````
---
title: "Driva et al. Bismarck Replication"
date: "2023"
output:
  html_document: default
  pdf_document: default
---

# Libraries

```{r load libraries, warning=FALSE, message=FALSE}
library(tidyverse)
library(haven)
library(plm)
library(modelsummary)
library(lmtest)
library(gridExtra)
library(grid)
library(gridtext)
library(stargazer)
library(webshot)
library(tibble)
library(dplyr)
library(gt)
library(scales)
library(cowplot)
library(sandwich)
library(svglite)
```


```{r}
#reset environment
rm(list = ls())

```

# Read the Dataset
```{r}
rm(list = ls())

WD_file = dirname(rstudioapi::getSourceEditorContext()$path)
setwd(WD_file)
getwd()
````

DiD
Data

```{r data DiD, message=FALSE, warning=FALSE}

#load data
df <- haven::read_dta('BDH-Bismarck-JEEA-spec1.dta')

#clean data up, rename columns more intuitively
df <- df %>%
  rename(bluecollar = industry,
         district_id = code,
         district_name = RB,
         deaths_tot = dbo_tot,
         deaths_am = dbo_am,
         deaths_af = dbo_af,
         deaths_km = dbo_km,
         deaths_kf = dbo_kf,
         deaths_pc = dbo_tot_pc,
         deaths_am_pc = dbo_am_pc,
         deaths_af_pc = dbo_af_pc,
         deaths_km_pc = dbo_km_pc,
         deaths_kf_pc = dbo_kf_pc,
```
```

```
        urb_rate = i_urb_pc)

#create panel data set
df_spec1 <- pdata.frame(df, index = c("id_occ", "id_year"))

#delete unneeded data sets
rm("df")

```

Parallel trends
```{r parallel trends, message=FALSE, warning=FALSE}

  #creating df with data for trend plot
  df_trends <- aggregate(deaths_pc ~ year + bluecollar,
                         data = df_spec1,
                         FUN = mean)

  #error message w/o this - unclear why it is necessary
  df_trends$year <- as.numeric(df_trends$year)

  #create new rows for counterfactual, assign number 2
  df_trends <- rbind(df_trends,
                     data.frame(year = seq(1877,1900),
                                bluecollar = 2,
                                deaths_pc = 0))

  ##assign the correct values:
  #up to and including 1884 values are equal to bluecollar
  df_trends <- within(df_trends,
    deaths_pc[bluecollar == 2 & year <= 1884] <-
      deaths_pc[bluecollar == 1 & year <= 1884])

  #from 1885 on the values are that of public servant +
  #the difference between bluecollar and public servant in 1884

  #calculate difference between mean death rate in 1884
  diff_1884 = with(df_trends,
                   deaths_pc[year == 1884 & bluecollar == 1] -
                     deaths_pc[year == 1884 & bluecollar == 0])

  #assign the values for post 1884
  df_trends <- within(df_trends,
                      deaths_pc[bluecollar == 2 & year >= 1885] <-
                        deaths_pc[bluecollar == 0 & year >= 1885] +
                        diff_1884)

  #necessary for plotting
  df_trends$bluecollar <- as.factor(df_trends$bluecollar)

  #plotting
  plot_parallel_trends <- ggplot(df_trends,
         aes(x = year,
             y = deaths_pc,
             color = bluecollar)) +
    geom_line(size = 1) +
    #draw blue collar line on top of counterfactual line
    geom_line(data = subset(df_trends,
                            bluecollar == 1),
                            color = "blue",
                            size = 1) +
    labs(x = "Time",
         y = "Mean Crude Death Rate") +
         #title = "Development of Mean Crude Death Rate by Sector") +
    scale_color_manual(values = c("red",
                                  "blue",
                                  "grey"),
                       labels = c("Public Servant",
                                  "Blue Collar",
```

```
                                    "Blue Collar - Counterfactual")) +
    geom_vline(xintercept = 1884,
               linetype = "dashed",
               color = "black") + #add the vertical line in 1884
    #theme_minimal() + #no gray background
    theme(plot.title = element_text(hjust = 0.5),
          plot.background = element_rect(fill = "white"),
          panel.background = element_rect(fill = "white"),
          panel.grid = element_line("#E0E0E0"),
          legend.position = "bottom",
          legend.title = element_blank(),
          legend.key = element_rect(fill = "transparent")) +
    guides(linetype = "none") #deleting legend for linetypes

  ggsave("plot_parallel_trends.png",
         plot_parallel_trends,
         height = 9,
         width = 16,
         units = "cm",
         dpi = 1000)

  #delete unneeded df & variable
  rm(list = c("diff_1884", "df_trends"))
```

