The Journey from Wild to Text Book Data: A Case Study from the National Longitudinal Survey of Youth

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

August 19, 2021

Abstract

NSLY79 is a prominent open data source that has been an important support in multidisciplinary research on longitudinal data. Subsets of this data can be found in numerous textbooks and research articles, however the steps and decisions taken to get from the original data to these subsets is never clearly articulated. This article describes our journey when trying to re-create a textbook example data set from the original data, with a goal to refresh it for use in the classroom. It thus demonstrates the process of initial data analysis (IDA) – tidying, cleaning, and documenting the process – to make data available for text book examples or research. Three new data sets, and the code to produce them are provided in an accompanying open source R package, called `yowie`. The code might be useful for future refreshing of this particular text book data. As a result of this process, some recommendations are made for the NLSY79 curators for incorporating data quality checks, and providing more convenient samples of the data to potential users.

Keywords: Data cleaning; Data tidying; Longitudinal data; NLSY79; Open data; Initial data analysis; Outlier detection; Robust linear regression

1 Introduction

"Open data" is data that are freely accessible, modifiable, and shareable by anyone for any purpose (Open Knowledge Foundation 2021). This type of data can be useful as example data in statistical text books and for research purposes. However, open data are often, what we might call "wild data," because it requires substantial cleaning and tidying to tame it into text book shape. Huebner Marianne, Vach, and Cessie (2016) emphasize that making the data cleaning process accountable and transparent is imperative. Documenting the data cleaning is essential for the integrity of downstream statistical analyses and model building (M. Huebner et al. 2020).

Data cleaning is part of what is called "initial data analysis (IDA)" (Chatfield 1985). The other IDA steps are to explore the data properties and check assumption of modeling to document and report the process for the later formal analysis. Hence, exploratory data analysis, which is defined by Tukey (1977) as a detective work to get the clue from data, either numerically or graphically, before confirmatory data analysis is performed, encompasses IDA. Dasu and Johnson (2003) say that data cleaning and exploration, without naming it as IDA, is a difficult task and consumes 80% of the data mining task.

Despite its importance, this IDA stage is often undervalued and neglected (Chatfield 1985). There are few research papers that document the data cleaning (Wickham 2014). Furthermore, the decisions made in this stage often go unreported in the sense that IDA is often performed in an unplanned and unstructured way, and is only shared among restricted parties (M. Huebner et al. 2020).

This paper aims to demonstrate the steps of IDA, tidying, cleaning, and documenting the process, for a prominent open data source, National Longitudinal Survey of Youth (NLSY) 1979, henceforth referred to NSLY79. This data 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), thousand of articles, and hundreds of book chapters and monographs has incorporated the NLSY 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 high school dropouts from the NLSY79 data as an example data set to illustrate longitudinal data modeling. Our aim is to refresh this text book data to be more up to date. Here, we investigate the process of getting from the raw NLSY79 to this text book data set. However, we are not able to create the exact same data set as published in their book since we do not have the information of what age threshold did they use to determine the high school dropouts.

This paper is structured to have 5 sections. Section 2 describes the original data source. Section 3 presents the steps of cleaning the data, including getting and tidying the data from the NLSY79 and initial data analysis to find and treat the anomalies in the data. We also gave examples of exploratory data analysis using the clean data in Section 4. Finally, Section 6 summarizes the paper.

2 The NLSY79

2.1 Database

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 U.S. Bureau of Labor Statistics, n.d.). 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).

2.2 Target Data

As mentioned in Section 1, this paper aims to refresh the NLSY79 data as used in Singer and Willet's text book (Singer and Willett (2003)). This data set contains the information of yearly mean hourly wages from the NLSY79 cohort, with education and race as covariates. This data set represents the measurement of male high-school dropouts, aged from 14 to 17 years old when first time measured. Hence, in this paper, after getting the data set of high school graduates, we subset it by only having the high school dropouts.

Since there is no information of high school drop-outs in Singer and Willet's book, we define 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).

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.
- 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).

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 2020b), `dplyr` (Wickham et al. 2020), and `stringr` (Wickham 2019) to do this job.

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

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 next step is tidying the `highest grade completed` variable. This variable came with several version in each year. We chose the revised May data because it seemed to have less missing and presumably has been checked. However, there was no revised May data for 2012, 2014, 2016, and 2018 so we just used the ordinary May data for these years. The code for tidying this variable is provided in the supplementary code whose details could be seen in the Supplementary Materials section of this paper.

Further, `highest grade completed` is measured and could be updated in each period of the survey. Hence, we only used the highest grade completed ever and derived the year when it is completed. The code for tidying this variable is also available in the supplementary code of this paper.

Finally, we join all of the tidy variables in a data set called `full_demographics` as displayed as follows:

```
##   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`.

`hours worked per week` initially has 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, in 1993, this variable 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))) %>%
```

```r
    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 way 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 a respondent 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. the result of the joining variables could be seen in the following tibble.

