79/NLSY79.R"))
```

According to Wickham (2014), a tidy data sets comply with three rules, the first is that each variable forms a column, the second is that each observation forms a row, and the last is that each type of observational unit forms a table. Unfortunately, the `new_data_qnames` did not meet these requirements in the way that the value for particular year and job are stored in different columns, hence the data contains a huge number of columns (686 columns). The example of the untidy of the data set is displayed in Table 1, which actually intended to display the hourly rate of each respondent by job (HRP1 to HRP5) and by year (1979 and 1980). The table implies that the column headers are values of the year and job, not variable names.

Consequently, the data should be tidied and wrangled first to extract the demographic and employment variables that we want to put in the final data set. We mainly used `tidyr` (Wickham 2020), `dplyr` (Wickham et al. 2020), and `stringr` (Wickham 2019) to do this job.

3.1.1 Tidying demographic variables

`Age in 1979`, `gender`, `race`, `highest grade completed` (factor and integer), and the `year when the highest grade completed` are the variables that we want to use in the cleaned data set.

Age in 1979 are derived from year of birth and month of birth variables in the raw data set. The variables have two versions, which are the 1979 version and the 1981 version. We only used the 1979 data and did a consistency check of those years and flag the inconsistent observations.

```
## tidy the date of birth data
dob_tidy <- new_data_qnames %>%
  dplyr::select(CASEID_1979,
         starts_with("Q1-3_A~")) %>%
  mutate(dob_year = case_when(
                    # if the years recorded in both sets match, take 79 data
                    `Q1-3_A~Y_1979` == `Q1-3_A~Y_1981` ~ `Q1-3_A~Y_1979`,
                    # if the year in the 81 set is missing, take the 79 data
```

```
                    is.na(`Q1-3_A~Y_1981`) ~ `Q1-3_A~Y_1979`,
                    # if the sets don't match for dob year, take the 79 data
                    `Q1-3_A~Y_1979` != `Q1-3_A~Y_1981` ~ `Q1-3_A~Y_1979`),
        dob_month = case_when(
                    # if months recorded in both sets match, take 79 data
                    `Q1-3_A~M_1979` == `Q1-3_A~M_1981` ~ `Q1-3_A~M_1979`,
                    # if month in 81 set is missing, take the 79 data
                    is.na(`Q1-3_A~M_1981`) ~ `Q1-3_A~M_1979`,
                    # if sets don't match for dob month, take the 79 data
                    `Q1-3_A~M_1979` != `Q1-3_A~M_1981` ~ `Q1-3_A~M_1979`),
        # flag if there is a conflict between dob recorded in 79 and 81
        dob_conflict = case_when(
                      (`Q1-3_A~M_1979` != `Q1-3_A~M_1981`) & !is.na(`Q1-3_A~M_1981`)
                      ~ TRUE,
                      (`Q1-3_A~Y_1979` != `Q1-3_A~Y_1981`) & !is.na(`Q1-3_A~Y_1981`)
                      ~ TRUE,
                      (`Q1-3_A~Y_1979` == `Q1-3_A~Y_1981`) &
                      (`Q1-3_A~M_1979` == `Q1-3_A~M_1981`) ~ FALSE,
                      is.na(`Q1-3_A~M_1981`) | is.na(`Q1-3_A~Y_1981`) ~ FALSE)) %>%
  dplyr::select(CASEID_1979,
         dob_month,
         dob_year,
         dob_conflict)
```

For `gender` and `race`, we only renamed these variables.

```
## tidy the gender and race variables
demog_tidy <- categories_qnames %>%
  dplyr::select(CASEID_1979,
         SAMPLE_RACE_78SCRN,
         SAMPLE_SEX_1979) %>%
  rename(gender = SAMPLE_SEX_1979,
         race = SAMPLE_RACE_78SCRN)
```

The `highest grade completed` came with several version in each year. We chose the revised May data because the May data seemed to have less missing and presumably the revised data has been checked. However, there was no revised May data for 2012, 2014, 2016, and 2018 so we just used the ordinary May data.

```
# tidy the grade
demog_education <- new_data_qnames %>%
  as_tibble() %>%
  # in 2018, the variable's name is Q3-4_2018, instead of HGC_2018
  rename(HGC_2018 = `Q3-4_2018`) %>%
  dplyr::select(CASEID_1979,
         starts_with("HGCREV"),
         "HGC_2012",
         "HGC_2014",
         "HGC_2016",
         "HGC_2018") %>%
  pivot_longer(!CASEID_1979,
               names_to = "var",
               values_to = "grade") %>%
  separate("var", c("var", "year"), sep = -4) %>%
  filter(!is.na(grade)) %>%
  dplyr::select(-var)
```

In the final data, we only used the highest grade completed ever and derived the year of when the highest grade completed its categorical value. Therefore, we wrangled the highest grade completed in each year to mutate these variables.

```
## getting the highest year of completed education ever
highest_year <- demog_education %>%
  group_by(CASEID_1979) %>%
  mutate(hgc_i = max(grade)) %>%
  filter(hgc_i == grade) %>%
  filter(year == first(year)) %>%
  rename(yr_hgc = year) %>%
  dplyr::select(CASEID_1979, yr_hgc, hgc_i) %>%
  ungroup() %>%
  mutate('hgc' = ifelse(hgc_i == 0, "NONE", ifelse(hgc_i == 1, "1ST GRADE",
                 ifelse(hgc_i == 2, "2ND GRADE", ifelse(hgc_i == 3, "3RD GRADE",
                 ifelse(hgc_i >= 4 & hgc_i <= 12, paste0(hgc_i,"TH GRADE"),
                 ifelse(hgc_i == 13, "1ST YEAR COL",
                 ifelse(hgc_i == 14, "2ND YEAR COL",
                 ifelse(hgc_i == 15, "3RD YEAR COL",
                 ifelse(hgc_i == 95, "UNGRADED",
                        paste0((hgc_i - 12), "TH YEAR COL")))))))))))
```

Finally, we join all the tidy variables in a data set called `full_demographics`.

