Employment Effects of the new German minimum wage (SOEP dataset)

Meret Borchmann
Jupp Kerschek
Albert Thieme
Timm Walz

Ladislaus von Bortkiewicz Chair of Statistics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de

Outline

1. Motivation
2. SOEP dataset
3. Implementation of data
4. Identifying treatment and control group
5. Characteristics of individuals
6. Future outlook: Regression analysis

Economic Theory

Goal: Find out employment effects of the introduction of German minimum wage

- Potential effects of minimum wage on labor demand depend on market structure
- Competitive price-taker setting: Negative employment effects
- Monopsonic price setting: Possibility of positive employment effects
- Empirical research necessary

What is the SOEP

- Sozio-oekonomische Panel
- Annual survey of over 30.000 individuals, conducted by DIW
- Used for a wide variety of research: Economics, social science

Why SOEP

- Representative sample of people living in Germany
- Panel data set
- Data on micro level for each individual
- For free

Allows us to identify an effect on employment

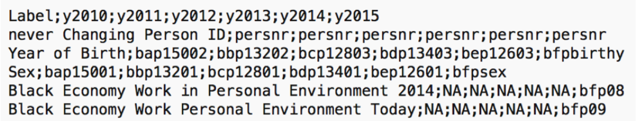
Variable Selection

- Surveys/datasets have different variables over the years
- Variable selection based on our research focus
- Only take the relevant datasets and extract the relevant information
- Current focus: datasets from 2010 and 2015
- Code must be dynamic to avoid future code adjustments (e.g. further years)

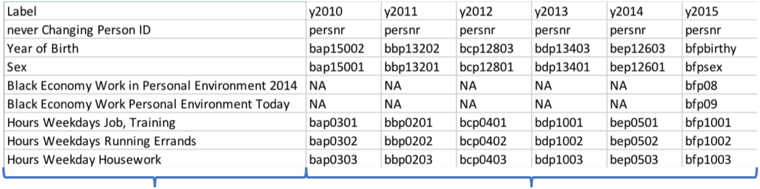
Procedure

- Import of variable list as .csv
- Store relevant data in a subfolder of *"input-data"* named by the year
- Code crawls the data and merges it by the unique identifier *"persnr"*
- Filtering of the dataset based on variable list
- Labeling of the data with comprehensive/understandable names

Procedure



```
Label;y2010;y2011;y2012;y2013;y2014;y2015
never Changing Person ID;persnr;persnr;persnr;persnr;persnr;persnr
Year of Birth;bap15002;bbp13202;bcp12803;bdp13403;bep12603;bfpbirthy
Sex;bap15001;bbp13201;bcp12801;bdp13401;bep12601;bfpsex
Black Economy Work in Personal Environment 2014;NA;NA;NA;NA;NA;bfp08
Black Economy Work Personal Environment Today;NA;NA;NA;NA;NA;bfp09
```



Label	y2010	y2011	y2012	y2013	y2014	y2015
never Changing Person ID	persnr	persnr	persnr	persnr	persnr	persnr
Year of Birth	bap15002	bbp13202	bcp12803	bdp13403	bep12603	bfpbirthy
Sex	bap15001	bbp13201	bcp12801	bdp13401	bep12601	bfpsex
Black Economy Work in Personal Environment 2014	NA	NA	NA	NA	NA	bfp08
Black Economy Work Personal Environment Today	NA	NA	NA	NA	NA	bfp09
Hours Weekdays Job, Training	bap0301	bbp0201	bcp0401	bdp1001	bep0501	bfp1001
Hours Weekdays Running Errands	bap0302	bbp0202	bcp0402	bdp1002	bep0502	bfp1002
Hours Weekday Housework	bap0303	bbp0203	bcp0403	bdp1003	bep0503	bfp1003

Labels, column = 1

Variables, column 2+, extendable

Advantages of the procedure

- Easy data preparation
- Resource-efficient, as only filtered data will stay in memory
- Not "hard coded": Extendable in the future by just adding directories and files

Quantlet: Import Data

```
# We need two control variables:

# i is to step through the list of years, beginning with 1

# k is always one digit higher as it reads the second column of the feature selection list (the first column is the label)
 i <- 1
 k <- 2

# List all directories within the input data, non-recursive
 list_dirs <- list.dirs(path="input-data", recursive=FALSE)
```

Quantlet: Import Data

```
# Extract the year name of the directories, so the last 4 digits
list_years <- str_sub(list_dirs, -4)

# Create Variable names for every merged year based on the style merged[year]
  list_varnames <- paste("merged", list_years, sep="")

# Load the feature list we cleaned manually in Excel as CSV
  soep_selection <- read.table("variable-selection/soep-feature-selection.csv", header = TRUE, sep = ";", check.names=FALSE)
```

