Research Assistantship - Data Collection

Federico Vicentini

01/05/2023

DATA COLLECTION

Real GDP Data

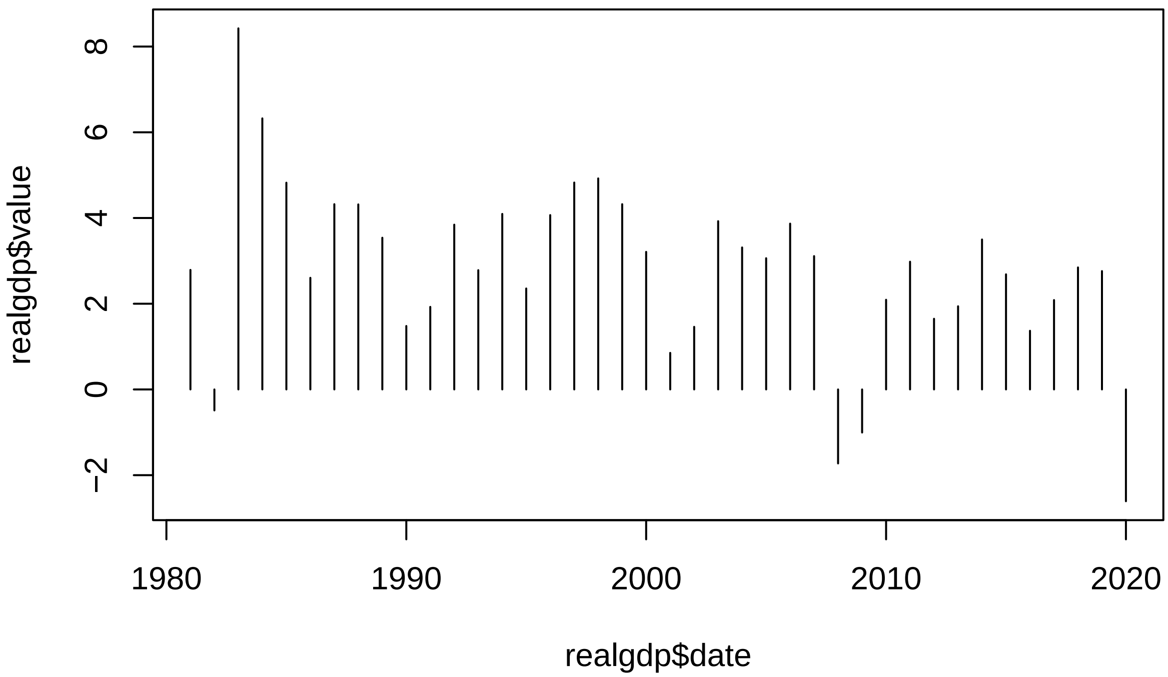
In this first part, we download nominal GDP data and GDP deflator data from 1980 to 2020 and then we divide it into the 2 section (1980s and 2010s) in order to calculate means and thus compare it to the data we got in the TTD Presentation.

```
## [1] 2.816635
```

```
## [1] 3.813955
```