```
## # 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 2. 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. The following tibble shows the result of this tidying process.

```
## # 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`.

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%)

3.2 Initial Data Analysis

According to Huebner Marianne, 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`. 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.

However, we can not be sure only by observing the summary statistics. Hence, we used visualisation techniques to investigate this matter. Figure 1 conveys a problem in the mean hourly wage values. Figure 1 A shows that some observations had an exceptionally high figure of wages, even more than US$10,000 per hour. In

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

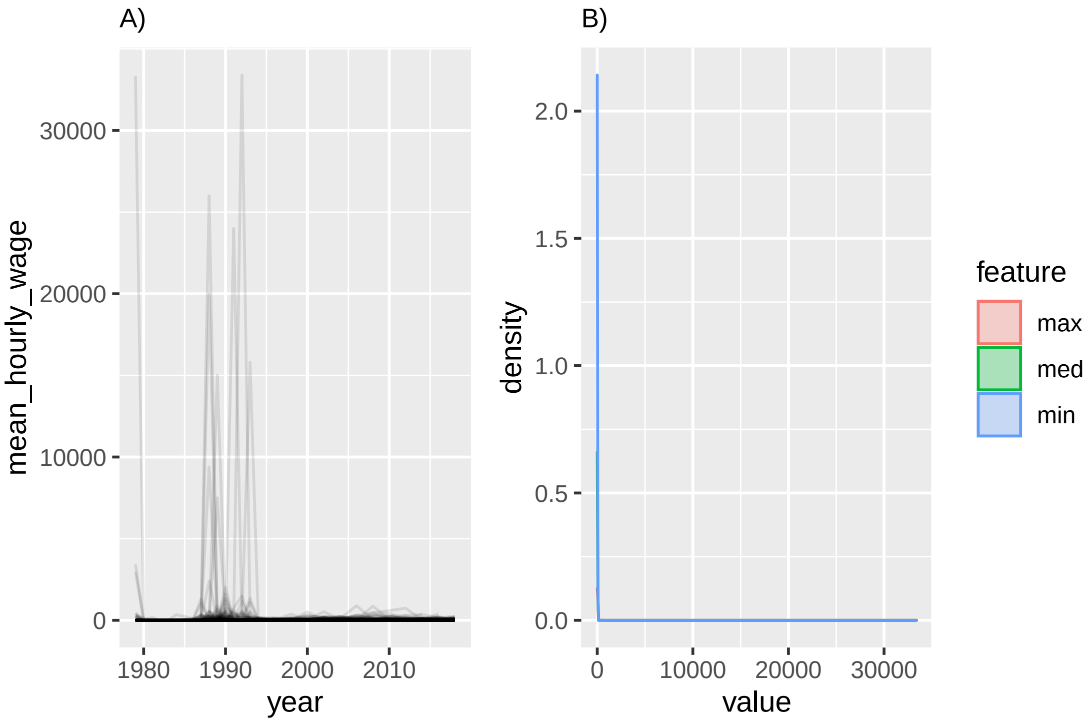


Figure 1: Two plots showing the distribution of the mean hourly wage. Plot A portrays the pattern of mean hourly wage of high school cohort from 1979 to 2018 of each ID in US Dollar; Plot B shows the distribution of their minimum, median, and maximum value. We can see that some IDs had an extremely high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

Figure 1 B, we barely see any difference in the minimum, median, and maximum value of the wages since the distribution is heavily skewed to the right.

In Figure 2, 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 on average. We found that these respondents only experienced one point of extremely high wages. Thus, we suspected that these high values are erroneous resulted from a data entry error.

Further, we took 36 samples randomly from the data and plotted them, as shown in Figure 3. It implies that not only that some observations earned extremely high figures of wages, but some also had reasonably fluctuated wages, for example, the IDs in panel numbers 5, 7, and 11. The plot also implies that the samples had a different pattern of mean hourly wages. Some had flat wages for years but had a sudden increase in one particular year, then it went down again, while the others experienced an 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 to the fluctuation of one's wages data.

3.2.1 Replacing extreme values

As part of the IDA, which is the model formulation, we built a robust linear regression model to handle the extreme values in the data. Robust linear regression yields 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-behaved" they are, known as robustness weight. Observations with lower robustness weight are suggested as an outlier by this method (Koller 2016).

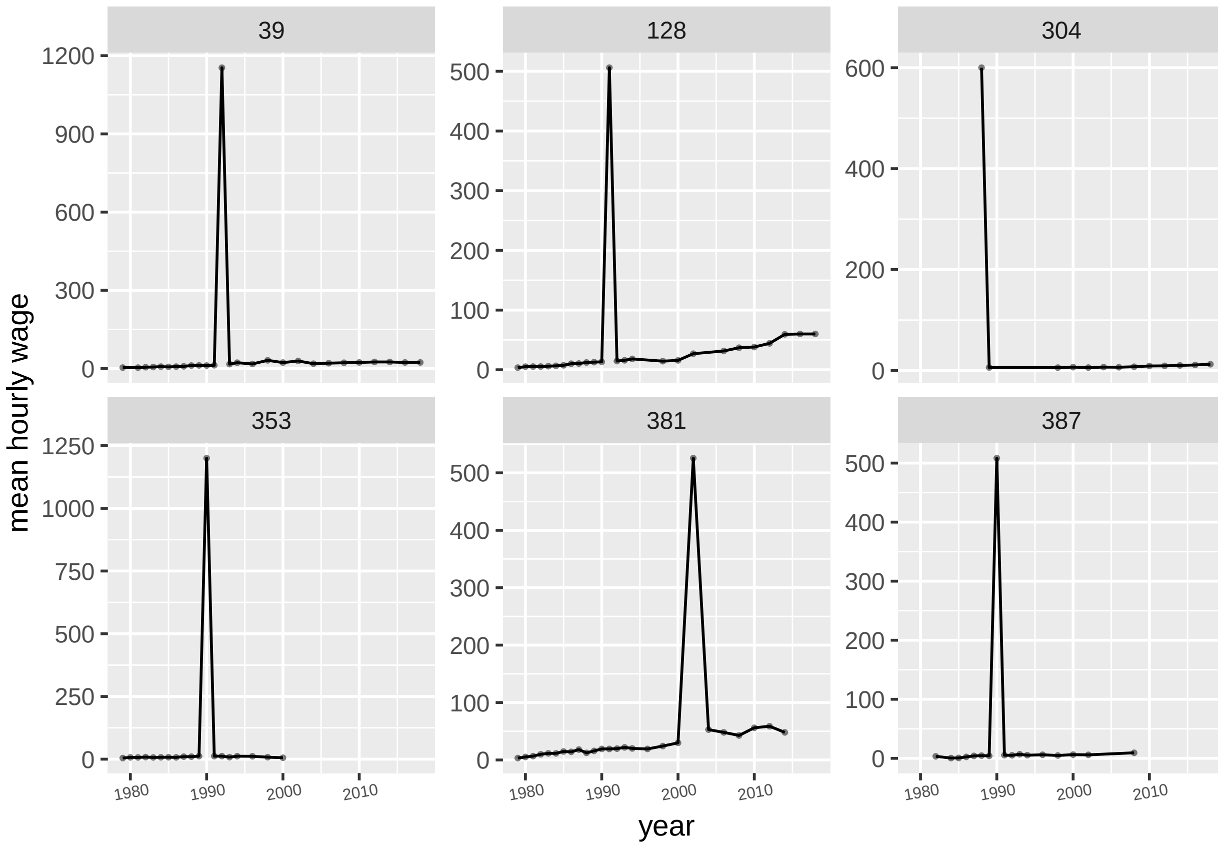


Figure 2: 6 out of 45 IDs with extremely high mean hourly wage. Most of the IDs only have one point of high wage.

Since we worked with longitudinal data, we built the model for each ID instead of the overall data. The robust mixed model is might be 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 2020b) and `purrr` (Henry and Wickham 2020) respectively.

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 a slight residual gets a weight of 1, while the larger the residual, the smaller the weight (less than 1) (UCLA: Statistical Consulting Group 2021). However, the challenging part of detecting the anomaly using the robustness weight is determining the threshold of the weight in which the observations are considered outliers. Moreover, it should be noted that not all the outliers are due to an error. Instead, it might be that one had reasonably increasing or decreasing wages in a particular period.