```
full_demographics <- full_join(dob_tidy, demog_tidy, by = "CASEID_1979") %>%
  full_join(highest_year, by = "CASEID_1979") %>%
  rename("id" = "CASEID_1979")

head(full_demographics)
```

```
##   id dob_month dob_year dob_conflict                    race gender yr_hgc
## 1  1         9       58        FALSE NON-BLACK, NON-HISPANIC FEMALE   1979
## 2  2         1       59        FALSE NON-BLACK, NON-HISPANIC FEMALE   1985
## 3  3         8       61        FALSE NON-BLACK, NON-HISPANIC FEMALE   1993
## 4  4         8       62        FALSE NON-BLACK, NON-HISPANIC FEMALE   1986
## 5  5         7       59        FALSE NON-BLACK, NON-HISPANIC   MALE   1984
## 6  6        10       60        FALSE NON-BLACK, NON-HISPANIC   MALE   1983
##   hgc_i          hgc
## 1    12   12TH GRADE
## 2    12   12TH GRADE
## 3    12   12TH GRADE
## 4    14 2ND YEAR COL
## 5    18 6TH YEAR COL
## 6    16 4TH YEAR COL
```

3.1.2 Tidying employment variables

The employment data comprises of three variables, i.e. `total hours of work per week`, `number of jobs that an individual has`, and `mean hourly wage`. For hours worked per week, initially only one version per job, no choice from 1979 to 1987 (QES-52A). From 1988 onward, when we had more options, we chose the variable for total hours including time spent working from home (QES-52D). However, 1993 did not have all the five D variables (the first one and the last one were missing), so we used QES-52A variable instead. In addition, 2008 only had jobs 1-4 for the QES-52D variable (whereas the other years had 1-5), so we just used these.

```
# make a list for years where we used the "QES-52A"
year_A <- c(1979:1987, 1993)
#function to get the hour of work
get_hour <- function(year){
  if(year %in% year_A){
```

```
   temp <- new_data_qnames %>%
    dplyr::select(CASEID_1979,
            starts_with("QES-52A") &
              ends_with(as.character(year)))}
  else{
    temp <- new_data_qnames %>%
    dplyr::select(CASEID_1979,
            starts_with("QES-52D") &
              ends_with(as.character(year)))}
    temp %>%
      pivot_longer(!CASEID_1979,
                   names_to = "job",
                   values_to = "hours_work") %>%
      separate("job", c("job", "year"), sep = -4) %>%
      mutate(job = paste0("job_", substr(job, 9, 10))) %>%
      rename(id = CASEID_1979)
}

# list to save the iteration result
hours <- list()
# getting the hours of work of all observations
for(ayear in c(1979:1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010,
               2012, 2014, 2016, 2018)) {
   hours[[ayear]] <- get_hour(ayear)
}
# unlist the hours of work
hours_all <- bind_rows(!!!hours)
```

The same algorithm was also deployed to tidy the rate of wage by year and by ID. The difference is that the hourly rate had only one version of each year. The hours of work and the hourly rate were then joined to calculate the number of jobs that an ID has and their mean hourly wage. Some observations had 0 in their hourly rate, which is considered as invalid value. Thus, their hourly rate set to be N.A.

```
get_rate <- function(year) {
  new_data_qnames %>%
    dplyr::select(CASEID_1979,
            starts_with("HRP") &
              ends_with(as.character(year))) %>%
    pivot_longer(!CASEID_1979, names_to = "job", values_to = "rate_per_hour") %>%
    separate("job", c("job", "year"), sep = -4) %>%
    mutate(job = paste0("job_0", substr(job, 4, 4))) %>%
    rename(id = CASEID_1979)
}
rates <- list()
for(ayear in c(1979:1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010,
               2012, 2014, 2016, 2018)) {
   rates[[ayear]] <- get_rate(ayear)
}
rates_all <- bind_rows(!!!rates)
# join hours and rates variable
hours_wages <- left_join(rates_all,
                         hours_all,
                         by = c("id", "year", "job")) %>%
  # set the 0 value in rate_per_hour as NA
  mutate(rate_per_hour = ifelse(rate_per_hour == 0, NA,
                                rate_per_hour))
head(hours_wages)
```

```
## # A tibble: 6 x 5
```

```
##      id job    year  rate_per_hour hours_work
##   <int> <chr>  <chr>         <int>      <int>
## 1     1 job_01 1979            328         38
## 2     1 job_02 1979             NA         15
## 3     1 job_03 1979             NA         NA
## 4     1 job_04 1979             NA         NA
## 5     1 job_05 1979             NA         NA
## 6     2 job_01 1979            385         35
```

Since our ultimate goal is to calculate the mean hourly wage, the number of jobs is calculate based on the availability of the `rate_per_hour` information. For example, the number of jobs of ID 1, based on `hours_work`, is two. However, since the information of hourly rate of `job_02` is not available, the number of job is considered as 1.

Further, we calculated the mean hourly wage for each ID in each year using a weighted mean with the hours of work as the weight. However, there are a lot of missing value in `hours_work` variable. In that case, we only calculated the mean hourly wage based on arithmetic/regular mean method. Hence, we created a new variable to flag whether the mean hourly wage is a weighted or a regular mean. Additionally, if an ID only had one job, we directly used their hourly wages information and flagged it as an arithmetic mean.