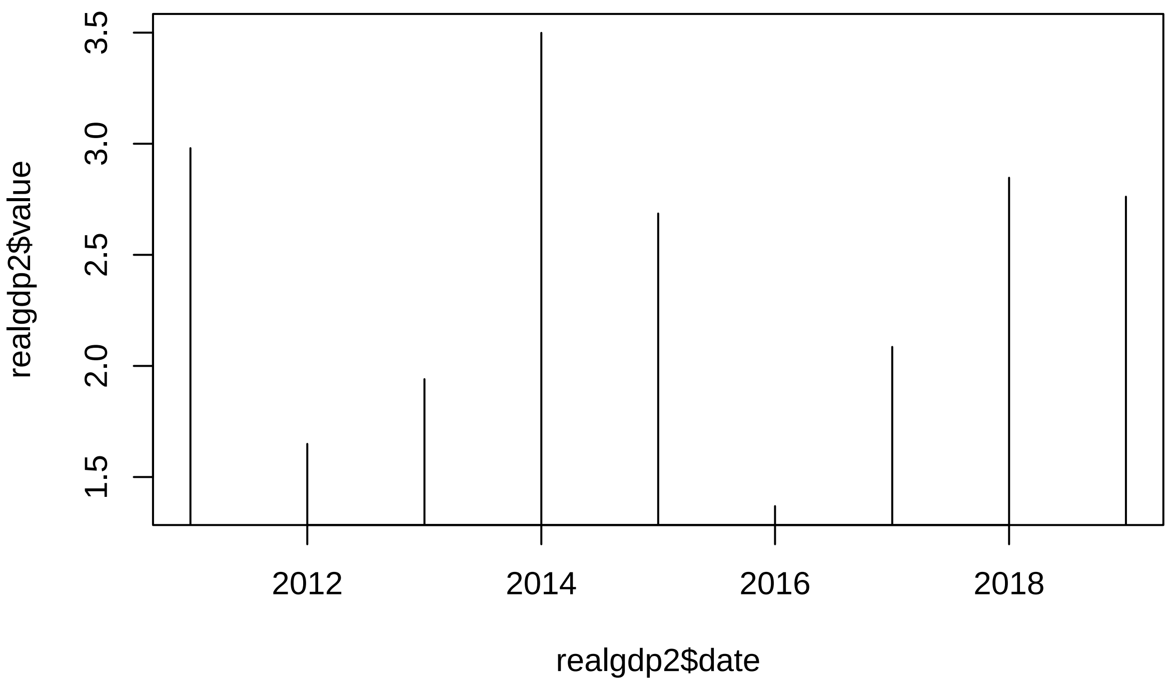
```
## [1] 2.424288
```

```
plot(realgdp$date, realgdp$value, type="h")
```




```
plot(realgdp1$date, realgdp1$value, type="h")
```


```
plot(realgdp2$date, realgdp2$value, type="h")
```



Labor Share

First attmpt here is to download the laborshare timeseries from fred. Howevere, this approach neglects some important aspects, such as the role of self-employed workers and correction to value added in the forms of indirect taxes and consumption of fixed capital.

```
labshare = fredr(
  series_id = "LABSHPUSA156NRUG",
  observation_start = as.Date("1981-01-01"),
  observation_end = as.Date("2020-12-31"),
  frequency = "a"
)

labshare1 = fredr(
  series_id = "LABSHPUSA156NRUG",
  observation_start = as.Date("1981-01-01"),
  observation_end = as.Date("1990-12-31"),
  frequency = "a"
)

labshare2 = fredr(
  series_id = "LABSHPUSA156NRUG",
  observation_start = as.Date("2011-01-01"),
  observation_end = as.Date("2019-12-31"),
```

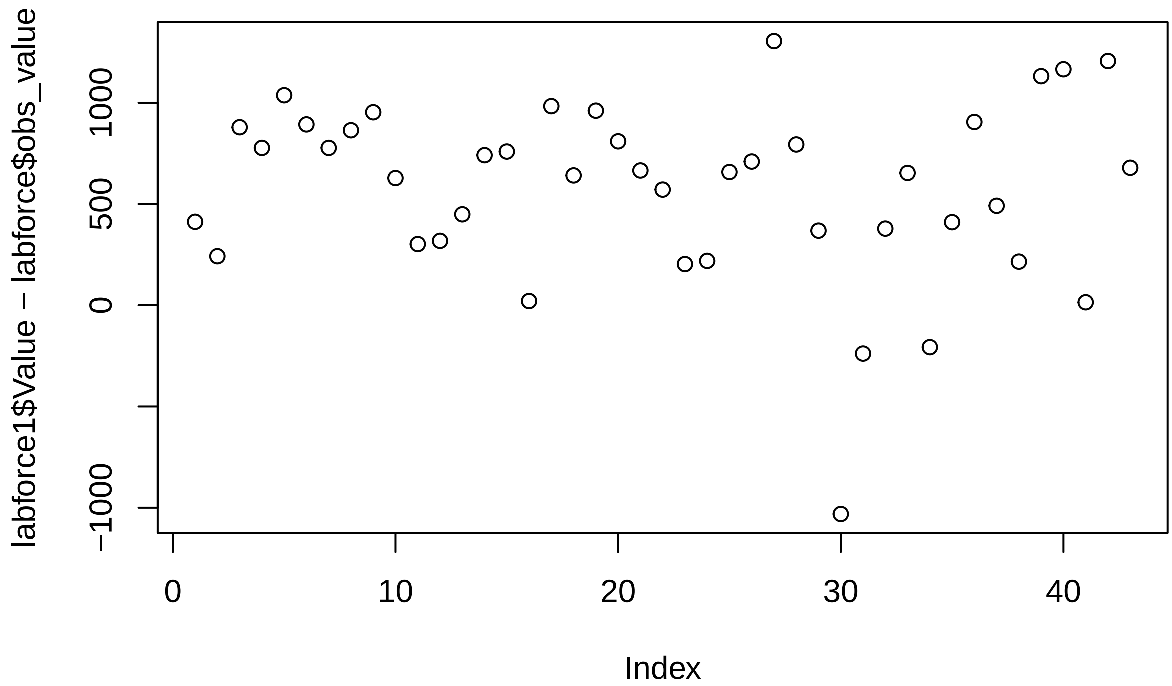
```r
  frequency = "a"
)


labshare = data.frame(labshare)

labshare1 = data.frame(labshare1)

labshare2 = data.frame(labshare2)
```