To minimize the risk of mistakenly regarding an outlier as an "erroneous outlier," we have simulated some thresholds and study how they affected the data. We found that 0.12 is the most reasonable value to be the threshold to minimize that drawback's risk because it still captures the sensible spikes in the data. In other words, we kept maintaining the natural variability of the wages while minimizing anomalies because of the error in the data recording. After deciding the threshold, we imputed the observations whose weight less than 0.12 with the models' predicted value. We then flagged those observations in a new variable called `is_pred`.

Figure 4 shows the mean hourly wage before and after the extreme values are replaced. It implies that the fluctuation can still be observed in the data after the treatment. However, the large spikes, which are considered "erroneous outliers," are already eliminated from the data. Hence, the model produces a data set with a more reasonable degree of fluctuation.

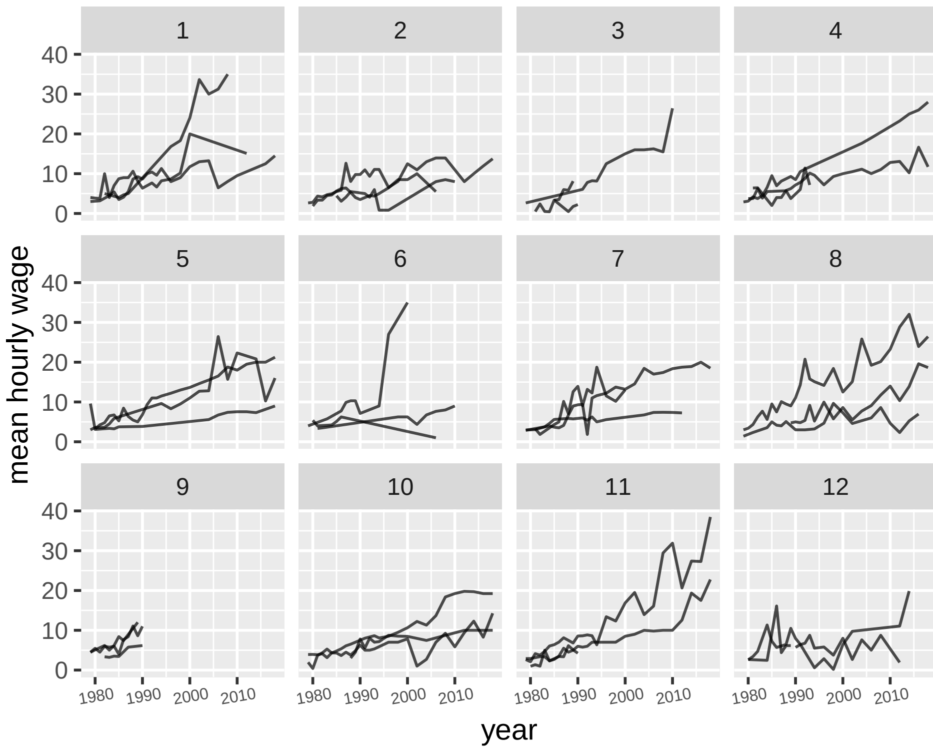


Figure 3: The mean hourly wages of some random samples are shown in twelve facets, three IDs per facet. It suggests that some IDs had a reasonably fluctuate wages.

Further, 5 A shows that after eliminating the extreme values, the highest value has decreased to be around \$350. The spikes were still observed but are not as extreme as the original data set. In Figure 5 B), we plotted the three features of mean hourly wages, namely the minimum, median, and maximum value, transformed to log scale. The plot implies that the skewness is slightly negative. We also see that the three features are overlapped each other. It indicates that some ID's minimum wages are higher than some ID's maximum wages.

Finally, we saved the imputed data and set the appropriate data type for the variables. As our target data is the mean hourly wage of the high-school dropouts, we then subset the high-school graduate data set to have only the male-high school dropout data. We also saved the NLSY79 cohort's demographic information in a separate dataset. We then make these three data sets and their processing documentation 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 6.

4 Exploratory Data Analysis

This part gives some examples of how the data might be used for data analysis in a textbook. We used `brolgar` (Tierney, Cook, and Prvan 2020) to perform exploratory data analysis. In the first example, we observed the relationship between wages and education regarding how the wages increase year by year. Using `key_slope` function in `brolgar`, we performed linear regression with the mean hourly wage as response variable and year as the predictor, extracted the regression slope of each ID, and plotted it according to the highest grade completed.

Figure 7 shows that the higher the education completed does not necessarily increase the wages more from year to year. We can see that people who completed 4th grade happen to have relatively the same median of slope as people who completed 11th grade, even though it is probably because some respondents in this group had extremely high mean hourly wages. Moreover, the annual increasing wages of people who completed 6th grade to 11th grade are relatively the same. People who completed 12th grade tend to have a higher slope of

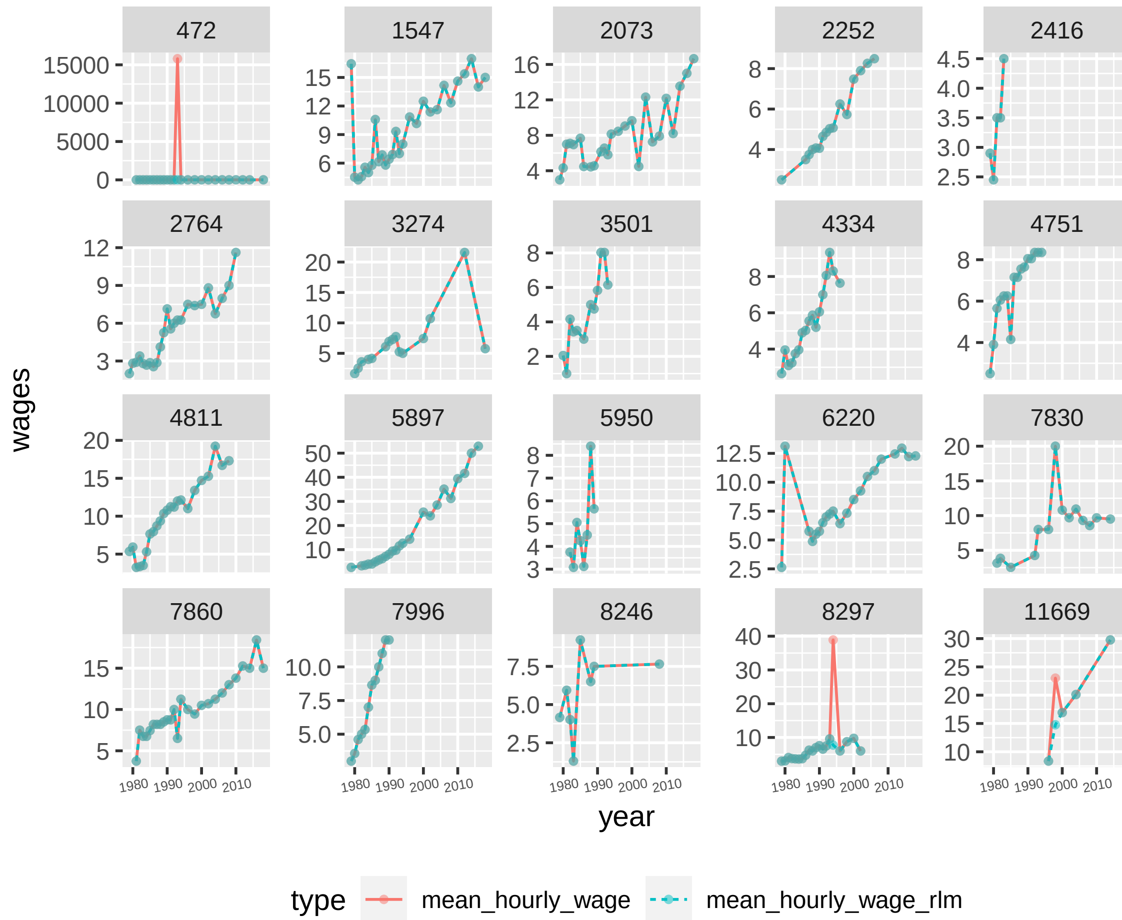


Figure 4: Comparison between the original and the treated mean hourly wage. The orange line portray the original value of mean hourly wage, while the turquoise line display the mean hourly wages value after the extreme values imputed with the robust linear model's prediction value. We can see that some extreme spikes has been reduced by the model.

the year than the other groups. Some outliers are also spotted in this group. Additionally, we can see that some respondents have a negative slope, meaning that their wages tend to decrease over the year.