Baseline model
```{r DiD baseline model, message=FALSE, warning=FALSE}

  #to use 1884 as baseline year, setting year to 0 if year is 1884
  df_spec1$year_normalized <- ifelse(df_spec1$year == 1884,
                                     0,
                                     df_spec1$year)

  #run regression
  model_did_baseline <- plm(deaths_pc ~ factor(year_normalized) * bluecollar,
                            data = df_spec1,
                            model = "within")

  #extract coefficient estimates
  coefficients <- coef(model_did_baseline)

  #clustered standard errors
  clustered_sd <- coeftest(model_did_baseline,
                           vcov=vcovHC(model_did_baseline,
                                       type="sss",
                                       cluster="group"))

  # create data frame with values and names of coefficients
  df_coef <- data.frame(coefficient_name = c(1877:1883, 1885:1900),
                        coefficient = coefficients[-(1:(length(coefficients)-23))],
                        standard_error = clustered_sd[-(1:
(length(clustered_sd[,2])-23)),2])

  # insert the row between 1883 and 1885
  df_coef <- rbind(df_coef[1:7,],
                   data.frame(coefficient_name = 1884,
                              coefficient = 0,
                              standard_error = 0),
                   df_coef[8:nrow(df_coef),])


  plot_baseline_model <- ggplot(data = df_coef,
         aes(x = coefficient_name,
             y = coefficient)) +
    theme(plot.title = element_text(hjust = 0.5),
      plot.background = element_rect(fill = "white"),
      panel.background = element_rect(fill = "white"),
      panel.grid = element_line("#E0E0E0"),
      legend.position = "bottom",
```

```
      legend.title = element_blank(),
      legend.key = element_rect(fill = "transparent")) +
    geom_hline(yintercept = 0,
               linetype = "dashed",
               color = "red") +
    geom_vline(xintercept = 1884,
               linetype = "dashed",
               color = "red") +
    scale_x_continuous(
      breaks = c(1880, 1884, 1890, 1900),
      labels = c("1880", "1884", "1890", "1900")) +
    scale_y_continuous(
      breaks = c(2, 0, -2, -4),
      labels = c(2, 0, -2, -4)) +
    ylim(-4, 2) +
    labs(x = NULL,
         y = "Coefficient Estimate") +
         #title = "Main Results: Baseline Model") +
    #95% conf. interval
    geom_errorbar(
      aes(ymin = coefficient - 1.96 * standard_error,
          ymax = coefficient + 1.96 * standard_error),
      width = 0.2,
      color = "blue") +
    geom_point() #add points in the end such that they are on

  ggsave("plot_baseline_model.png",
         plot_baseline_model,
         width = 16,
         height = 9,
         units = "cm")

  #remove unneeded objects
  rm(list=c("df_coef",
            "coefficients",
            "clustered_sd",
            "model_did_baseline"))
```

Heterogeneity age & gender

There might be heterogeneous effects for children or women (Figure 5 in paper)

```{r heterogeneity age & gender, warning=FALSE, message=FALSE}

#list of dependent variables for the regressions
formulas <- c("deaths_am_pc ~ factor(year_normalized) * bluecollar",
              "deaths_af_pc ~ factor(year_normalized) * bluecollar",
              "deaths_km_pc ~ factor(year_normalized) * bluecollar",
              "deaths_kf_pc ~ factor(year_normalized) * bluecollar")

#list of titles for the respective plots
plot_titles <- c("Adult Males",
                 "Adult Females",
                 "Boys < 14",
                 "Girls < 14")

#create empty list to store plots
plots <- list()

#loop through the four variables, run regression and create plot for each
for (i in 1:4) {

  #run the regression using the defined reg. formula
  model <- plm(formula = formulas[[i]],
               data = df_spec1,
               model = "within")

  #extract coefficient estimates
```

```
  coefficients <- coef(model)

  #get clustered standard errors
  clustered_sd <- coeftest(model,
                           vcov=vcovHC(model,
                                       type="sss",
                                       cluster="group"))

  #create data frame with values and names of coefficients
  df_coef <- data.frame(coefficient_name = c(1877:1883, 1885:1900),
                        #keep only last 23 entries (relevant estimates of interaction
term)
                        coefficient = coefficients[-(1:(length(coefficients)-23))],
                        #keep only the last 23 values of the second column
                        standard_error = clustered_sd[-(1:
(length(clustered_sd[,2])-23)),2])

  # insert the row between 1883 and 1885
  df_coef <- rbind(df_coef[1:7,],
                   data.frame(coefficient_name = 1884,
                              coefficient = 0,
                              standard_error = 0),
                   df_coef[8:nrow(df_coef),])

  #create plot
  plot <- ggplot(data = df_coef,
                 aes(x = coefficient_name,
                     y = coefficient)) +
    theme_minimal() +
    geom_hline(yintercept = 0,
               linetype = "dashed",
               color = "red",
               size = 0.5) +
    geom_vline(xintercept = 1884,
               linetype = "dashed",
               color = "red",
               size = 0.5) +
    scale_x_continuous(
      breaks = c(1880, 1884, 1890, 1900),
      labels = c("1880", "1884", "1890", "1900")) +
    scale_y_continuous(
      breaks = c(2, 0, -2, -4),
      labels = c(2, 0, -2, -4)) +
    ylim(-4, 2) +
    theme(plot.title = element_text(hjust = 0.5,
                                    size = 10)) +
    labs(x = NULL,
         y = NULL,
         title = plot_titles[i]) +
    #95% conf. interval
    geom_errorbar(
      aes(ymin = coefficient - 1.96 * standard_error,
          ymax = coefficient + 1.96 * standard_error),
      width = 0.25,
      size = 0.25,
      color = "blue") +
    geom_point(size=0.25)

  plots[[i]] <- plot

}

#arrange list of plots in one plot
plot_heterogeneity_gender_age <-
  grid.arrange(grobs = plots,
               nrow=2,
               ncol=2,
               left = textGrob("Coefficient Estimates",
                               rot = 90,
```

```
                                gp = gpar(fontsize = 12)))

#save combined plot to png file
ggsave("plot_heterogeneity_gender_age.png",
       plot_heterogeneity_gender_age,
       width = 16,
       height = 9,
       units = "cm",
       dpi = 1000)

#get rid of no longer needed objects
rm(list = c("model",
            "df_coef",
            "clustered_sd",
            "plot",
            "plot_name",
            "formula",
            "plot_titles",
            "dep_variables",
            "coefficients",
            "plots",
            "i",
            "formulas"))
```

