The Euro Area

First of all, we gather the following databases:

1. The database based on @Faga01 and used in the bayesian estimation of the DSGE model developped in @Smet03 for the Euro area : https://macro.cepremap.fr/article/2015-10/sw03-data/

2. The financial database similar to @Chri14a, but for the Euro area : https://macro.cepremap.fr/article/2016-06/cmr14-EA-data/

3. The fiscal database for the Euro Area, based on the paper of @ppp09 : https://macro.cepremap.fr/article/2019-11/fipu-EA-data/

4. The foreign block database for the Eurozone : https://macro.cepremap.fr/article/2019-12/open-EA-data/

```
sw03 <- read_csv("https://shiny.cepremap.fr/data/EA_SW_rawdata.csv") %>%
  filter(period >="1980-01-01")

fipu <- read_csv("https://shiny.cepremap.fr/data/EA_Fipu_rawdata.csv")

finance <- read_csv("https://shiny.cepremap.fr/data/EA_Finance_rawdata.csv")

open <- read_csv("https://shiny.cepremap.fr/data/EA_Open_rawdata.csv")

EA_rawdata <-
  sw03 %>%
  inner_join(fipu,by="period") %>%
  inner_join(finance,by="period") %>%
  inner_join(open,by="period") %>%
  rename(unempbenef=unemp) %>%
  mutate(pop=1000*pop) %>%
  add_column(country="EA")
```

France, Germany, Italy & Spain

Additionnally to the Euro area data, country data for France, Germany, Italy and Spain is needed. Different sources have been used: Eurostat, IMF (WEO and IFS), BIS, OECD and ECB.

Data Retrieval & Seasonal Adjustment

In this section we get country data on :

- Gross Domestic Product
- consumption
- investment
- GDP deflator
- compensation of employees
- hours worked

- investment deflator
- loans to non-financial corporations
- entrepreneurial net worth
- short-term interest rate
- lending rate
- total government expenditure
- government consumption
- government investment
- government social transfers
- government interest payments
- government debt
- world demand
- total government revenue
- unemployment benefits
- nominal effective exchange rate
- imports
- exports
- population

As in the Euro area database, except oil prices which are the same by assumption.

Some of these data may not be seasonally adjusted, therefore, we use the seasonal package developed by Sax (2016) to remove the seasonal component.

Compensation of Employees

We use the `namq_10_a10` database from Eurostat to obtain these data.

```
wage_de_fr <-
  rdb("Eurostat","namq_10_a10",mask = "Q.CP_MEUR.SA.TOTAL.D1.DE+FR")
%>%
  add_column(var="wage")

wage_es_it <-
  rdb("Eurostat","namq_10_a10", mask = "Q.CP_MEUR.SCA.TOTAL.D1.ES+IT")
%>%
  add_column(var="wage")
```

Hours Worked

We use the `namq_10_a10_e` database from Eurostat to retrieve these data.

```
hours <-
  rdb("Eurostat","namq_10_a10_e",mask = "Q.THS_HW.TOTAL.SCA.EMP_DC.IT+
DE+FR+ES") %>%
  add_column(var="hours")
```

Gross Domestic Product

We use the `namq_10_gdp` database from Eurostat to obtain these data.

```
gdp <-
  rdb("Eurostat","namq_10_gdp",mask = "Q.CLV10_MEUR.SCA.B1GQ.IT+DE+
FR+ES") %>%
  add_column(var="gdp")
```

Consumption

We use the `namq_10_gdp` database from Eurostat to retrieve these data.

```
conso <-
  rdb("Eurostat","namq_10_gdp",mask = "Q.CLV10_MEUR.SCA.P31_S14_S15.IT+DE+FR+ES") %>%
  add_column(var="conso")
```

Investment

We use the `namq_10_gdp` database from Eurostat to obtain these data.

```
inves <-
  rdb("Eurostat","namq_10_gdp",mask = "Q.CLV10_MEUR.SCA.P51G.IT+DE+FR+ES") %>%
  add_column(var="inves")
```

GDP Deflator

We use the `namq_10_gdp` database from Eurostat to retrieve these data.

```
defgdp <-
  rdb("Eurostat","namq_10_gdp",mask = "Q.PD10_EUR.SCA.B1GQ.IT+DE+FR+ES") %>%
  add_column(var="defgdp")
```

Investment Deflator

We use the `namq_10_gdp` database from Eurostat to obtain these data.

```
definves <-
  rdb("Eurostat","namq_10_gdp",mask = "Q.PD10_EUR.SCA.P51G.IT+DE+FR+ES") %>%
  add_column(var="definves")
```

Population

Given the scarcity of data to build long quarterly series for this variable, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `lfsq_pganws` & `demo_pjanbroad`.

```
pop_recent <-
  rdb("Eurostat","lfsq_pganws",mask = "Q.THS.T.TOTAL.Y15-64.POP.IT+DE+FR+ES") %>%
  add_column(var="pop_recent")

pop_old <-
  rdb("Eurostat","demo_pjanbroad",mask = "A.NR.Y15-64.T.IT+DE+FR+ES") %>%
  add_column(var="pop_old")
```

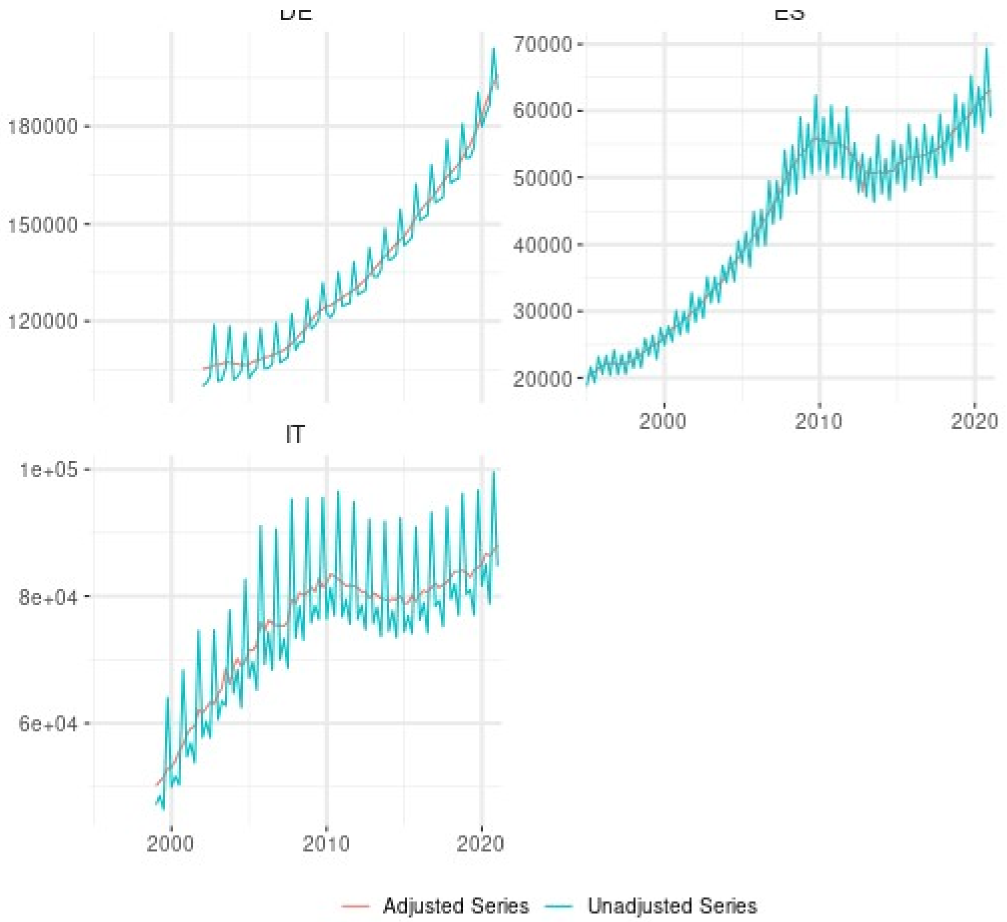
Government Consumption

Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

```
pubcons_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.P3.FR") %>%
  add_column(var="pubcons_recent")

pubcons_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.P3.IT+DE+ES")
%>%
  add_column(var="pubcons_recent")

pubcons_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.P3.IT+DE") %>%
  add_column(var="pubcons_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy and Germany, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  pubcons_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Government Consumption")
```



```
pubcons_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="pubcons_recent")
```

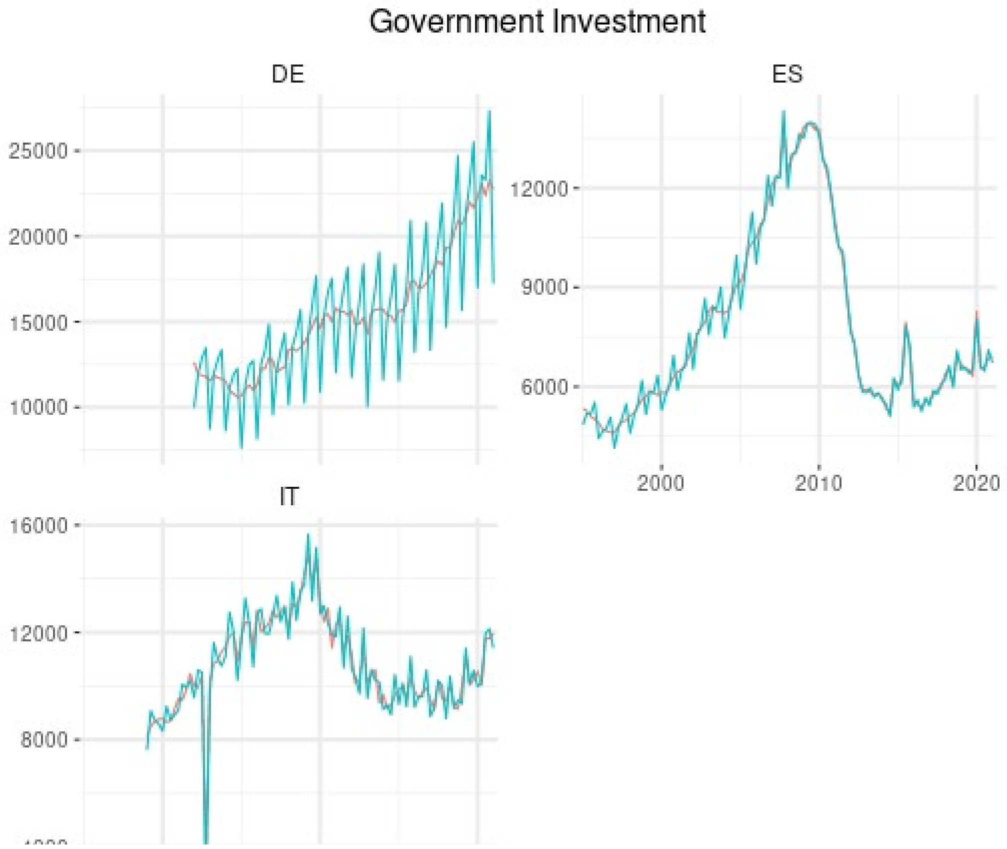
Government Investment

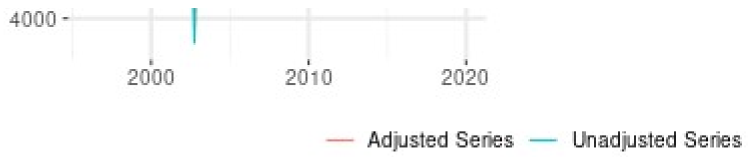
Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

```
pubinves_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.P51G.FR")
%>%
  add_column(var="pubinves_recent")

pubinves_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.P51G.IT+DE+
ES") %>%
  add_column(var="pubinves_recent")

pubinves_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.P51G.IT+DE") %>%
  add_column(var="pubinves_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy and Germany, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  pubinves_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Government Investment")
```





```
pubinves_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="pubinves_recent")
```

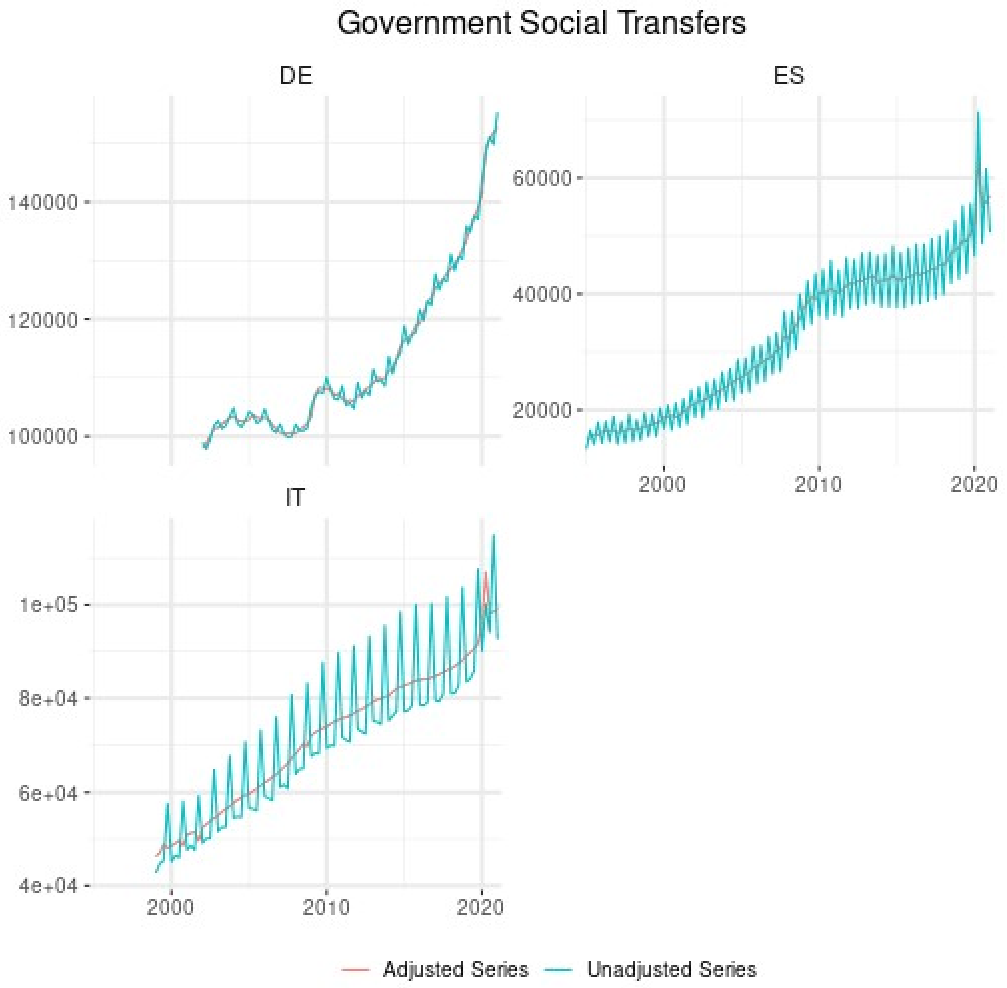
Government Social Transfers

Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

```
tfs_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.D62PAY.FR")
%>%
  add_column(var="tfs_recent")

tfs_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.D62PAY.IT+
DE+ES") %>%
  add_column(var="tfs_recent")

tfs_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.D62PAY.IT+DE") %>%
  add_column(var="tfs_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy and Germany, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  tfs_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Government Social Transfers")
```



```
tfs_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="tfs_recent")
```

Government Interest Payments

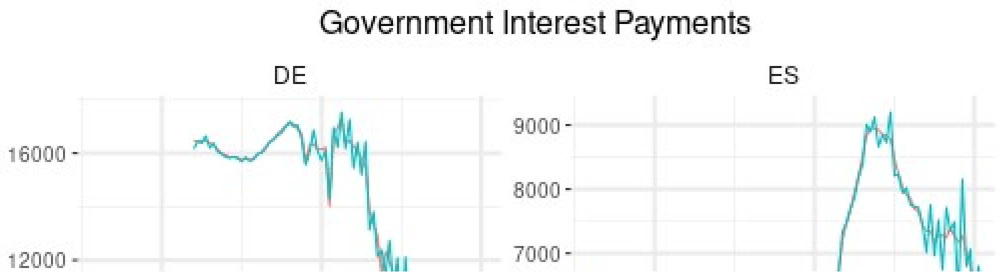
Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

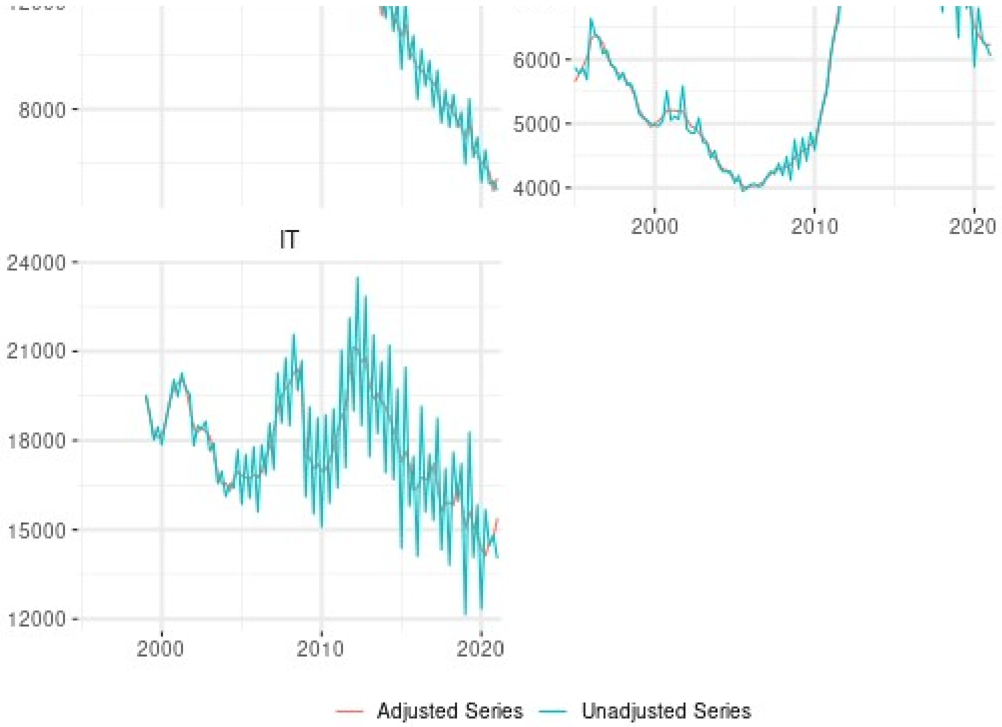
```
intpay_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.D41PAY.FR")
```

```
%>%
  add_column(var="intpay_recent")

intpay_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.D41PAY.IT+
DE+ES") %>%
  add_column(var="intpay_recent")

intpay_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.D41PAY.DE+IT") %>%
  add_column(var="intpay_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy and Germany, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  intpay_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Government Interest Payments")
```





```
intpay_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="intpay_recent")
```

Total Government Expenditure

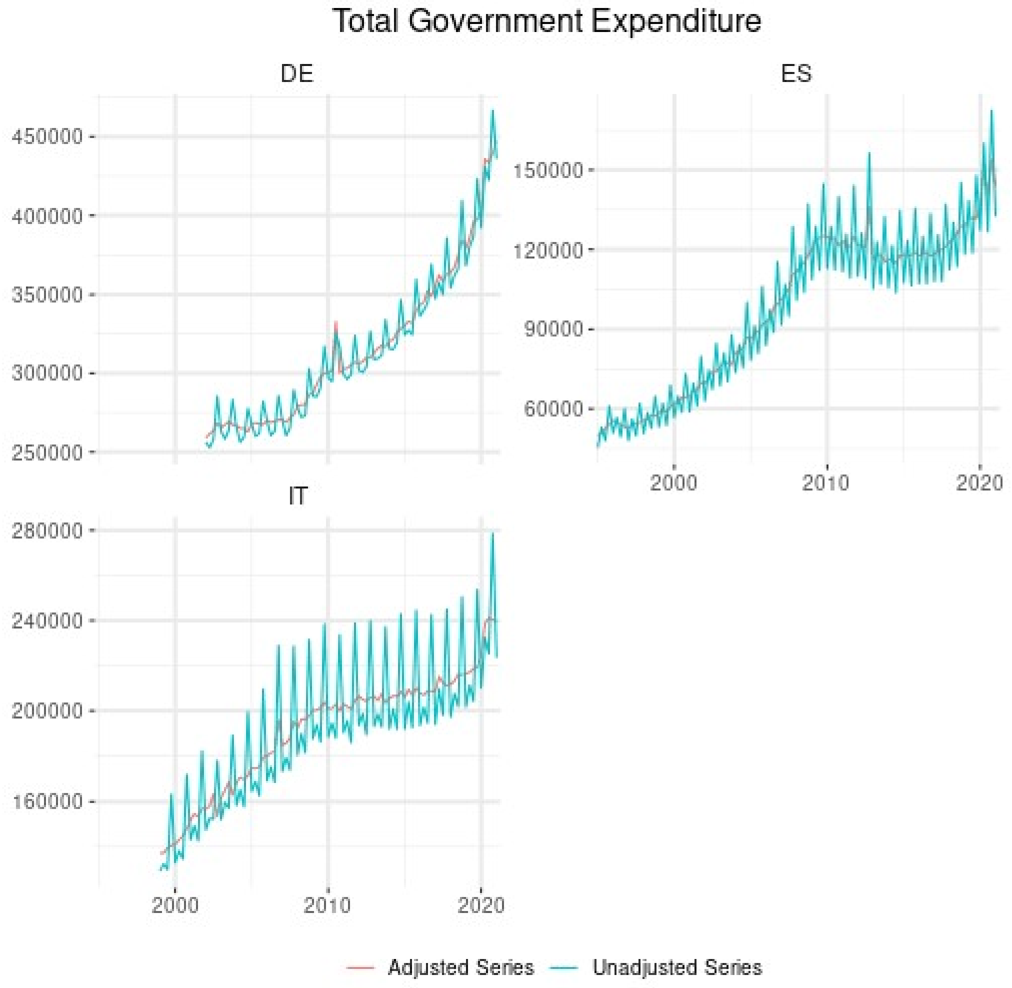
Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

```
totexp_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.TE.FR") %>%
  add_column(var="totexp_recent")

totexp_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.TE.IT+DE+ES")
%>%
  add_column(var="totexp_recent")

totexp_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.TE.DE+IT") %>%
  add_column(var="totexp_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy, Germany, and Spain, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  totexp_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
```

```
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Total Government Expenditure")
```



```
totexp_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="totexp_recent")
```