```
ggplot(wages_slope, aes(x = .slope_year0)) +
  geom_histogram()
```

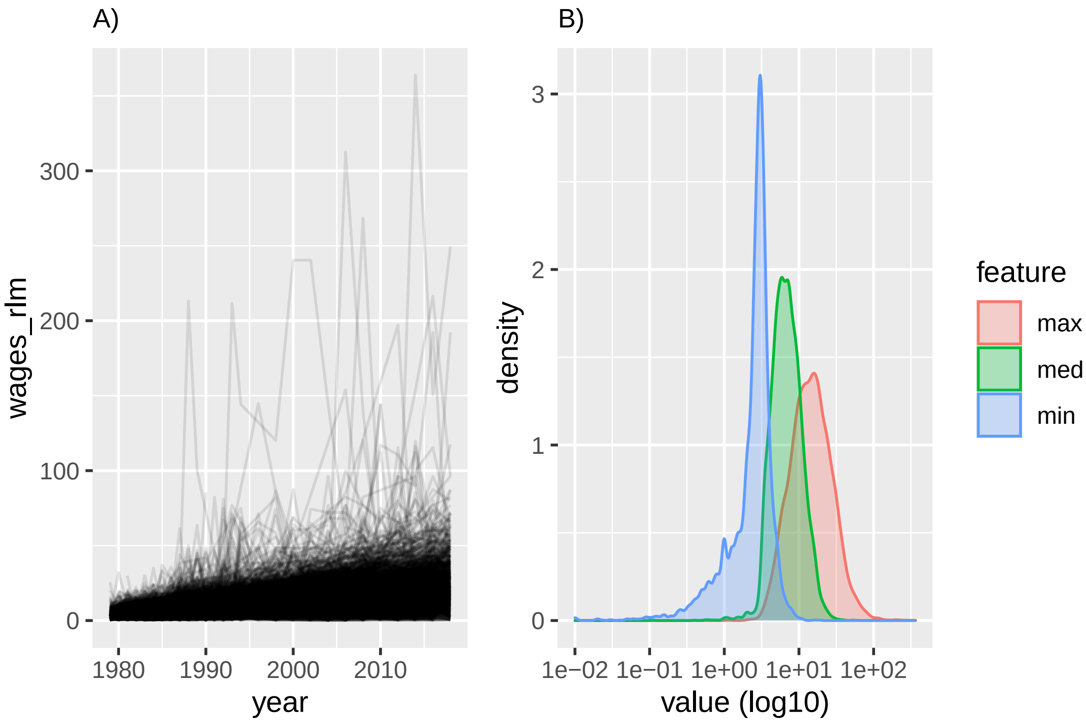


Figure 5: The distribution of the mean hourly wage after the extreme values are replaced. Plot A portrays the pattern of mean hourly wage of high school cohort from 1979 to 2018 of each ID in US Dollar; Plot B shows the distribution of their minimum, median, and maximum value transformed to log10 scale. We can see that some observations still had reasonbaly higher wages than the others. Also, some IDs' have a minimum wages that is higher than others' maximum wages.

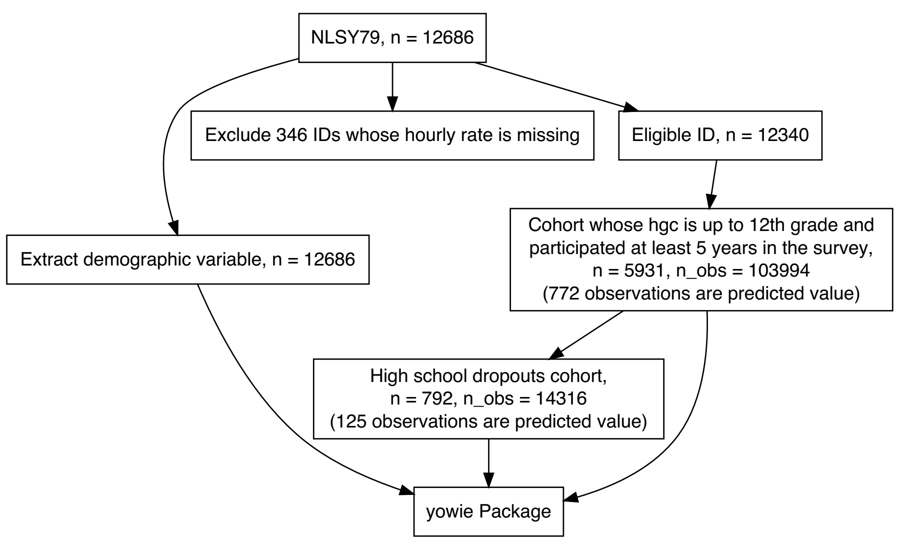