This is precisely why I tried to replicate the laborshare measure provided by Guerriero (2019), defined as:

$$LS6 = \frac{compensation\ of\ employees * \left(\frac{workforce - employers}{employees}\right)}{value\ added - ind.\ taxes - fixed\ cap.\ cons.}$$

```r
#API key for bls is: 7b39505b098044f886d056883d2bd925

#library(blsAPI)
library(wbstats)

s=wb_search("compensation of employees")
comp=wb_data(country="US","GC.XPN.COMP.CN")

s=fredr_series_search_text("compensation")
#View(s)

labforce1=read.csv("civilianlaborforce.csv", sep=",")
labforce1=subset(labforce1, labforce1$Period=="M12")

nomgdp = fredr(
  series_id = "GDP",
  observation_start = as.Date("1981-01-01"),
  observation_end = as.Date("2020-12-31"),
  frequency = "a",
  units = "lin",
  aggregation_method = "eop"
)

library(Rilostat)

#This should be the correct indicator to use for the labor force level
labforce=get_ilostat("EAP_TEAP_SEX_AGE_NB_A")
labforce=subset(labforce, labforce$indicator=="EAP_TEAP_SEX_AGE_NB"&
                          labforce$ref_area=="USA"&
                          labforce$classif1=="AGE_AGGREGATE_TOTAL"&
                          labforce$sex=="SEX_T"&
                          labforce$time>=1980)
toc=get_ilostat_toc()

# Load the dplyr package
library(dplyr)

# Assume your dataframe is called "my_df" and your time column is called "time"
```

```
# Sort the dataframe in ascending order of time
labforce <- labforce %>% arrange(labforce$time)
plot(labforce1$Value-labforce$obs_value)
```