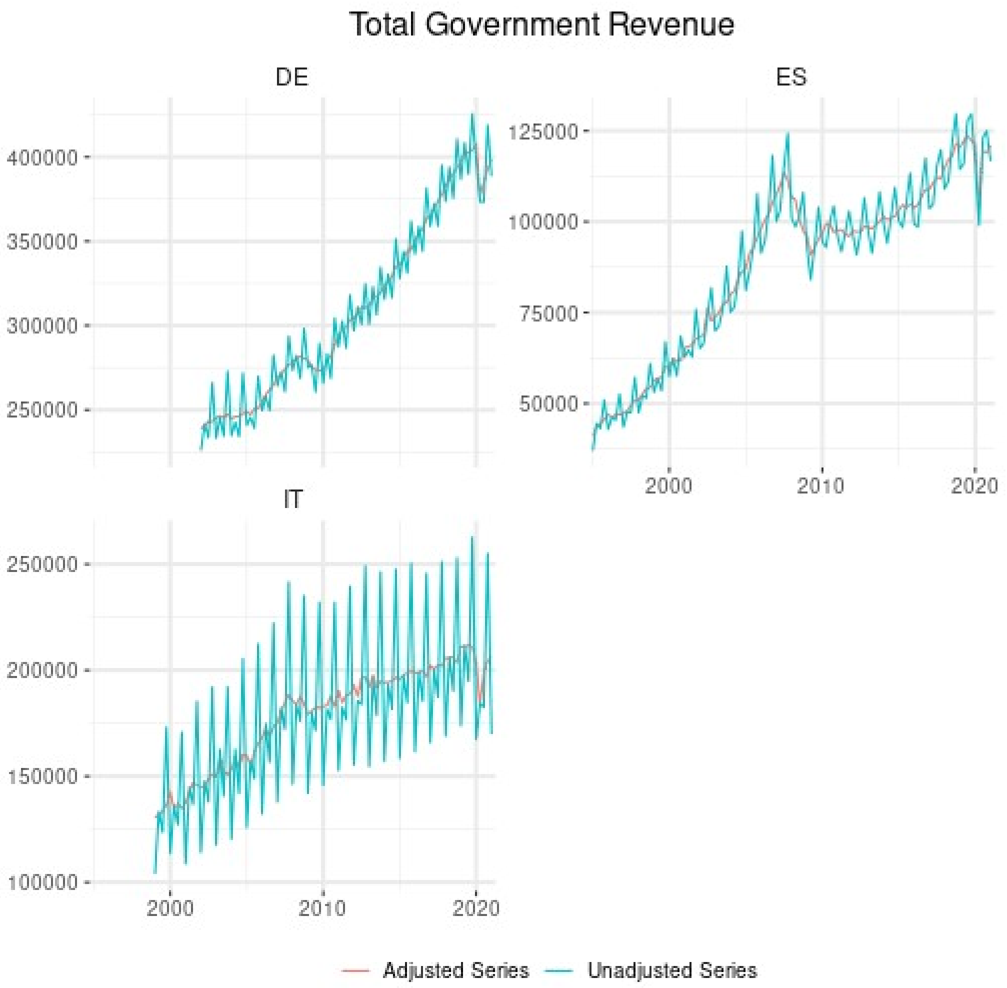
Total Government Revenue

Given the scarcity of data to build long quarterly series for Italy and Germany, we use 2 databases from Eurostat that we will chain and interpolate by country in the next section: `gov_10q_ggnfa` & `gov_10a_main`.

```
totrev_recent_fr <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.SCA.S13.TR.FR") %>%
  add_column(var="totrev_recent")

totrev_recent_it_de_es_nsa <-
  rdb("Eurostat","gov_10q_ggnfa",mask = "Q.MIO_EUR.NSA.S13.TR.IT+DE+ES")
%>%
  add_column(var="totrev_recent")

totrev_old_it_de <-
  rdb("Eurostat","gov_10a_main",mask = "A.MIO_EUR.S13.TR.DE+IT") %>%
  add_column(var="totrev_old")
```

Since the data retrieved from the `gov_10q_ggnfa` database is not seasonally adjusted for Italy, Germany, and Spain, we use the seasonal package developed by Sax (2016):

```
df_nsa_q <-
  totrev_recent_it_de_es_nsa %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Total Government Revenue")
```

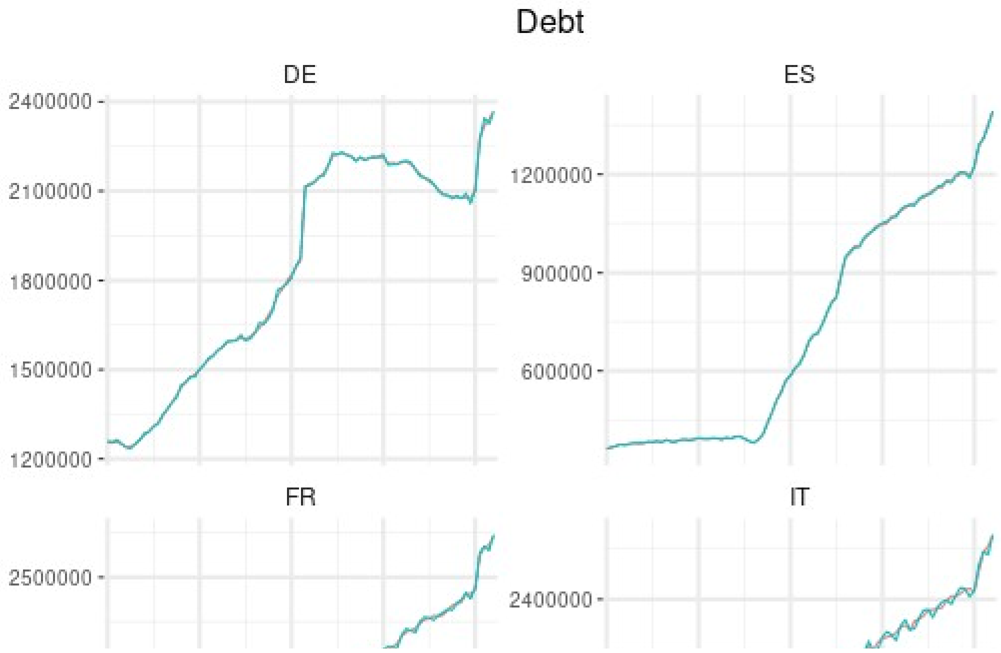


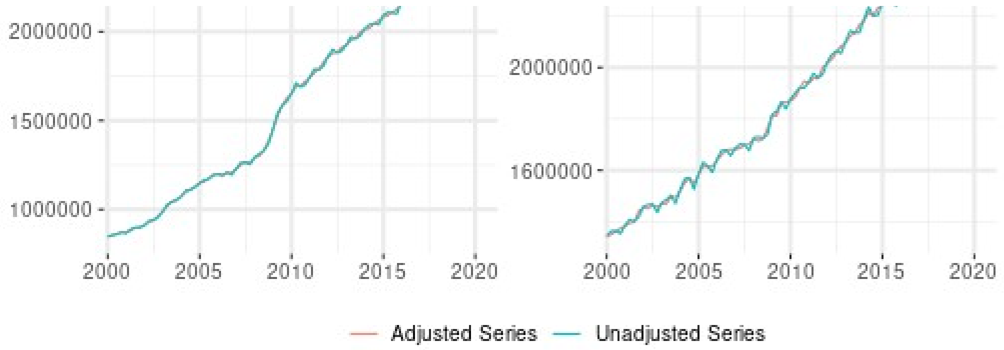
```
totrev_recent_it_de_es <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="totrev_recent")
```

Government Debt

Given the scarcity of data to build long quarterly series for this variable, we use 2 databases, one from Eurostat and the other from the IMF, that we will chain and interpolate by country in the next section: `gov_10q_ggdebt` & `WEO`, respectively.

```
debt_recent <-
  rdb("Eurostat","gov_10q_ggdebt",mask = "Q.GD.S13.MIO_EUR.IT+DE+FR+ES")
%>%
  add_column(var="debt_recent")

debt_old <-
  rdb(provider_code = "IMF", dataset_code = "WEO:latest", mask =
"DEU+ESP+FRA+ITA.GGXWDG") %>%
  add_column(var="debt_old") %>%
  select(geo='weo-country',period,value,var) %>%
  mutate(geo=str_sub(geo,1,2)) %>%
  filter(year(period)<=max(debt_recent$period))

df_nsa_q <-
  debt_recent %>%
  select(period,country=geo, value)

to_deseason <-
  df_nsa_q %>%
  spread(country, value)

deseasoned_q <-
  bind_rows(lapply(unique(df_nsa_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

df_nsa_q %<>% mutate(Origin = "Unadjusted Series")

plot_df <-
  bind_rows(df_nsa_q,deseasoned_q) %>%
  na.omit()

ggplot(plot_df,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Debt")
```





```
debt_recent <-
  deseasoned_q %>%
  filter(Origin=="Adjusted Series") %>%
  select(country,-Origin,value,period) %>%
  mutate(var="debt_recent")
```

Loans to Non-Financial Corporations

We use the `CNFS` database from the Bank for International Settlements to obtain these data.

```
loans_nfc <-
  rdb("BIS","total_credit",mask = "Q.IT+DE+FR+ES.N.A.M.XDC.A") %>%
  add_column(var="loans_nfc") %>%
  select(geo=BORROWERS_CTY,period,value,var)
```

Entrepreneurial net worth

```
networth <-
  rdb("OECD","MEI", mask = "FRA+DEU+ITA+ESP.SPASTT01.IXOB.Q") %>%
  select(period,value,geo=LOCATION) %>%
  add_column(var="networth") %>%
  mutate(geo=case_when(
    geo=="FRA" ~ "FR",
    geo=="DEU" ~ "DE",
    geo=="ESP" ~ "ES",
    geo=="ITA" ~ "IT"))
```

Short-term interest rate

```
shortrate <-
  rdb("OECD","MEI", mask = "FRA+DEU+ITA+ESP.IR3TIB01.ST.Q") %>%
  select(period,value,geo=LOCATION) %>%
  add_column(var="shortrate") %>%
  mutate(geo=case_when(
    geo=="FRA" ~ "FR",
    geo=="DEU" ~ "DE",
    geo=="ESP" ~ "ES",
    geo=="ITA" ~ "IT"))
```

Lending Rate

Given the scarcity of data to build long quarterly series for this variable, we use 2 databases, one from the European Central Bank and the other from the IMF, that we will chain and interpolate by country in the next section: `MIR` & `IFS`, respectively.

```
lendingrate_recent <-
  rdb("ECB","MIR",mask = "M.IT+DE+FR+ES.B.A2A.A.R.A.2240.EUR.N") %>%
  select(geo=REF_AREA,period,value) %>%
  mutate(period=paste(year(period),quarter(period))) %>%
  group_by(geo,period) %>%
  summarize(value=mean(value,na.rm = T)) %>%
  mutate(var="lendingrate_recent",
         period=yq(period))

lendingrate_old <-
  rdb("IMF","IFS",mask = "Q.IT+DE+FR+ES.FILR_PA") %>%
  add_column(var="lendingrate_old") %>%
  select(geo=REF_AREA,period,value,var)
```

World Demand

We use the foreign demand specific to the four countries, that we also built for this project : https://macro.cepremap.fr/article/2020-02/foreign-demand-euro-countries-data/

```
world_demand <-
  read_csv("https://shiny.cepremap.fr/data/Foreign_demand.csv") %>%
  rename(geo=country)
```

Unemployment benefits

Given the scarcity of data to build long quarterly series for this variable, we proceed in this way: we first determine quarterly series using the ratio of quarterly social expenditures on annual unemployment benefits, and then we use the seasonal package developed by Sax to remove the seasonal componenet from the series. Then we retrieve annual data on unemployment benefits using the `spr_exp_sum` database from Eurostat.

In the first place, we retrieve government social expenditures and compute their quaterly share for each year:

```
filter <- "Q.MIO_EUR.NSA.S13.D62PAY.IT+DE+FR+ES"
df <- rdb("Eurostat","gov_10q_ggnfa",mask = filter)

socialexp <-
  df %>%
  mutate(year=year(period),country=geo) %>%
  select(period,value,year,country) %>%
  group_by(year,country) %>%
  mutate(sum=sum(value),
         ratio=value/sum) %>%
  ungroup() %>%
  select(-value,-year,-sum)
```

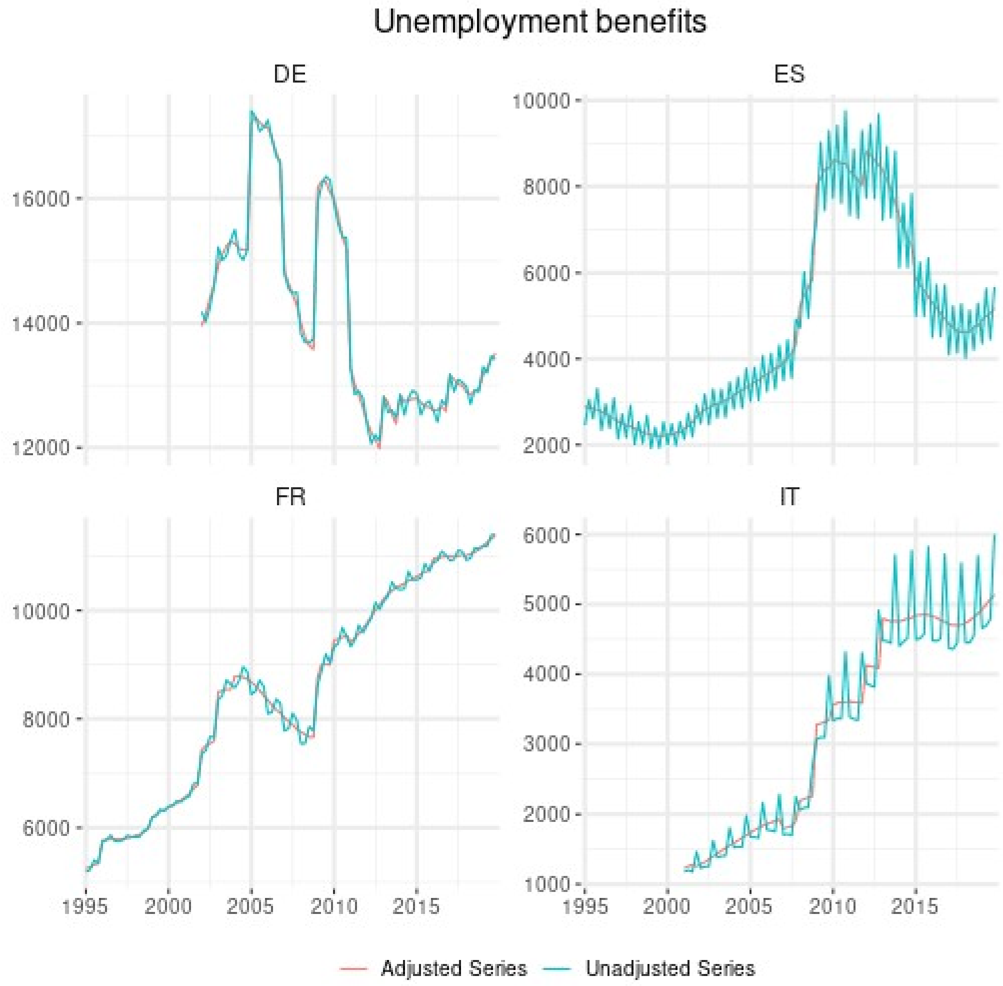
Then, we retrieve the latest annual data on unemployment benefits, put them in a quarterly table and use the previous ratio of quarterly social expenditures to compute quarterly unemployment benefits:

```
filter <- "A.MIO_EUR.S13.GF1005.TE.IT+DE+FR+ES"
```

```
df <- rdb("Eurostat","gov_10a_exp",mask = filter)

recent_unempbenef <-
  df %>%
  mutate(year=year(period),country=geo) %>%
  select(period,value,year,country) %>%
  spread(country, value)

recent_unempbenef_q <-
  tibble(period=seq(min(recent_unempbenef$period),
                    length.out=nrow(recent_unempbenef)*4,
                    by = "quarter"),
         year=year(period)) %>%
  left_join(recent_unempbenef,by="year") %>%
  select(-period.y,-year) %>%
  rename(period=period.x) %>%
  gather(country,value,-period)

unempbenef_q <-
  recent_unempbenef_q %>%
  inner_join(socialexp,by=c("period"="period","country"="country")) %>%
  mutate(value=value*ratio) %>%
  select(-ratio)%>%
  na.omit()
```

Since we have unadjusted data, we use the seasonal package developed by Sax (2016) on these data:

```
to_deseason <-
  unempbenef_q %>%
  spread(country, value)

unempbenef_q_deseasoned <-
  bind_rows(lapply(unique(unempbenef_q$country),
                   function(var) deseason(var_arrange = var,
                                          source_df = to_deseason)))%>%
  mutate(Origin = "Adjusted Series",country=var)%>%
  select(-var)

unempbenef_q %<>% mutate(Origin = "Unadjusted Series")

plot_unempbenef <-
  bind_rows(unempbenef_q,unempbenef_q_deseasoned) %>%
  na.omit()

ggplot(plot_unempbenef,aes(period,value,colour=Origin))+
  geom_line()+
  facet_wrap(~country ,scales ="free_y",ncol = 2)+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  theme(strip.text = element_text(size=12)) +
  theme(plot.title = element_text(size=16)) +
  ggtitle("Unemployment benefits")
```



```
unempbenef_recent <-
  unempbenef_q_deseasoned %>%
  filter(Origin=="Adjusted Series") %>%
  select(geo=country,-Origin,value,period) %>%
  mutate(var="unempbenef_recent")
```

Now, using the `spr_exp_sum` database from Eurostat we retrieve the annual series, that will be interpolated and chained to the quarterly series in the next section.

```
filter <- "A.UNEMPLOY.MIO_EUR.IT+DE+FR+ES"
df <- rdb("Eurostat","spr_exp_sum",mask=filter)

unempbenef_old <-
  df %>%
  add_column(var="unempbenef_old") %>%
  select(period,value,geo,var)
```

Nominal effective exchange rate

```
df <- rdb("BIS","eer",mask="M.N.B.IT+DE+FR+ES")

neer <-
  df %>%
  select(period,value,geo=REF_AREA) %>%
  mutate(period=paste(year(period),quarter(period))) %>%
  group_by(geo,period) %>%
  summarize(value=mean(value)) %>%
  mutate(period=yq(period),
         var="neer")
```