Figure 6: The stages of data filtering from the raw data to get three datasets that are contained in an R package called yowie, n means the number of ID, while n_obs means the number of observations.

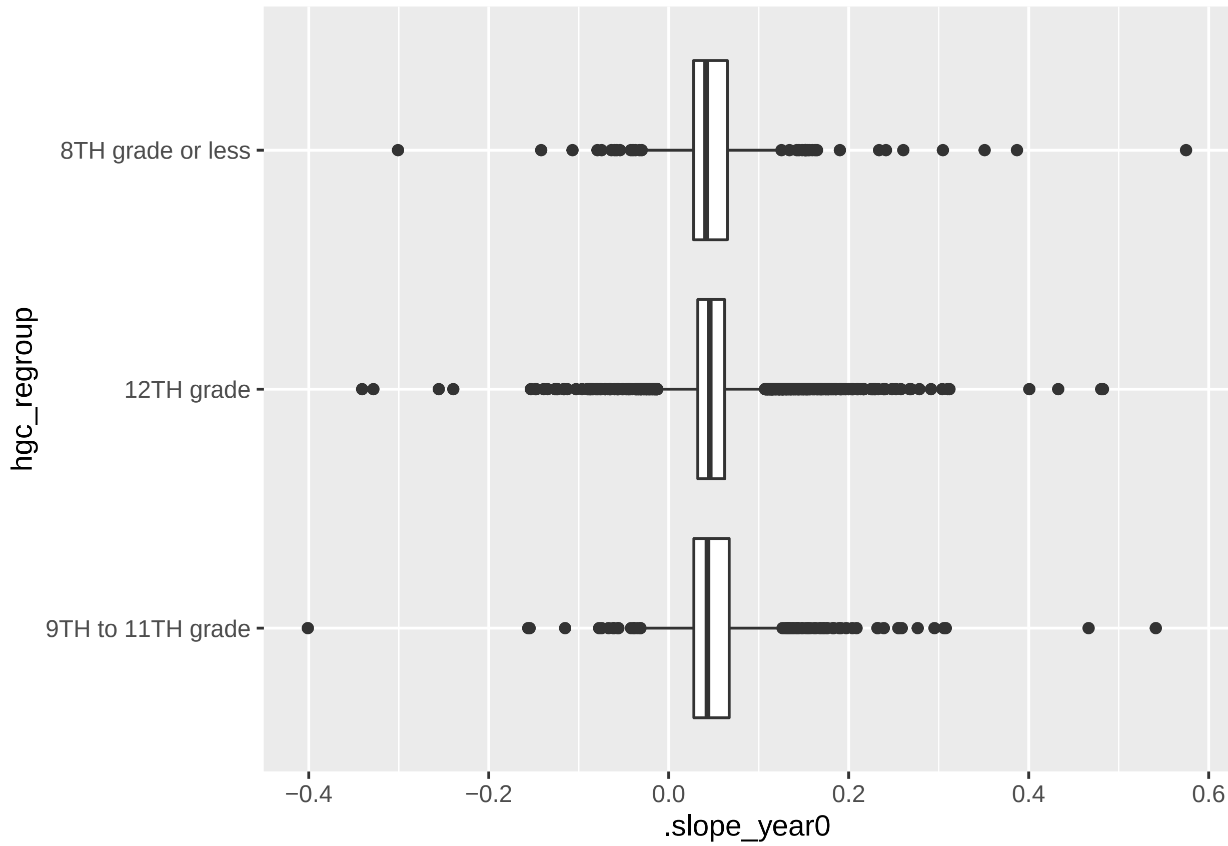
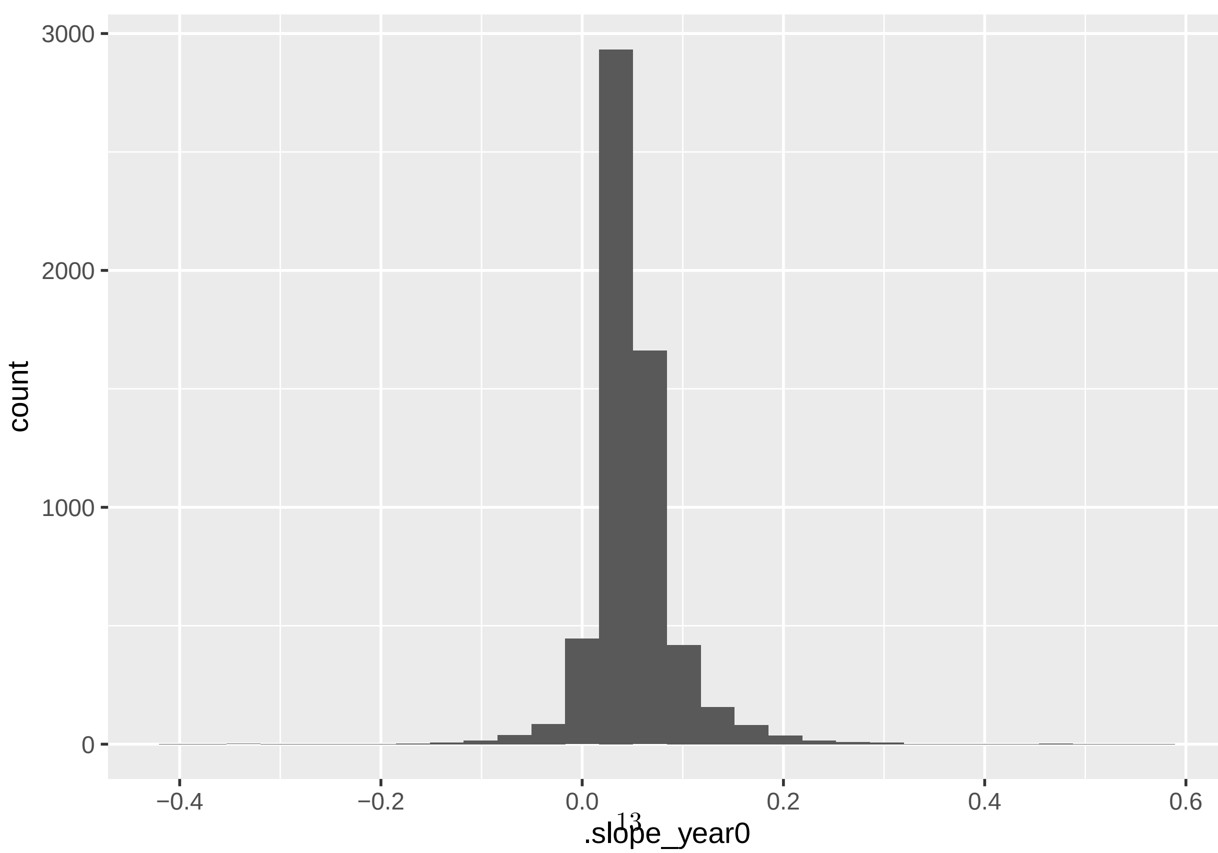
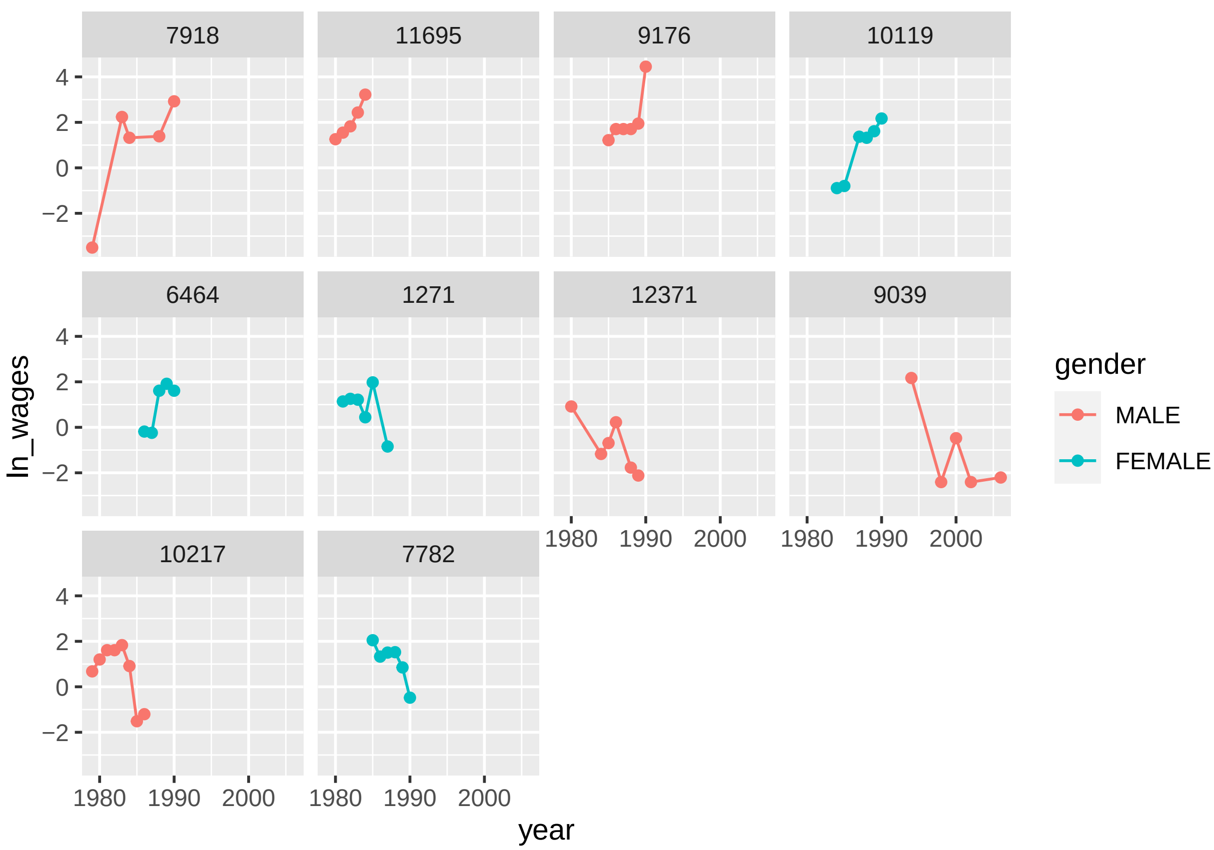


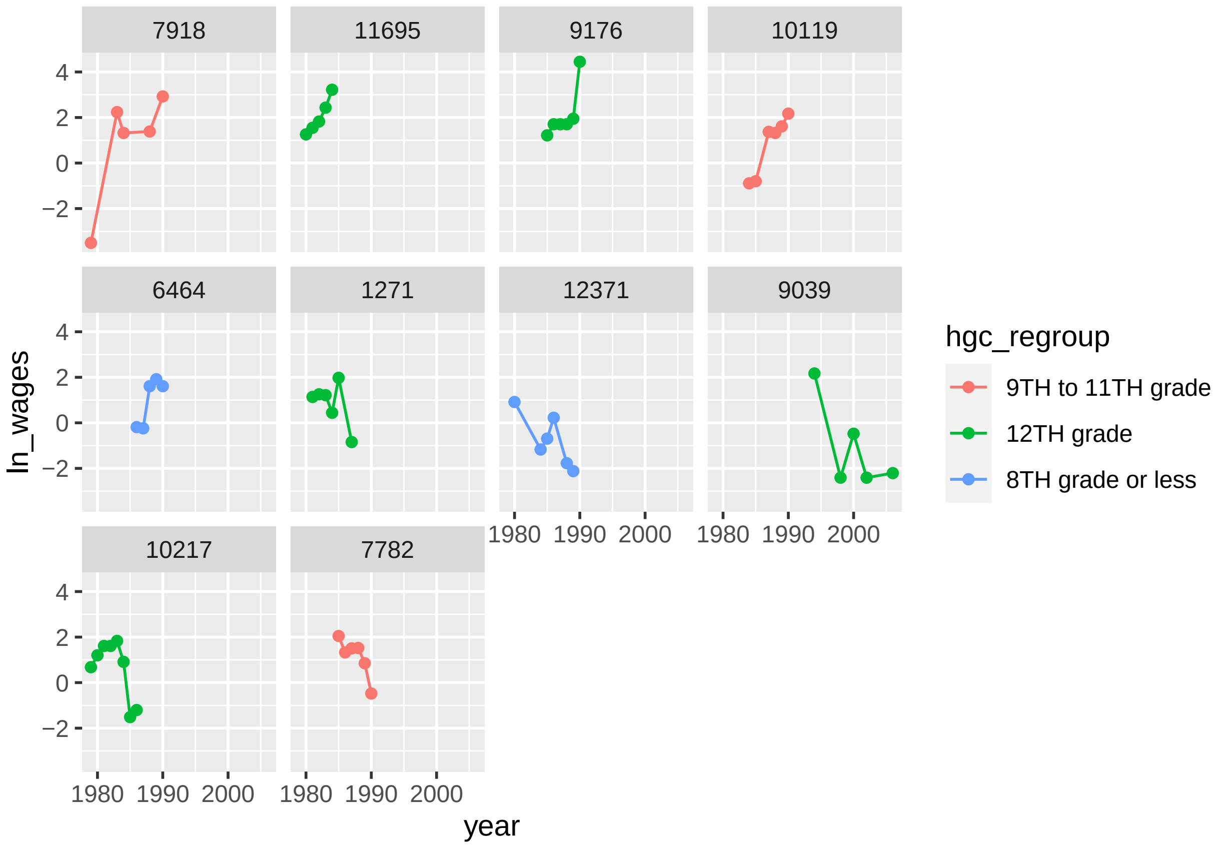
Figure 7: Regression slope of mean hourly wage regression by highest grade completed. The regression model has mean hourly wage as response variable and year as predictor. Respondents who completed 12th grade has the highest median of slope.



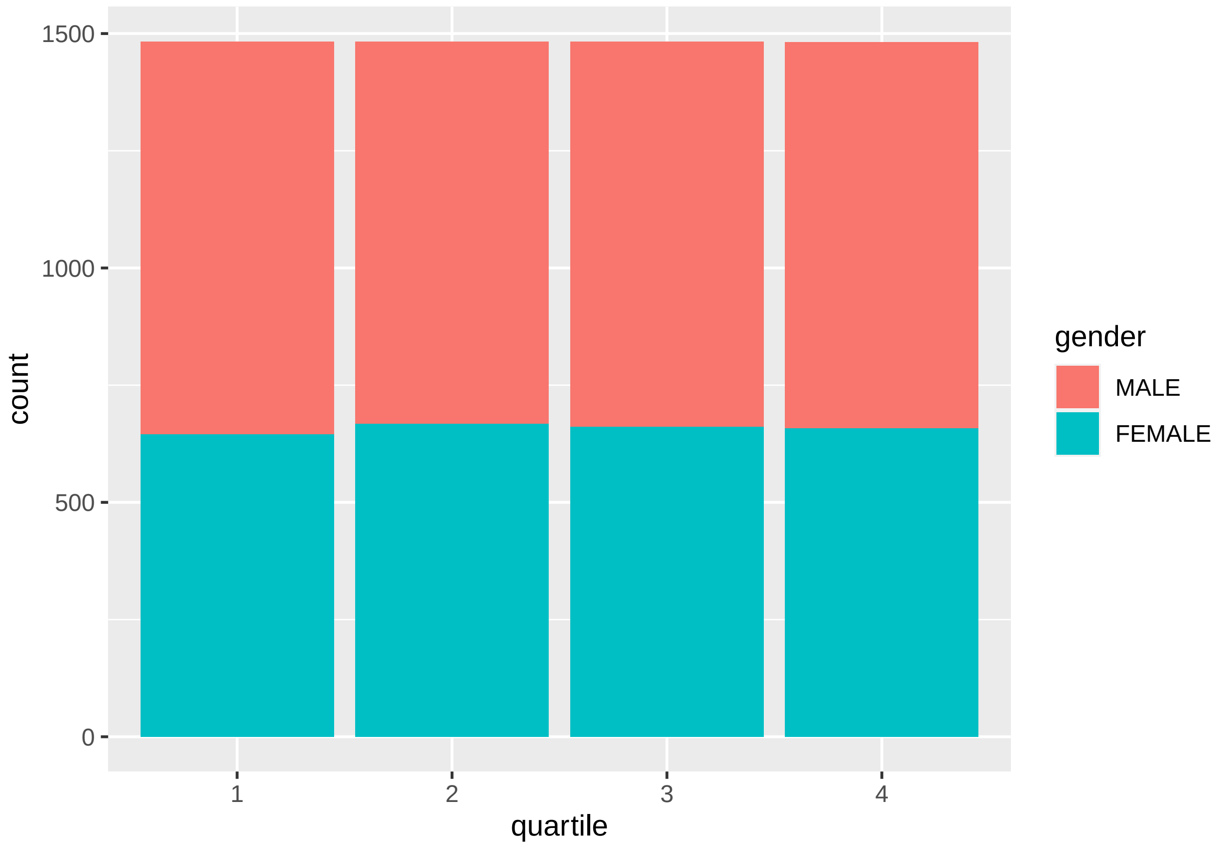
```
slope_bottom_5<- wages_slope %>%
  arrange(.slope_year0) %>%
  slice_head(n = 5)