```r
# calculate number of jobs that a person has in one year
no_job <- hours_wages %>%
  filter(!is.na(rate_per_hour)) %>%
  group_by(id, year) %>%
  summarise(no_jobs = length(rate_per_hour))

# filter the observations with available rate per hour
eligible_wages <- hours_wages %>%
  filter(!is.na(rate_per_hour)) %>%
  left_join(no_job, by = c("id", "year"))

# calculate the mean_hourly_wage
# flag1 = code 1 for weighted mean
# code 0 for arithmetic mean
mean_hourly_wage <-
  eligible_wages %>%
  group_by(id, year) %>%
  #calculate the weighted mean if the number of jobs > 1
  mutate(wages = ifelse(no_jobs == 1, rate_per_hour/100,
                        weighted.mean(rate_per_hour, hours_work, na.rm = TRUE)/100)) %>%
  #give the flag if it the weighted mean
  mutate(flag1 = ifelse(!is.na(wages) & no_jobs != 1, 1,
                        0)) %>%
  #calculate the arithmetic mean for the na
  mutate(wages = ifelse(is.na(wages), mean(rate_per_hour)/100,
                        wages)) %>%
  group_by(id, year) %>%
  summarise(wages = mean(wages),
            total_hours = sum(hours_work),
            number_of_jobs = mean(no_jobs),
            flag1 = mean(flag1)) %>%
  mutate(year = as.numeric(year)) %>%
  ungroup() %>%
  rename(mean_hourly_wage = wages) %>%
  mutate(is_wm = ifelse(flag1 == 1, TRUE,
                        FALSE)) %>%
  dplyr::select(-flag1)

head(mean_hourly_wage, n = 10)
```

```
## # A tibble: 10 x 6
##       id  year mean_hourly_wage total_hours number_of_jobs is_wm
##    <int> <dbl>            <dbl>       <int>          <dbl> <lgl>
##  1     1  1979             3.28          38              1 FALSE
##  2     1  1981             3.61          NA              1 FALSE
##  3     2  1979             3.85          35              1 FALSE
##  4     2  1980             4.57          NA              1 FALSE
##  5     2  1981             5.14          NA              1 FALSE
##  6     2  1982             5.71          35              1 FALSE
##  7     2  1983             5.71          NA              1 FALSE
##  8     2  1984             5.14          NA              1 FALSE
##  9     2  1985             7.71          NA              1 FALSE
## 10     2  1986             7.69          NA              1 FALSE
```

The `mean_hourly_wage` and `full_demographic` data are then joined. We also filtered the data to only have the cohort who completed the education up to 12th grade and participated at least five rounds in the survey and save it to an object called `wages_demog_hs`.