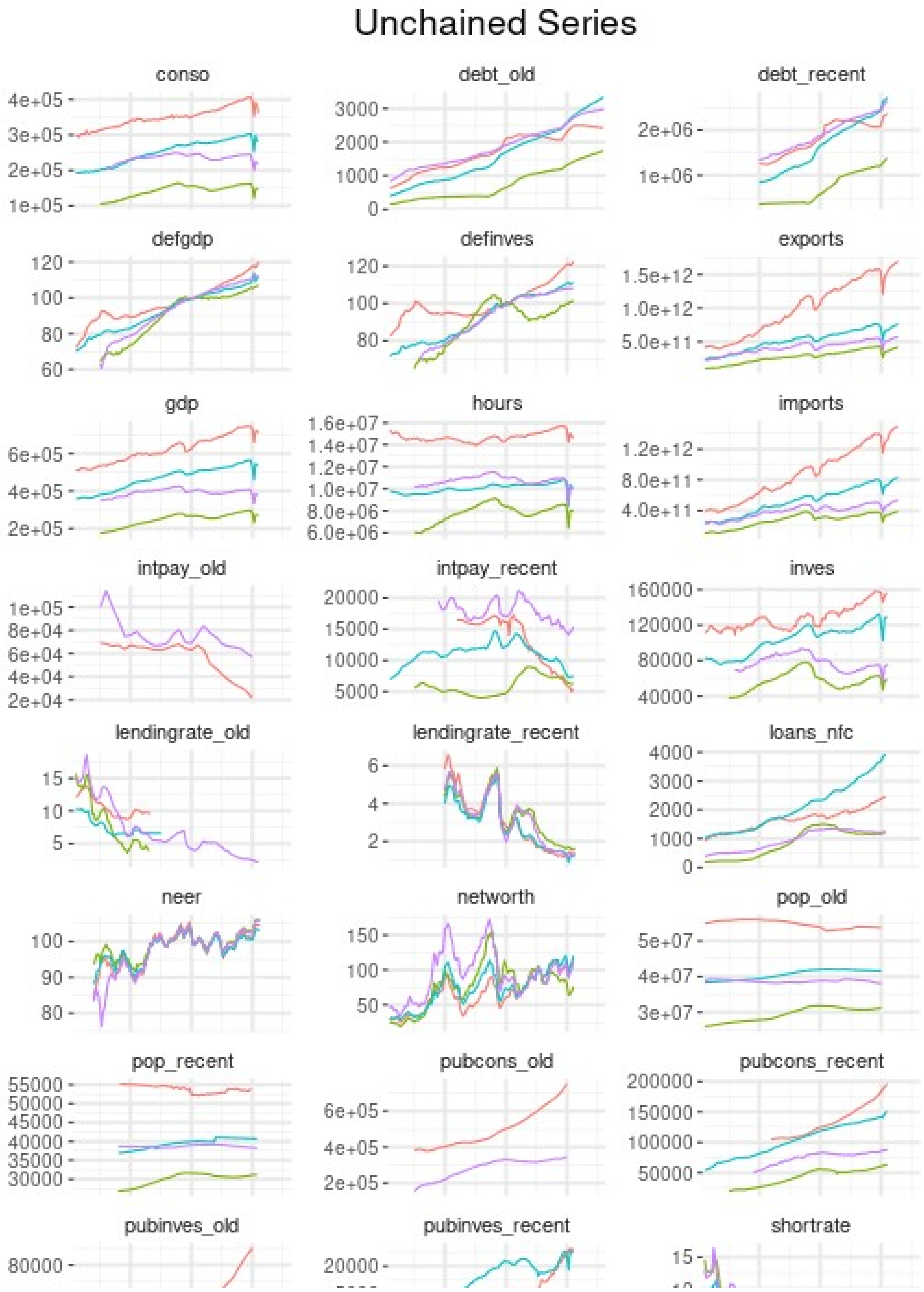
Imports and exports

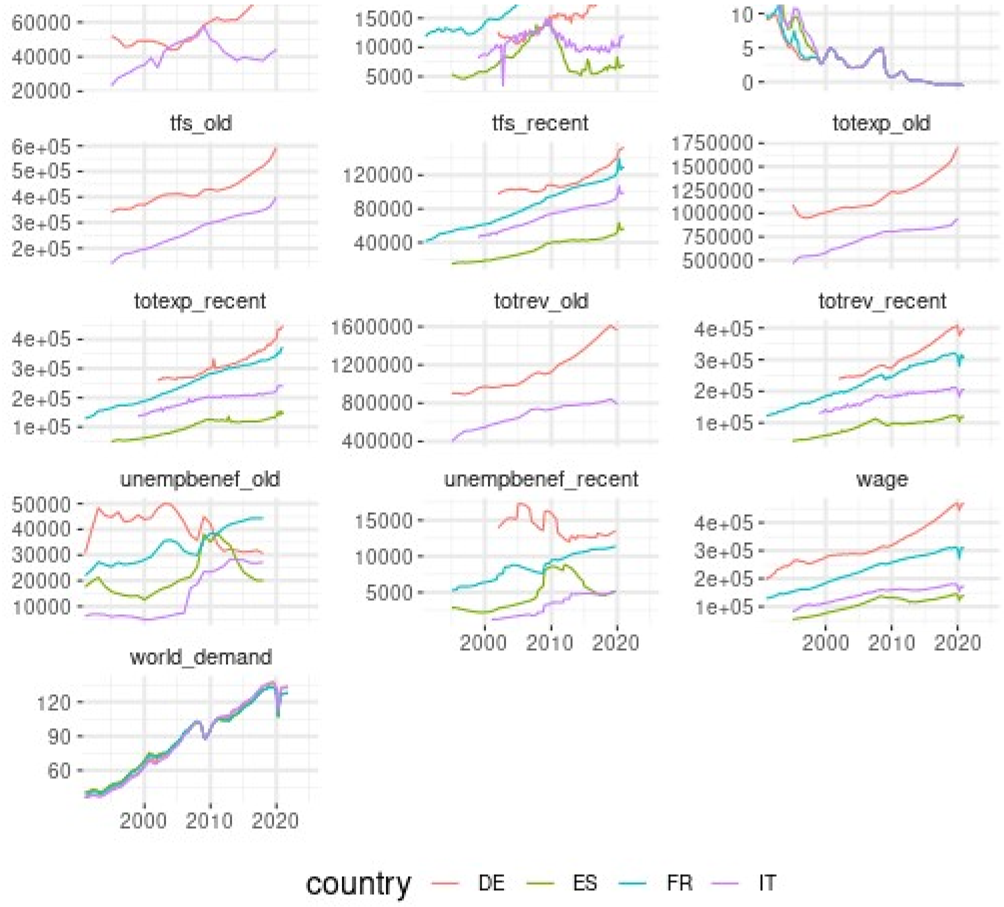
```
df <- rdb("OECD","EO",mask = "FRA+DEU+ITA+ESP.MGSV+XGSV.Q")

imports_exports_volume <-
  df %>%
  select(period,value,geo=LOCATION,value,var=VARIABLE) %>%
  mutate(var=case_when(
    var=="MGSV" ~ "imports",
    var=="XGSV" ~ "exports")) %>%
  mutate(geo=case_when(
    geo=="FRA" ~ "FR",
    geo=="DEU" ~ "DE",
    geo=="ESP" ~ "ES",
    geo=="ITA" ~ "IT"))
```

Merging data frames

We merge all the data frames with all the series, to chain and interpolate the special cases in the following section. The figure below shows the unchained series for France, Germany, Italy and Spain.

```
df <-
  bind_rows(wage_de_fr,wage_es_it,
            hours,
            gdp,
            conso,
            inves,
            defgdp,
            definves,
            pop_recent,pop_old,
            debt_old,
            lendingrate_recent,lendingrate_old,
            networth,
            shortrate,
            loans_nfc,
            totexp_recent_fr, totexp_old_it_de,
            totrev_recent_fr, totrev_old_it_de,
            intpay_recent_fr, intpay_old_it_de,
            pubcons_recent_fr, pubcons_old_it_de,
            pubinves_recent_fr, pubinves_old_it_de,
            tfs_recent_fr,tfs_old_it_de,
            world_demand,
            unempbenef_recent,unempbenef_old,
            neer,
            imports_exports_volume) %>%
  select(period,value,country=geo,var) %>%
  bind_rows(pubcons_recent_it_de_es, pubinves_recent_it_de_es,
tfs_recent_it_de_es, debt_recent,
            totexp_recent_it_de_es, intpay_recent_it_de_es,
totrev_recent_it_de_es) %>%
  na.omit() %>%
  filter(year(period)>=1991) %>%
  arrange(var, period)

ggplot(df,aes(period,value,colour=country))+
  geom_line()+
  facet_wrap(~var,ncol=3,scales = "free_y")+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + theme(strip.text=element_text(size=11)) +
  xlab(NULL) + ylab(NULL)+
  ggtitle("Unchained Series")
```





France: Chaining & Interpolating Data

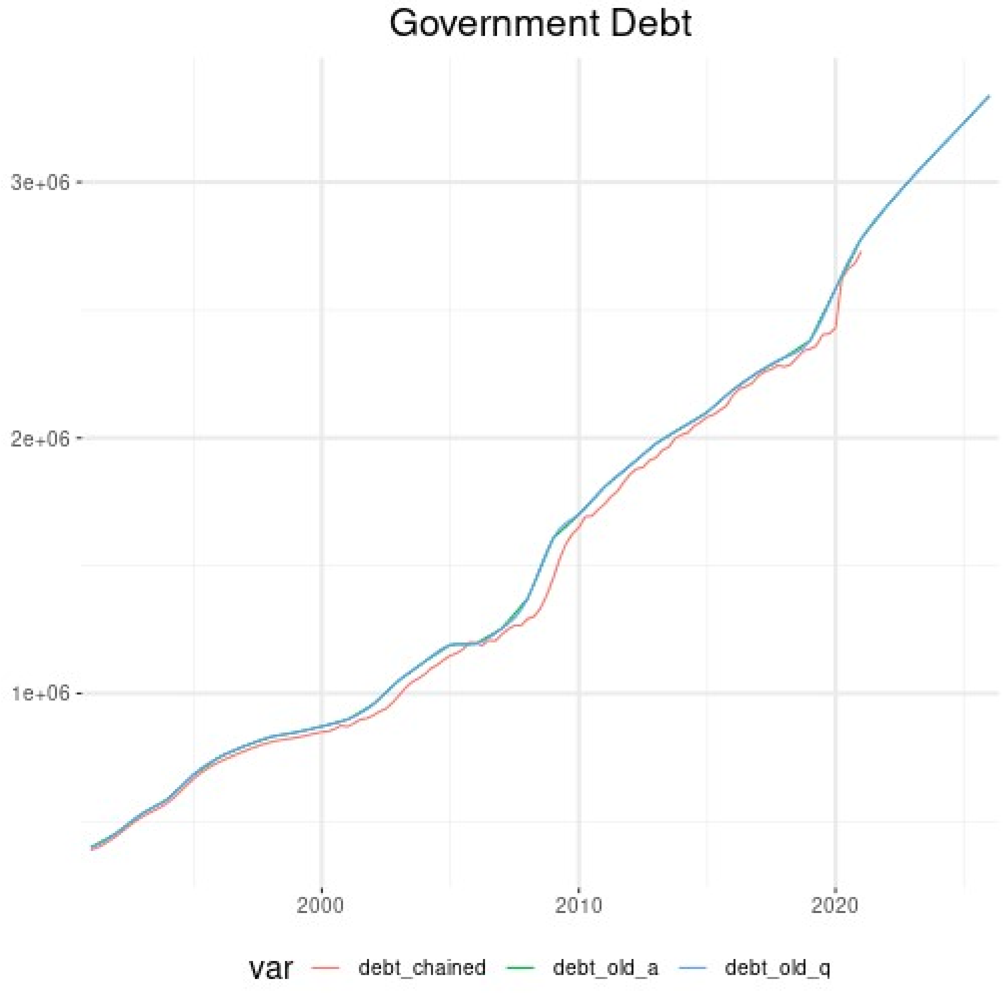
Before chaining and interpolating the special cases, we first gather all data for France, and then we proceed case by case.

```
df_fr <-
  df %>%
  filter(country=="FR") %>%
  select(-country)
```

Government Debt

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

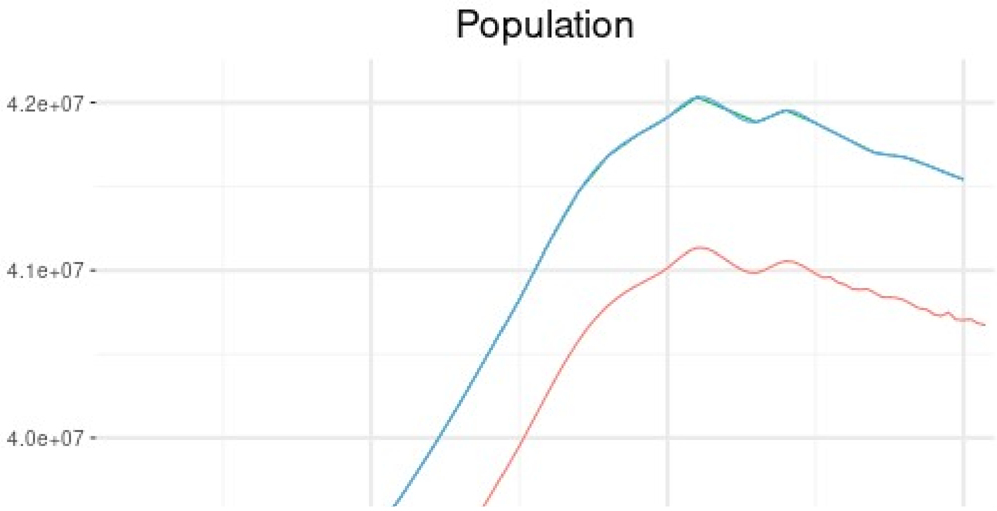
```
debt_old_a <-
  df_fr %>%
  filter(var=="debt_old") %>%
  mutate(value=1000*value) %>%
  select(-var)

debt_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(debt_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(debt_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="debt")

debt_recent <-
  df_fr %>%
  filter(var=="debt_recent") %>%
  mutate(var="debt")

minDateDebtRecent <- min(debt_recent$period)

debt <-
  chain(basis = debt_recent,
        to_rebase= debt_old_q,
        date_chain=minDateDebtRecent) %>%
  mutate(var="debt")

plot_df <- bind_rows(add_column(debt_old_a,var="debt_old_a"),
                     mutate(debt_old_q,var="debt_old_q"),
                     mutate(debt,var="debt_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Debt")
```

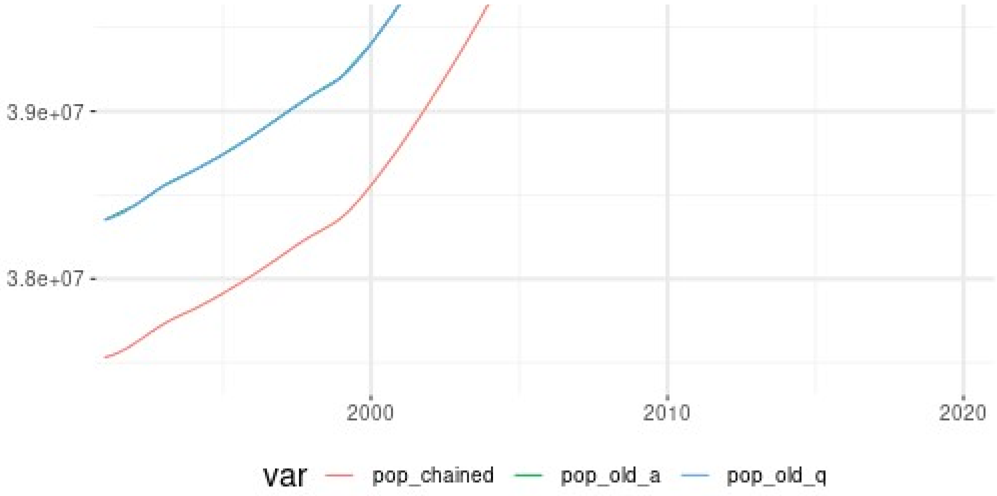


Population

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pop_old_a <-
  df_fr %>%
  filter(var=="pop_old")

pop_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(pop_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pop_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="pop")

pop_recent <-
  df_fr %>%
  filter(var=="pop_recent") %>%
  mutate(var="pop",value=value*1000)

pop <- chain(basis = pop_recent,
             to_rebase= pop_old_q,
             date_chain="2015-01-01")

plot_df <- bind_rows(mutate(pop_old_a,var="pop_old_a"),
                     mutate(pop_old_q,var="pop_old_q"),
                     mutate(pop,var="pop_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Population")
```



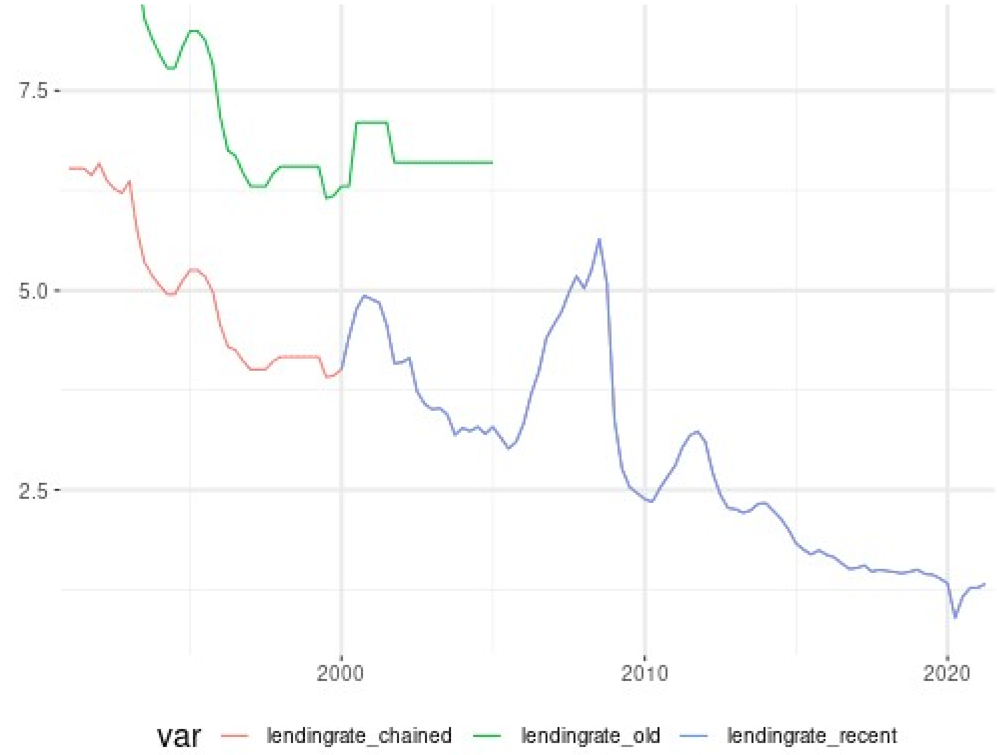


Lending Rate

We chain the quarterly series from the two different databases.

```
lendingrate_old <-
  df_fr %>%
  filter(var=="lendingrate_old") %>%
  mutate(var="lendingrate")

lendingrate_recent <-
  df_fr %>%
  filter(var=="lendingrate_recent") %>%
  mutate(var="lendingrate")

minDateLendingRateRecent <- min(lendingrate_recent$period)

lendingrate <-
  chain(basis = lendingrate_recent,
        to_rebase= lendingrate_old,
        date_chain=minDateLendingRateRecent) %>%
  mutate(var="lendingrate")

plot_df <- bind_rows(mutate(lendingrate_old,var="lendingrate_old"),
                     mutate(lendingrate_recent,var=
"lendingrate_recent"),
                     mutate(lendingrate,var="lendingrate_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Lending Rate")
```

Lending Rate
10.0

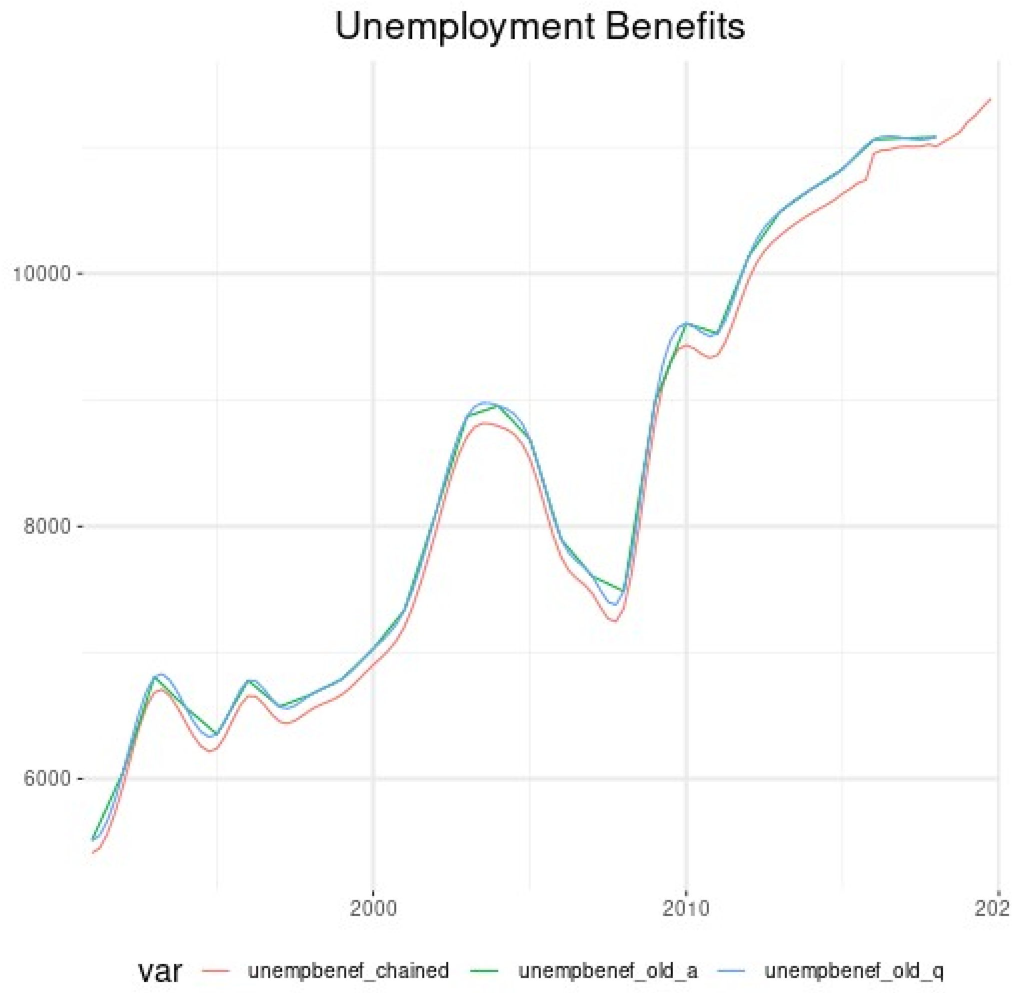


Unemployment Benefits

We first interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series.

```
unempbenef_old_a <-
  df_fr %>%
  filter(var=="unempbenef_old") %>%
  mutate(value=value/4)

unempbenef_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(unempbenef_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(unempbenef_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="unempbenef")

unempbenef_recent <-
  df_fr %>%
  filter(var=="unempbenef_recent") %>%
  mutate(var="unempbenef")

unempbenef <- chain(basis = unempbenef_recent,
                    to_rebase= unempbenef_old_q,
                    date_chain="2015-01-01")

plot_df <- bind_rows(mutate(unempbenef_old_a,var="unempbenef_old_a"),
                     mutate(unempbenef_old_q,var="unempbenef_old_q"),
                     mutate(unempbenef,var="unempbenef_chained"))
```

```
ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Unemployment Benefits")
```



Merging French Data

We gather all the final series in a dataframe.