Heterogeneity industries

The effects might be confounded by wage effects in the industries. A consistent effect across all industries would serve as evidence against this caveat.

```{r heterogeneity industries, warning=FALSE, message=FALSE}

#set public servants to 0 for them to be the basegroup
df_spec1$occ_normalized <- ifelse(df_spec1$occ == "14",
                                  0,
                                  df_spec1$occ)

#run regression similar to baseline model, but with occupations instead of bluecollar
model_did_heterogeneity_industries <-
  plm(deaths_pc ~ factor(year_normalized) * factor(occ_normalized),
      data = df_spec1,
      model = "within")

#extract coefficient estimates
coefficients <- coef(model_did_heterogeneity_industries)

#compute clustered standard errors
clustered_sd <- coeftest(model_did_heterogeneity_industries,
                         vcov=vcovHC(model_did_heterogeneity_industries,
                                     type="sss",
                                     cluster="group"))

#names of industries
industries = c("Mining & turf", "Minerals", "Metals", "Machinery", "Chemicals", "Fossil
fuels", "Textiles", "Paper & leather", "Wood", "Food", "Appareal & cleaning",
"Construction", "Printing")

#share of workers in respective occupations
shares = c("12.3%", "5.2%", "9.0%", "4.6%", "0.9%", "0.6%", "9.5%", "3.0%", "8.3%",
"10.6%", "17.6%", "17.4%", "0.8%")

#create empty list to store plots
plots <- list()

#loop through all 13 occupations and create a plot for each
for (i in 1:13) {
```

```
  #calculate where the relevant coefficients and std. errors are in the vectors
  start = i*23+1
  end = i*23+23

  #crate data frame with coefficients
  df <- data.frame(coefficient_name = c(1877:1883, 1885:1900),
                   coefficient = coefficients[start:end],
                   standard_error = clustered_sd[start:end,2])

  #insert a row between 1883 and 1885 (baseline year 1884) w/ values of 0
  df <- rbind(df[1:7,],
              data.frame(coefficient_name = 1884,
                         coefficient = 0,
                         standard_error = 0),
              df[8:nrow(df),])

  #create plot
  plot <- ggplot(data = df,
                 aes(x = coefficient_name,
                     y = coefficient)) +
    theme_minimal() +
    theme(plot.title = element_text(size = 10, hjust = 0.5)) +
    geom_hline(yintercept = 0,
               linetype = "dashed",
               color = "red",
               size = 0.25) +
    geom_vline(xintercept = 1884,
               linetype = "dashed",
               color = "red",
               size = 0.25) +
    ylim(-11, 7) +
    labs(x=NULL,
         y=NULL,
         title = paste(industries[i], "\n(", shares[i], ")", sep="")) +
    scale_x_continuous(
      breaks = c(1879, 1884, 1889, 1894, 1899),
      labels = c("'79", "'84", "'89", "'94", "'99")) +
    #add conf. intervals after lines s.t. they're on top
    geom_errorbar(
      aes(ymin = coefficient - 1.96 * standard_error,
          ymax = coefficient + 1.96 * standard_error),
      width = 0,
      size = 0.25,
      color = "blue") +
    #add points in the end s.t. they're on top
    geom_point(size=0.25)

  plots[[i]] <- plot
}

#exclude industries with shares < 1%
excluded_industries <- c(5, 6, 13)
plots <- plots[-excluded_industries]

#add all created plots to the grid.arrange object
plot_heterogeneity_industries <-
  grid.arrange(grobs = plots,
               #arrange them in a 2x5 grid
               nrow = 2,
               ncol = 5,
               #add a y-axis
               left = textGrob("Coefficient Estimates",
                               rot = 90,
                               gp = gpar(fontsize = 12)),
               #set fixed sizes of each plot to prevent stretched plots
               widths = unit(c(4,4,4,4,4), "cm"),
               heights = unit(c(4,4), "cm"))

#save plot
```

```
ggsave("plot_heterogeneity_industries.png",
       plot_heterogeneity_industries,
       width = 22,
       height = 10,
       units = "cm",
       dpi = 1000)

#get rid of no longer needed objects
rm(list = c("df",
            "clustered_sd",
            "coefficients",
            "plot",
            "plots",
            "end",
            "start",
            "excluded_industries",
            "i",
            "industries",
            "shares",
            "model_did_heterogeneity_industries"))
```

County Fixed Effects

Main issue with DiD is that selection into occupation groups after 1884 cannot be excluded.
Both selection that would lead to upward bias (sicker, more care needing people select into bluecollar occupations to get insurance) as well as selection leading to downward bias (younger people select into bluecollar occupations as the job becomes more attractive due to insurance) are possible.
Thus, need different approach such that one can fix treatment assignment/intensity pre treatment: county fixed effects.