slope_top_5<- wages_slope %>%
  arrange(desc(.slope_year0)) %>%
  slice_head(n = 5)

keep_contrast <- rbind(slope_bottom_5, slope_top_5) %>%
  dplyr::select(.slope_year0, id)

contrast_slope <- left_join(keep_contrast, wages_model_data, by = "id") %>%
  arrange(.slope_year0) %>%
  mutate(id = factor(id, levels = c("7918", "11695", "9176", "10119", "6464", "1271",
                                    "12371", "9039", "10217", "7782")))


ggplot(contrast_slope, aes(x = year, y = ln_wages, color = gender)) +
  geom_line() +
  geom_point() +
  facet_wrap(~id)
```



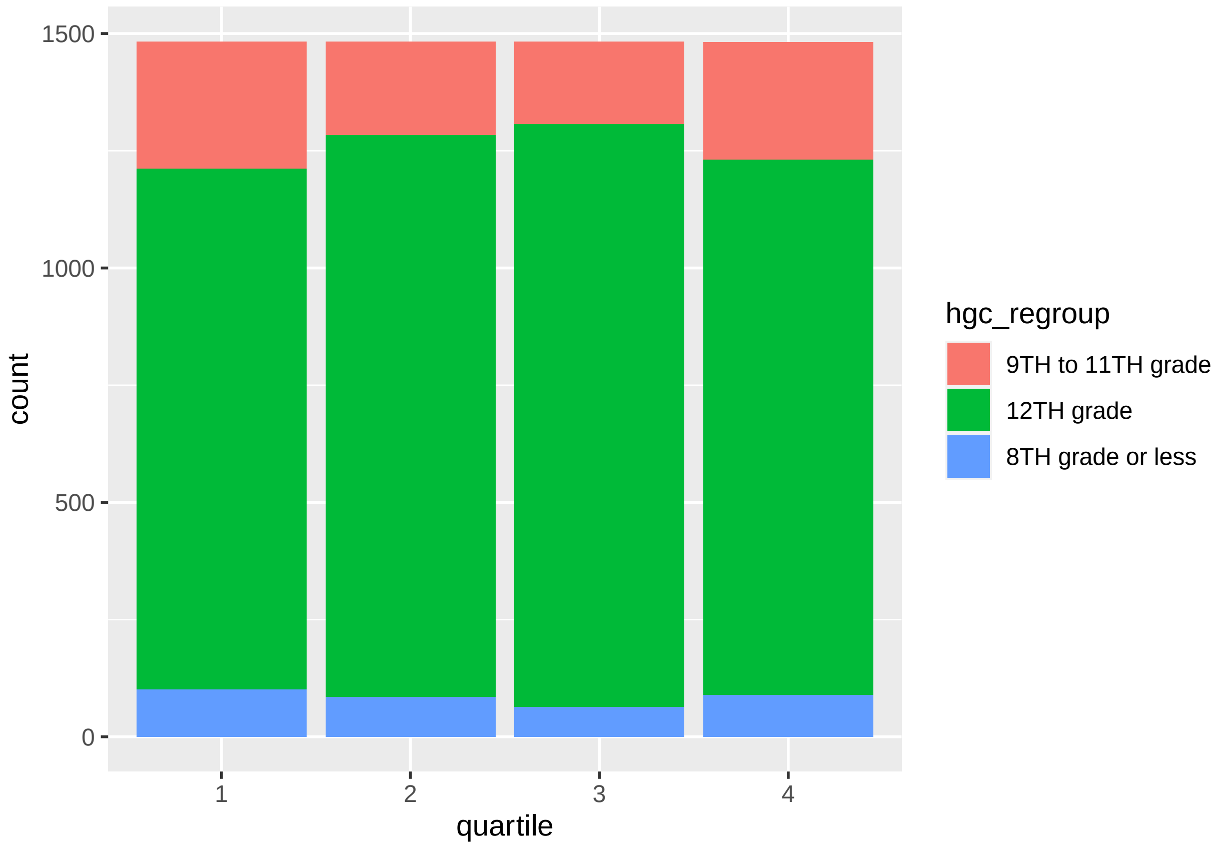

```
ggplot(contrast_slope, aes(x = year, y = ln_wages, color = hgc_regroup)) +
  geom_line() +
  geom_point() +
  facet_wrap(~id)
```




```
quartile <- wages_slope %>%
  mutate(quartile = as.factor(ntile(.slope_year0, 4)))

ggplot(quartile) +
  geom_bar(aes(x = quartile, fill = gender, position = "fill"))
```