```
FR_rawdata <-
  df_fr %>%
  filter(! var %in% c("lendingrate_old","lendingrate_recent",
                      "pop_old","pop_recent",
                      "debt_old","debt_recent",
                      "unempbenef_old","unempbenef_recent")) %>%
  bind_rows(lendingrate,pop,debt,unempbenef) %>%
  mutate(var=
           case_when(var=="pubcons_recent" ~ "pubcons",
                     var=="pubinves_recent" ~ "pubinves",
                     var=="tfs_recent" ~ "tfs",
                     var=="totexp_recent" ~ "totexp",
                     var=="intpay_recent" ~ "intpay",
                     var=="totrev_recent" ~ "totrev",
                     TRUE ~ var)) %>%
  spread(var,value) %>%
  add_column(country="FR")
```

Spain: Chaining & Interpolating Data

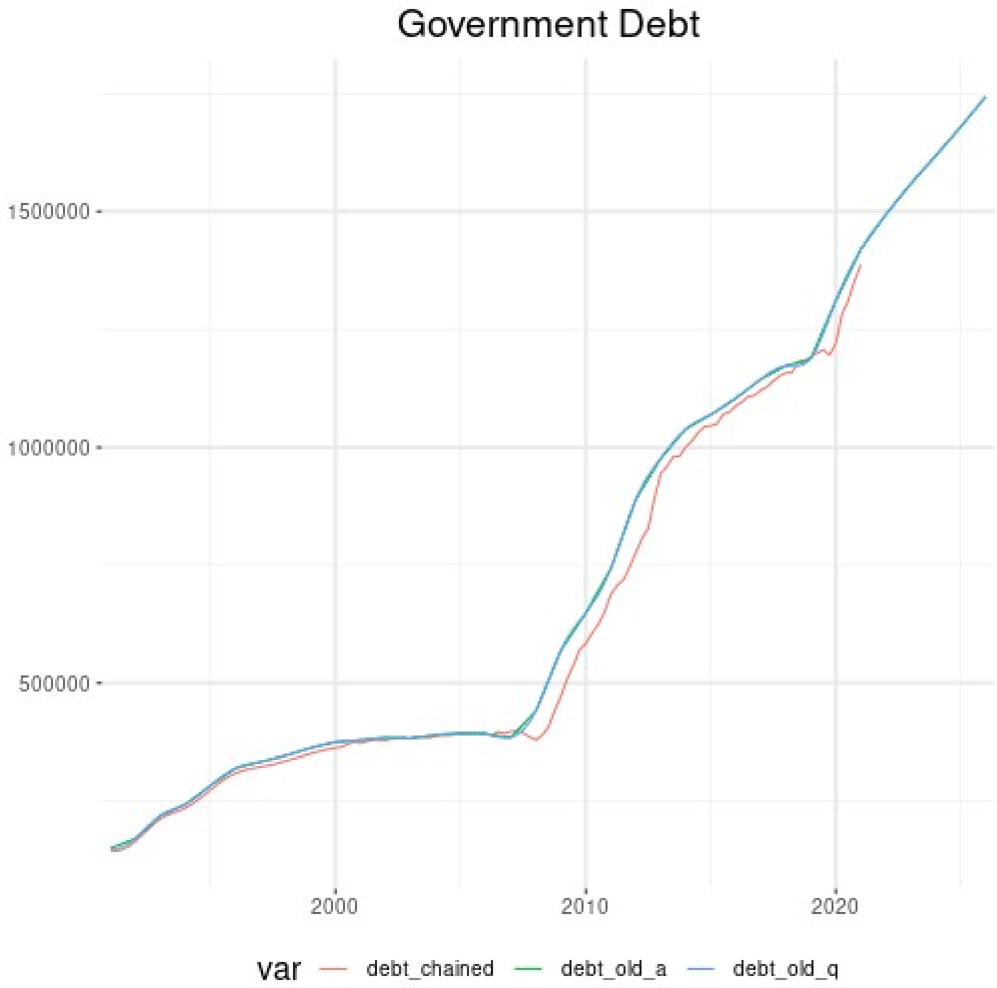
Before chaining and interpolating the special cases, we first gather all data for France, and then we proceed case by case.

```
df_es <-
  df %>%
  filter(country=="ES") %>%
  select(-country)
```

Government Debt

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
debt_old_a <-
  df_es %>%
  filter(var=="debt_old") %>%
  mutate(value=1000*value) %>%
  select(-var)

debt_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(debt_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(debt_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="debt")

debt_recent <-
  df_es %>%
  filter(var=="debt_recent") %>%
  mutate(var="debt")

minDateDebtRecent <- min(debt_recent$period)

debt <-
  chain(basis = debt_recent,
        to_rebase= debt_old_q,
        date_chain=minDateDebtRecent) %>%
  mutate(var="debt")

plot_df <- bind_rows(add_column(debt_old_a,var="debt_old_a"),
                     mutate(debt_old_q,var="debt_old_q"),
                     mutate(debt,var="debt_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Debt")
```

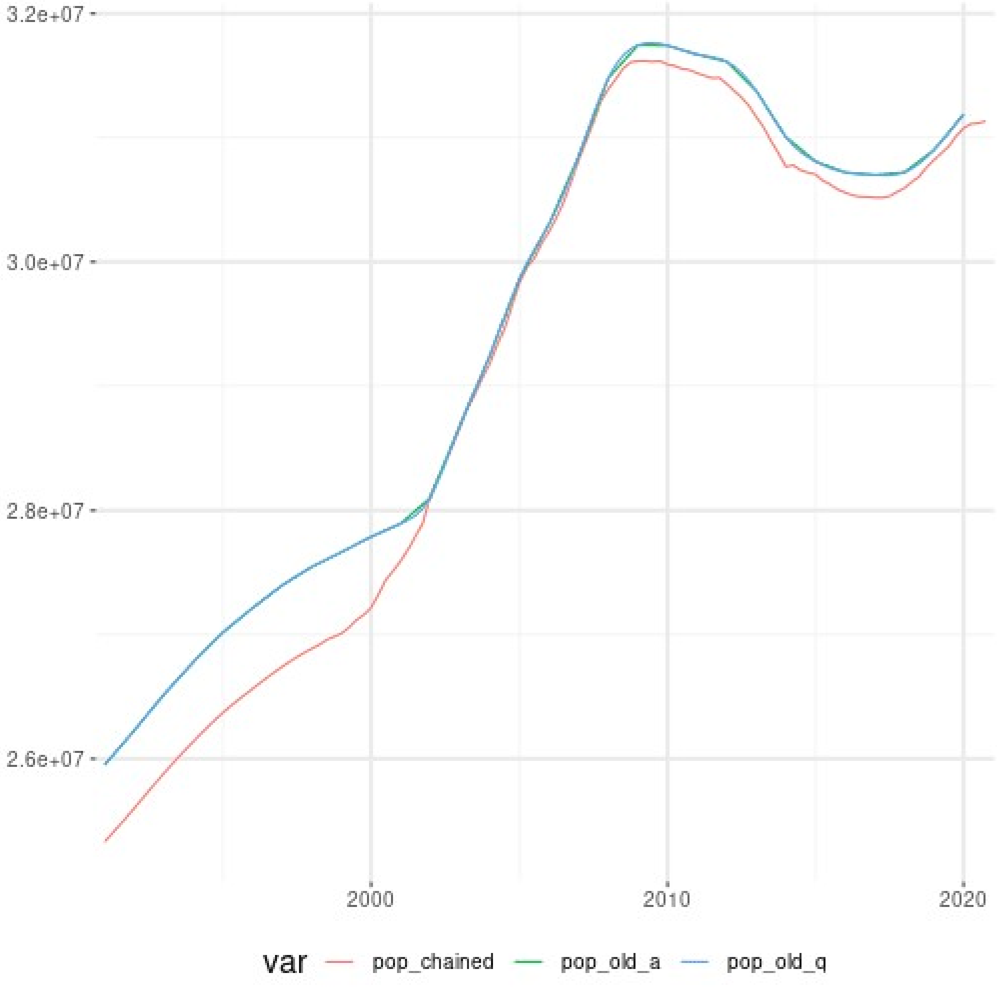


Population

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pop_old_a <-
  df_es %>%
  filter(var=="pop_old")

pop_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(pop_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pop_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="pop")

pop_recent <-
```

```
  df_es %>%
  filter(var=="pop_recent") %>%
  mutate(var="pop",value=value*1000)

pop <- chain(basis = pop_recent,
             to_rebase= pop_old_q,
             date_chain="1998-01-01")

plot_df <- bind_rows(mutate(pop_old_a,var="pop_old_a"),
                     mutate(pop_old_q,var="pop_old_q"),
                     mutate(pop,var="pop_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)
```

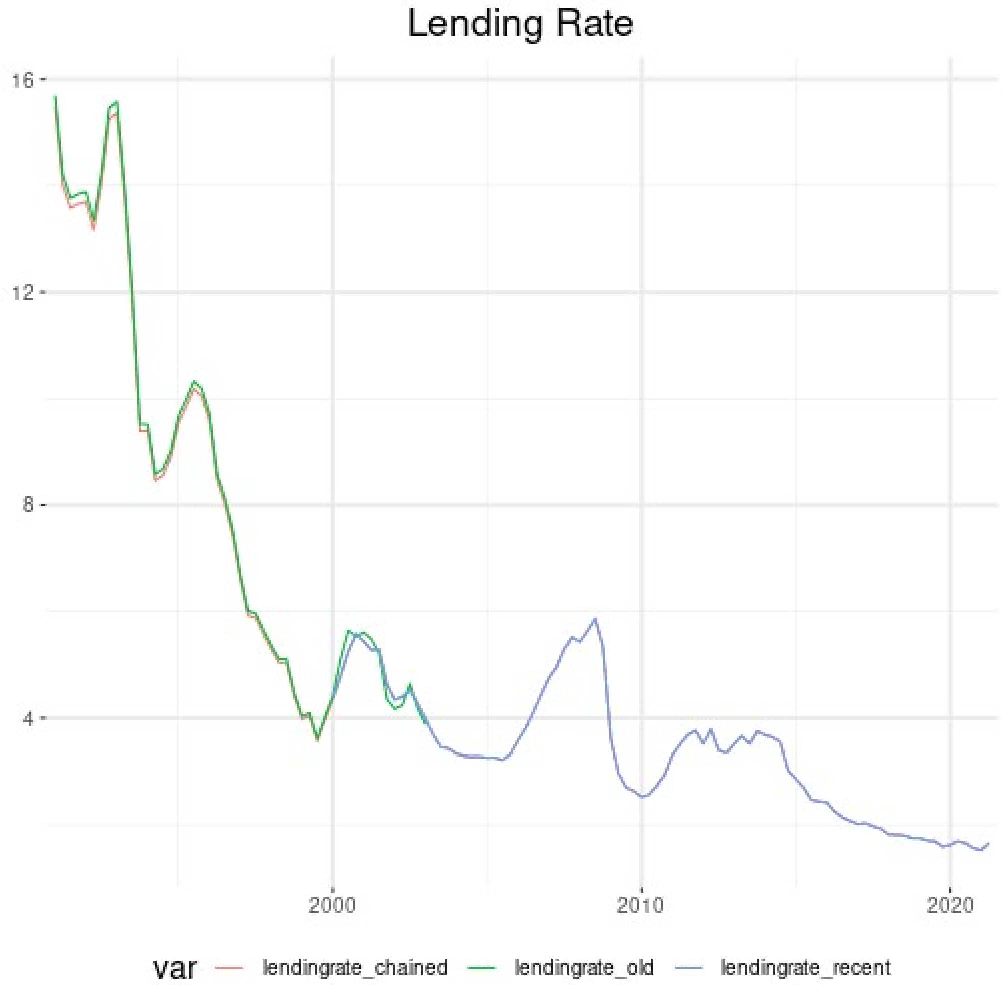


Lending Rate

We chain the quarterly series from the two different databases.

```
lendingrate_old <-
  df_es %>%
  filter(var=="lendingrate_old") %>%
  mutate(var="lendingrate")
```

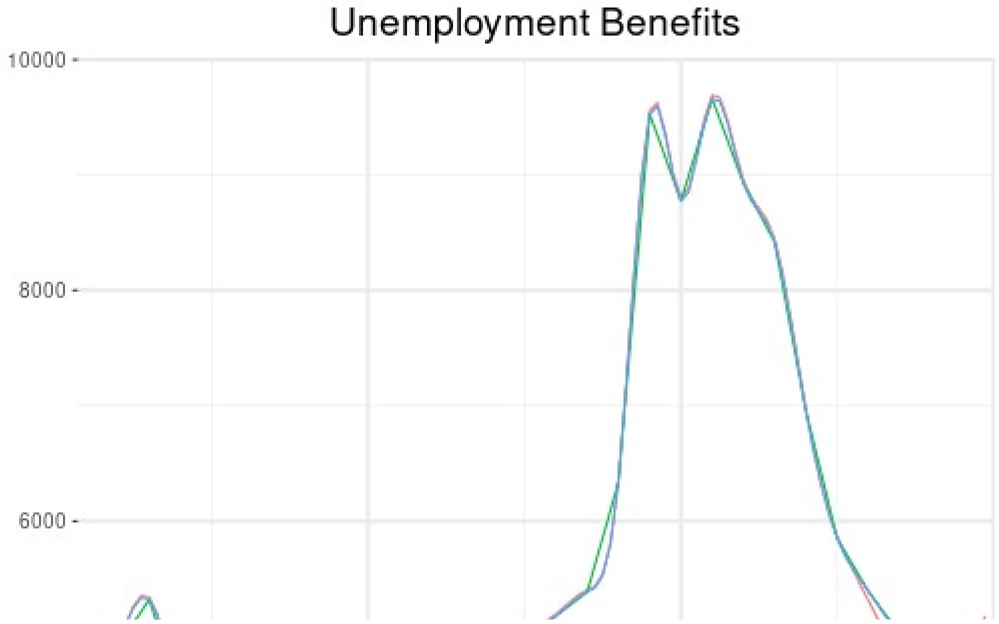
```
lendingrate_recent <-
  df_es %>%
  filter(var=="lendingrate_recent") %>%
  mutate(var="lendingrate")

minDateLendingRateRecent <- min(lendingrate_recent$period)

lendingrate <-
  chain(basis = lendingrate_recent,
        to_rebase= lendingrate_old,
        date_chain=minDateLendingRateRecent) %>%
  mutate(var="lendingrate")

plot_df <- bind_rows(mutate(lendingrate_old,var="lendingrate_old"),
                     mutate(lendingrate_recent,var=
"lendingrate_recent"),
                     mutate(lendingrate,var="lendingrate_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Lending Rate")
```

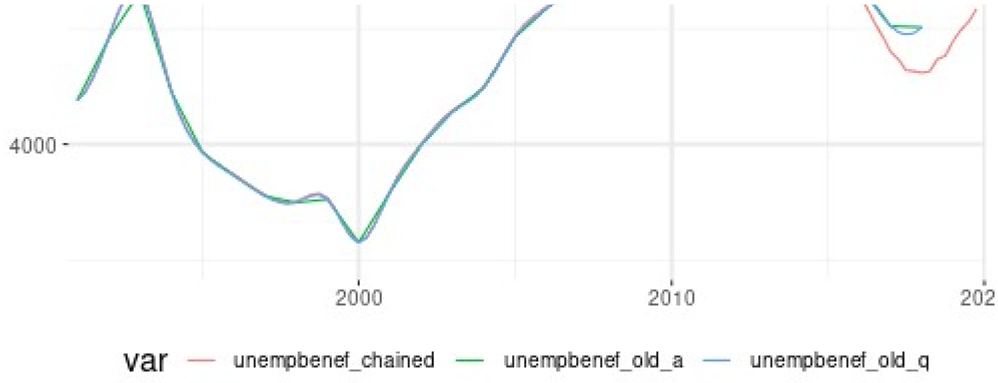


Unemployment Benefits

We first interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series.

```
unempbenef_old_a <-
  df_es %>%
  filter(var=="unempbenef_old") %>%
  mutate(value=value/4)

unempbenef_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(unempbenef_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(unempbenef_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="unempbenef")

unempbenef_recent <-
  df_es %>%
  filter(var=="unempbenef_recent") %>%
  mutate(var="unempbenef")

unempbenef <- chain(basis = unempbenef_recent,
                    to_rebase= unempbenef_old_q,
                    date_chain="2015-01-01")

plot_df <- bind_rows(mutate(unempbenef_old_a,var="unempbenef_old_a"),
                     mutate(unempbenef_old_q,var="unempbenef_old_q"),
                     mutate(unempbenef,var="unempbenef_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Unemployment Benefits")
```





Merging Spanish Data

We gather all the final series in a dataframe.

```
ES_rawdata <-
  df_es %>%
  filter(! var %in% c("lendingrate_old","lendingrate_recent",
                      "pop_old","pop_recent",
                      "debt_old","debt_recent",
                      "unempbenef_old","unempbenef_recent")) %>%
  bind_rows(lendingrate,pop,debt,unempbenef) %>%
  mutate(var=
           case_when(var=="pubcons_recent" ~ "pubcons",
                     var=="pubinves_recent" ~ "pubinves",
                     var=="tfs_recent" ~ "tfs",
                     var=="totexp_recent" ~ "totexp",
                     var=="totrev_recent" ~ "totrev",
                     var=="intpay_recent" ~ "intpay",
                     TRUE ~ var)) %>%
  spread(var,value) %>%
  add_column(country="ES")
```

Germany: Chaining & Interpolating Data

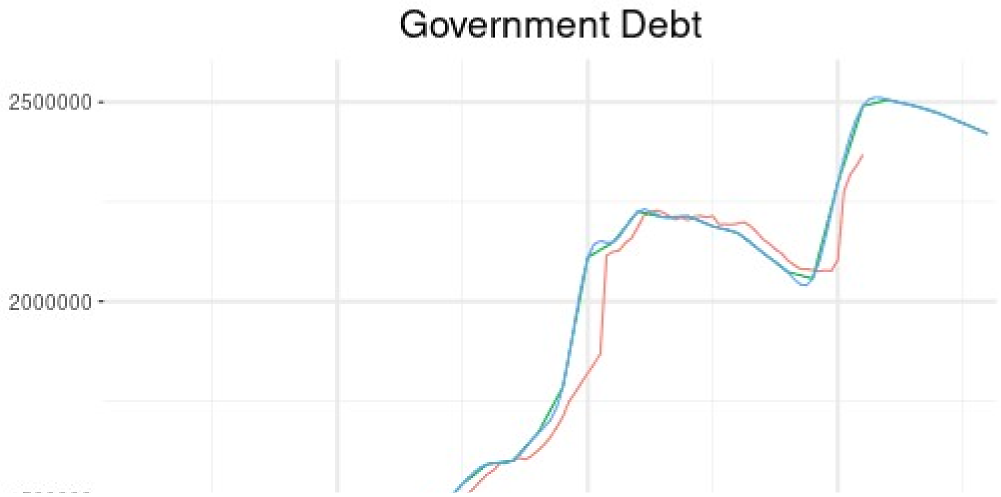
Before chaining and interpolating the special cases, we first gather all data for France, and then we proceed case by case.

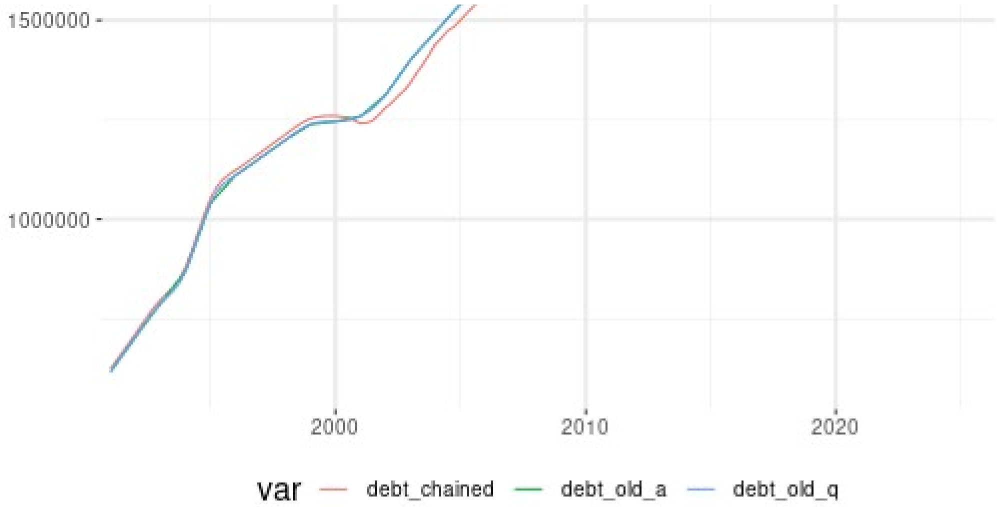
```
df_de <-
  df %>%
  filter(country=="DE") %>%
  select(-country)
```

Government Debt

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
debt_old_a <-
  df_de %>%
  filter(var=="debt_old") %>%
  mutate(value=1000*value) %>%
  select(-var)

debt_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(debt_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(debt_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="debt")

debt_recent <-
  df_de %>%
  filter(var=="debt_recent") %>%
  mutate(var="debt")

minDateDebtRecent <- min(debt_recent$period)

debt <-
  chain(basis = debt_recent,
        to_rebase= debt_old_q,
        date_chain=minDateDebtRecent) %>%
  mutate(var="debt")

plot_df <- bind_rows(add_column(debt_old_a,var="debt_old_a"),
                     mutate(debt_old_q,var="debt_old_q"),
                     mutate(debt,var="debt_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Debt")
```