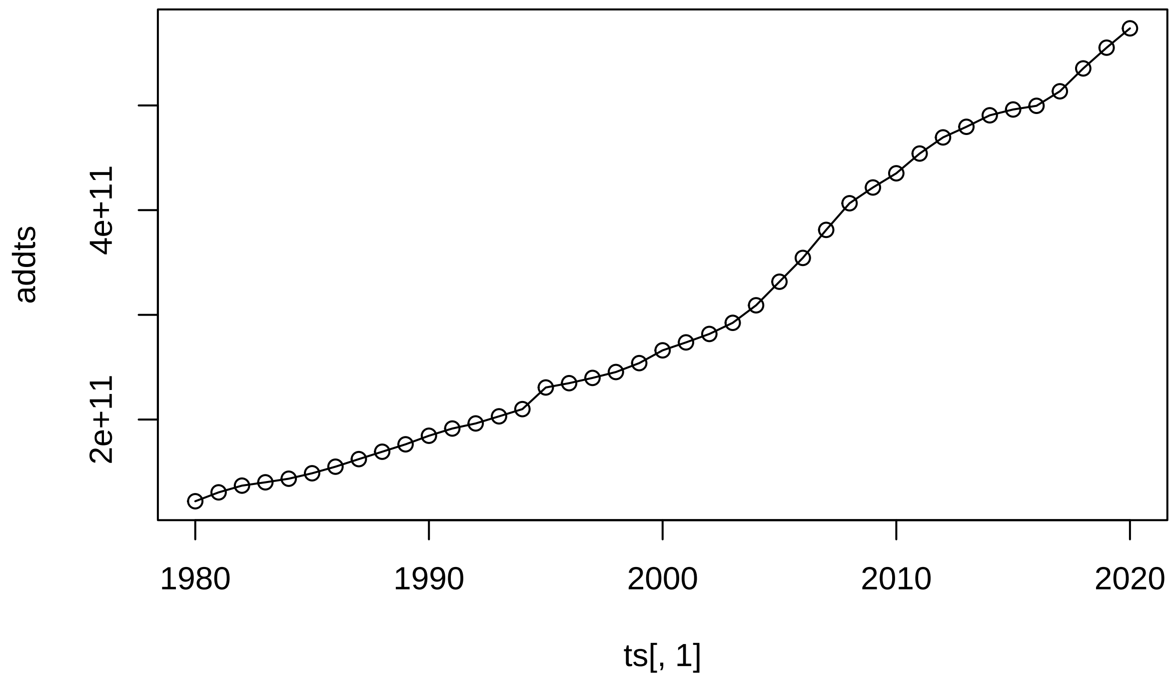

```
#Download and filter the dataset with all employed by status
empl=get_ilostat("EMP_TEMP_SEX_AGE_STE_NB_A")
empl=subset(empl, empl$indicator=="EMP_TEMP_SEX_AGE_STE_NB"&
                  empl$ref_area=="USA"&
                  empl$classif1=="AGE_AGGREGATE_TOTAL"&
                  empl$classif2 == "STE_AGGREGATE_TOTAL"&
                  empl$sex=="SEX_T"&
                  empl$time>=1980)

#To get only employees, filter for STE_ICSE93_1
emplee=get_ilostat("EMP_TEMP_SEX_AGE_STE_NB_A")
emplee=subset(emplee, emplee$indicator=="EMP_TEMP_SEX_AGE_STE_NB"&
                      emplee$ref_area=="USA"&
                      emplee$classif1=="AGE_AGGREGATE_TOTAL"&
                      emplee$classif2 == "STE_ICSE93_1"&
                      emplee$sex=="SEX_T"&
                      emplee$time>=1980)

#Filter for STE_ICSE93_3
icse93_3=get_ilostat("EMP_TEMP_SEX_AGE_STE_NB_A")
icse93_3=subset(icse93_3, icse93_3$indicator=="EMP_TEMP_SEX_AGE_STE_NB"&
```

```
                      icse93_3$ref_area=="USA"&
                      icse93_3$classif1=="AGE_AGGREGATE_TOTAL"&
                      icse93_3$classif2 == "STE_ICSE93_3"&
                      icse93_3$sex=="SEX_T"&
                      icse93_3$time>=1980)

#Filter for STE_ICSE93_5
icse93_5=get_ilostat("EMP_TEMP_SEX_AGE_STE_NB_A")
icse93_5=subset(icse93_5, icse93_5$indicator=="EMP_TEMP_SEX_AGE_STE_NB"&
                      icse93_5$ref_area=="USA"&
                      icse93_5$classif1=="AGE_AGGREGATE_TOTAL"&
                      icse93_5$classif2 == "STE_ICSE93_5"&
                      icse93_5$sex=="SEX_T"&
                      icse93_5$time>=1980)

#Filter for STE_AGGREGATE_SLF
aggslf=get_ilostat("EMP_TEMP_SEX_AGE_STE_NB_A")
aggslf=subset(aggslf, aggslf$indicator=="EMP_TEMP_SEX_AGE_STE_NB"&
                      aggslf$ref_area=="USA"&
                      aggslf$classif1=="AGE_AGGREGATE_TOTAL"&
                      aggslf$classif2 == "STE_AGGREGATE_SLF"&
                      aggslf$sex=="SEX_T"&
                      aggslf$time>=1980)

#Merge the dataframes
# merged_df <- merge(df1, df2["Date", "Value", drop = FALSE], by = "Date", all = TRUE)

empl = select(empl, time, obs_value)
names(empl)[2]="empl"

emplee = select(emplee, time, obs_value)
names(emplee)[2]="emplee"

icse93_3 = select(icse93_3, time, obs_value)
names(icse93_3)[2]="icse93_3"

icse93_5 = select(icse93_5, time, obs_value)
names(icse93_5)[2]="icse93_5"

aggslf = select(aggslf, time, obs_value)
names(aggslf)[2]="aggslf"

labforce = select(labforce, time, obs_value)
names(labforce)[2]="labforce"

mergemp <- empl %>%
            merge(labforce, by = "time") %>%
            merge(emplee, by = "time") %>%
            merge(icse93_3, by = "time") %>%
            merge(icse93_5, by = "time") %>%
            merge(aggslf, by = "time")

toc=get_ilostat_toc()
```

```r
# Calculate the number of employers as aggslf - icse93_3 - icse93_5
# this is under the assumption that icse93_4 is negligible
# in fact is around 8k workers for the US economy as a whole

mergemp$emplrs = mergemp$aggslf - mergemp$icse93_3 - mergemp$icse93_5


# For lack of available data, we will use  LS5 and not LS6
#Now, calculate the multiplier of the compensation for every year
mergemp$mult = mergemp$labforce/mergemp$emplee


#Now retrieve data from the un on aggregates
un = read.csv("un-nsa-aggregates.txt", sep=";")
names(un)[6]="time"
un = un[order(un$time), ]



#Use the time period of the interregnum (1995-2011), and maybe cut in two (1995-2008)
#Regression with sna2008 as dependent var and sna1993 as regressor
#Coefficient will be our conversion factor.
#Evaluate the model, then check if it can replicate the other series
#and check for violations of assumptions

#Apply to convert the vintage period

library(stargazer)

findconversion = function(s1,s2){
  train = lm(s1 ~ s2)
  stargazer(train, type = "text")
  coef = train$coefficients
  return(coef)
}

# First is the fixed cap cons, then gross value added
# then compensation and lastly indirect taxes


codelist = c("K.1", "B.1g", "D.1", "D.2-D.3")

codenames = c("time","fcc", "gdp", "wages", "ind_tax")

ts=data.frame(seq(1980,2020,1))


for(i in 1:length(codelist)){
  ts93 = subset(un, un$SNA93.Item.Code == codelist[i] &
                      un$Series == 100 & un$time <= 2011 & un$time >= 1995)
  ts08 = subset(un, un$SNA93.Item.Code == codelist[i] &
                      un$Series == 1000 & un$time <= 2011 & un$time >= 1995)
  tsold = subset(un, un$SNA93.Item.Code == codelist[i] &
```

