Introduction to Applied Data Science

Lecture 8: Work Experience on Income

Bas Machielsen
Utrecht University

2024-05-13

Work Experience on Income

Lecture 8: Getting Data, API's &

- Overview of this class:
  - Lecture 1: Introduction to Data Science & R
  - Lecture 2: Introduction to Programming
  - Lecture 3: Getting Data, API & Databases
  - Lecture 4: Getting Data, Web Scraping
  - Lecture 5: Transforming and Cleaning Data
  - Lecture 6: Spatial and Network Data
  - Lecture 7: Text Data & Text Mining
  - *This lecture*: Lecture 8: Data Science Project

Work Experience on Income

- As a data science project, we picked the following research question:

  > What is the effect of additional work experience on income?

- The structure of the project is as follows:
  - We are first going to look for **data** to use. Our data source is the IPUMS database.
  - Then, we read the **literature** to see how and what we should estimate
  - We **clean**, **tidy** and then **report** our data, and **estimate** our preferred models
  - We **interpret** our findings in light of the theory and policy implications

Data Source: IPUMS

> IPUMS provides census and survey data from around the world integrated across time and space. IPUMS integration and documentation makes it easy to study change, conduct comparative research, merge information across data types, and analyze individuals within family and community contexts. Data and services available free of charge.

- We will be using IPUMS International
- We need to access IPUMS through an *API key*
- In order to do so, you need to create an account at https://international.ipums.org/international/
- Once registered, you can create an API key

API Key

- API keys should be kept **secret**
- Hence, I should also not show my API key to you
- So what I do is:
    - I put my API key in a file
    - I read the file in R
    - You see the code I use to read the file but not the content of the file itself
    - I provide the content to the `set_ipums_api_key` function:

```r
library(ipumsr)
api_key ← read_lines('api_key_ipums.txt')
```

```r
set_ipums_api_key(api_key, save=TRUE)
```

Why API?

- Why would we extract our data via R, rather than going to the website and manually download files?

- From the `ipumsr` website:

Use of the IPUMS API enables the adoption of a programmatic workflow that can help users to:

- Precisely recreate the specifications of previous extract requests, making analysis scripts reproducible and self-contained
- Save extract request definitions that can be shared with others without violating IPUMS conditions
- Integrate the extract download process with functions to load data into R
- Quickly identify and explore available IPUMS data sources

How does this work?

- Again from the `ipumsr` website:

The basic workflow for interacting with the IPUMS API is as follows:

- Define the parameters of an extract request
- Submit the extract request to the IPUMS API
- Wait for an extract to complete
- Download a completed extract

Find Data We Need

- From the `ipumsr` package, we know a function exists to browse through the IPUMS data: `ipums_data_collections()`

```r
ipums_data_collections() |>
  head(5)
```

```
## # A tibble: 5 × 4
##   collection_name     collection_type code_for_api api_support
##   <chr>               <chr>           <chr>        <lgl>
## 1 IPUMS USA           microdata       usa          TRUE
## 2 IPUMS CPS           microdata       cps          TRUE
## 3 IPUMS International microdata       ipumsi       TRUE
## 4 IPUMS NHGIS         aggregate data  nhgis        TRUE
## 5 IPUMS IHGIS         aggregate data  ihgis        FALSE
```

- We are looking for the "IPUMS International" collection of **microdata**, which (fortunately) has API support!
- Presumably, we need to `code_for_api`, `ipumsi` to start working on this

Find Data We Need

- Now, we want to look for suitable datasets to conduct our analysis on
- In the documentation, we can again read that:

> Every microdata extract definition must contain a set of requested samples and variables. In an IPUMS microdata collection, a sample refers to a distinct combination of records and variables. A record is a set of values that describe the characteristics of a single unit of measurement (e.g. a single person or a single household), and variables define the characteristics that were measured.

- We can see what's available by using `get_sample_info("ipumsi")` and using `dplyr` functions to filter this

Find Data We Need

- It appears the most detailed microdata is available for Italy
- We will try to use these

```
get_sample_info("ipumsi") |>
  filter(str_detect(description, "Italy"))
```

```
## # A tibble: 12 × 2
##    name    description
##    <chr>   <chr>
##  1 it2001a Italy 2001
##  2 it2011a Italy 2011
##  3 it2011h Italy 2011 Q1 LFS
##  4 it2012h Italy 2012 Q1 LFS
##  5 it2013h Italy 2013 Q1 LFS
##  6 it2014h Italy 2014 Q1 LFS
##  7 it2015h Italy 2015 Q1 LFS
##  8 it2016h Italy 2016 Q1 LFS
##  9 it2017h Italy 2017 Q1 LFS
## 10 it2018h Italy 2018 Q1 LFS
## 11 it2019h Italy 2019 Q1 LFS
## 12 it2020h Italy 2020 Q1 LFS
```