```
# join the wages information and the demographic information by case id.
wages_demog <- left_join(mean_hourly_wage, full_demographics, by="id")
# calculate the years in work force and the age of the subjects in 1979
wages_demog <- wages_demog %>%
  mutate(yr_hgc = as.numeric(yr_hgc)) %>%
  mutate(years_in_workforce = year - yr_hgc) %>%
  mutate(age_1979 = 1979 - (dob_year + 1900))
# filter only the id with high school education
wages_demog_hs <- wages_demog  %>% filter(grepl("GRADE", hgc))
# calculate the number of observation
keep_me <- wages_demog_hs %>%
  count(id) %>%
  filter(n > 4)
wages_demog_hs <- wages_demog_hs %>%
  filter(id %in% keep_me$id)
```

3.2 Initial Data Analysis

According to Huebner, Vach, and Cessie (2016), Initial Data Analysis (IDA) is the step of inspecting and screening the data after being collected to ensure that the data is clean, valid, and ready to be deployed in the later formal statistical analysis. Moreover, Chatfield (1985) argued that the two main objectives of IDA is data description, which is to assess the structure and the quality of the data; and model formulation without any formal statistical inference.

In this paper, we conducted an IDA or a preliminary data analysis to assess the consistency of the data with the cohort information that is provided by the NLSY. In addition, we also aimed to find the anomaly in the wages values using this approach. We mainly used graphical summary to do the IDA using `ggplot2`(Wickham 2016) and `brolgar` (Tierney, Cook, and Prvan 2020).

As stated previously, the respondents' ages ranged from 12 to 22 when first interviewed in 1979. Hence, we would like to validate whether all of the respondents were in this range. Additionally, the NLSY also provided the number of the survey cohort by their gender (6,403 males and 6,283 females) and race (7,510 Non-Black/Non-Hispanic; 3,174 Black; 2,002 Hispanic). To validate this, we used the `full_demographic` i.e. the data with the survey years 1979 sample. Table 2 and Table 3 suggest that the demographic data we had is consistent with the sample information in the database.

The next step is that we explored the mean hourly wage data, in this case, we only explored the wages data in `wages_demog_hs`. Figure **??** conveys that there is clearly a problem in the mean hourly wage values. Figure **??** A shows that some observations had an exceptionally high figure of wages, even more than US$10,000 per hour. In Figure **??** B, we barely see any difference in the minimum, median, an maximum value of the wages since the distribution is heavily skewed to the right. Additionally, Table 4 shows that the overall wages median of the cohort is only 7.2, while the mean is 11.87. It indicates that the data might contains a lot of extreme values.

Table 2: Age Distribution of the NLSY79 samples

Age	Number of Sample
15	1265
16	1550
17	1600
18	1530
19	1662
20	1722
21	1677
22	1680

Table 3: Gender and Race Distribution of the NLSY79 Samples

	Race			
Gender	Hispanic	Black	Non-Black, Non-Hispanic	Total
Male	1000 (15.62%)	1613 (25.19%)	3790 (59.19%)	6403 (100.00%)
Female	1002 (15.95%)	1561 (24.84%)	3720 (59.21%)	6283 (100.00%)
Total	2002 (15.78%)	3174 (25.02%)	7510 (59.20%)	12686 (100.00%)

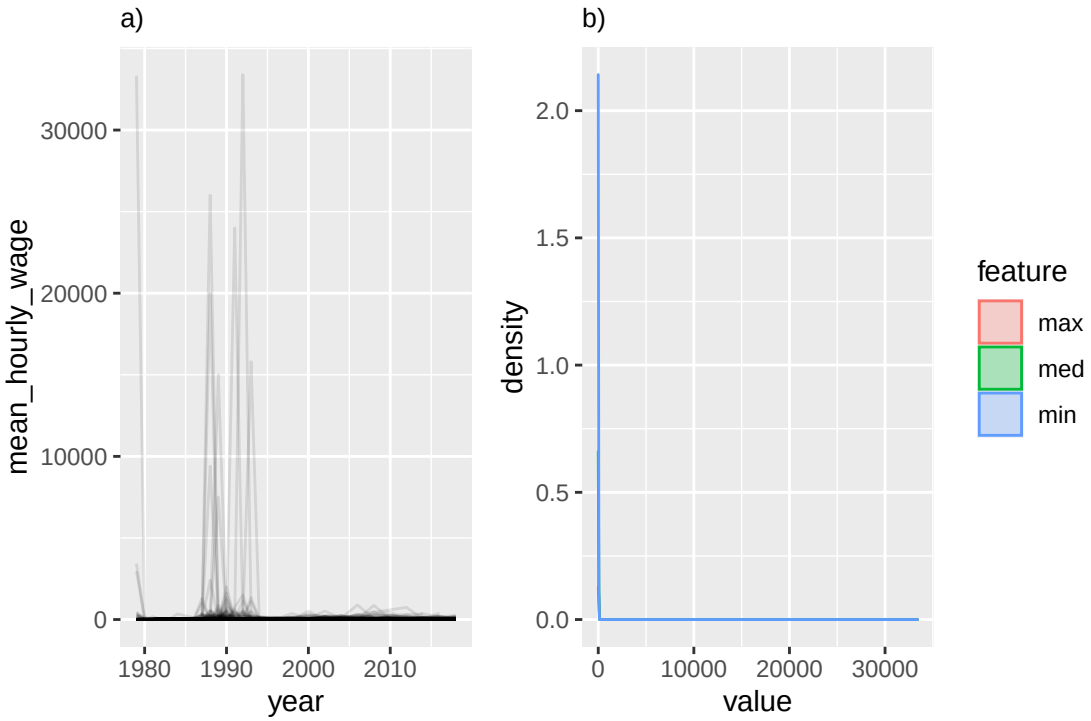

Figure 1: Two plots showing the distribution of the mean hourly wage. Plot A portrays the pattern of mean hourly wage of high school cohort from 1979 to 2018 of each ID in US Dollar; Plot B shows the distribution of their minimum, median, and maximum value. We can see that some IDs had an extremely high of wages and it made the distribution of the three features is extremely skewed.

Table 4: Summary Statistics of Wages of High School Data

Statistics	Value
Min.	0.01000
1st Qu.	4.50000
Median	7.20000
Mean	11.86578
3rd Qu.	11.74000
Max.	33400.00000

In Figure **??**, we plotted some respondents with a high value of mean hourly wages. We filtered all of the IDs who earned more than US$ 500 per hour averagely. We found that most of these respondents only experienced one point of extremely high wages. Thus, we suspected that these high values are erroneous values resulted from a data entry error.

Further, we took 36 samples randomly from the data and plotted it as is seen in Figure **??**. It implies that not only that some observations earned an extremely high figures of wages, but some also had a reasonably fluctuate wages, for example the IDs in panel number 5, 7, and 11. The plot also implies that the samples had a different pattern of mean hourly wages. Some had a flat wages for years but had a sudden increase in a year than it went down again, while the other experienced a upsurge in their wage, for instance the IDs in panel 9.

According to Pergamit et al. (2001), one of the flaws of the NLSY79 employment data is that since the NLSY79 collect the information of the working hours since the last interview, it might be challenging for the respondents to track the within-job hours changes that happens between survey year, especially for the respondents with fluctuate working hours or whose job is seasonal. It even has been more challenging since 1994, where the respondents had to recall two years period. This shortcoming might also contribute on the fluctuation of one's wages data.

3.2.1 Robust Linear Model for Noises Treatment

As it is seen from figure **??**, there are many spikes in the mean hourly wage data. As part of the IDA, which is the model formulation, we built a robust linear regression model to address this issue. The notion of robust linear regression is to yield an estimation that is robust to the influence of noise or contamination (Koller 2016). It also aims to detect the contamination by weighting each observation based on how "well-behave" they are, known as robustness weight. Observations with lower robustness weight are suggested as an outliers by this method (Koller 2016).

In this paper, we built the model using the `rlm` function from `MASS` package (Venables and Ripley 2002). We set the `mean_hourly_wage` and `year` as the dependent and predictor respectively. Furthermore, we used M-Estimation with Huber weighting where the observation with small residual get a weight of 1, while the larger the residual, the smaller the weight (less than 1) (UCLA: Statistical Consulting Group 2021).

Since we worked with longitudinal data, we should built the model for each ID, instead of the overall data. The robust mixed model is actually the best model to be employed in this case. However, this method is too computationally and memory expensive, especially for a large data set, like the NLSY79 data. Thus, the model for each ID is built utilizing the `nest` and `map` function from `tidyr` (Wickham 2020) and `purrr` (Henry and Wickham 2020) respectively.

The challenging part of detecting the anomaly using the robustness weight is to determine the threshold of the weight where the observations considered as outliers. Moreover, it should be noted that not all the outliers is due to an error, instead it might be that one had a reasonably increasing or decreasing wages. To, minimize the risk of being mistakenly regard an outlier as an error outlier, we have simulated some threshold and study the behavior of the spikes in each threshold. We found that 0.1 is the most reasonable value to be the threshold to minimize the risk of that drawback because it still capture the sensible spikes in the data. After deciding the threshold, we flagged the observations with the weight less than 0.1, and imputed their mean hourly wage with the models' predicted value.