```
ggplot(quartile) +
  geom_bar(aes(x = quartile, fill = hgc_regroup, position = "fill"))
```



In the second example, we examined the pattern of mean hourly wage corresponding to different groups of gender and race. We took 240 samples of females and males equally and spread them into twelve facets, as shown in Figure 8 using `facet_strata` function from `brolgar` (Tierney, Cook, and Prvan 2020). We learn that the mean hourly wage of females and males fluctuated over the years. For example, in facets 4, 5, and 10, males tend to earn more wages than females.

To further investigate this, we calculated the median wage of females relative to males, which is called gender wage gap. According to OECD (n.d.), gender pay gap is the difference between median wage of males and females relative to median wage of males. We find that, as shown in 9, over the years, females tend to earn fewer than males. For example, in 2018 females earn about 78 cents for every $1 earned by males. The widest gender gaps were observed in 1990s.

For the final example, we showed modeling the longitudinal data using linear mixed model with `lme4` package (Bates et al. 2015).

We fitted a more flexible model to fit into the data since the relationship between mean hourly wage and year is not linear due to the spikes. Since this approach is computationally expensive, we only showed the model in a small fraction of the data. We sampled 1 percent of the total respondents using `brolgar's` `sample_frac_keys` function. Further, we plotted some IDs overlayed by the fitted GAM model in Figure 10. We learn that the model is flexible enough to deal with the fluctuation in the data. Moreover, the data with small spikes tend to be linearly fitted.

5 Summary

This paper has performed a set of stages to make open data suitable for textbook data or make it ready for research. In the first stage, we showed the steps performed to get the data from the NLSY79 database. Since the data format is untidy, we showed how the data had been tidied. After that, we conducted initial data analysis to investigate and screen the quality of the data. Using the robust linear regression model, we found and fixed the anomalous observations in the data set with its predicted values. We also performed an example of exploratory data analysis using the cleaned data set.

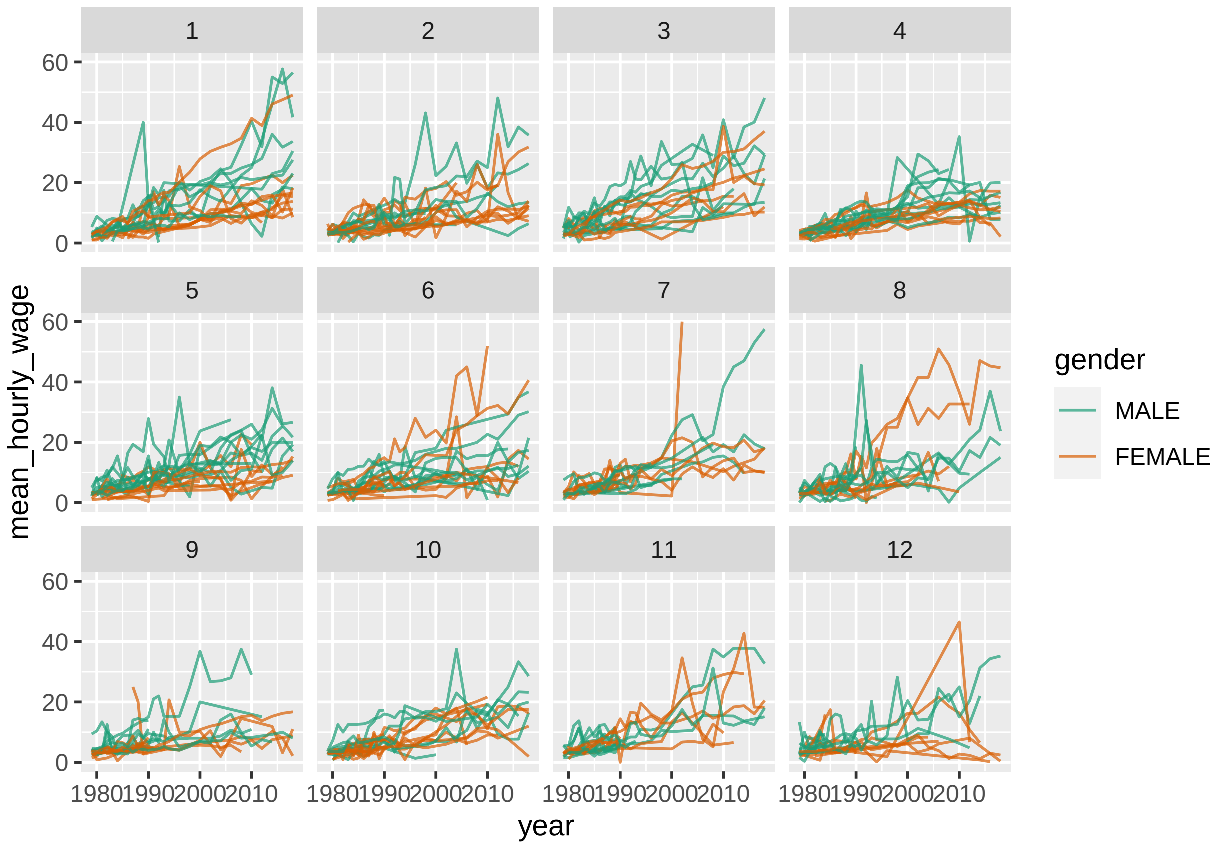


Figure 8: The pattern of mean hourly wages overtime displayed in 12 facets coloured by gender. The respondents belong to high school cohort are randomly sampled. We learn that both male and female respondents had a fluctuate mean hourly wage over the year.

This paper has also demonstrated data cleaning documentation by providing all of the codes in performing data tidying and initial data analysis. Thus, this paper provided an opportunity to continuously refresh the textbook data whenever the updated data is published in the NLSY79 database. It could be done by following the documentation of the code that is provided in this paper.

Moreover, the documentation also includes how we generated the robustness weight and how we decided the threshold of anomalous observations. It is also documented along with the flag of whether an observation is imputed value or not. Accordingly, if somebody wishes to make another decision, it can be done by making a small change in the code provided. Further, this paper is also supplemented by a `shiny` (Chang et al. 2020) app as a simulation tool to customize the weight threshold.

Finally, this paper implies that data providers should design a database that is able to produce tidy data sets. A data provider should also check for data anomalies before the data publishing or at least provides a set of rules or threshold values. For example, in this case, is the threshold of reasonable wages. This will greatly support the data users to validate and set the same understanding of which data are considered outliers. Moreover, providing validation rules would facilitate any established data validation tool, such as `validate` (van der Loo and de Jonge 2021) package. In this case, we cannot use this handy validation package due to the absence of validation rules.

6 Acknowledgements

We would like to thank Aarathy Babu for the insight and discussion during the writing of this paper.

The entire analysis is conducted using `R` (R Core Team 2020) in `rstudio` using these packages: `tidyverse` (Wickham et al. 2019), `ggplot2` (Wickham 2016), `dplyr` (Wickham et al. 2020), `readr` (Wickham and Hester 2020), `tidyr` (Wickham 2020b), `stringr` (Wickham 2019), `purrr` (Henry and Wickham 2020), `broom`

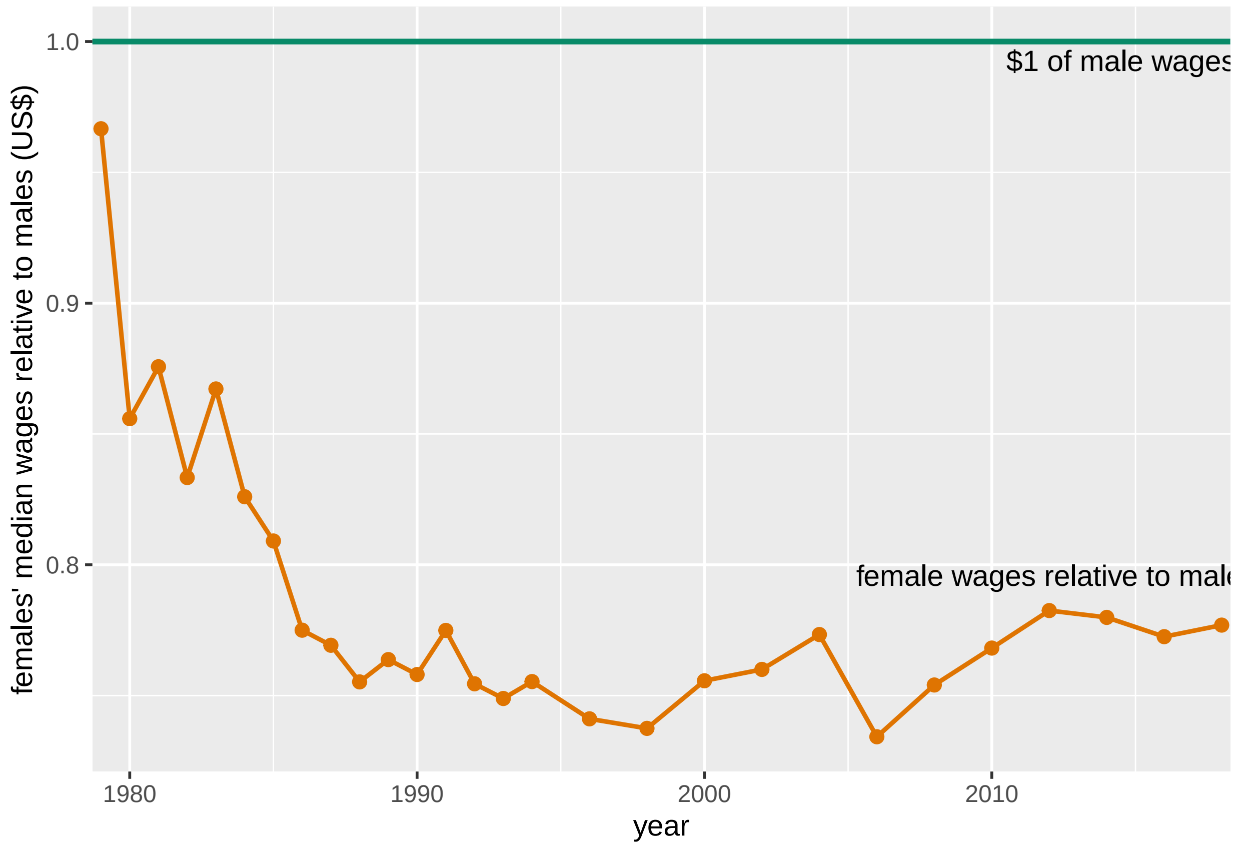