Quantlet: Import Data

```
# Get all Labels, unfiltered
  labels <- soep_selection[,1]

# Create a vector to put object names of all years in it
  datalist <- c()

# Loop through all the years, import the data, merge, clean and label them
  for (years in list_years) {

  # Define Current List of import data based on the "i" value
    list_files <- list.files(path=list_dirs[i], pattern = "", full.names=TRUE)
```

Quantlet: Import Data

```
    # Import all the data from the current list with the read.dta-Function for SPSS-Files
    list_import <- lapply(list_files, read.dta)

    # Merge it into one file
    data_merged <- Reduce(function(x, y) merge(x, y, by=’persnr’, all.x=TRUE), list_import)

    # Cut the .x and y. values from the merge process, so that we have clean column names
    colnames(data_merged) <- gsub("\\.x|\\.y", "", colnames(data_merged))

    # Get the feature list of the current year
    current_list <- sort(soep_selection[,k])
```

Quantlet: Import Data

```
    # Delete all columns where no data exists (as
      the surveys differed every year) -> not needed
       as import function excludes missing values
    # shortlist <- na.omit(current_list)
    # ONLY take the data shortlisted for the current
       year
    cleaned <- data_merged[ ,which(names(data_merged
      ) %in% current_list==TRUE)]

    # Select the Label Column and the Variable
      Column of the current Year
    soep_subcrit <- c(1, k)
```

Quantlet: Import Data

```
    # Subset the Feature list so that only the label and the current year exist
    soep_selection_sub <- soep_selection[soep_subcrit]

    # Delete NA-Values from the list
    soep_selection_sub <- na.omit(soep_selection_sub)

    # Create a subset of the clean labels, where all codenames match, to make sure that the labels are correct
    clean_labels <- subset(soep_selection_sub, sort(soep_selection_sub[,2]) ==  sort(names(cleaned)))
```

Quantlet: Import Data

```
    # Order Dataframe alphabetically
    clean_sorted <- cleaned[ , order(names(cleaned)) ]

    # Order Frame with the Labels based on the ID
    ordered_colnames <- clean_labels[order(clean_labels[2]), ]

    # Label the columns properly
    colnames(clean_sorted) <- ordered_colnames[,1]

    # Assign data_merged to current merge[year]
      assign(list_varnames[i], clean_sorted)
```

Quantlet: Import Data

```
   # Add Year Variable to a list so that we can access all years by a loop
    datalist <- c(datalist, list_varnames[i])

   # Update our variables for the next round
    i <- i + 1
    k <- k + 1
  }
```

Idea

Treatment

- ⊡ 2015: Individuals affected by minimum wage
- ⊡ Before 2015: Individuals with hourly wage < 8.50€

Control Group

- ⊡ 2010 - 2015: Individuals with hourly wage > 8.50€
- ⊡ Three different control groups
  - ▶ 1. Hourly Wage: 8.50€$< C_1 <$ 9.00€
  - ▶ 2. Hourly Wage: 8.50€$< C_2 <$ 9.50€
  - ▶ 3. Hourly Wage: 8.50€$< C_3 <$ 10.00€

Procedure

- Generate 4 dummy variables that indicate whether individual belongs to Treatment or Control groups
  - Example: Value *"1"* if in treatment, otherwise value *"0"*
- Need to generate a *"hourly wage"* variable
  - $\frac{\text{Current Gross Labor Income in Euro}}{4 * \text{Actual Work Time per Week}}$

Quantlet: Generating treatment and control identifier

```
# Treatment dummy

  # 2015
      # Minimum wage earnings are split in 2 variables "Minimum Wage EUR" and "Minimum Wage Cent"
      # The values "-2" and "-1" refer to "Does not apply" and "No answer" respectively
      merged2015$‘Minimum Wage‘ <- merged2015$‘Minimum Wage EUR‘ + merged2015$‘Minimum Wage Cent‘/100
```

Quantlet: Generating treatment and control identifier

```
# Create the treatment dummy
merged2015$‘Treatment‘[merged2015$‘Minimum Wage‘ > 0 &  merged2015$‘Minimum Wage‘ <= 8.5] <- 1

merged2015$Treatment[is.na(merged2015$Treatment)] <- 0
```

Quantlet: Generating treatment and control identifier

```
# 2010 - 2014
    # Loop function
    # Generate variable "Hourly earnings"
    # Use "Current Gross Labor Income in Euro" and "Actual Work Time Per Week"
```