```
# nest the data by id to build a robust linear model
by_id <- wages_demog_hs %>%
  dplyr::select(id, year, mean_hourly_wage) %>%
```

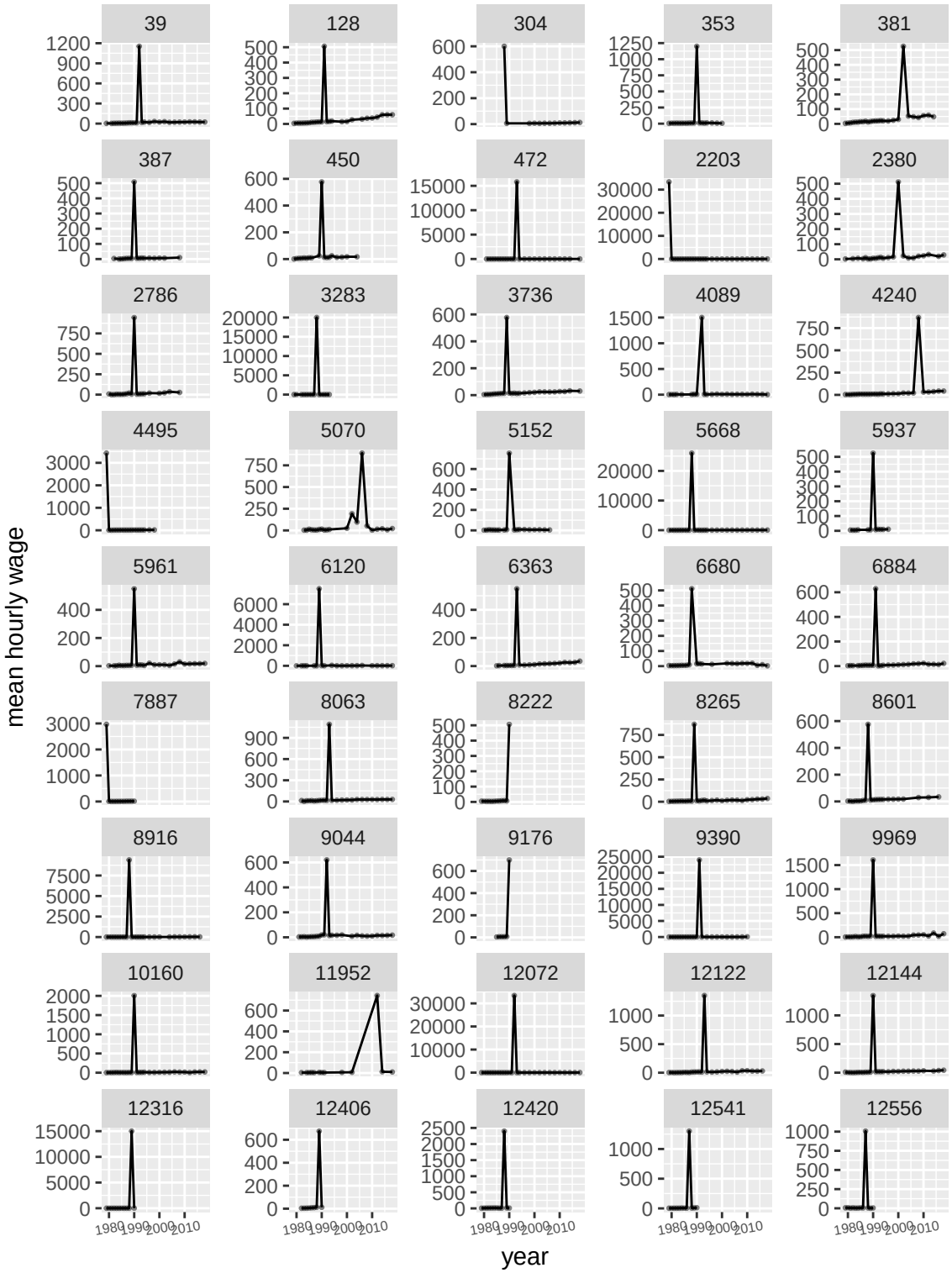

Figure 2: Some observations with extremely high mean hourly wage. Most of the IDs only have one point of high wage.

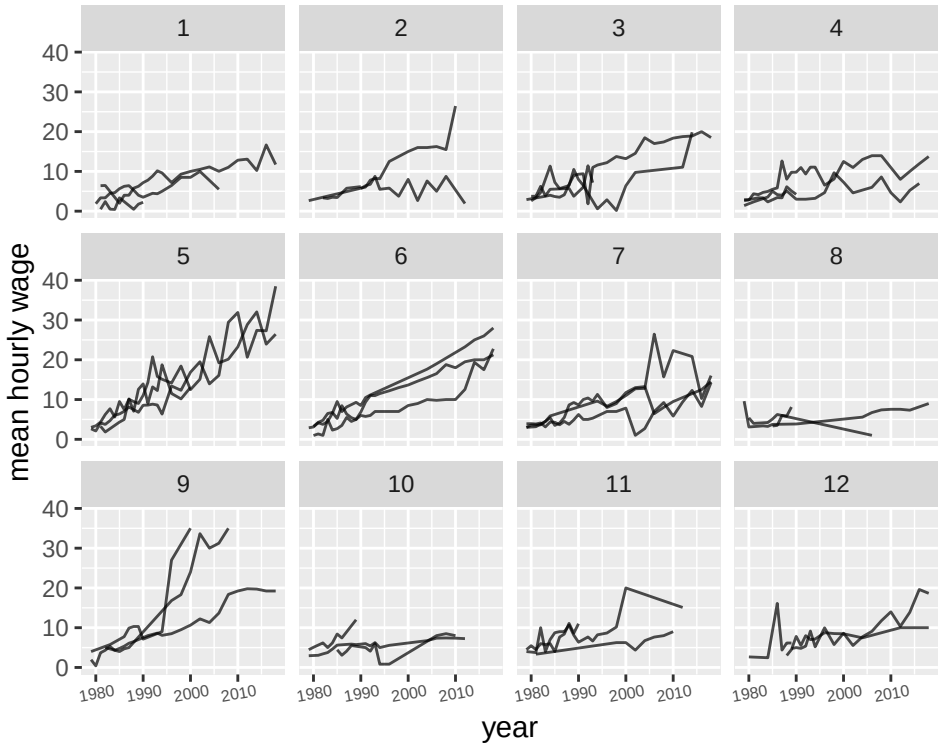

Figure 3: The mean hourly wages of some random samples are shown in twelve facets, three IDs per facet. It suggests that some IDs had a reasonably fluctuate wages.