Data

```{r data fixed effects}
#load data
#pick path for spec2 data
df <- haven::read_dta('BDH-Bismarck-JEEA-spec2.dta')

#clean data up, rename columns more intuitively
df <- df %>%
  rename(period1875 = yr1,
         period1880 = yr2,
         period1885 = yr3,
         period1890 = yr4,
         period1895 = yr5,
         period1900 = yr6,
         treat1875 = b_industry_w_pc1875,
         treat1880 = b_industry_w_pc1880,
         treat1885 = b_industry_w_pc1885,
         treat1890 = b_industry_w_pc1890,
         treat1895 = b_industry_w_pc1895,
         treat1900 = b_industry_w_pc1900,
         deaths_pc = dth_pc,
         county = code1867,
         bluecollar_1882 = b_industry_w_pc,
         deaths_male = dthm_pc,
         deaths_female = dthf_pc,
         deaths_infants_tot = dth1_pb,
         deaths_infants_leg = dthleg_pb,
         deaths_infants_illeg = dthbas_pb,
         bluecollar_1882_self = b_industry_self_pc,
         public_1882 = e_public_w_pc)

#create panal data set
df_spec2 <- pdata.frame(df, index = c("county", "year"))
```

```
#delete unneeded data frame
rm("df")
```


Model
```{r warning=FALSE, message=FALSE}
#create new year variable which is 0 whenever year = 1880 as this is the
#omitted reference period
df_spec2$year_normalized <-
  ifelse(df_spec2$year == 1880,
         0,
         df_spec2$year)

#create list with specs of regressions
formulas <- list("deaths_pc ~ bluecollar_1882 * factor(year_normalized)",
                 "deaths_pc ~ bluecollar_1882*factor(year_normalized) + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_male ~ bluecollar_1882*factor(year_normalized) + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_female ~ bluecollar_1882*factor(year_normalized) + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_infants_tot ~ bluecollar_1882*factor(year_normalized) + urb_pc
+ waterwork_pc + sewage_pc",
                 "deaths_infants_leg ~ bluecollar_1882*factor(year_normalized) + urb_pc
+ waterwork_pc + sewage_pc",
                 "deaths_infants_illeg ~ bluecollar_1882*factor(year_normalized) +
urb_pc + waterwork_pc + sewage_pc",
                 "deaths_pc ~ bluecollar_1882_self * factor(year_normalized) + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_pc ~ public_1882 * factor(year_normalized) + urb_pc +
waterwork_pc + sewage_pc")

#list of names for models
model_names <- c("Baseline",
                 "Controls",
                 "Males",
                 "Females",
                 "Infants",
                 "Legitimate Infants",
                 "Illegitimate Infants",
                 "Self-Employed",
                 "Public Servants")

#create empty list to store models
models <- list()

#loop through all 9 specs
for (i in 1:9){
  #run regression
  models[[i]] <- plm(formula = formulas[[i]],
                data = df_spec2,
                model = "within")
  #calculate clustered SE
  models[[i]]$vcov = vcovHC(models[[i]],
                            type = "sss",
                            cluster = "group")
}

names(models) <- model_names

#create new rows which need to be added to regression table
rows <- tribble(~term, ~`[[1]]`, ~`[[2]]`, ~`[[3]]`, ~`[[4]]`, ~`[[5]]`, ~`[[6]]`,
~`[[7]]`, ~`[[8]]`, ~`[[9]]`,
                'Controls', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Yes',
                'County FE', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Yes',
```

```
                'Time FE', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'Yes',
                'Counties', '441', '441', '441', '441', '441', '441', '441', '441',
'441',
                'Periods', '6', '6', '6', '6', '6', '6', '6', '6', '6')

attr(rows, 'position') <- c(11, 12, 13, 15, 16)


# create regression table
table_fe_results <-
  modelsummary(models,
               #name all three different sets of coefficients the same such
               #that the table consists of only five rows of coefficients
               coef_map =
                 c('bluecollar_1882:factor(year_normalized)1' = 'Treat × 1875',
                   'bluecollar_1882:factor(year_normalized)3' = 'Treat × 1885',
                   'bluecollar_1882:factor(year_normalized)4' = 'Treat × 1890',
                   'bluecollar_1882:factor(year_normalized)5' = 'Treat × 1895',
                   'bluecollar_1882:factor(year_normalized)6' = 'Treat × 1900',
                   'bluecollar_1882_self:factor(year_normalized)1' = 'Treat × 1875',
                   'bluecollar_1882_self:factor(year_normalized)3' = 'Treat × 1885',
                   'bluecollar_1882_self:factor(year_normalized)4' = 'Treat × 1890',
                   'bluecollar_1882_self:factor(year_normalized)5' = 'Treat × 1895',
                   'bluecollar_1882_self:factor(year_normalized)6' = 'Treat × 1900',
                   'public_1882:factor(year_normalized)1' = 'Treat × 1875',
                   'public_1882:factor(year_normalized)3' = 'Treat × 1885',
                   'public_1882:factor(year_normalized)4' = 'Treat × 1890',
                   'public_1882:factor(year_normalized)5' = 'Treat × 1895',
                   'public_1882:factor(year_normalized)6' = 'Treat × 1900'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               #add the previously defined additional rows
               add_rows = rows,
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
               output = "gt")

table_fe_results <- table_fe_results %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Heterogeneity Tests', columns = 4:8) %>%
  tab_spanner(label = 'Placebo Tests', columns = 9:10)

gtsave(table_fe_results, file = "table_fe_results.docx")

#remove unused objects
rm(list = c("rows",
            "formulas",
            "i",
            "models",
            "model_names"))
```