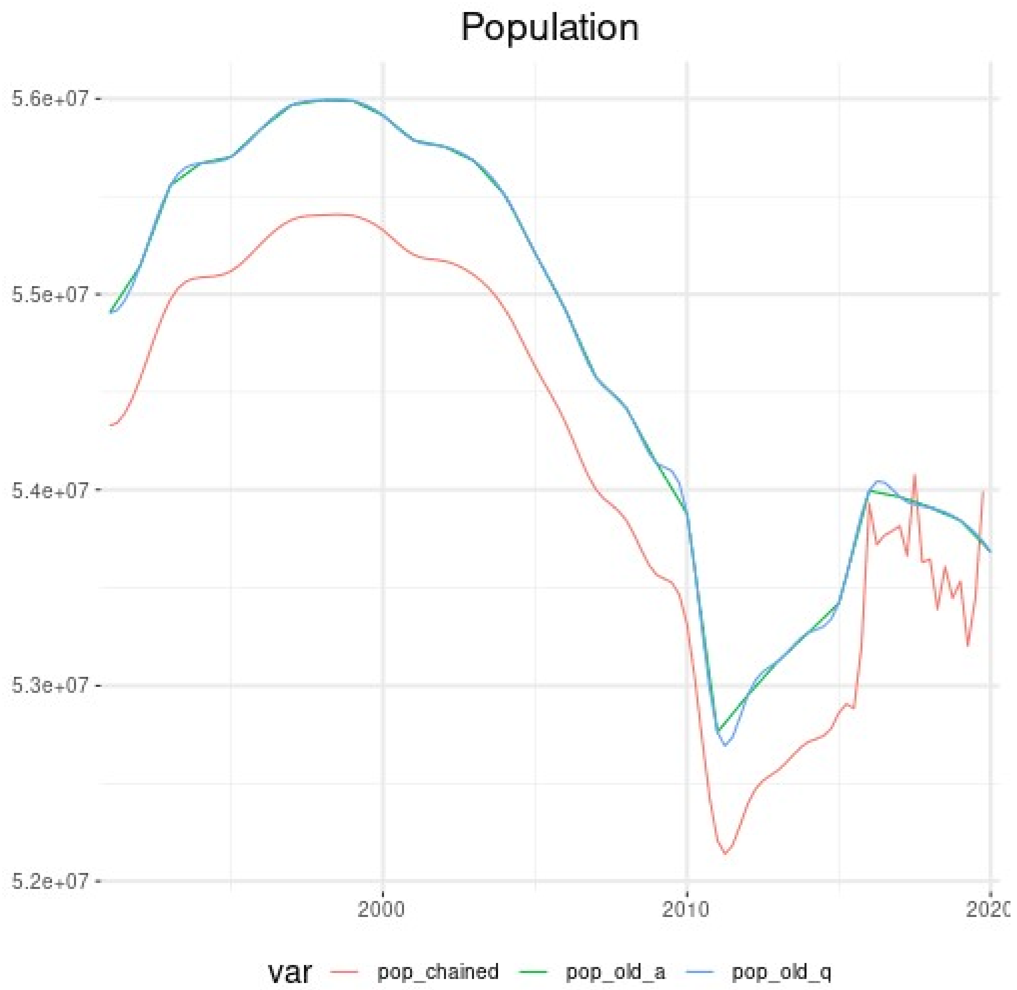




Population

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pop_old_a <-
  df_de %>%
  filter(var=="pop_old")

pop_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(pop_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pop_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="pop")

pop_recent <-
  df_de %>%
  filter(var=="pop_recent") %>%
  mutate(var="pop",value=value*1000)

pop <- chain(basis = pop_recent,
             to_rebase= pop_old_q,
             date_chain="2015-01-01")

plot_df <- bind_rows(mutate(pop_old_a,var="pop_old_a"),
                     mutate(pop_old_q,var="pop_old_q"),
                     mutate(pop,var="pop_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Population")
```

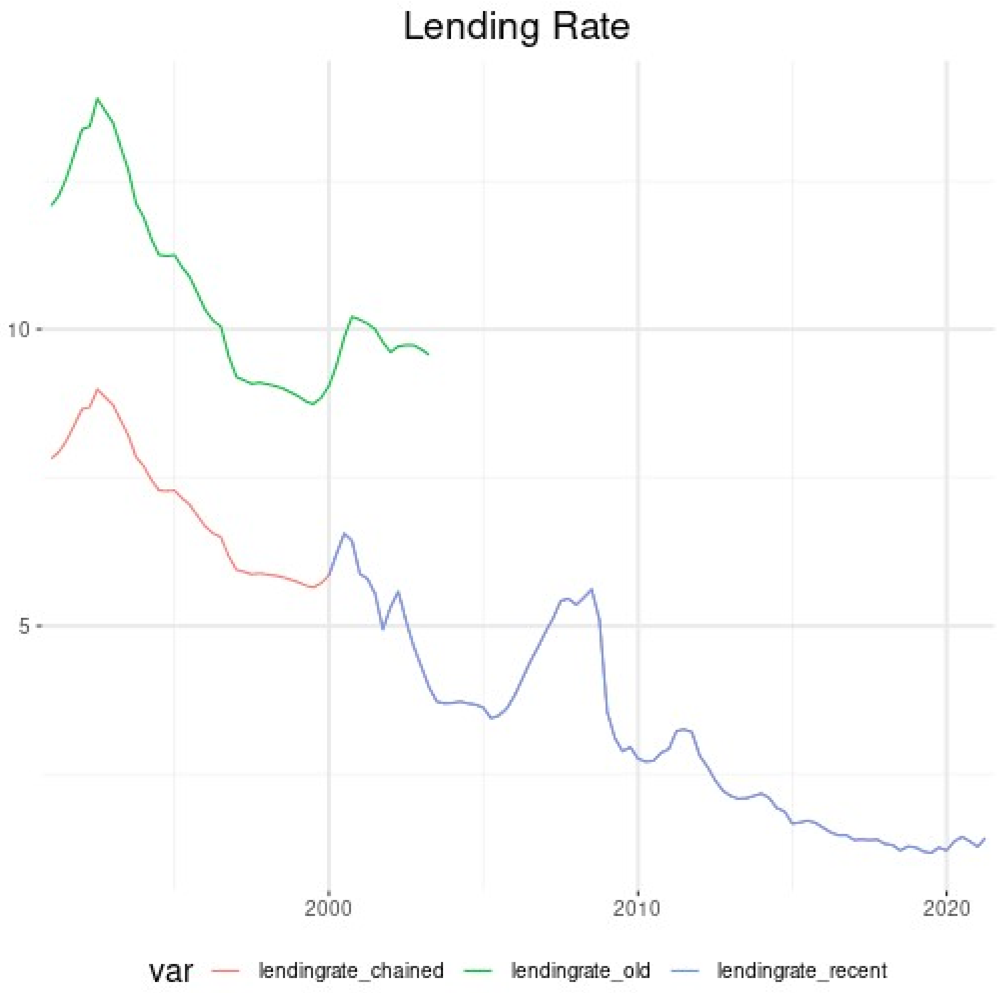


Lending Rate

We chain the quarterly series from the two different databases.

```
lendingrate_old <-
  df_de %>%
  filter(var=="lendingrate_old") %>%
  mutate(var="lendingrate")

lendingrate_recent <-
  df_de %>%
  filter(var=="lendingrate_recent") %>%
  mutate(var="lendingrate")

minDateLendingRateRecent <- min(lendingrate_recent$period)

lendingrate <-
  chain(basis = lendingrate_recent,
        to_rebase= lendingrate_old,
        date_chain=minDateLendingRateRecent) %>%
  mutate(var="lendingrate")

plot_df <- bind_rows(mutate(lendingrate_old,var="lendingrate_old"),
                     mutate(lendingrate_recent,var=
"lendingrate_recent"),
                     mutate(lendingrate,var="lendingrate_chained"))
```

```
ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Lending Rate")
```

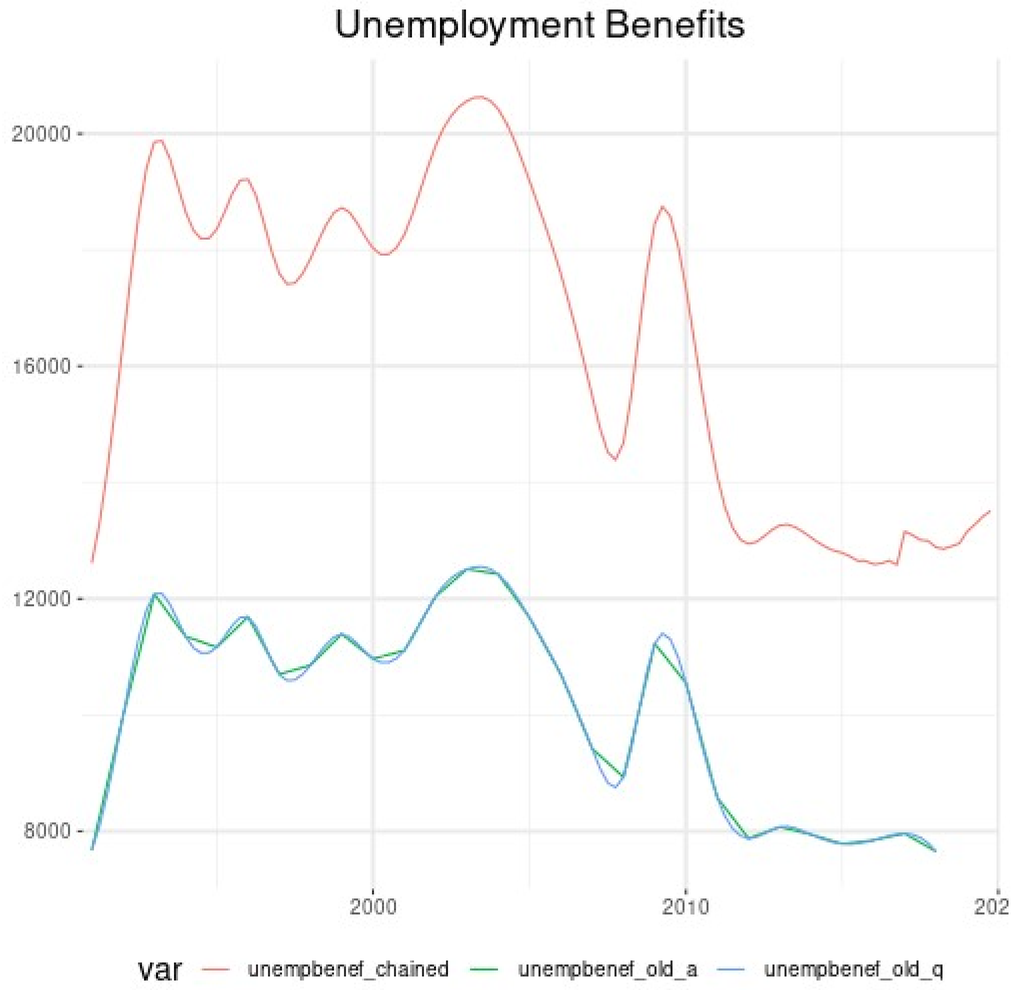


Unemployment Benefits

We first interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series.

```
unempbenef_old_a <-
  df_de %>%
  filter(var=="unempbenef_old") %>%
  mutate(value=value/4)

unempbenef_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(unempbenef_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(unempbenef_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="unempbenef")
```

```
unempbenef_recent <-
  df_de %>%
  filter(var=="unempbenef_recent") %>%
  mutate(var="unempbenef")

unempbenef <- chain(basis = unempbenef_recent,
                    to_rebase= unempbenef_old_q,
                    date_chain="2015-01-01")

plot_df <- bind_rows(mutate(unempbenef_old_a,var="unempbenef_old_a"),
                     mutate(unempbenef_old_q,var="unempbenef_old_q"),
                     mutate(unempbenef,var="unempbenef_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Unemployment Benefits")
```

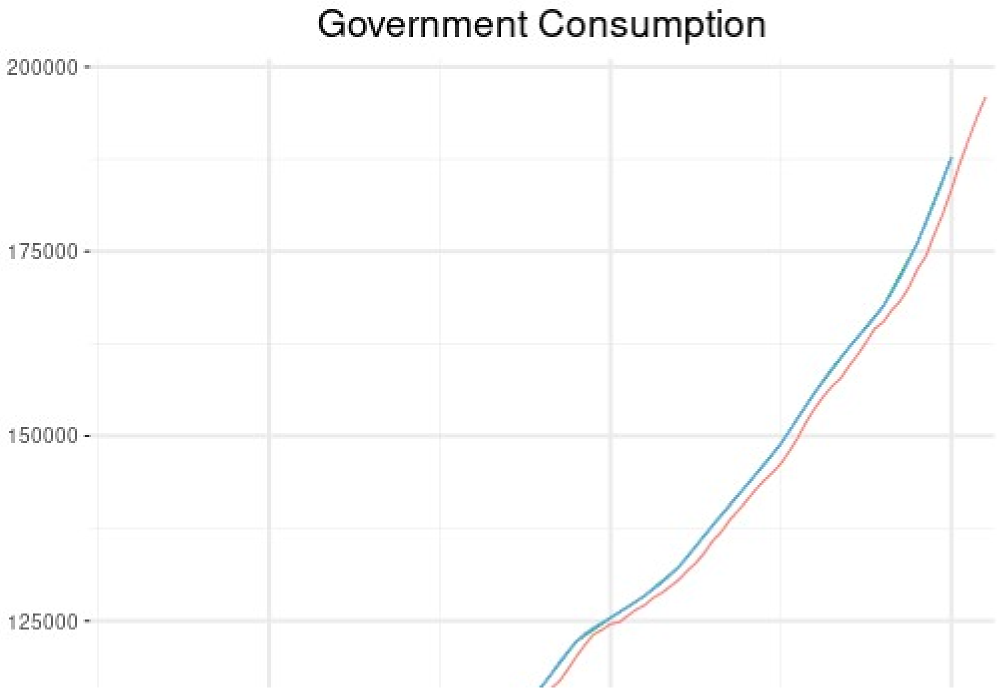


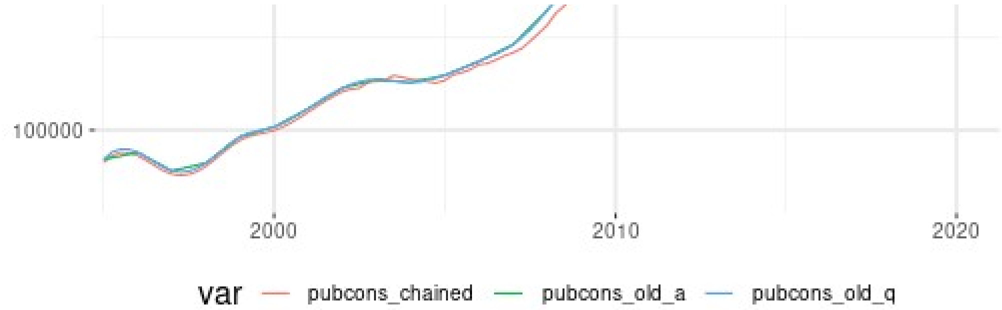
Government Consumption

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pubcons_old_a <-
```

```
  df_de %>%
  filter(var=="pubcons_old") %>%
  mutate(value=value/4)

pubcons_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(pubcons_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pubcons_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="pubcons")

pubcons_recent <-
  df_de %>%
  filter(var=="pubcons_recent") %>%
  mutate(var="pubcons")

minDatePubConsRecent <- min(pubcons_recent$period)

pubcons <- chain(basis = pubcons_recent,
                    to_rebase= pubcons_old_q,
                    date_chain=minDatePubConsRecent)

plot_df <- bind_rows(mutate(pubcons_old_a,var="pubcons_old_a"),
                     mutate(pubcons_old_q,var="pubcons_old_q"),
                     mutate(pubcons,var="pubcons_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Consumption")
```

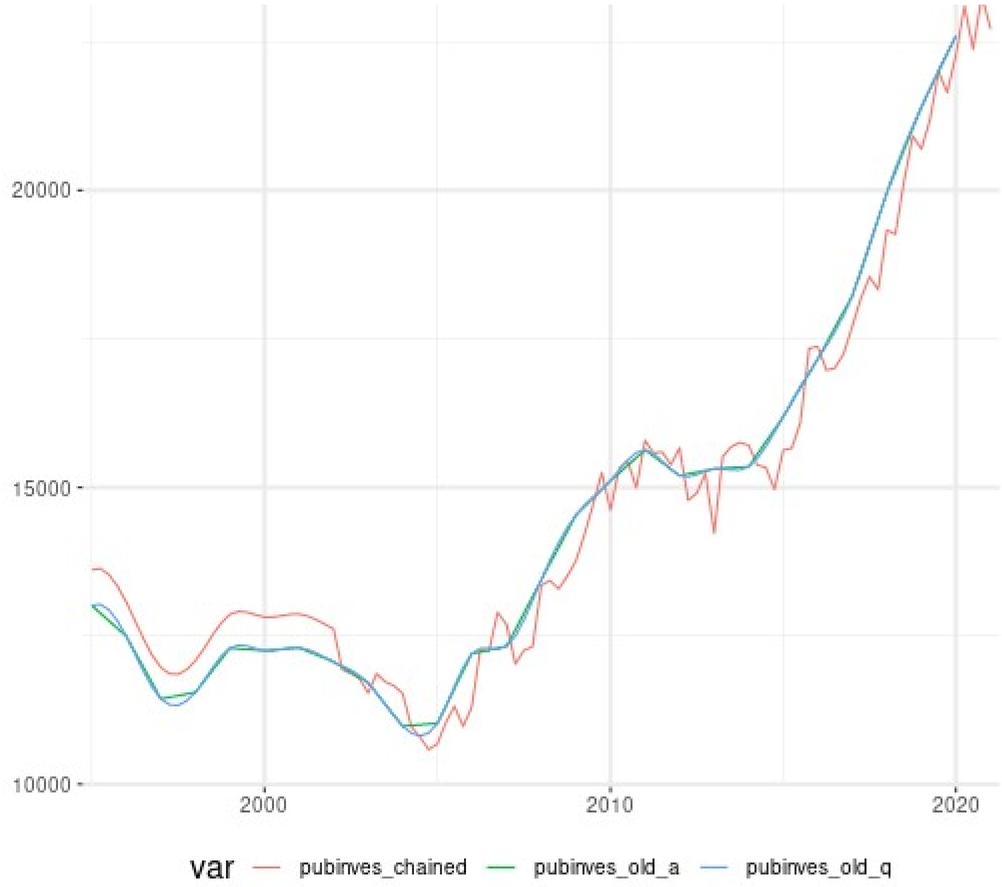




Government Investment

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

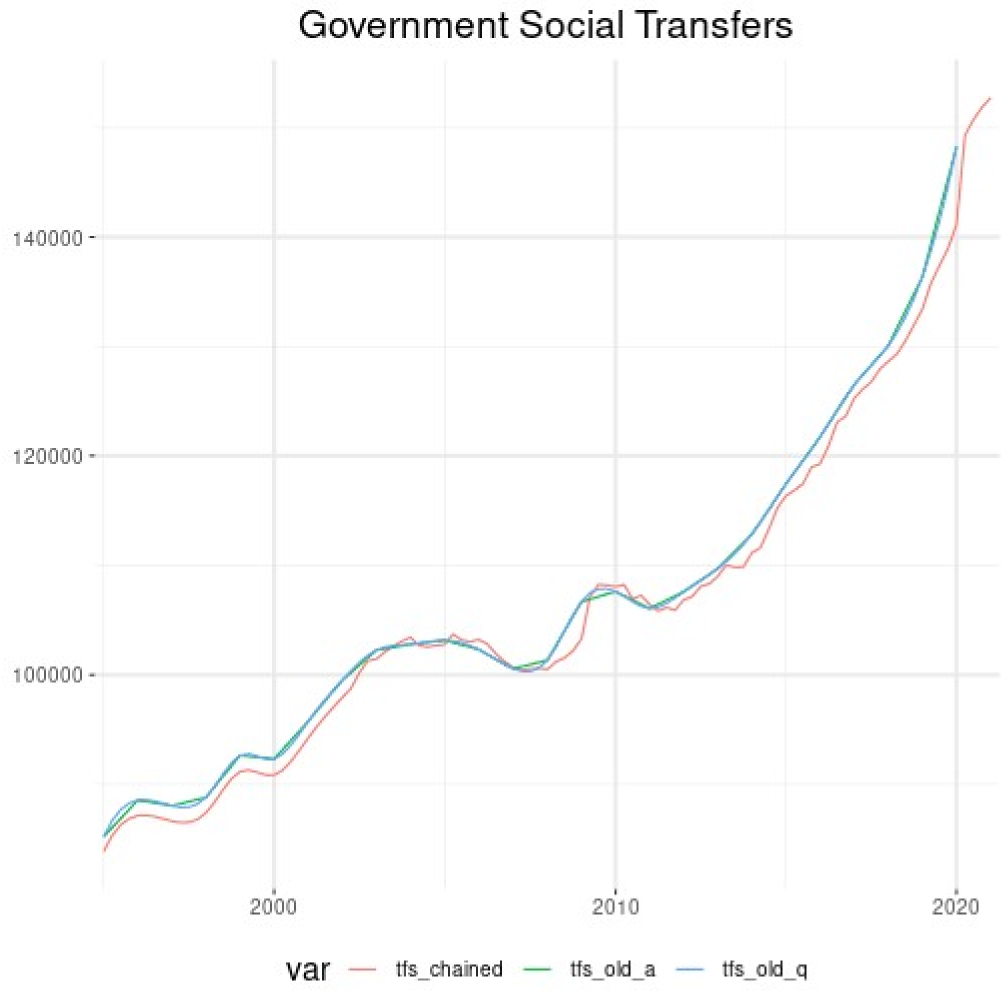
```
pubinves_old_a <-
  df_de %>%
  filter(var=="pubinves_old") %>%
  mutate(value=value/4)

pubinves_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(pubinves_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pubinves_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="pubinves")

pubinves_recent <-
  df_de %>%
  filter(var=="pubinves_recent") %>%
  mutate(var="pubinves")

minDatePubInvesRecent <- min(pubinves_recent$period)

pubinves <- chain(basis = pubinves_recent,
                    to_rebase= pubinves_old_q,
                    date_chain=minDatePubInvesRecent)

plot_df <- bind_rows(mutate(pubinves_old_a,var="pubinves_old_a"),
                     mutate(pubinves_old_q,var="pubinves_old_q"),
                     mutate(pubinves,var="pubinves_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Investment")
```