```
  group_by(id) %>%
  nest()

# build a robust linear model
id_rlm <- by_id %>%
  mutate(model = map(.x = data,
                     .f = function(x){
                       rlm(mean_hourly_wage ~ year, data = x)
                     }))
# extract the property of the regression model
id_aug <- id_rlm %>%
  mutate(augmented = map(model, broom::augment)) %>%
  unnest(augmented)

# extract the weight of each observation
id_w <- id_rlm %>%
  mutate(w = map(.x = model,
                 .f = function(x){
                   x$w
                 })) %>%
  unnest(w) %>%
  dplyr::select(w)

# bind the property of each observation with their weight
id_aug_w <- cbind(id_aug, id_w) %>%
  dplyr::select(`id...1`,
                year,
```

```
                mean_hourly_wage,
                .fitted,
                .resid,
                .hat,
                .sigma,
                w) %>%
  rename(id = `id...1`)

# if the weight < 1, the mean_hourly_wage is replaced by the model's fitted/predicted value.
# and add the flag whether the observation is predicted value or not.
# since the fitted value is sometimes <0, and wages value could never be negative,
# we keep the mean hourly wage value even its weight < 1.

wages_rlm_dat <- id_aug_w %>%
  mutate(wages_rlm = ifelse(w < 0.1  & .fitted >= 0, .fitted,
                            mean_hourly_wage)) %>%
  mutate(is_pred = ifelse(w < 0.1 & .fitted >= 0, TRUE,
                          FALSE)) %>%
  dplyr::select(id, year, wages_rlm, is_pred)

# join back the `wages_rlm_dat` to `wages_demog_hs`

wages_demog_hs <- left_join(wages_demog_hs, wages_rlm_dat, by = c("id", "year"))
```

Figure **??** A shows that after imputing the "error outliers" with the models' predicted value, the highest wages value has decreased to be around US$250. The spikes were still observed, but are not as extreme as the original data set. In Figure **??** B) although the distributions of the features are still positively skewed, we can still examine it clearly the difference shape of those features. The minimum value is heavily skewed, means that most the subjects have a small minimum value of wages, but there are still extreme cases where their minimum wages was extremely higher than others. Moreover, these cases had a minimum wages that is higher than others' maximum wages.

Furthermore, the robust linear regression has reduce the level of noise in the data set as is seen in Figure **??**. The figure also implies that after the treatment, the fluctuation can still be observed in the data and only the large spikes, which are considered as "error outliers", are eliminated from the data. Hence, the model results a data set with the reasonable degree of fluctuation.

Finally, we saved the imputed data and set the appropriate data type for the variables. We also saved the NLSY79 cohort's demographic information and the high school dropout cohort in a separate data sets. Here, we defined high school dropouts as respondents who only completed either 9^{th}, 10^{th}, 11^{th} grade, or who completed 12^{th} when their age are at least 19 years old (means that these IDs being dropped out in certain year, but they returned back to high school to complete it). We also filtered the data to be only male and aged between 14 and 17 years old to refresh the high school dropouts data in the `brolgar` package.

We then make these three data sets and its processing documentation to be publicly available through an R data container package called `yowie`. The complete flow from the raw data to these data set is displayed in Figure **??**fig:flowchart).

```
# select out the old value of mean hourly wage and change it with the wages_rlm value
wages_demog_hs <- wages_demog_hs %>%
  dplyr::select(-mean_hourly_wage) %>%
  rename(mean_hourly_wage = wages_rlm)