Robustness checks
Data

```{r data robustness checks}
#load data
#pick path for spec3 data
df <- haven::read_dta('BDH-Bismarck-JEEA-spec3.dta')

#clean data up, rename columns more intuitively
df <- df %>%
  rename(district_id = code,
         deaths_pc = dth_pc,
         insured_1885 = hi_initial,
         bluecollar_1882 = b_industry_w_pc,
         share_age_1_9 = sh_birthage1_9,
```

```
         share_age_10_19 = sh_birthage10_19,
         share_age_20_29 = sh_birthage20_29,
         share_age_30_39 = sh_birthage30_39,
         share_age_40_49 = sh_birthage40_49,
         share_age_50_59 = sh_birthage50_59,
         share_age_60_69 = sh_birthage60_69,
         share_age_70_plus = sh_birthage70_plus,
         share_spd = sh_spd)

#create panel data set
df_spec3 <- pdata.frame(df, index = c("district_id", "year"))

#remove unneeded data set
rm("df")
```

Checks

```{r robustness checks, warning=FALSE, message=FALSE}
#controls: urb_pc waterwork_pc sewage_pc
#baseline period: 1880
#treatment variables: treat1875 treat1885 treat1890 treat1895 treat1900

#can't set baseline year as in stata --> need to set all 1880 periods to 0
df_spec3$year_normalized <-
  ifelse(df_spec3$year == 1880,
         0,
         df_spec3$year)

#create list with specs of regressions
formulas <- list("deaths_pc ~ factor(year_normalized) * bluecollar_1882 + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_pc ~ factor(year_normalized) * bluecollar_1882 + urb_pc +
waterwork_pc + sewage_pc + avg_age",
                 "deaths_pc ~ factor(year_normalized) * bluecollar_1882 + urb_pc +
waterwork_pc + sewage_pc + share_age_1_9 + share_age_10_19 + share_age_20_29 +
share_age_30_39 + share_age_40_49 + share_age_50_59 + share_age_60_69",
                 "deaths_pc ~ factor(year_normalized) * bluecollar_1882 + urb_pc +
waterwork_pc + sewage_pc + share_spd",
                 "deaths_pc ~ factor(year_normalized) * insured_1885 + urb_pc +
waterwork_pc + sewage_pc",
                 "deaths_pc ~ factor(year_normalized) * insured_1885 + urb_pc +
waterwork_pc + sewage_pc + avg_age",
                 "deaths_pc ~ factor(year_normalized) * insured_1885 + urb_pc +
waterwork_pc + sewage_pc + share_age_1_9 + share_age_10_19 + share_age_20_29 +
share_age_30_39 + share_age_40_49 + share_age_50_59 + share_age_60_69",
                 "deaths_pc ~ factor(year_normalized) * insured_1885 + urb_pc +
waterwork_pc + sewage_pc + share_spd")

#list of names for models
model_names <- c("Controls",
                 "Average age",
                 "Age groups",
                 "SPD vote",
                 "Controls",
                 "Average age",
                 "Age groups",
                 "SPD vote")

#create list to store models in
models <- list()

for (i in 1:8){
    #run regression
    models[[i]] <- plm(formula = formulas[[i]],
                       data = df_spec3,
                       model = "within")
    #calculate SE
    models[[i]]$vcov <- vcovHC(models[[i]],
```

```
                                type = "sss",
                                cluster = "group")
}

#assign names to models
names(models) = model_names


#create new rows which need to be added to regression table
rows <- tribble(~term, ~`[[1]]`, ~`[[2]]`, ~`[[3]]`, ~`[[4]]`, ~`[[5]]`, ~`[[6]]`,
~`[[7]]`, ~`[[8]]`,
                'Average age', 'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No',
                'Age groups', 'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No',
                'SPD vote', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes',
                'Districts', '36', '36', '36', '36', '36', '36', '36', '36',
                'Periods', '6', '6', '6', '6', '6', '6', '6', '6')

attr(rows, 'position') <- c(11, 12, 13, 15, 16)

table_fe_robust <-
  modelsummary(models,
               #name all three different sets of coefficients the same such
               #that the table consists of only five rows of coefficients
               coef_map = c('factor(year_normalized)1:bluecollar_1882' = 'Treat × 1875',
                            'factor(year_normalized)3:bluecollar_1882' = 'Treat × 1885',
                            'factor(year_normalized)4:bluecollar_1882' = 'Treat × 1890',
                            'factor(year_normalized)5:bluecollar_1882' = 'Treat × 1895',
                            'factor(year_normalized)6:bluecollar_1882' = 'Treat × 1900',
                            'factor(year_normalized)1:insured_1885' = 'Treat × 1875',
                            'factor(year_normalized)3:insured_1885' = 'Treat × 1885',
                            'factor(year_normalized)4:insured_1885' = 'Treat × 1890',
                            'factor(year_normalized)5:insured_1885' = 'Treat × 1895',
                            'factor(year_normalized)6:insured_1885' = 'Treat × 1900'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               #add the previously defined additional rows
               add_rows = rows,
               #add some notes
               notes = "Included in (1)-(8): Standard controls, district FE, time FE",
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
               output = "gt")

table_fe_robust <- table_fe_robust %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Initial blue collar workers (1882)', columns = 2:5) %>%
  tab_spanner(label = 'Initial insured (1885)', columns = 6:9)

gtsave(table_fe_robust, file = "table_fe_robust.docx")

rm(list = c("formulas",
            "models",
            "rows",
            "i",
            "model_names"))