Quantlet: Generating treatment and control identifier

```
    #Create hourly wage:
        y <- 1
        for (years in c(datalist)) {
            # Rewrite the minus values as NA
            current_year <- datalist[y]

            current_data <- get(current_year)

            current_data$‘Actual Work Time Per Week
              ‘[current_data$‘Actual Work Time Per
              Week‘ == -1] <- NA

            current_data$‘Actual Work Time Per Week
              ‘[current_data$‘Actual Work Time Per
              Week‘ == -2] <- NA
```

Quantlet: Generating treatment and control identifier

```
          current_data$`Actual Work Time Per Week`[current_data$`Actual Work Time Per Week` == -3] <- NA

          current_data$`Current Gross Labor Income in Euro`[current_data$`Current Gross Labor Income in Euro` == -1] <- NA

          current_data$`Current Gross Labor Income in Euro`[current_data$`Current Gross Labor Income in Euro` == -2] <- NA
```

Quantlet: Generating treatment and control identifier

```
        # Variable "Hourly earnings"
        current_data$‘Hourly earnings‘ <-
          current_data$‘Current Gross Labor Income
          in Euro‘ / (4 * current_data$‘Actual Work
          Time Per Week‘)

        current_data$‘Treatment‘[current_data$‘
          Hourly earnings‘ < 8.5] <- 1

        current_data$Treatment[is.na(
          current_data$Treatment)] <- 0

        assign(current_year, current_data)
        y <- y + 1
          }
```

Quantlet: Generating treatment and control identifier

```
# Control Dummies
  # Use a loop function
  # Use the generated "hourly earnings" variable

    y <- 1
    for (years in c(datalist)) {
      current_year <- datalist[y]

      current_data <- get(current_year)
```

Quantlet: Generating treatment and control identifier

```
  # 1. Control Dummy

      current_data$‘Control_1‘[current_data$‘
        Hourly earnings‘ >= 8.5 & current_data$‘
        Hourly earnings‘ < 9] <- 1

      current_data$‘Control_1‘[is.na(
        current_data$Control_1)] <- 0
```

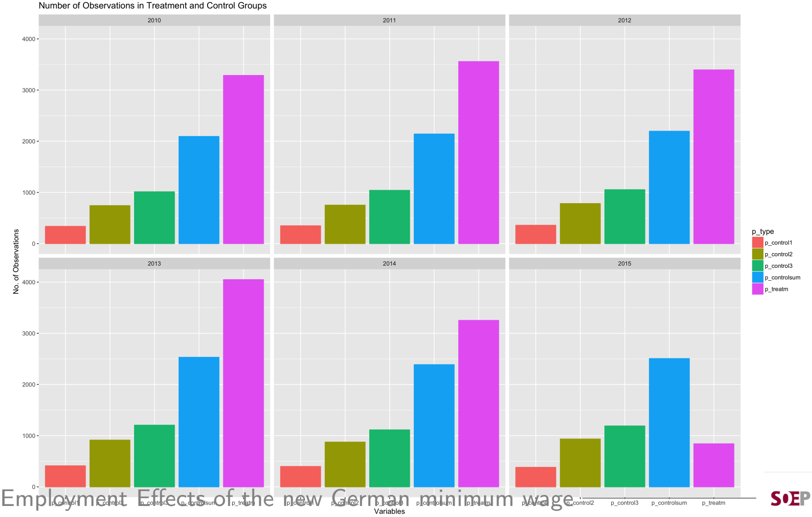
Quantlet: Generating treatment and control identifier

```
    # 2. Control Dummy

        current_data$‘Control_2‘[current_data$‘
          Hourly earnings‘ >= 8.5 & current_data$‘
          Hourly earnings‘ < 9.5] <- 1

        current_data$‘Control_2‘[is.na(
          current_data$Control_2)] <- 0
```

Quantlet: Generating treatment and control identifier

```
    # 3. Control Dummy

         current_data$‘Control_3‘[current_data$‘
           Hourly earnings‘ >= 8.5 & current_data$‘
           Hourly earnings‘ < 10] <- 1

        current_data$‘Control_3‘[is.na(
          current_data$Control_3)] <- 0

     # Assign it to the correct year
            assign(current_year, current_data)

            y <- y + 1
      }
```

Result


Number of Observations in Treatment and Control Groups

Result

- Convenient number of observations for analysis
  - Treatment Group: 3200 - 4000 obs. (2010 - 2014), 842 obs. (2015)
  - Control Groups: 300 - 1200 obs. (2010 - 2015)
- Code for graph quantlet can be found in Git-directory

Idea

- Economic theory: Wage depends on various factors
  - Human capital: Education, Working Experience, Age...
- Find out whether these factors are similar across groups