Find Data We Need

- Let us combine the names for future reference:

```
names <- get_sample_info("ipumsi") |>
  filter(str_detect(description, "Italy")) |>
  pull(name) |>
  magrittr::extract(3:12)
```

Extract Data

- Now that we have identified the samples we're going to use, we can start extracting the data
- This is done using the four aforementioned steps: *define, submit, wait, download*
- We start with the first step:

```
define_extract_ipumsi(description="Italy Work Experience",
                      samples = names) # We saved these before
```

```
## Error in define_extract_micro(collection = "ipumsi", description = description, : ar
```

- Oops! It turns out we have to specify which variables we want. But how do we know which variables are contained in the survey?

Finding Variables

- Fortunately, we can access the variables in a notebook called the *DDI file*
- Once we obtain this file, we can read it with `read_ipums_ddi()`
- So far, this DDI file is not available yet in the `ipumsr` package but can be downloaded through the website
- By going to the Select Data Tab on the IPUMS International website, we can manually select variables and get a hold of this DDI file:
- I download the DDI file and import it in R using `read_ipums_ddi()`:

```
ddi_file ← read_ipums_ddi('ipumsi_00001.xml')
```

- This DDI file contains important information about the variables we select

Checking Variables

- We can now check the definitions of the variables:

```
variables ← ipums_var_info(ddi_file)
variables ▷ head(10)
```

```
## # A tibble: 10 × 10
##    var_name var_label                         var_desc val_labels code_instr start
##    <chr>    <chr>                             <chr>    <list>     <chr>      <dbl> <d
##  1 COUNTRY  Country                           "COUNTR… <tibble>    <NA>          1
##  2 YEAR     Year                              "YEAR g… <tibble>    <NA>          4
##  3 SAMPLE   IPUMS sample identifier           "SAMPLE… <tibble>    <NA>          8
##  4 SERIAL   Household serial number           "SERIAL… <tibble>   "SERIAL i…    17
##  5 HHWT     Household weight                  "HHWT i… <tibble>   "HHWT is …    29
##  6 PERNUM   Person number                     "PERNUM… <tibble>   "PERNUM i…    37
##  7 PERWT    Person weight                     "PERWT … <tibble>   "PERWT is…    41
##  8 RELATE   Relationship to household head … "RELATE… <tibble>    <NA>         49
##  9 RELATED  Relationship to household head … "RELATE… <tibble>    <NA>         50
## 10 ERELATE  Relationship to head, Europe      "ERELAT… <tibble>    <NA>         54
```

Downloading Data

- Now, we can finally download the data.
- We can do so with the help of the variables mentioned in the DDI file:
- We are looking for: a person identifier, age, sex, wage income, employment status, employment categories, educational attainment and tenure at employer

```
vars ←  c("PERNUM", "AGE", "SEX", "INCWAGE",
          "EEMPSTAT", "OCCISCO", "EDATTAIN", "WRKTENURE")
```

- First, create an extract

```
italy_extract ← define_extract_ipumsi("Italy Work Experience",
                                      samples=names,
                                      variables =vars)
```

Downloading Data

- Then, submit an extract:

```
italy_extract_submitted <- submit_extract(italy_extract)
```

- It may take some time for the IPUMS servers to process your extract request. You can ensure that an extract has finished processing before you attempt to download its files by using `wait_for_extract()`.
    - This polls the API regularly until processing has completed (by default, each interval increases by 10 seconds). It then returns an ipums_extract object containing the completed extract definition.

```
italy_extract_complete <- wait_for_extract(italy_extract_submitted)
italy_extract_complete$status
```

Downloading Data

- Finally, once completed, we can import the data into R:

```r
# By default, downloads to your current working directory
filepath ← download_extract(italy_extract_submitted)

# Import the file on the basis of the DDI File
ddi ← read_ipums_ddi(filepath)
micro_data ← read_ipums_micro(ddi)
```