```

Channels

Causes of death

```{r}

#continue to use spec3 data with normalized year

#create list with specs for regressions
formulas <- list("cod_accident_pc ~ factor(year_normalized) * bluecollar_1882 + urb_pc +
```

```
waterwork_pc + sewage_pc",
                  "cod_water_pc ~ factor(year_normalized) *  bluecollar_1882 + urb_pc +
waterwork_pc + sewage_pc",
                  "cod_air_pc ~ factor(year_normalized) * bluecollar_1882  + urb_pc +
waterwork_pc + sewage_pc",
                  "cod_lung_pc ~ factor(year_normalized) * bluecollar_1882  + urb_pc +
waterwork_pc + sewage_pc",
                  "cod_tuber_pc ~ factor(year_normalized) * bluecollar_1882  + urb_pc +
waterwork_pc + sewage_pc",
                  "cod_noninfec_pc ~ factor(year_normalized) * bluecollar_1882  + urb_pc
+ waterwork_pc + sewage_pc",
                  "cod_matern_pc ~ factor(year_normalized) * bluecollar_1882  + urb_pc +
waterwork_pc + sewage_pc",
                  "cod_other_unkn_dis_pc ~ factor(year_normalized) * bluecollar_1882  +
urb_pc + waterwork_pc + sewage_pc")

#list of names for models
model_names <- c("Accident",
                  "Waterborne",
                  "Airborne",
                  "Lung",
                  "TB+Scrofula",
                  "Noninfectious",
                  "Maternal",
                  "Unknown")

#create list to store models in
models <- list()

for (i in 1:8){
    #run regression
    models[[i]] <- plm(formula = formulas[[i]],
                       data = df_spec3,
                       model = "within")
    #calculate SE
    models[[i]]$vcov <- vcovHC(models[[i]],
                               type = "sss",
                               cluster = "group")
}

#assign names to models
names(models) = model_names

#create new rows which need to be added to regression table
rows <- tribble(~term, ~`[[1]]`, ~`[[2]]`, ~`[[3]]`, ~`[[4]]`, ~`[[5]]`, ~`[[6]]`,
~`[[7]]`, ~`[[8]]`,
                'Districts', '36', '36', '36', '36', '36', '36', '36', '36',
                'Periods', '6', '6', '6', '6', '6', '6', '6', '6')

attr(rows, 'position') <- c(12, 13, 1)

table_channels_cod <-
  modelsummary(models,
               #name all eight different sets of coefficients the same such
               #that the table consists of only five rows of coefficients
               coef_map = c('factor(year_normalized)1:bluecollar_1882' = 'Treat × 1875',
                            'factor(year_normalized)3:bluecollar_1882' = 'Treat × 1885',
                            'factor(year_normalized)4:bluecollar_1882' = 'Treat × 1890',
                            'factor(year_normalized)5:bluecollar_1882' = 'Treat × 1895',
                            'factor(year_normalized)6:bluecollar_1882' = 'Treat ×
1900'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               #add the previously defined additional rows
               add_rows = rows,
               #add some notes
               notes = "Included in (1)-(8): Standard controls, district FE, time FE",
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
```

```
               output = "gt")

table_channels_cod <- table_channels_cod %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Infectious diseases', columns = 3:6)

gtsave(table_channels_cod, file = "table_channels_cod.docx")

#delete unneeded objects
rm(list = c("formulas",
            "models",
            "rows",
            "i",
            "model_names"))