Government Social Transfers

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

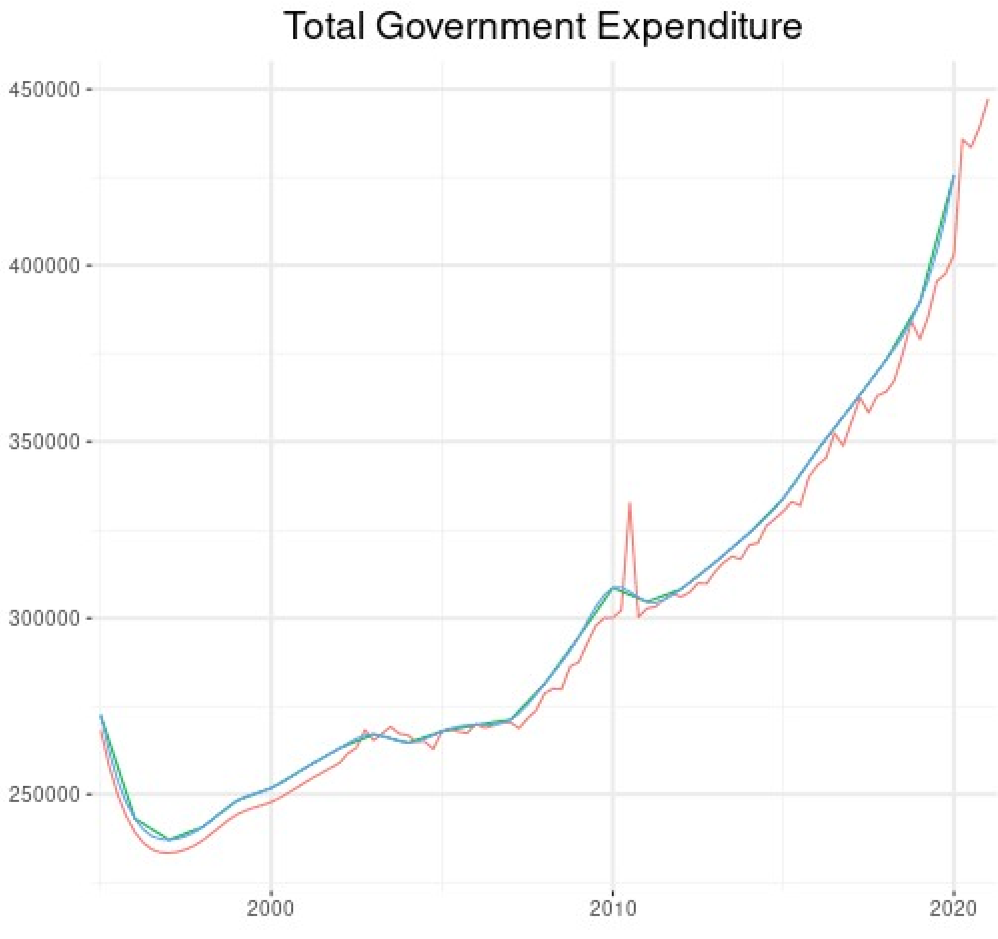
```
tfs_old_a <-
  df_de %>%
  filter(var=="tfs_old") %>%
  mutate(value=value/4)

tfs_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(tfs_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(tfs_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="tfs")

tfs_recent <-
  df_de %>%
  filter(var=="tfs_recent") %>%
  mutate(var="tfs")

minDateTfsRecent <- min(tfs_recent$period)

tfs <- chain(basis = tfs_recent,
                    to_rebase= tfs_old_q,
                    date_chain=minDateTfsRecent)

plot_df <- bind_rows(mutate(tfs_old_a,var="tfs_old_a"),
                     mutate(tfs_old_q,var="tfs_old_q"),
                     mutate(tfs,var="tfs_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Social Transfers")
```



Total Government Expenditure

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

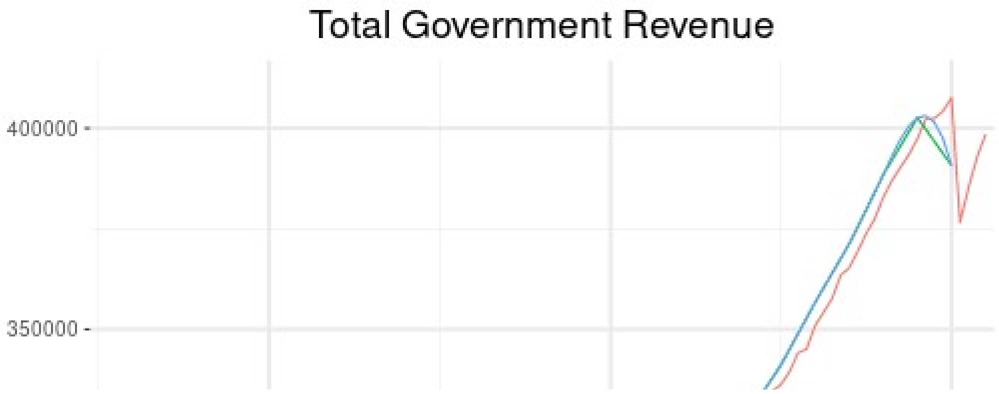
```
totexp_old_a <-
  df_de %>%
  filter(var=="totexp_old") %>%
  mutate(value=value/4)

totexp_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(totexp_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(totexp_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="totexp")

totexp_recent <-
  df_de %>%
  filter(var=="totexp_recent") %>%
  mutate(var="totexp")

minDateTotExpRecent <- min(totexp_recent$period)

totexp <- chain(basis = totexp_recent,
                    to_rebase= totexp_old_q,
                    date_chain=minDateTotExpRecent)

plot_df <- bind_rows(mutate(totexp_old_a,var="totexp_old_a"),
                     mutate(totexp_old_q,var="totexp_old_q"),
                     mutate(totexp,var="totexp_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Total Government Expenditure")
```

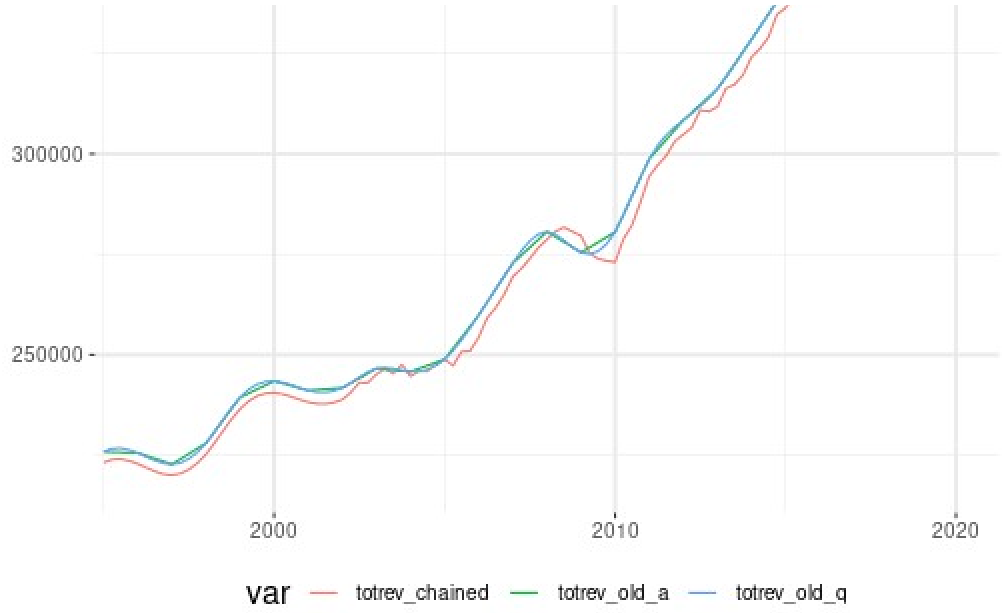


Total Government Revenue

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
totrev_old_a <-
  df_de %>%
  filter(var=="totrev_old") %>%
  mutate(value=value/4)

totrev_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(totrev_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(totrev_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="totrev")

totrev_recent <-
  df_de %>%
  filter(var=="totrev_recent") %>%
  mutate(var="totrev")

minDateTotRevRecent <- min(totrev_recent$period)

totrev <- chain(basis = totrev_recent,
                    to_rebase= totrev_old_q,
                    date_chain=minDateTotRevRecent)

plot_df <- bind_rows(mutate(totrev_old_a,var="totrev_old_a"),
                     mutate(totrev_old_q,var="totrev_old_q"),
                     mutate(totrev,var="totrev_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Total Government Revenue")
```



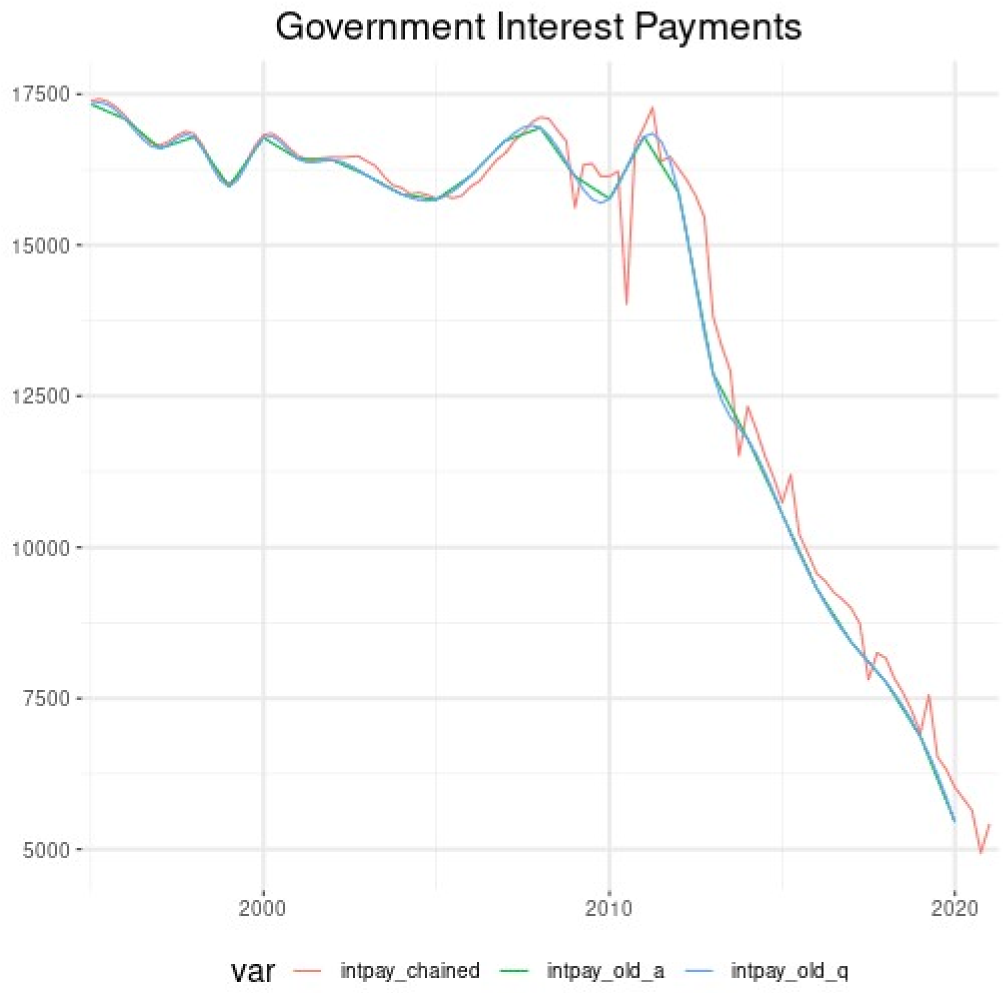


Government Interest Payments

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
intpay_old_a <-
  df_de %>%
  filter(var=="intpay_old") %>%
  mutate(value=value/4)

intpay_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(intpay_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(intpay_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="intpay")

intpay_recent <-
  df_de %>%
  filter(var=="intpay_recent") %>%
  mutate(var="intpay")

minDateIntPayRecent <- min(intpay_recent$period)

intpay <- chain(basis = intpay_recent,
                    to_rebase= intpay_old_q,
                    date_chain=minDateIntPayRecent)

plot_df <- bind_rows(mutate(intpay_old_a,var="intpay_old_a"),
                     mutate(intpay_old_q,var="intpay_old_q"),
                     mutate(intpay,var="intpay_chained"))
```

```
ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Interest Payments")
```



Merging German Data

We gather all the final series in a dataframe.

```
DE_rawdata <-
  df_de %>%
  filter(! var %in% c("lendingrate_old","lendingrate_recent",
                      "pop_old","pop_recent",
                      "debt_old","debt_recent",
                      "unempbenef_old","unempbenef_recent",
                      "totexp_recent", "totexp_old",
                      "intpay_recent", "intpay_old",
                      "totrev_recent", "totrev_old",
                      "pubcons_recent", "pubcons_old",
                      "pubinves_recent","pubinves_old",
                      "tfs_recent","tfs_old")) %>%
  bind_rows(lendingrate,pop,debt,unempbenef,totexp,totrev,
intpay,pubcons,pubinves,tfs) %>%
  spread(var,value) %>%
  add_column(country="DE")
```

Italy: Chaining & Interpolating Data

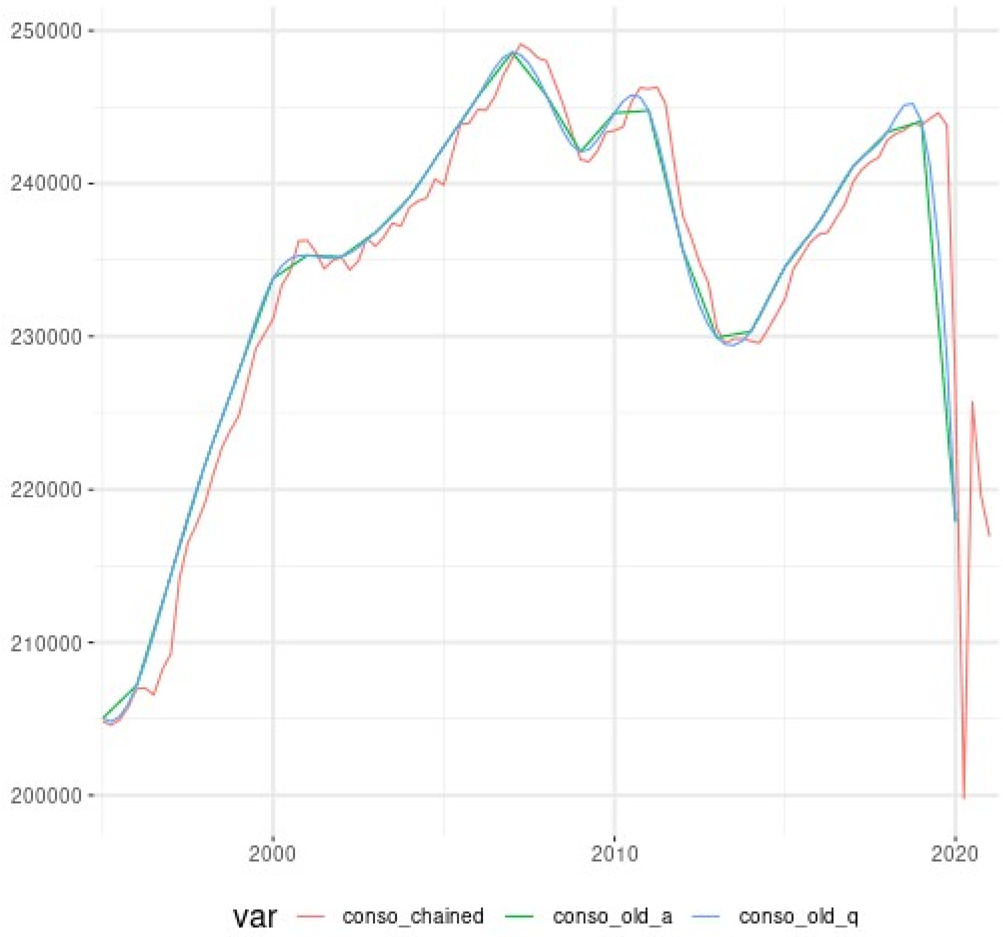
Before chaining and interpolating the special cases, we first gather all data for France, and then we proceed case by case.

```
df_it <-
  df %>%
  filter(country=="IT") %>%
  select(-country)
```

Consumption

Quarterly data on consumption is not available for Italy before 1996, but there is annual data. We interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series in 1996-01-01.

```
conso_old_a <-
  rdb("Eurostat","nama_10_gdp",mask = "A.CLV10_MEUR.P31_S14_S15.IT") %>%
  select(period, value) %>%
  add_column(var="conso_old") %>%
  mutate(value=value/4)

conso_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(conso_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(conso_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="conso")

conso_recent <-
  df_it %>%
  filter(var=="conso")

minDateConsoRecent <- min(conso_recent$period)

conso<- chain(basis = conso_recent,
                    to_rebase= conso_old_q,
                    date_chain=minDateConsoRecent)

plot_df <- bind_rows(mutate(conso_old_a,var="conso_old_a"),
                     mutate(conso_old_q,var="conso_old_q"),
                     mutate(conso,var="conso_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Consumption")
```

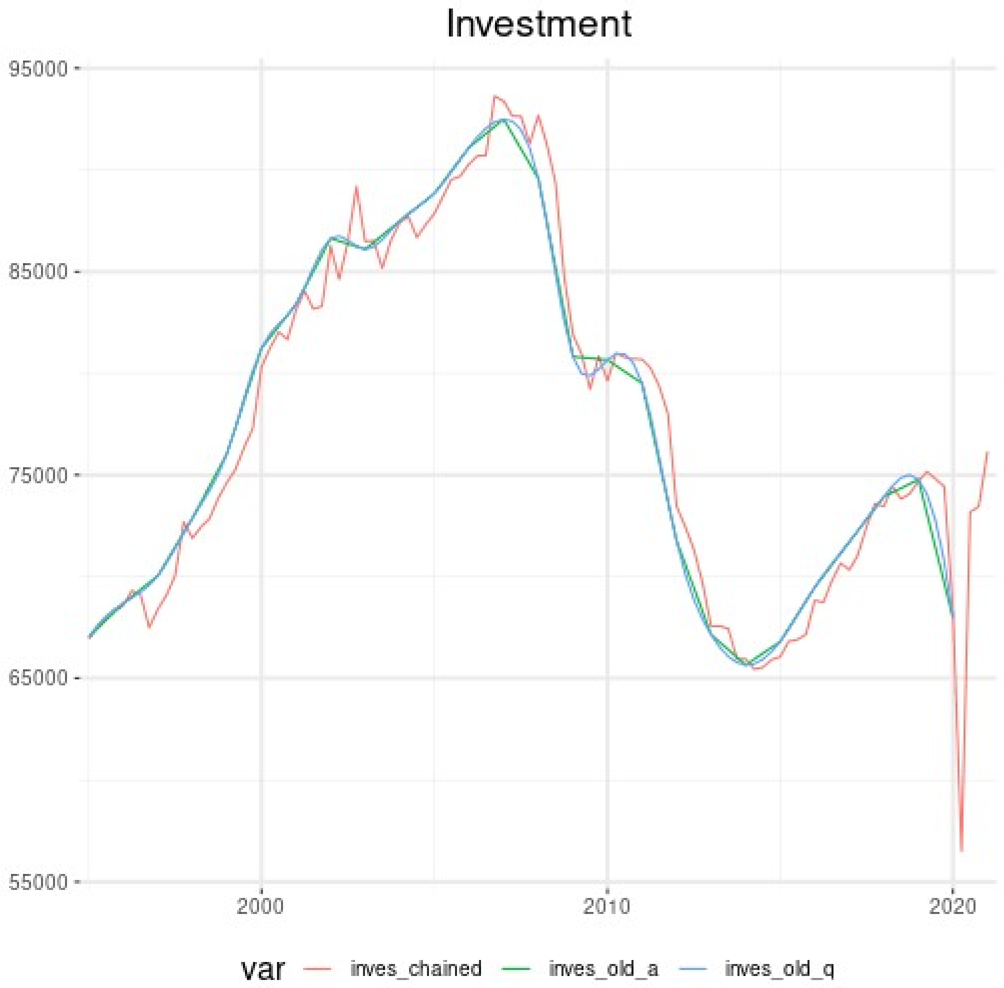


Investment

Quarterly data on investment is not available for Italy before 1996, but there is annual data. We interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series in 1996-01-01.

```
inves_old_a <-
  rdb("Eurostat","nama_10_gdp",mask = "A.CLV10_MEUR.P51G.IT") %>%
  select(period, value) %>%
  add_column(var="inves_old") %>%
  mutate(value=value/4)

inves_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(inves_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(inves_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="inves")

inves_recent <-
  df_it %>%
  filter(var=="inves")

minDateInvesRecent <- min(inves_recent$period)
```

```
inves<- chain(basis = inves_recent,
                    to_rebase= inves_old_q,
                    date_chain=minDateInvesRecent)

plot_df <- bind_rows(mutate(inves_old_a,var="inves_old_a"),
                     mutate(inves_old_q,var="inves_old_q"),
                     mutate(inves,var="inves_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Investment")
```

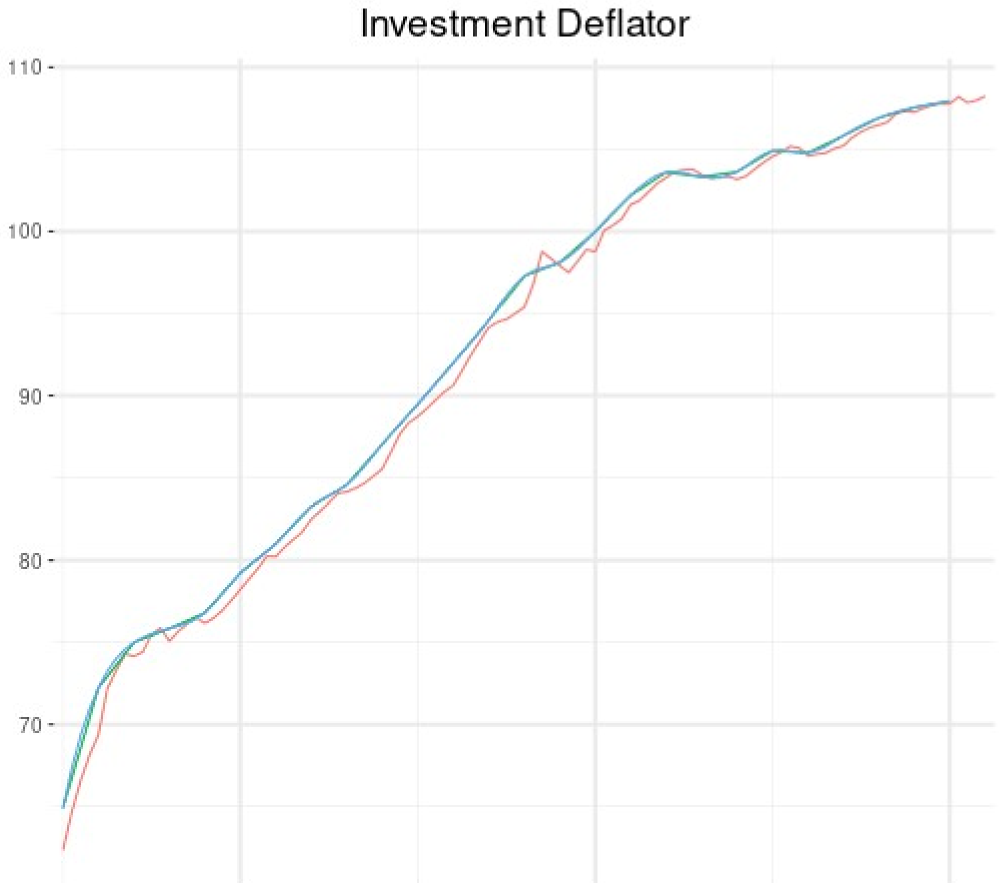


Investment Deflator

Quarterly data on investment deflator is not available for Italy before 1996, but there is annual data. We interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series in 1996-01-01.

```
definves_old_a <-
  rdb("Eurostat","nama_10_gdp",mask = "A.PD10_EUR.P51G.IT")%>%
  select(period, value) %>%
  add_column(var="definves_old")
```