# rename and select the wages in tidy
wages_hs2020 <- wages_demog_hs %>%
  dplyr::select(id, year, mean_hourly_wage, age_1979, gender, race, hgc, hgc_i, yr_hgc,
                number_of_jobs, total_hours, is_wm, is_pred) %>%
  mutate(hgc = as.factor(hgc),
         year = as.integer(year),
         age_1979 = as.integer(age_1979),
```

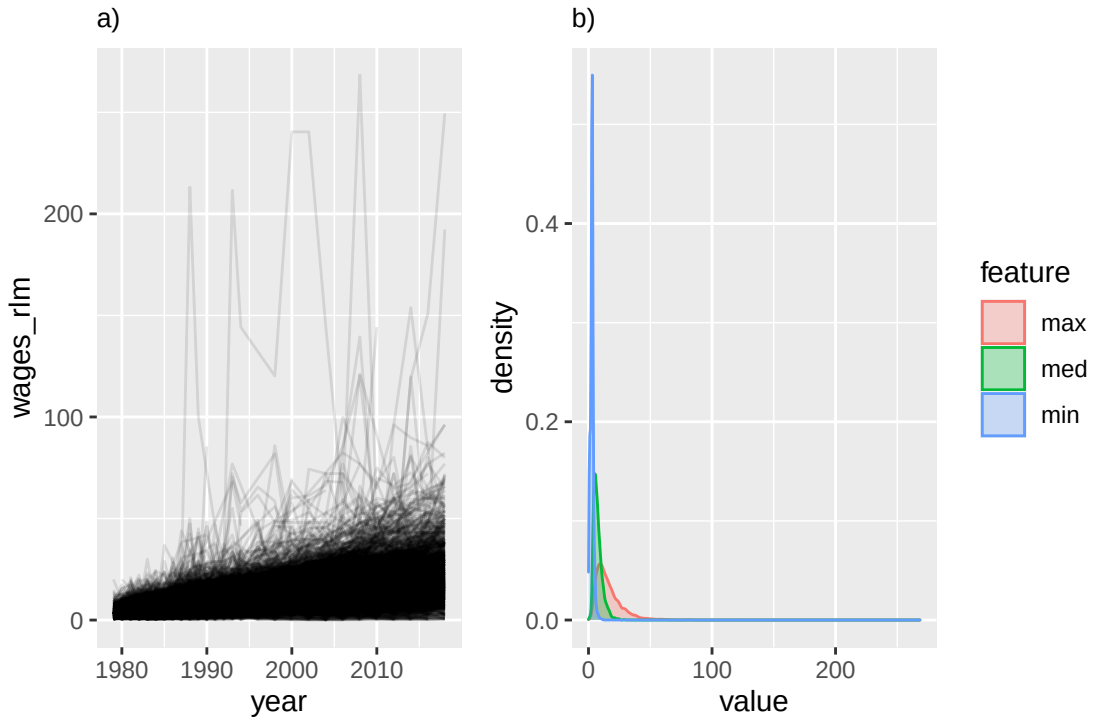

Figure 4: The distribution of the mean hourly wage after treating the extreme values. Plot A portrays the pattern of mean hourly wage of high school cohort from 1979 to 2018 of each ID in US Dollar; Plot B shows the distribution of their minimum, median, and maximum value. We can see that some observations still had reasonbaly higher wages than the others. The minimum, median, and maximum distribution is positively skewed, where some IDs' have a minimum wages that is higher than others' maximum wages.

```
         yr_hgc = as.integer(yr_hgc),
         number_of_jobs = as.integer(number_of_jobs))

# Create a data set for demographic variables
demographic_nlsy79 <- full_demographics %>%
  mutate(age_1979 = 1979 - (dob_year + 1900)) %>%
  dplyr::select(id,
         age_1979,
         gender,
         race,
         hgc,
         hgc_i,
         yr_hgc) %>%
  mutate(age_1979 = as.integer(age_1979),
         hgc = as.factor(hgc),
         yr_hgc = as.integer(yr_hgc))

# Create a data set for the high school dropouts cohort
wages_hs_dropout <- wages_hs2020 %>%
  mutate(dob = 1979 - age_1979,
         age_hgc = yr_hgc - dob) %>%
  filter((hgc %in% c("9TH GRADE",
                     "10TH GRADE",
                     "11TH GRADE")) |
```

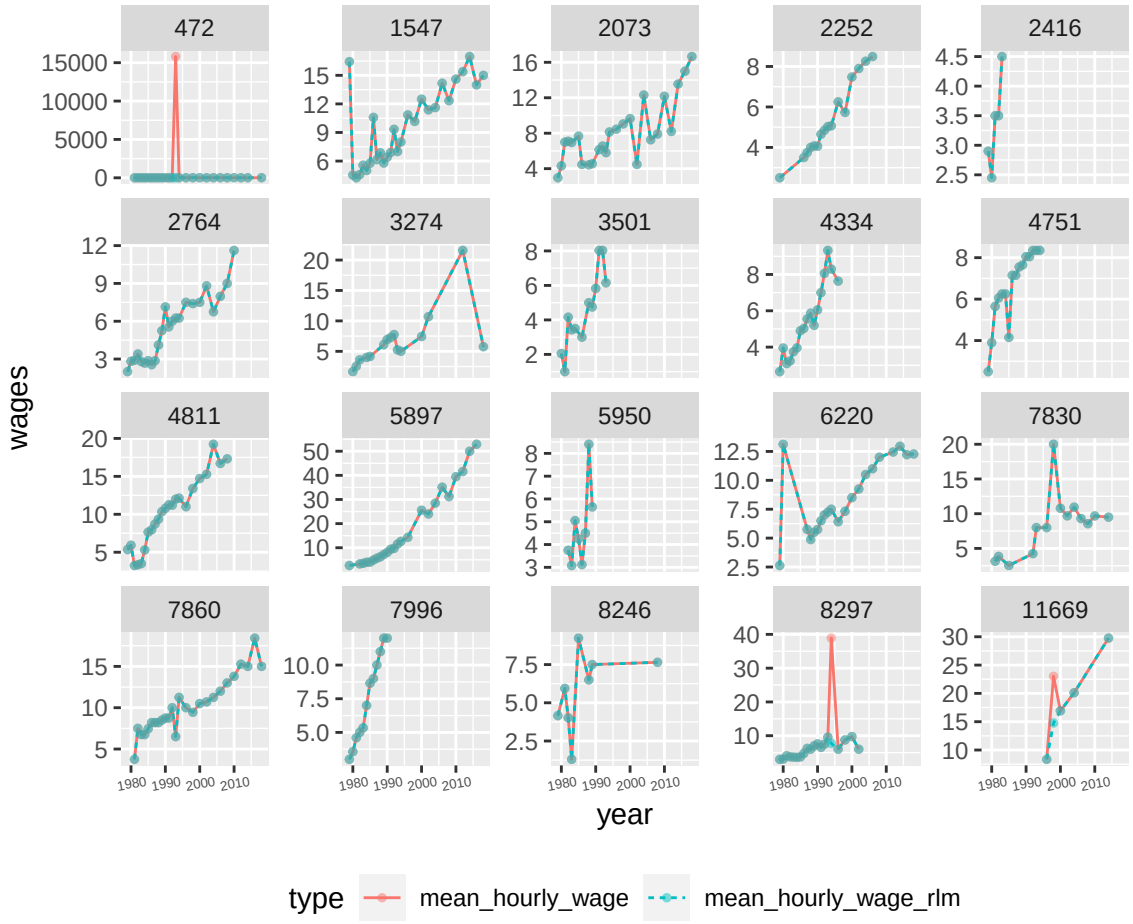

Figure 5: Comparison between the original and the treated mean hourly wage. The orange line portray the original value of mean hourly wage, while the turquoise line display the mean hourly wages value after the extreme values imputed with the robust linear model's prediction value. We can see that some extreme spikes has been reduced by the model.