```

Supply of health

```{r}

#two regressions are run with the spec2 data, two with spec3

#create list with specs for regressions
formulas <- list("deaths_pc ~ factor(year_normalized) * bluecollar_1882 * doc_pc +
urb_pc + waterwork_pc + sewage_pc",
                 "deaths_pc ~ factor(year_normalized) * bluecollar_1882 * doc1882_pc +
urb_pc + waterwork_pc + sewage_pc",
                 "deaths_pc ~ factor(year_normalized) *  bluecollar_1882 * ln_distuni +
urb_pc + waterwork_pc + sewage_pc",
                 "deaths_pc ~ factor(year_normalized) * bluecollar_1882 * uni + urb_pc +
waterwork_pc + sewage_pc")

#list of names for models
model_names <- list("Medical professionals 1882",
                    "Approbated doctors 1882",
                    "Distance to university",
                    "University dummy")

#list with respective dataframes
dataframes <- list(df_spec2,
                   df_spec3,
                   df_spec2,
                   df_spec3)

#create list to store models in
models <- list()

for (i in 1:4){
    #run regression
    models[[i]] <- plm(formula = formulas[[i]],
                       data = dataframes[[i]],
                       model = "within")
    #calculate SE
    models[[i]]$vcov <- vcovHC(models[[i]],
                               type = "sss",
                               cluster = "group")
}

#assign names to models
names(models) = model_names

#create new rows which need to be added to regression table
rows <- tribble(
  ~term, ~`[[1]]`, ~`[[2]]`, ~`[[3]]`, ~`[[4]]`,
  'Counties/districts', '441', '36', '441', '36',
  'Periods', '6', '6', '6', '6'
)
```

```
attr(rows, 'position') <- c(22, 23)

table_channels_supply <-
  modelsummary(models,
               #name all eight different sets of coefficients the same such
               #that the table consists of only five rows of coefficients
               coef_map = c('factor(year_normalized)1:bluecollar_1882' = 'Treat × 1875',
                            'factor(year_normalized)3:bluecollar_1882' = 'Treat × 1885',
                            'factor(year_normalized)4:bluecollar_1882' = 'Treat × 1890',
                            'factor(year_normalized)5:bluecollar_1882' = 'Treat × 1895',
                            'factor(year_normalized)6:bluecollar_1882' = 'Treat × 1900',
                            'factor(year_normalized)1:bluecollar_1882:doc_pc' = 'Treat ×
Health supply × 1875',
                            'factor(year_normalized)3:bluecollar_1882:doc_pc' = 'Treat ×
Health supply × 1885',
                            'factor(year_normalized)4:bluecollar_1882:doc_pc' = 'Treat ×
Health supply × 1890',
                            'factor(year_normalized)5:bluecollar_1882:doc_pc' = 'Treat ×
Health supply × 1895',
                            'factor(year_normalized)6:bluecollar_1882:doc_pc' = 'Treat ×
Health supply × 1900',
                            'factor(year_normalized)1:bluecollar_1882:ln_distuni' =
'Treat × Health supply × 1875',
                            'factor(year_normalized)3:bluecollar_1882:ln_distuni' =
'Treat × Health supply × 1885',
                            'factor(year_normalized)4:bluecollar_1882:ln_distuni' =
'Treat × Health supply × 1890',
                            'factor(year_normalized)5:bluecollar_1882:ln_distuni' =
'Treat × Health supply × 1895',
                            'factor(year_normalized)6:bluecollar_1882:ln_distuni' =
'Treat × Health supply × 1900',
                            'factor(year_normalized)1:bluecollar_1882:doc1882_pc' =
'Treat × Health supply × 1875',
                            'factor(year_normalized)3:bluecollar_1882:doc1882_pc' =
'Treat × Health supply × 1885',
                            'factor(year_normalized)4:bluecollar_1882:doc1882_pc' =
'Treat × Health supply × 1890',
                            'factor(year_normalized)5:bluecollar_1882:doc1882_pc' =
'Treat × Health supply × 1895',
                            'factor(year_normalized)6:bluecollar_1882:doc1882_pc' =
'Treat × Health supply × 1900',
                            'factor(year_normalized)1:bluecollar_1882:uni' = 'Treat ×
Health supply × 1875',
                            'factor(year_normalized)3:bluecollar_1882:uni' = 'Treat ×
Health supply × 1885',
                            'factor(year_normalized)4:bluecollar_1882:uni' = 'Treat ×
Health supply × 1890',
                            'factor(year_normalized)5:bluecollar_1882:uni' = 'Treat ×
Health supply × 1895',
                            'factor(year_normalized)6:bluecollar_1882:uni' = 'Treat ×
Health supply × 1900'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               #add the previously defined additional rows
               add_rows = rows,
               #add some notes
               notes = "Included in (1)-(4): Standard controls, county/district FE, time
FE",
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
               output = "gt")

table_channels_supply <- table_channels_supply %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Supply of health services', columns = 2:3) %>%
  tab_spanner(label = 'Supply of health knowledge', columns = 4:5)
```

```
gtsave(table_channels_supply, file= "table_channels_supply.docx")

rm(list = c("dataframes",
            "formulas",
            "model_names",
            "models",
            "rows",
            "i"))