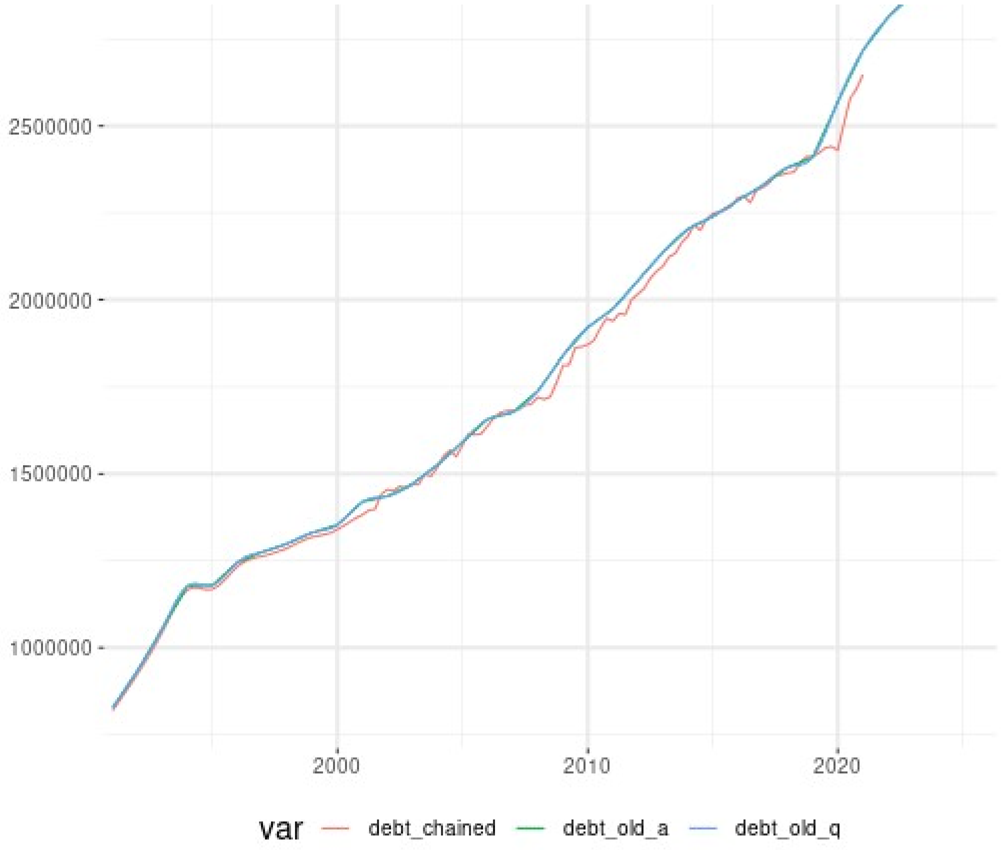
```
definves_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(definves_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(definves_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="definves")

definves_recent <-
  df_it %>%
  filter(var=="definves")

minDateDefInvesRecent <- min(definves_recent$period)

definves<- chain(basis = definves_recent,
                    to_rebase= definves_old_q,
                    date_chain=minDateDefInvesRecent)

plot_df <- bind_rows(mutate(definves_old_a,var="definves_old_a"),
                     mutate(definves_old_q,var="definves_old_q"),
                     mutate(definves,var="definves_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Investment Deflator")
```



Government Debt

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
debt_old_a <-
  df_it %>%
  filter(var=="debt_old") %>%
  mutate(value=1000*value) %>%
  select(-var)

debt_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(debt_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(debt_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="debt")

debt_recent <-
  df_it %>%
  filter(var=="debt_recent") %>%
  mutate(var="debt")

minDateDebtRecent <- min(debt_recent$period)

debt <-
  chain(basis = debt_recent,
        to_rebase= debt_old_q,
        date_chain=minDateDebtRecent) %>%
  mutate(var="debt")

plot_df <- bind_rows(add_column(debt_old_a,var="debt_old_a"),
                     mutate(debt_old_q,var="debt_old_q"),
                     mutate(debt,var="debt_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Debt")
```

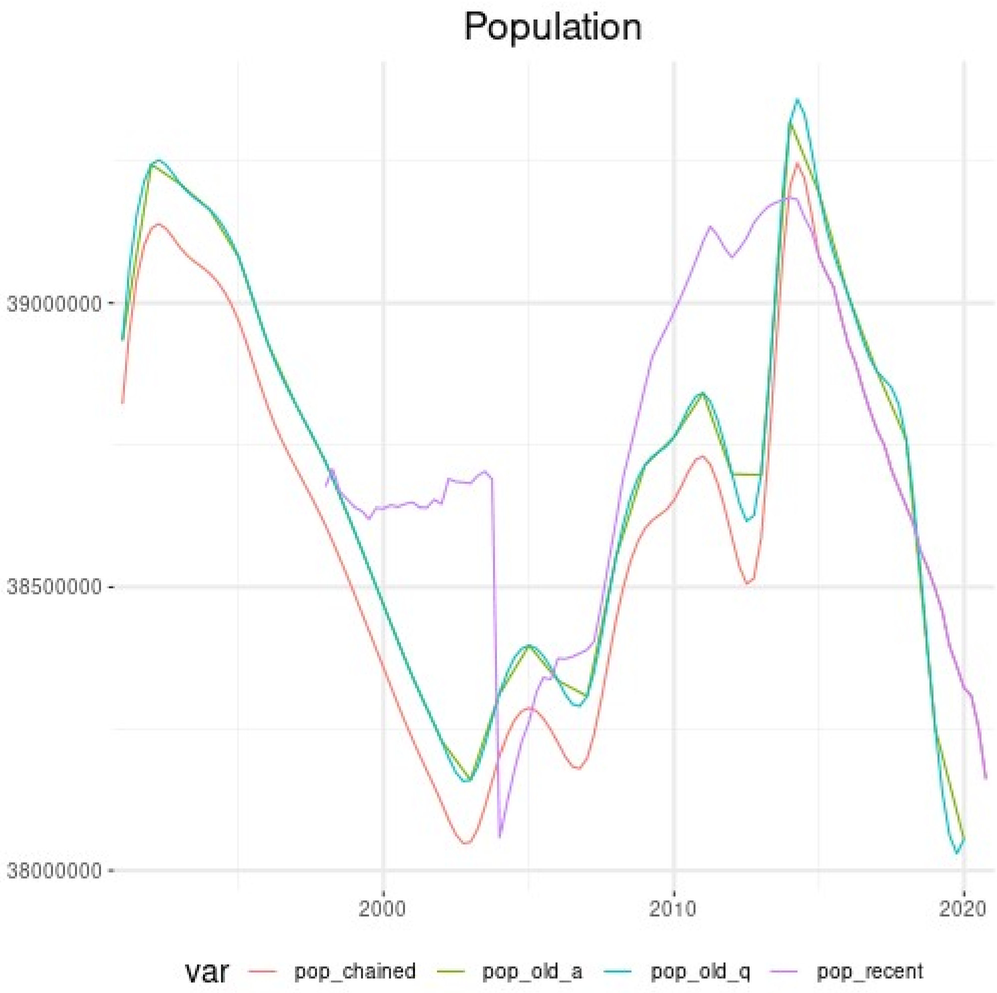


Population

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pop_old_a <-
  df_it %>%
  filter(var=="pop_old")

pop_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(pop_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pop_old_a, by="period") %>%
  select(-value.x) %>%
  rename(value=value.y) %>%
  mutate(value=na.spline(value),
         var="pop")

pop_recent <-
  df_it %>%
  filter(var=="pop_recent") %>%
  mutate(var="pop",value=value*1000)

pop <- chain(basis = pop_recent,
             to_rebase= pop_old_q,
             date_chain="2015-01-01")
```

```
plot_df <- bind_rows(mutate(pop_old_a,var="pop_old_a"),
                     mutate(pop_old_q,var="pop_old_q"),
                     mutate(pop,var="pop_chained"),
                     mutate(pop_recent,var="pop_recent"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Population")
```

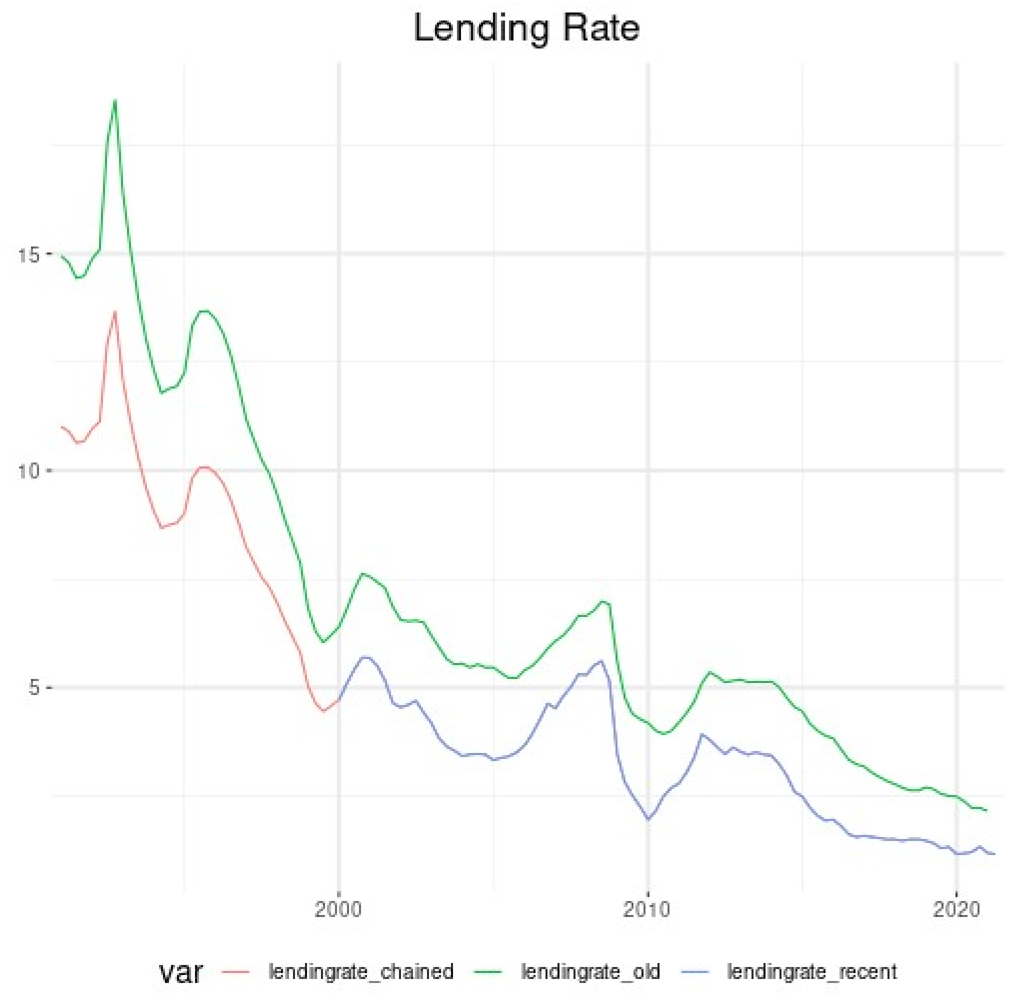


Lending Rate

We chain the quarterly series from the two different databases.

```
lendingrate_old <-
  df_it %>%
  filter(var=="lendingrate_old") %>%
  mutate(var="lendingrate")

lendingrate_recent <-
  df_it %>%
  filter(var=="lendingrate_recent") %>%
  mutate(var="lendingrate")

minDateLendingRateRecent <- min(lendingrate_recent$period)
```

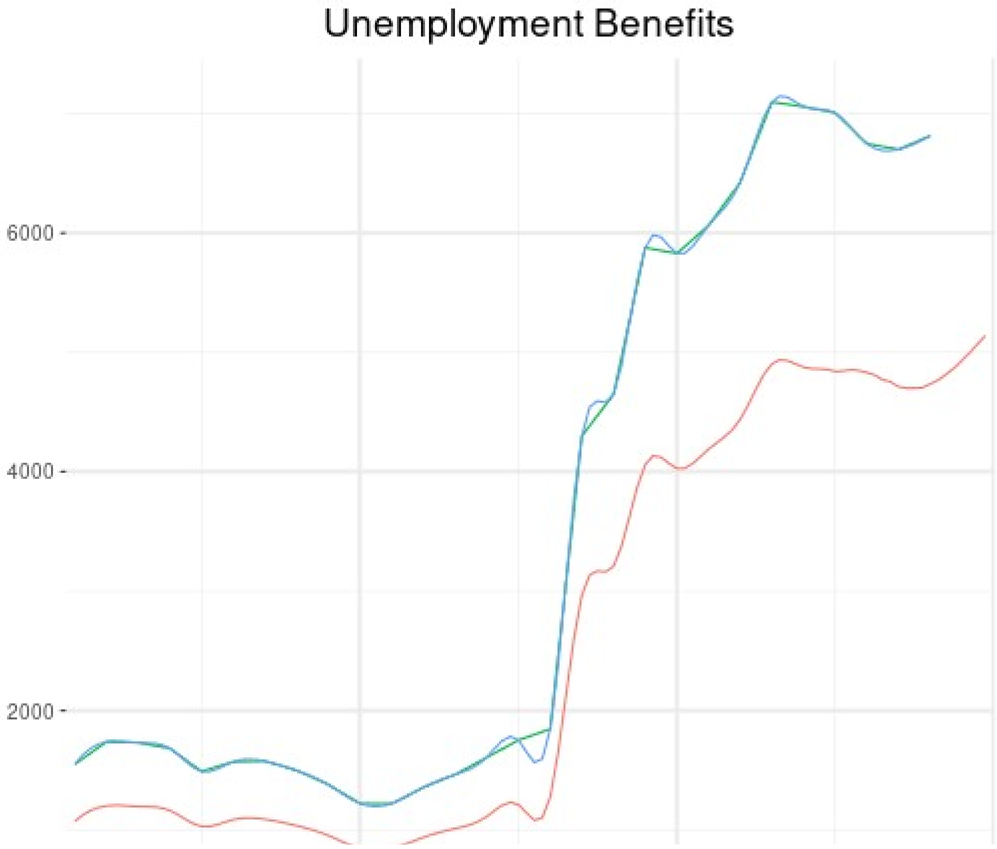
```
lendingrate <-
  chain(basis = lendingrate_recent,
        to_rebase= lendingrate_old,
        date_chain=minDateLendingRateRecent) %>%
  mutate(var="lendingrate")

plot_df <- bind_rows(mutate(lendingrate_old,var="lendingrate_old"),
                     mutate(lendingrate_recent,var=
"lendingrate_recent"),
                     mutate(lendingrate,var="lendingrate_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Lending Rate")
```



Unemployment Benefits

We first interpolate the annual series in order to obtain quarterly series, and then we chain the two quarterly series.

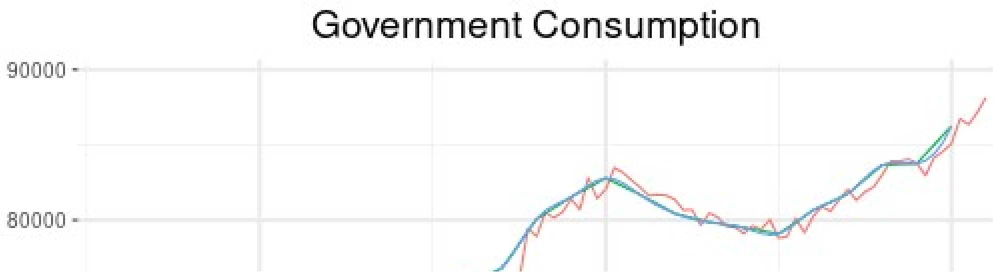
```
unempbenef_old_a <-
  df_it %>%
  filter(var=="unempbenef_old") %>%
  mutate(value=value/4)

unempbenef_old_q <-
  tibble(period=seq(as.Date("1991-01-01"),
                    length.out = (nrow(unempbenef_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(unempbenef_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="unempbenef")

unempbenef_recent <-
  df_it %>%
  filter(var=="unempbenef_recent") %>%
  mutate(var="unempbenef")

unempbenef <- chain(basis = unempbenef_recent,
                    to_rebase= unempbenef_old_q,
                    date_chain="2015-01-01")

plot_df <- bind_rows(mutate(unempbenef_old_a,var="unempbenef_old_a"),
                     mutate(unempbenef_old_q,var="unempbenef_old_q"),
                     mutate(unempbenef,var="unempbenef_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Unemployment Benefits")
```

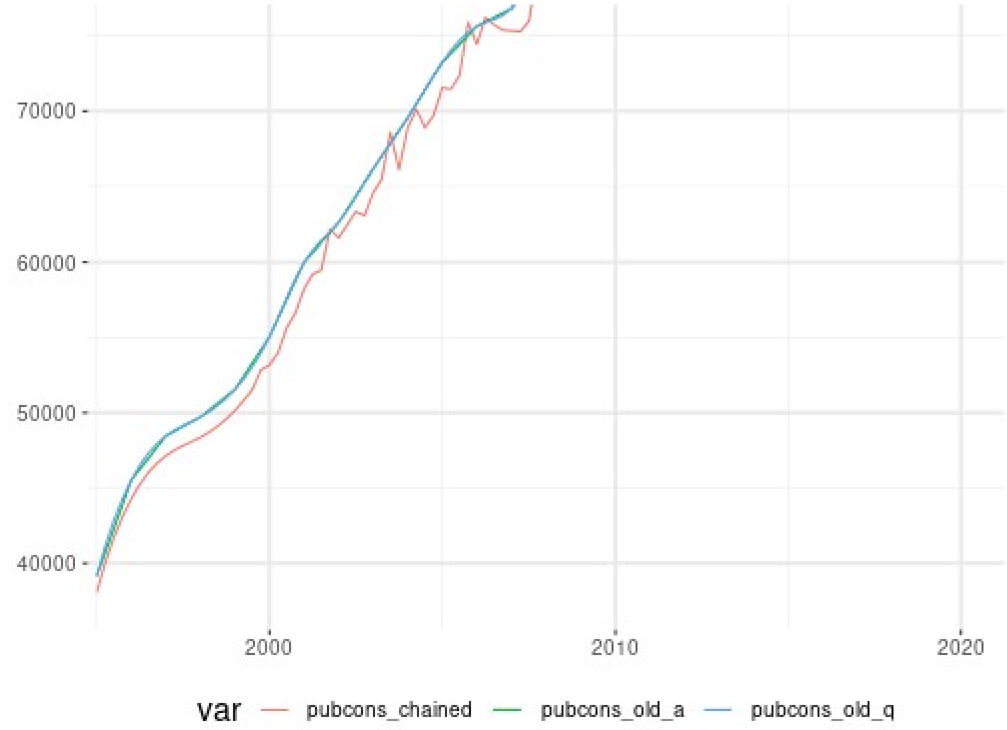


Government Consumption

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pubcons_old_a <-
  df_it %>%
  filter(var=="pubcons_old") %>%
  mutate(value=value/4)

pubcons_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(pubcons_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pubcons_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="pubcons")

pubcons_recent <-
  df_it %>%
  filter(var=="pubcons_recent") %>%
  mutate(var="pubcons")

minDatePubConsRecent <- min(pubcons_recent$period)

pubcons <- chain(basis = pubcons_recent,
                    to_rebase= pubcons_old_q,
                    date_chain=minDatePubConsRecent)

plot_df <- bind_rows(mutate(pubcons_old_a,var="pubcons_old_a"),
                     mutate(pubcons_old_q,var="pubcons_old_q"),
                     mutate(pubcons,var="pubcons_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Consumption")
```

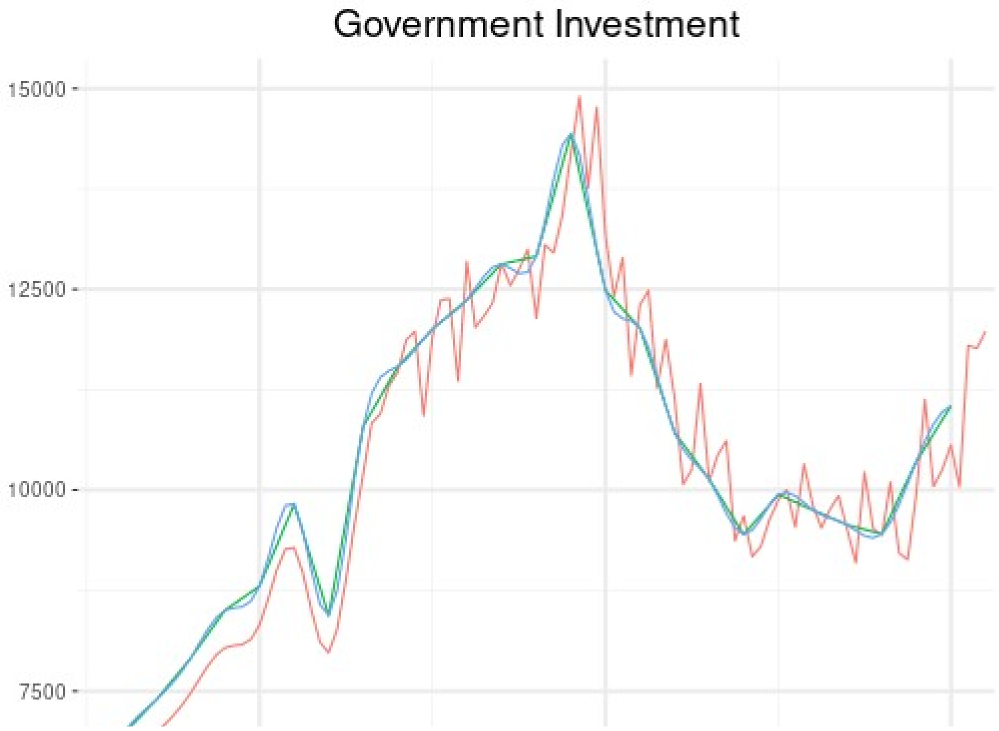


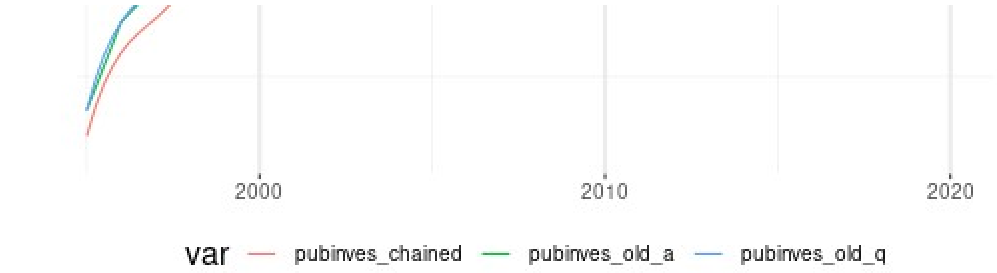


Government Investment

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
pubinves_old_a <-
  df_it %>%
  filter(var=="pubinves_old") %>%
  mutate(value=value/4)

pubinves_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(pubinves_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(pubinves_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="pubinves")

pubinves_recent <-
  df_it %>%
  filter(var=="pubinves_recent") %>%
  mutate(var="pubinves")

pubinves <- chain(basis = pubinves_recent,
                  to_rebase= pubinves_old_q,
                  date_chain="2003-01-01")

plot_df <- bind_rows(mutate(pubinves_old_a,var="pubinves_old_a"),
                     mutate(pubinves_old_q,var="pubinves_old_q"),
                     mutate(pubinves,var="pubinves_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Investment")
```