Inspecting Data

- Now that we have the data, we can inspect it:

```
micro_data |> head(10)
```

```
## # A tibble: 10 × 15
##    COUNTRY      YEAR SAMPLE           SERIAL  HHWT PERNUM PERWT AGE   SEX     EDATTA
##    <int+lbl>   <int> <int+lbl>         <dbl> <dbl>  <dbl> <dbl> <int> <int+l> <int+l
##  1 380 [Italy]  2011 380201121 [Ital…   1000  506.      1  506. 75     2 [Fem… 1 [Les
##  2 380 [Italy]  2011 380201121 [Ital…   2000  536.      1  536. 75     2 [Fem… 2 [Pri
##  3 380 [Italy]  2011 380201121 [Ital…   3000  787.      1  787. 47     1 [Mal… 2 [Pri
##  4 380 [Italy]  2011 380201121 [Ital…   3000  787.      2  787. 47     2 [Fem… 2 [Pri
##  5 380 [Italy]  2011 380201121 [Ital…   3000  787.      3  787. 22     2 [Fem… 2 [Pri
##  6 380 [Italy]  2011 380201121 [Ital…   3000  787.      4  787. 17     2 [Fem… 2 [Pri
##  7 380 [Italy]  2011 380201121 [Ital…   4000  707.      1  707. 37     1 [Mal… 3 [Sec
##  8 380 [Italy]  2011 380201121 [Ital…   4000  707.      2  707. 37     2 [Fem… 3 [Sec
##  9 380 [Italy]  2011 380201121 [Ital…   4000  707.      3  707.  8     2 [Fem… 0 [NIU
## 10 380 [Italy]  2011 380201121 [Ital…   4000  707.      4  707.  4     1 [Mal… 0 [NIU
## # ℹ 2 more variables: WRKTENURE <dbl+lbl>, INCWAGE <dbl+lbl>
```

Variables Info

- We can also have a look at the variables which we have extracted:

```r
var_info <- ipums_var_info(ddi)
var_info |> head(10)
```

```
## # A tibble: 10 × 10
##    var_name var_label                          var_desc val_labels code_instr start
##    <chr>    <chr>                              <chr>    <list>     <chr>      <dbl> <d
##  1 COUNTRY  Country                            "COUNTR… <tibble>    <NA>           1
##  2 YEAR     Year                               "YEAR g… <tibble>    <NA>           4
##  3 SAMPLE   IPUMS sample identifier            "SAMPLE… <tibble>    <NA>           8
##  4 SERIAL   Household serial number            "SERIAL… <tibble>   "SERIAL i…     17
##  5 HHWT     Household weight                   "HHWT i… <tibble>   "HHWT is …     29
##  6 PERNUM   Person number                      "PERNUM… <tibble>   "PERNUM i…     37
##  7 PERWT    Person weight                      "PERWT … <tibble>   "PERWT is…     41
##  8 AGE      Age                                "AGE gi… <tibble>    <NA>          49
##  9 SEX      Sex                                "SEX re… <tibble>    <NA>          52
## 10 EDATTAIN Educational attainment, interna… "EDATTA… <tibble>    <NA>          53
```

Theoretical Framework

Standard Approach

- Altonji and Shakotko (1987) and Topel (1991) developed methodologies to deal with the inherent problem that the job match component in a standard log wage equation is not exogenous to **tenure** and **experience**:

$$W_{ijt} = \beta_x X_{ijt} + \beta_T T_{ijt} + \epsilon_{ijt}$$

where X_{ijt} represents the accumulated labor market experience and T_{ijt} tenure for person i in job j at time t

Also a job-person-time specific return, and a person-specific return:

$$\epsilon_{ijt} = \phi_{ijt} + \mu_i + \upsilon_{ijt}$$

And the job-person-time specific depends itself on experience and tenure:

$$\phi_{ijt} = \alpha_0 + \alpha_x X_{ijt} + \alpha_T T_{ijt} + \eta_{ijt}$$

We potentially want to know β_x but also β_T.

Empirical Strategy

- What these authors suggest is to do the following:

- First, estimate *within-job wage growth*: this gives us $\beta_T + \beta_x$

- Then, estimate the first equation using observations for *the first period for each job*
  - In this case, $T_{ijt} = 0$. By substituting equations 2 and 3 into equation 1, this will then give you $\beta_x + \alpha_x$
  - By subtracting the first estimate from the second, $\beta_T + \beta_x - [\beta_x + \alpha_x]$, we can also find $\beta_T - \alpha_x$.
  - If we then think that α_x, the effect of experience on job matching, is "small" (or zero), we can say we have found β_x and β_T.

- This is what we will attempt to do!