```

Health expenditures

```{r}

#load data
#pick path for spec4 data
df <- haven::read_dta('BDH-Bismarck-JEEA-spec4.dta')

#render the df a panel data frame
df_spec4 <- pdata.frame(df,
                        index = c("code",
                                  "year"))

##### Panel A #####

#define the lagged variables
df_spec4$i_urb_pc_lag <- lag(df_spec4$i_urb_pc, n = 1)
df_spec4$waterwork_pc_lag <- lag(df_spec4$waterwork_pc, n = 1)
df_spec4$sewage_pc_lag <- lag(df_spec4$sewage_pc, n = 1)
df_spec4$ins_all_exp_doc_pi_std_lag <- lag(df_spec4$ins_all_exp_doc_pi_std, n = 1)
df_spec4$ins_all_exp_med_pi_std_lag <- lag(df_spec4$ins_all_exp_med_pi_std, n = 1)
df_spec4$ins_all_exp_hos_pi_std_lag <- lag(df_spec4$ins_all_exp_hos_pi_std, n = 1)
df_spec4$ins_all_exp_pay_pi_std_lag <- lag(df_spec4$ins_all_exp_pay_pi_std, n = 1)
df_spec4$ins_all_exp_mat_pi_std_lag <- lag(df_spec4$ins_all_exp_mat_pi_std, n = 1)
df_spec4$ins_all_exp_dth_pi_std_lag <- lag(df_spec4$ins_all_exp_dth_pi_std, n = 1)
df_spec4$ins_all_exp_adm_pi_std_lag <- lag(df_spec4$ins_all_exp_adm_pi_std, n = 1)
df_spec4$sick_days_pi_std_lag <- lag(df_spec4$sick_days_pi_std, n = 1)


#set up list with all the regression specifications
formulas <-
  list("dth_pc_std ~ ins_all_exp_doc_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       "dth_pc_std ~ ins_all_exp_med_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       "dth_pc_std ~ ins_all_exp_hos_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       #"dth_pc_std ~ ins_all_exp_mpay_pi_std + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag", #not included
       "dth_pc_std ~ ins_all_exp_pay_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       "dth_pc_std ~ ins_all_exp_mat_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       "dth_pc_std ~ ins_all_exp_dth_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       "dth_pc_std ~ ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag",
       #"dth_pc_std ~ ins_all_exp_care_pi_std + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag", #not included
       #"dth_pc_std ~ ins_all_exp_comp_pi_std + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag", #not included
       "dth_pc_std ~ sick_days_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag +
sewage_pc_lag")

#create list to store models in
models <- list()

#define names for models
```

```
model_names <- c("Doctor visits",
                 "Medication",
                 "Hospitalization",
                 "Sick pay",
                 "Maternity benefits",
                 "Death benefits",
                 "Administration expenditures",
                 "Days of sick leave")

#loop through the eight regression specs
for (i in 1:8){
    #run regression
    models[[i]] <- plm(formula = formulas[[i]],
                       data = df_spec4,
                       model = "within",
                       effect = "twoways")
    #calculate SE
    models[[i]]$vcov <- vcovHC(models[[i]],
                               type = "sss",
                               cluster = "group")
}

names(models) = model_names

table_expenditures_A <-
  modelsummary(models,
               #naming all coefficients identically such that they all show
               #up in the same row
               coef_map = c('ins_all_exp_doc_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_med_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_hos_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_pay_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_mat_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_dth_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'ins_all_exp_adm_pi_std_lag' = 'Lagged Expenditure (Std.)',
                            'sick_days_pi_std_lag' = 'Lagged Expenditure (Std.)'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               notes = "Included in (1)-(8): Standard controls, district FE, time FE",
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
               output = "gt")

table_expenditures_A <- table_expenditures_A %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Health care expenditures', columns = 2:4) %>%
  tab_spanner(label = 'Compensation expenditures', columns = 5:7)

gtsave(table_expenditures_A, file= "table_expenditures_A.docx")


##### Panel B #####

#define the other lagged variables
df_spec4$ins_all_exp_care_pi_std_lag <- lag(df_spec4$ins_all_exp_care_pi_std, n = 1)
df_spec4$ins_all_exp_comp_pi_std_lag <- lag(df_spec4$ins_all_exp_comp_pi_std, n = 1)
df_spec4$ins_all_exp_adm_pi_std_lag <- lag(df_spec4$ins_all_exp_adm_pi_std, n = 1)

#set up list with all the regression specifications
formulas <-
  list("cod_tot_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_accident_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag
+ ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_water_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_air_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
```

```
       "cod_lung_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_tuber_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_noninfec_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag
+ ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag",
       "cod_matern_pc_std ~ ins_all_exp_care_pi_std_lag + ins_all_exp_comp_pi_std_lag +
ins_all_exp_adm_pi_std_lag + i_urb_pc_lag + waterwork_pc_lag + sewage_pc_lag")

#create list to store models in
models <- list()

#define names for models
model_names <- c("All",
                 "Accident",
                 "Waterborne",
                 "Airborne",
                 "Lung",
                 "TB + Scrufola",
                 "Noninfectious",
                 "Maternal")

#loop through the eight regression specs
for (i in 1:8){
    #run regression
    models[[i]] <- plm(formula = formulas[[i]],
                       data = df_spec4,
                       model = "within",
                       effect = "twoways")
    #calculate SE
    models[[i]]$vcov <- vcovHC(models[[i]],
                               type = "sss",
                               cluster = "group")
}

names(models) = model_names

table_expenditures_B <-
  modelsummary(models,
               #naming all coefficients identically such that they all show
               #up in the same row
               coef_map = c('ins_all_exp_care_pi_std_lag' = 'Lagged Health Care Exp.
(Std.)',
                            'ins_all_exp_comp_pi_std_lag' = 'Lagged Compensation Exp.
(Std.)',
                            'ins_all_exp_adm_pi_std_lag' = 'Lagged Administration Exp.
(Std.)'),
               #add number of observations and r^2 statistics
               gof_map = c("nobs", "r.squared"),
               notes = "Included in (1)-(8): Standard controls, district FE, time FE",
               #have the regression table being put out in the gt format s.t.
               #more adjustments can be made to it
               output = "gt")

table_expenditures_B <- table_expenditures_B %>%

  #assign column labels and span them over multiple columns
  tab_spanner(label = 'Infectious diseases', columns = 3:6)

gtsave(table_expenditures_B, file= "table_expenditures_B.docx")

#remove unneeded objects
rm(list = c("df",
            "formulas",
            "models",
            "i",
            "model_names"))
```

```
```