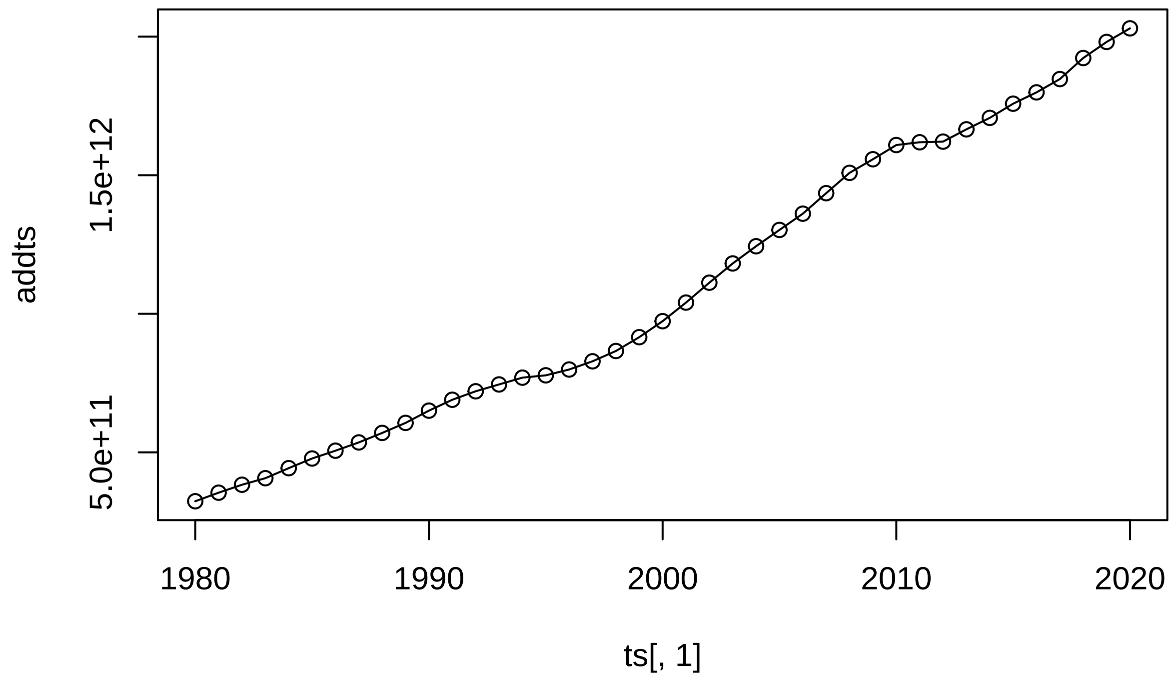
```r
                      un$Series == 100 & un$time < 1995)
  tsnew = subset(un, un$SNA93.Item.Code == codelist[i] &
                      un$Series == 1000 & un$time >= 1995)
  coef = findconversion(ts08$Value, ts93$Value)
  simts08 = tsold$Value * coef[2] + coef[1]
  addts=c(simts08, tsnew$Value)
  ts[,i+1]=addts
  plot(ts[,1], addts, type = "o")
}
```

```
##
## ================================================
##                         Dependent variable:
##                     ----------------------------
##                                  s1
## ------------------------------------------------
## s2                             1.571***
##                                (0.037)
##
## Constant                 63,836,459,006.000***
##                           (6,316,653,900.000)
##
## ------------------------------------------------
## Observations                     17
## R2                             0.992
## Adjusted R2                    0.991
## Residual Std. Error 7,243,450,444.000 (df = 15)
## F Statistic          1,759.565*** (df = 1; 15)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```