Cleaning Data

Cleaning Data

- But first, we have to clean the data.
- By looking at the variable definitions and their labels, we can see that there are many missing values coded as "99999" etc. We have to get these out:

```
md <- micro_data |>
  filter(INCWAGE < 9999999,
         EDATTAIN != 9,
         OCCISCO < 97,
         WRKTENURE < 998,
         AGE < 999,
         EEMPSTAT == 110)
```

Descriptive Statistics

Descriptive Statistics

- First, let us show what the data look like. We can do so with the help of the `modelsummary` package
- We want to show some "descriptive statistics" for each continuous variable:

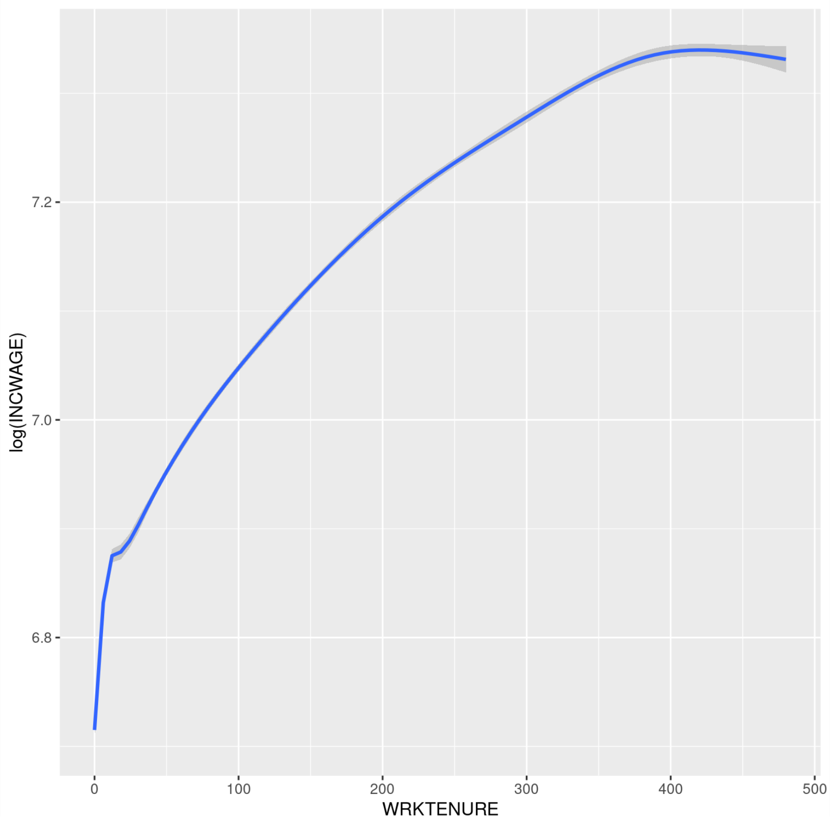
```
library(modelsummary)
datasummary(formula=YEAR + PERNUM + AGE + SEX + (`Educ. Attainment`=EDATTAIN) + WR
                 mean + median + sd + min + max + N, data=md)
```

	mean	**median**	**sd**	**min**	**max**	**N**
YEAR	2014.97	2015.00	2.95	2011.00	2020.00	289594
PERNUM	1.73	1.00	0.89	1.00	10.00	289594
Age	43.58	44.00	10.89	16.00	75.00	289594
Sex	1.47	1.00	0.50	1.00	2.00	289594
Educ. Attainment	2.82	3.00	0.68	1.00	4.00	289594
Tenure in current job (months)	154.06	120.00	127.98	0.00	480.00	289594
Wage and salary income	1306.02	1290.00	521.95	125.00	3000.00	289594

Descriptive Statistics

- In addition to testing our theories formally, we might also want to investigate some of the patterns in the data. For example, we might want to see how wage is correlated with job experience without applying the method mentioned before:

```
md ▷ ggplot(aes(x=WRKTENURE, y=log(INCWAGE))) + geom_smooth()
```



Discussion of Descriptives

- There seems to be preliminary evidence for our theory that work experience causes increase wages
- There seems to be a *marginally* decreasing benefit of experience, consistent with decreasing marginal returns to experience
- But how to separate experience in general from tenure at the job? Let us try to do that now

Empirical Approach

- Remember that we first wanted to estimate **within-job wage growth**. This gives us the combined effect of experience and tenure, $\beta_x + \beta_T$. We thus need a sample of individuals that haven't changed jobs between two (or more) waves of the survey. How to do that?