```
         (hgc == "12TH GRADE" &
            age_hgc > 19)) %>%
filter(age_1979 <= 17,
       gender == "MALE") %>%
dplyr::select(-dob,
       -age_hgc)
```

4 Exploratory Data Analysis

5 Summary and Discussion

One take away from this paper is to make an open data set to be suitable for a text book data or to make it ready to be incorporated in a research, an IDA should be performed and documented. The IDA shows that the NLSY79 data has two problems. Firstly, the downloaded data format is untidy, in term of its column name contains the year values, not just the variable names. Secondly, some anomalies i.e. an extreme high hourly wages are found. We addressed the first problem by tidying the data according to three principles of a tidy data. For the latter problem, we fixed the anomalies using the predicted value of the robust linear regression model. Further, we kept maintaining the natural variability of the wages while minimizing the

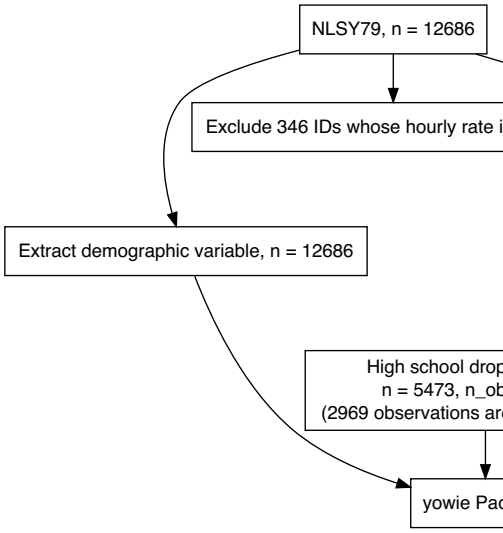

Figure 6: The stages of data filtering from the raw data to get three datasets that are contained in an R package called yowie, n means the number of ID, while n_obs means the number of observations.

presence of anomalies because of the error in the data recording. As an effort to provide the good practice IDA documentation, we also provided two additional variables in the final data sets, which is `is_wm` and `is_pred`. These variables could be a precaution for the data users in analyzing the data.

In addition to demonstrating the importance of IDA to transform the open data to be a analysis-ready data, the finding in this paper also implies that data providers should also design the database that is able to produce tidy data sets. A data provider should also check for data anomalies prior the data publishing or at least provides a set of rules or threshold value, for example, in this case is the threshold of reasonable wages. This will greatly support the data users to carry out further validation and set the same line of which data are considered as outliers by all of the data users. Moreover, some data cleaning tools also require validation rules. For example, `validates`, an R package that is powerful enough to perform a data validation, cannot be used in this case.

For the future study, we suggested the usage of linear mixed model to find and fixed the anomalies in the data set.

References

Chatfield, C. 1985. "The Initial Examination of Data." *Journal of the Royal Statistical Society. Series A. General* 148 (3): 214–53.

Cooksey, Elizabeth C. 2017. "Using the National Longitudinal Surveys of Youth (Nlsy) to Conduct Life Course Analyses." In *Handbook of Life Course Health Development*, edited by Richard M. Lerner Neal Halfon Christoper B. Forrest, 561–77. Cham: Springer. `https://doi.org/https://doi.org/10.1007/978-3-319-47143-3_23`.

Dasu, Tamraparni, and Theodore Johnson. 2003. *Exploratory Data Mining and Data Cleaning*. Wiley Series in Probability and Statistics. Hoboken: WILEY.

Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Huebner, Marianne, Werner Vach, Saskia le Cessie, Carsten Oliver Schmidt, and Lara Lusa. 2020. "Hidden Analyses: A Review of Reporting Practice and Recommendations for More Transparent Reporting of Initial Data Analyses." *BMC Medical Research Methodology* 20 (1): 61–61.

Huebner, PhD, Marianne, Dr rer. nat Vach Werner, and PhD le Cessie Saskia. 2016. "A Systematic Approach to Initial Data Analysis Is Good Research Practice." *The Journal of Thoracic and Cardiovascular Surgery* 151 (1): 25–27.

Koller, Manuel. 2016. "Robustlmm: An R Package for Robust Estimation of Linear Mixed-Effects Models." *Journal of Statistical Software* 75 (6): 1–24.

Open Knowledge Foundation. n.d. "Open Definition. Defining Open in Open Data, Open Content, and Open Knowledge." Accessed March 3, 2021. http://opendefinition.org/od/2.1/en/.

Pergamit, Michael R., Charles R. Pierret, Donna S. Rothstein, and Jonathan R. Veum. 2001. "Data Watch: The National Longitudinal Surveys." *The Journal of Economic Perspectives* 15 (2): 239–53.

Singer, Judith D, and John B Willett. 2003. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford u.a: Oxford Univ. Pr.

Tierney, Nicholas, Di Cook, and Tania Prvan. 2020. *Brolgar: BRowse over Longitudinal Data Graphically and Analytically in R*. https://github.com/njtierney/brolgar.

UCLA: Statistical Consulting Group. 2021. "Robust Regression | R Data Analysis Examples." February 2021. https://stats.idre.ucla.edu/r/dae/robust-regression/.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth. New York: Springer. http://www.stats.ox.ac.uk/pub/MASS4.

Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10): 1–23.

———. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2020. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.