Procedure

- Analyze following variables: Sex, Education, Year of birth, Working experience, Minimum Wage, Nationality, Labor income, Work time
- Convert variables to numeric
- Value cleaning
  - Negative values to NA
  - Adjust scale
- Use of summary command for analysis

Quantlet: Characteristics of Individuals

```
# Analysis of treatment & control group
    y = 1
    for (years in c(datalist)) {
     current_year = datalist[y]
     current_data = get(current_year)
     print(current_year)
```

Quantlet: Characteristics of Individuals

```
# rewrite Sex as numeric and shows the gender variable of the Treatment group (1 for women)
        current_data$Sexnum = NA
        current_data$Sexnum = as.numeric(current_data$Sex) - 7
        current_data$Sexnum[current_data$Sexnum <= -1] = NA
        summary(current_data$Sexnum[current_data$Treatment == 1])
        table(current_data$Sexnum[current_data$Treatment == 1])
```

Similar code for other variables

Quantlet: Characteristics of Individuals

```
# results: treatment group
summary(current_data$Sexnum[current_data$Treatment == 1])
summary(current_data$‘Amount of Education Or Training in Years‘[current_data$Treatment == 1])
summary(current_data$German[current_data$Treatment == 1])
summary(current_data$‘Current Gross Labor Income in Euro‘[current_data$Treatment == 1])
summary(current_data$‘Actual Work Time Per Week‘[current_data$Treatment == 1])
```

Quantlet: Characteristics of Individuals

```r
summary(current_data$working_exp_with_NA[
  current_data$Treatment == 1])
summary(current_data$`Year of Birth`[current_data$`
  Year of Birth` > 0 & current_data$Treatment == 1])
if("merged2015" == current_year) {
summary(current_data$Minwagenona[
  current_data$Treatment == 1])
summary(current_data$getminwage[
  current_data$Treatment == 1])
```

Full Code in Git - directory

Results: 2010

Table 1: Descriptive Statistic of Treatment Group 2010

Statistic	N	Mean	St. Dev.	Min	Max
Amount of Education Or Training in Years	3,076	11.649	2.197	7.000	18.000
Actual Work Time Per Week	3,286	32.821	16.817	1.000	80.000
Current Gross Labor Income in Euro	3,286	703.556	493.470	0	2,500
Sexnum	2,172	0.584	0.493	0	1
German	3,286	0.928	0.259	0	1
working_exp_with_NA	3,049	9.754	11.259	0.000	59.300
Birth Year	2,172	1,971.259	14.895	1,922	1,992

Results: 2010

Table 2: Descriptive Statistic of Control Group 2010

Statistic	N	Mean	St. Dev.	Min	Max
Amount of Education Or Training in Years	987	11.727	2.038	7.000	18.000
Actual Work Time Per Week	1,013	36.256	15.756	2.000	80.000
Current Gross Labor Income in Euro	1,013	1,334.210	583.430	70	3,116
Sexnum	681	0.589	0.492	0	1
German	1,013	0.926	0.262	0	1
working_exp_with_NA	939	13.005	11.000	0.000	48.000
Birth Year	681	1,968.326	12.412	1,932	1,991

Results: 2015

Table 3: Descriptive Statistic of Treatment Group 2015

Statistic	N	Mean	St. Dev.	Min	Max
Amount of Education Or Training in Years	3,111	11.651	2.371	7.000	18.000
Actual Work Time Per Week	3,425	32.838	15.091	1.000	80.000
Current Gross Labor Income in Euro	3,441	980.551	983.931	0	18,159
Sexnum	3,148	0.589	0.492	0	1
German	3,441	0.807	0.395	0	1
working_exp_with_NA	3,394	9.215	11.496	0.000	54.000
Minwagenona	912	8.508	0.587	3.000	13.950
Birth Year	3,148	1,976.929	14.645	1,928	1,997

Results: 2015

Table 4: Descriptive Statistic of Control Group 2015

Statistic	N	Mean	St. Dev.	Min	Max
Amount of Education Or Training in Years	1,136	11.539	2.187	7.000	18.000
Actual Work Time Per Week	1,190	32.064	15.384	1.000	80.000
Current Gross Labor Income in Euro	1,190	1,177.287	568.195	34	3,000
Sexnum	1,052	0.663	0.473	0	1
German	1,190	0.768	0.422	0	1
working_exp_with_NA	1,179	11.483	11.366	0.000	55.700
Birth Year	1,052	1,972.520	12.466	1,937	1,997

Results - To Do

- Clean observations with missing values
- Calculate differences of the variables
- Check whether these are significant

Future Outlook: Regression analysis

- Use regression discontinuity to estimate the effect of minimum wage on employment
- Robustness checks and tests to validate significance and robustness of estimation