Figure 9: Gender wage gap or the median wages of females relatives to median wages of males. Females tend to earn fewer than males. For example, in 2018 females earn about 78 cents for every $1 earned by males.

(Robinson, Hayes, and Couch 2021), `blorgar` (Tierney, Cook, and Prvan 2020), `patchwork` (Pedersen 2020), `kableExtra` (Zhu 2019), `MASS` (Venables and Ripley 2002), `janitor` (Firke 2020), `DiagrammeR` (Iannone 2020), `rsvg` (Ooms 2020), `webshot` (Chang 2019), `mgcv` (Wood 2003), `tsibble` (Wang, Cook, and Hyndman 2020), and `modelr` (Wickham 2020a). The paper are generated using `knitr` (Xie 2014), `rmarkdown` (Xie, Dervieux, and Riederer 2020), and `rticles` (Allaire et al. 2021).

7 Supplementary Materials

- **Codes** : R script to reproduce data tidying and cleaning are available in this Github Repository.
- **R Package `yowie`**:`yowie` is a data container R package that contains 3 datasets, namely the high school mean hourly wage data, high school dropouts mean hourly wage data, and demographic data of the NLSY79 cohort. This package could be accessed here.
- **shiny app**: An web interactive `shiny` app to run a simulation to customize the weight threshold. This app could be accessed here.

References

Allaire, JJ, Yihui Xie, R Foundation, Hadley Wickham, Journal of Statistical Software, Ramnath Vaidyanathan, Association for Computing Machinery, et al. 2021. *Rticles: Article Formats for r Markdown*. `https://CRAN.R-project.org/package=rticles`.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. `https://doi.org/10.18637/jss.v067.i01`.

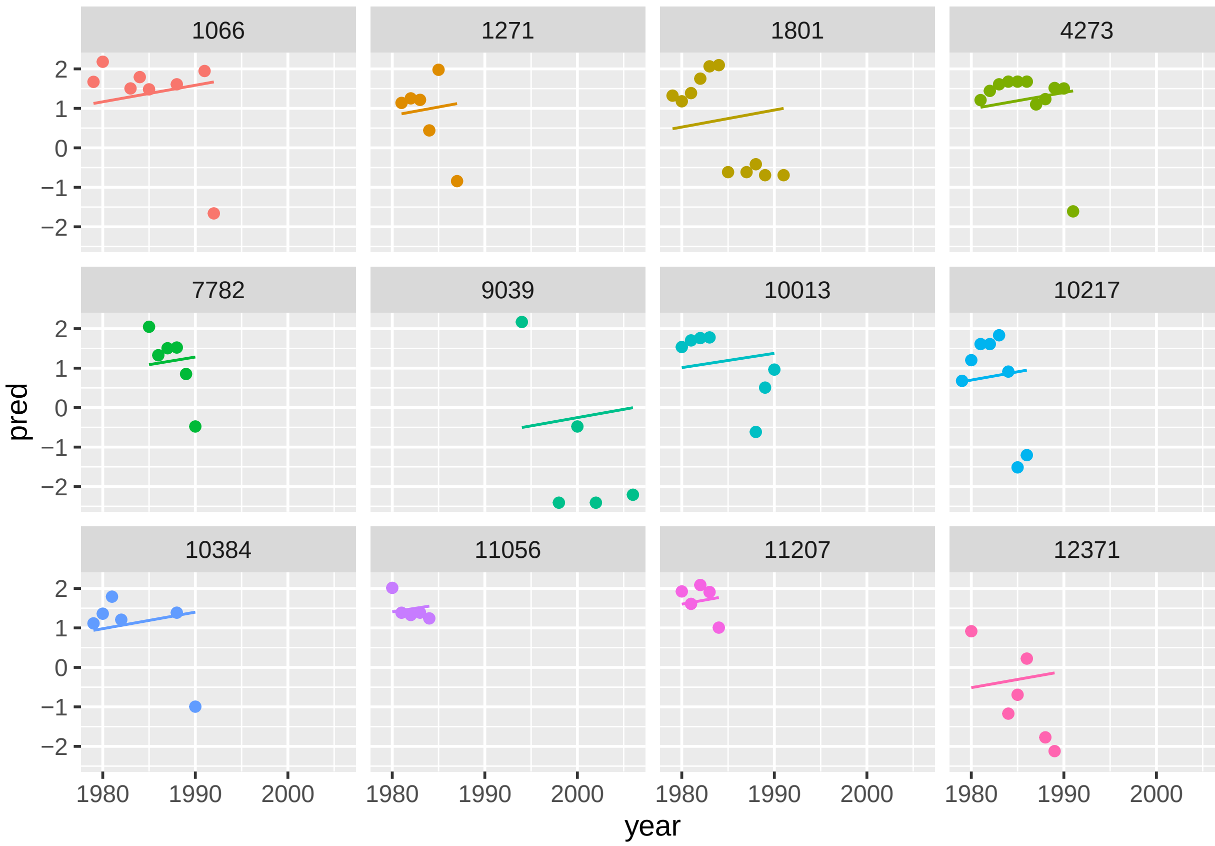


Figure 10: Exploration of wages data by fitting a GAM. The fitted model displayed by blue line. The fitted line shows that the model flexible enough to follow the pattern of the data.

Chang, Winston. 2019. *Webshot: Take Screenshots of Web Pages*. `https://CRAN.R-project.org/package=webshot`.

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. 2020. *Shiny: Web Application Framework for r*. `https://CRAN.R-project.org/package=shiny`.

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.

Firke, Sam. 2020. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. `https://CRAN.R-project.org/package=janitor`.

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

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

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.

Iannone, Richard. 2020. *DiagrammeR: Graph/Network Visualization*. https://CRAN.R-project.org/package=DiagrammeR.

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

OECD. n.d. "Gender Wage Gap." https://data.oecd.org/earnwage/gender-wage-gap.htm.

Ooms, Jeroen. 2020. *Rsvg: Render SVG Images into PDF, PNG, PostScript, or Bitmap Arrays*. https://CRAN.R-project.org/package=rsvg.

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

Pedersen, Thomas Lin. 2020. *Patchwork: The Composer of Plots*. https://CRAN.R-project.org/package=patchwork.

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.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David, Alex Hayes, and Simon Couch. 2021. *Broom: Convert Statistical Objects into Tidy Tibbles*. https://CRAN.R-project.org/package=broom.

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

The U.S. Bureau of Labor Statistics. n.d. "National Longitudinal Survey of Youth 1979." Available at https://www.nlsinfo.org/content/cohorts/nlsy79 (2021/25/02).

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

Tukey, John W. (John Wilder). 1977. *Exploratory Data Analysis*. Addison-Wesley Series in Behavioral Science. Reading, Mass.: Addison-Wesley Pub. Co.

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

van der Loo, Mark P. J., and Edwin de Jonge. 2021. "Data Validation Infrastructure for R." *Journal of Statistical Software* 97 (10): 1–31. https://doi.org/10.18637/jss.v097.i10.

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.

Wang, Earo, Dianne Cook, and Rob J Hyndman. 2020. "A New Tidy Data Structure to Support Exploration and Modeling of Temporal Data." *Journal of Computational and Graphical Statistics* 29 (3): 466–78. https://doi.org/10.1080/10618600.2019.1695624.

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.

———. 2020a. *Modelr: Modelling Functions That Work with the Pipe*. https://CRAN.R-project.org/package=modelr.

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

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

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

Wickham, Hadley, and Jim Hester. 2020. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wood, S. N. 2003. "Thin-Plate Regression Splines." *Journal of the Royal Statistical Society (B)* 65 (1): 95–114.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook.

Zhu, Hao. 2019. *kableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.