```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                 s1
## -----------------------------------------------
## s2                           1.082***
##                               (0.004)
##
## Constant               65,624,974,467.000***
##                         (5,480,194,745.000)
##
## -----------------------------------------------
## Observations                    17
## R2                             1.000
## Adjusted R2                    1.000
## Residual Std. Error 5,394,598,383.000 (df = 15)
## F Statistic         71,594.070*** (df = 1; 15)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```


```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                  s1
## ------------------------------------------------
## s2                             1.003***
##                                (0.009)
##
## Constant                  17,021,340,784.000
##                          (10,988,233,590.000)
##
## ------------------------------------------------
## Observations                      17
## R2                              0.999
## Adjusted R2                     0.999
## Residual Std. Error 10,847,339,997.000 (df = 15)
## F Statistic          11,967.850*** (df = 1; 15)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```




```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                 s1
## -----------------------------------------------
## s2                           1.024***
##                               (0.004)
##
## Constant                -14,792,373,436.000***
##                           (3,318,888,812.000)
##
## -----------------------------------------------
## Observations                    17
## R2                             1.000
## Adjusted R2                    1.000
## Residual Std. Error 2,953,524,347.000 (df = 15)
## F Statistic         60,797.620*** (df = 1; 15)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```


```
names(ts) = codenames


##########################################
############ SECOND METHOD ###############
##########################################

# Per il secondo metodo:
# ricontrolla il paper di Autor sulla fall of labor share
# and rise of superstar firms
```