- We need to condition the sample on the people whose increase in tenure (in months) is longer than the increase in two subsequent survey waves the person is in

- First, let us make a variable indicating how many times this person has taken the survey:

```
md <- md |>
  group_by(SERIAL, PERNUM) |>
  mutate(HOWMANY = n())
```

Empirical Approach

- Second, let us sort the data and figure out how many months have been between two different surveys, and whether the increase in tenure has been at least as long, or longer?
    - Those are the **within-job wage growth** individuals we want to keep. Hence, I create a variable called `KEEP`.

```
md ← md ▷
  group_by(SERIAL, PERNUM) ▷
  arrange(SERIAL, PERNUM, YEAR) ▷
  mutate(KEEP = WRKTENURE/12 - lag(WRKTENURE)/12 > YEAR - lag(YEAR))
```

Estimate A Model

- Now, we can estimate a model with tenure and experience and estimate $\beta_x + \beta_T$
  - For convenience's sake, let us transform the tenure variable to years as well:

```
md ← md ▷
  mutate(WRKTENURE = WRKTENURE/12)
```

```
library(fixest)
model ← feols(log(INCWAGE) ~ WRKTENURE | SERIAL:PERNUM + YEAR, data = md ▷ filte

modelsummary(model, gof_map = c("r.squared", "nobs"), stars=stars)
```

	(1)
WRKTENURE	0.010***
	(0.000)
R2	0.715
Num.Obs.	59539

* p < 0.1, ** p < 0.05, *** p < 0.01

Interpretation

- What we find is a $\hat{\beta}$ coefficient of about 0.01.
- This implies that a year increase in tenure/experience is associated with about a 1% increase in wages
- Note that this represents the *combined* effect of tenure and experience, $\beta_x + \beta_T$.
- The next thing that we need to do is estimate equation (1) for the first period in each job.
    - We will thus focus on observations for which `WRKTENURE` (tenure) is lower than 1 year:
    - This will ultimately give us an estimate of $\beta_x + \boldsymbol{\alpha}_x$.
    - To do this, we need to focus on observations with a low tenure.
    - We don't have a direct variable indicating work experience. However, we will focus on age as a proxy for work experience.
    - Conditional on education (a control variable we will add), age is extremely highly correlated with work experience.

Second Stage

- Let us implement this:

```r
model_2 <- feols(log(INCWAGE) ~ AGE | SERIAL:PERNUM + YEAR + as.factor(EDATTAIN),
                 data = md |> filter(WRKTENURE < 1))

modelsummary(model_2,  gof_map = c("r.squared", "nobs"), stars=stars)
```

	(1)
AGE	0.004***
	(0.001)
R2	0.879
Num.Obs.	29125
* p < 0.1, ** p < 0.05, *** p < 0.01	

Conclusion

Interpretation

- What we find here is that the effect of work experience wage is 0.003
    - Which means that an additional year in experience implies a wage increase of 0.3%.
    - Hence, in our previous terminology, $\boldsymbol{\beta_x + \alpha_x \approx 0.003}$
    - And $\boldsymbol{\beta_T - \alpha_x \approx 0.007}$
    - If we assume that $\boldsymbol{\alpha_x \approx 0}$, then we can say that:
- An increase in *work experience* can be decomposed into an increase in *tenure* at the same employer, and an increase in work experience irrespective of the employer
- We find that the observed effect is largely due to *increases at the same employer*, as evidenced by the $\boldsymbol{\beta_T \approx 0.007 > \beta_x \approx 0.003}$.
- This implies that a year of additional experience at the same employer (tenure) means a 0.7% increase in wages, whereas a year of additional experience in general implies a 0.3% increase in wages

References

Altonji, Joseph G. and Shakotko, R.A. (1987). Do Wages Rise with Job Seniority? Review of Economic Studies 54 (July): 437-439.

Connolly, H., & Gottschalk, P. (2000). Returns to Tenure and Experience Revisited--Do Less Educated Workers Gain Less from Work Experience?. Working Papers in Economics, 147.

Steven Ruggles, Lara Cleveland, Rodrigo Lovaton, Sula Sarkar, Matthew Sobek, Derek Burk, Dan Ehrlich, Quinn Heimann, Jane Lee. Integrated Public Use Microdata Series, International: Version 7.5 [dataset]. Minneapolis, MN: IPUMS, 2024. https://doi.org/10.1 [dataset]. Minneapolis, MN: IPUMS, 2024. https://doi.org/10.18128/D020.V7.5

Topel, Robert. Specific Capital, Mobility, and Wages: Wages Rise with Job Seniority . Journal of Political Economy, 1991, vol. 99, no 1