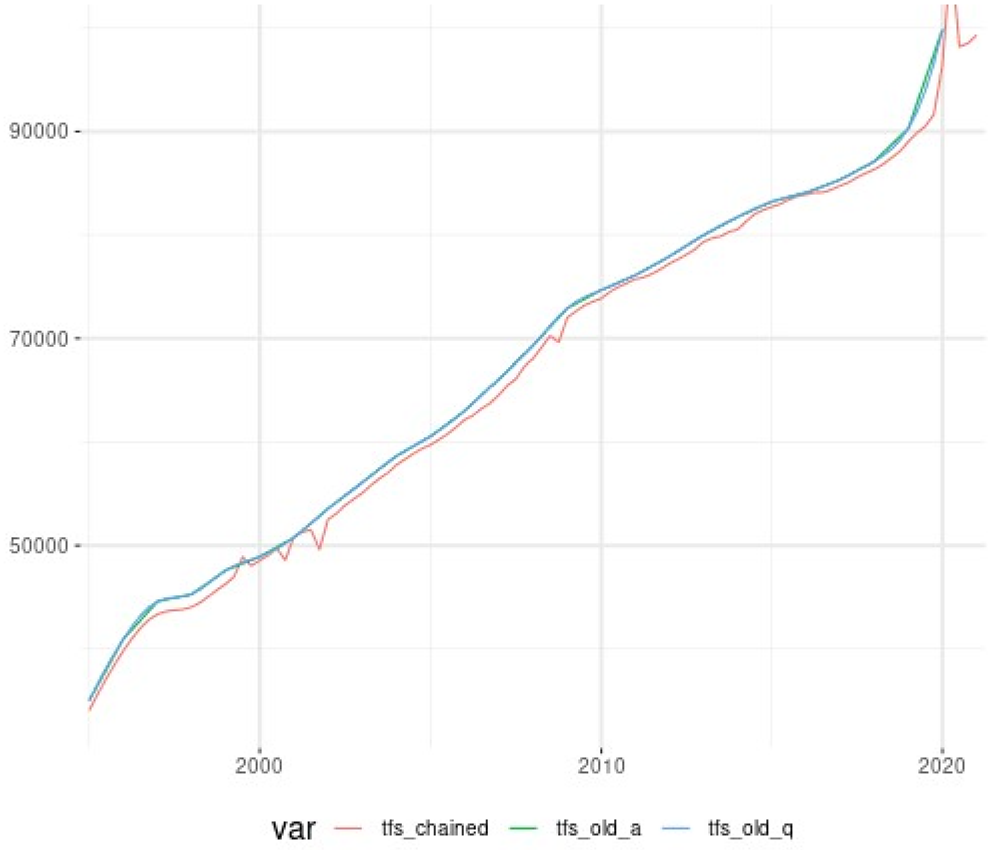




Government Social Transfers

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

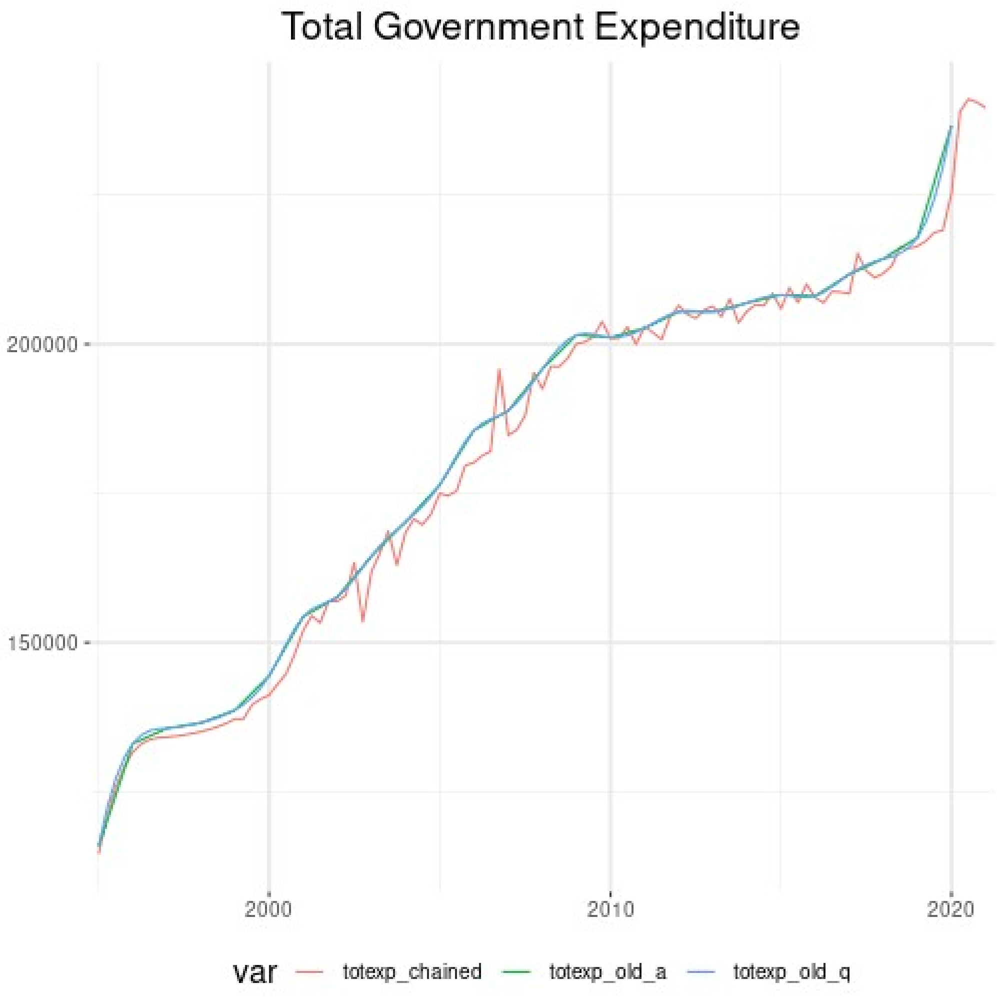
```
tfs_old_a <-
  df_it %>%
  filter(var=="tfs_old") %>%
  mutate(value=value/4)

tfs_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(tfs_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(tfs_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="tfs")

tfs_recent <-
  df_it %>%
  filter(var=="tfs_recent") %>%
  mutate(var="tfs")

minDateTfsRecent <- min(tfs_recent$period)

tfs <- chain(basis = tfs_recent,
                    to_rebase= tfs_old_q,
                    date_chain=minDateTfsRecent)

plot_df <- bind_rows(mutate(tfs_old_a,var="tfs_old_a"),
                     mutate(tfs_old_q,var="tfs_old_q"),
                     mutate(tfs,var="tfs_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Social Transfers")
```



Total Government Expenditure

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

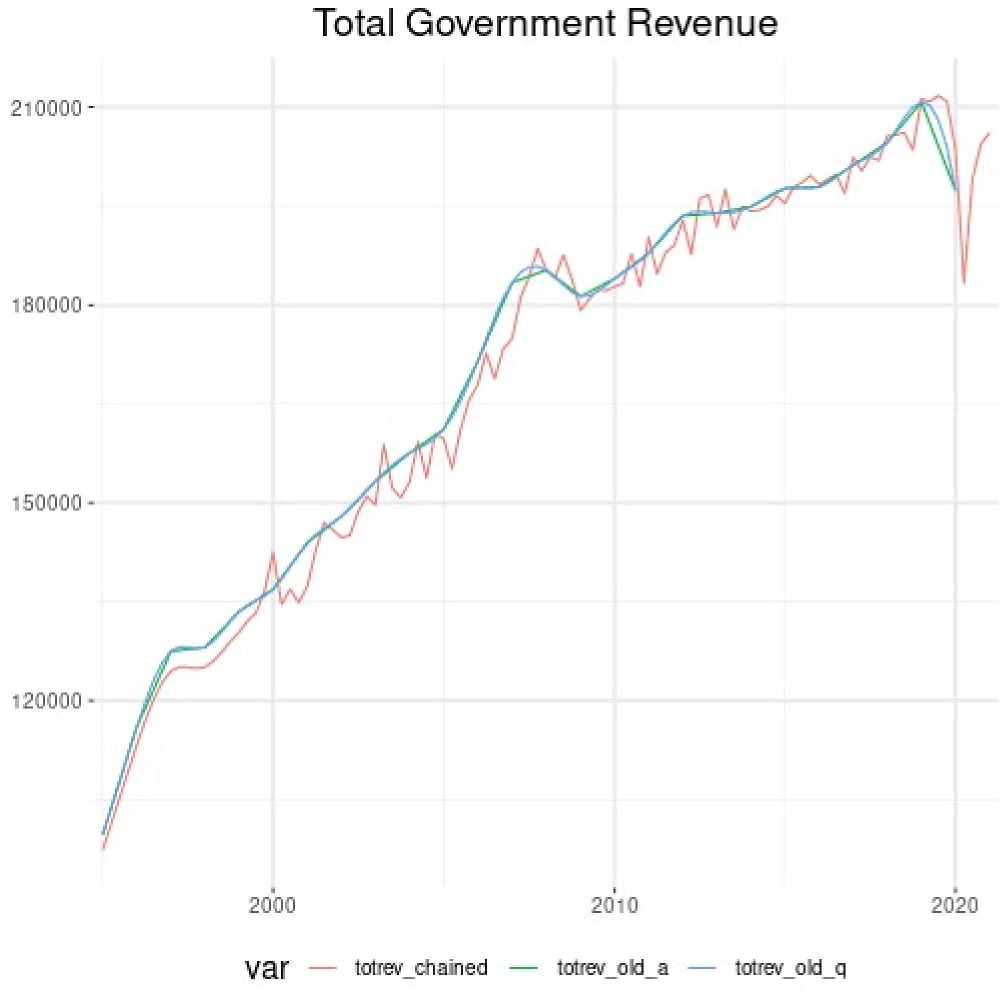
```
totexp_old_a <-
  df_it %>%
  filter(var=="totexp_old") %>%
  mutate(value=value/4)

totexp_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(totexp_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(totexp_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="totexp")

totexp_recent <-
  df_it %>%
  filter(var=="totexp_recent") %>%
  mutate(var="totexp")

minDateTotExpRecent <- min(totexp_recent$period)

totexp <- chain(basis = totexp_recent,
                    to_rebase= totexp_old_q,
                    date_chain=minDateTotExpRecent)

plot_df <- bind_rows(mutate(totexp_old_a,var="totexp_old_a"),
                     mutate(totexp_old_q,var="totexp_old_q"),
                     mutate(totexp,var="totexp_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Total Government Expenditure")
```



Total Government Revenue

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

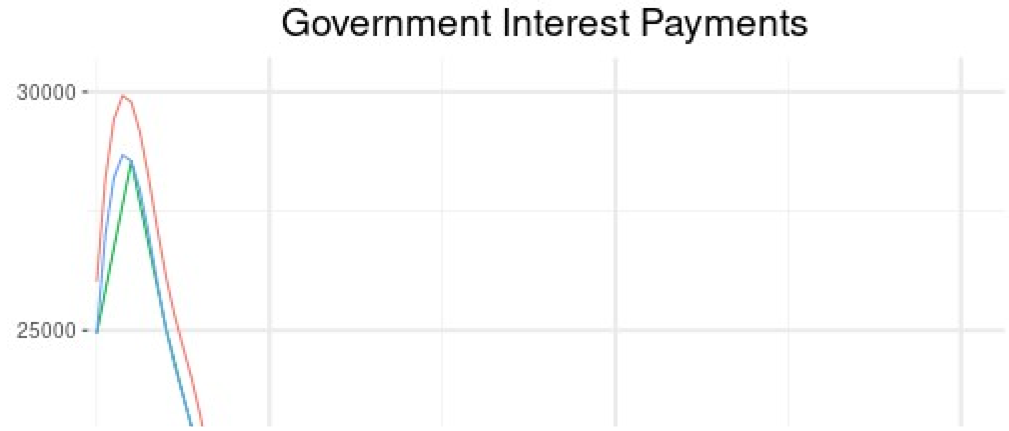
```
totrev_old_a <-
  df_it %>%
  filter(var=="totrev_old") %>%
  mutate(value=value/4)

totrev_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(totrev_old_a)-1)*4+1,
                    by = "quarter"),
           value=NA) %>%
    left_join(totrev_old_a, by="period") %>%
    select(-value.x,value=value.y) %>%
    mutate(value=na.spline(value),
           var="totrev")

totrev_recent <-
    df_it %>%
    filter(var=="totrev_recent") %>%
    mutate(var="totrev")

minDateTotRevRecent <- min(totrev_recent$period)

totrev <- chain(basis = totrev_recent,
                    to_rebase= totrev_old_q,
                    date_chain=minDateTotRevRecent)

plot_df <- bind_rows(mutate(totrev_old_a,var="totrev_old_a"),
                     mutate(totrev_old_q,var="totrev_old_q"),
                     mutate(totrev,var="totrev_chained"))

ggplot(plot_df,aes(period,value,col=var))+
    geom_line()+
    scale_x_date(expand = c(0.01,0.01)) +
    theme + xlab(NULL) + ylab(NULL)+
    ggtitle("Total Government Revenue")
```

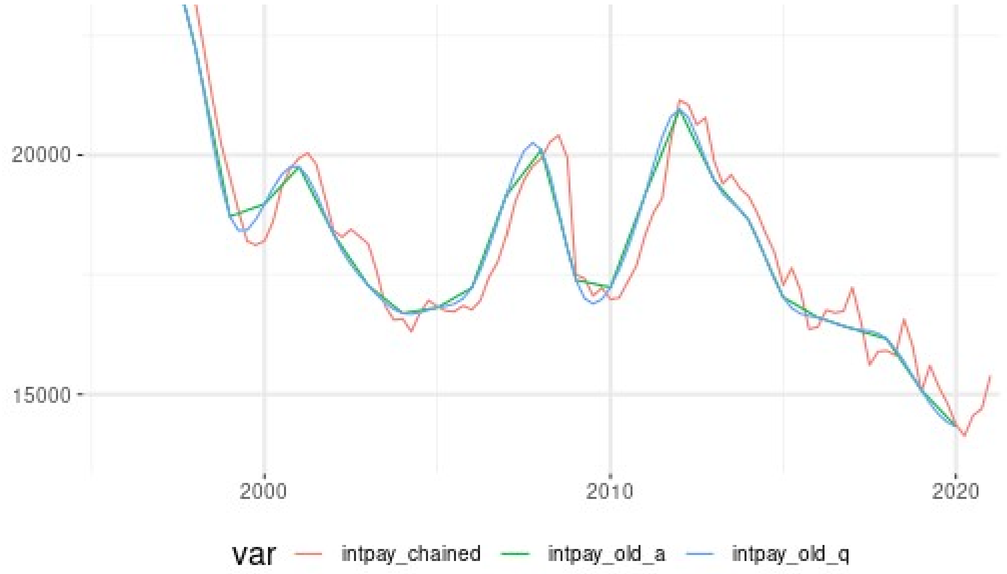


Government Interest Payments

We first interpolate the annual series in order to obtain quarterly series, and then we chain the quarterly series from the two different databases.

```
intpay_old_a <-
  df_it %>%
  filter(var=="intpay_old") %>%
  mutate(value=value/4)

intpay_old_q <-
  tibble(period=seq(as.Date("1995-01-01"),
                    length.out = (nrow(intpay_old_a)-1)*4+1,
                    by = "quarter"),
         value=NA) %>%
  left_join(intpay_old_a, by="period") %>%
  select(-value.x,value=value.y) %>%
  mutate(value=na.spline(value),
         var="intpay")

intpay_recent <-
  df_it %>%
  filter(var=="intpay_recent") %>%
  mutate(var="intpay")

minDateIntPayRecent <- min(intpay_recent$period)

intpay <- chain(basis = intpay_recent,
                    to_rebase= intpay_old_q,
                    date_chain=minDateIntPayRecent)

plot_df <- bind_rows(mutate(intpay_old_a,var="intpay_old_a"),
                     mutate(intpay_old_q,var="intpay_old_q"),
                     mutate(intpay,var="intpay_chained"))

ggplot(plot_df,aes(period,value,col=var))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  ggtitle("Government Interest Payments")
```





Merging Italian Data

We gather all the final series in a dataframe.

```
IT_rawdata <-
  df_it %>%
  filter(! var %in% c("conso","inves","definves","
lendingrate_old","lendingrate_recent",
                      "pop_old","pop_recent",
                      "debt_old","debt_recent",
                      "unempbenef_old","unempbenef_recent",
                      "totexp_recent", "totexp_old",
                      "intpay_recent", "intpay_old",
                      "totrev_recent", "totrev_old",
                      "pubcons_recent", "pubcons_old",
                      "pubinves_recent","pubinves_old",
                      "tfs_recent","tfs_old")) %>%
  bind_rows(conso,inves,definves,lendingrate,pop,debt,
unempbenef,totexp,totrev,intpay,pubcons,pubinves,tfs) %>%
  spread(var,value) %>%
  add_column(country="IT")
```

Final database for the estimation

Implicit tax rates

We retrieve the data on the implicit tax rates (ITR) on consumption, labour and corporate incomes, that we built specifically for this project : https://macro.cepremap.fr/article/2019-11/implicit_tax_rates/

```
itr <- read_csv("https://shiny.cepremap.fr/data/ITR_eurodata.csv") %>%
  rename(year=period)

itrq <-
  tibble(period=EA_rawdata$period) %>%
  mutate(year=year(period)) %>%
  left_join(itr,by="year") %>%
  na.omit() %>%
  select(-year) %>%
  gather(var,value,-period) %>%
  separate(var,c("country","var")) %>%
  spread(var,value)
```

Merge raw data

```
rawdata_var <- colnames(DE_rawdata)
EA_rawdata_short <-
  EA_rawdata %>%
  select(rawdata_var,oil_prices)
```

Then, we gather the datasets for France, Germany, Italy, Spain and the Euro area in a unique data frame.

```
rawdata_df <-
  bind_rows(EA_rawdata_short,DE_rawdata,ES_rawdata,FR_rawdata,
IT_rawdata) %>%
  left_join(itrq,
            by = c("period", "country")) %>%
  arrange(country,period) %>%
  filter(year(period)>=1995,
         period <= last_date)

save(rawdata_df, file = "rawdata.RData")
```

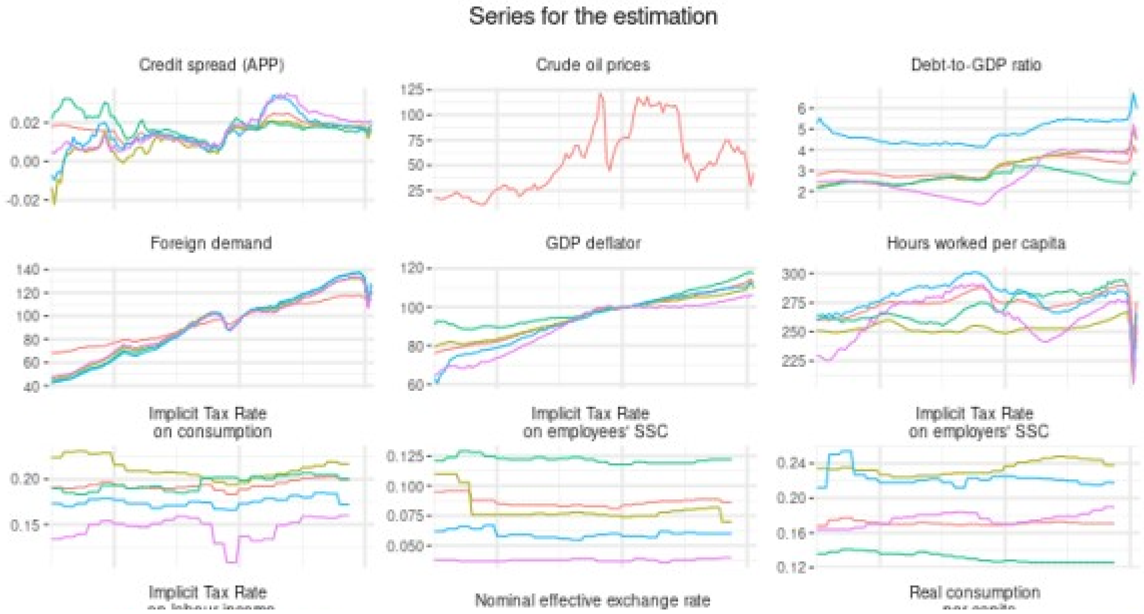
Normalize data

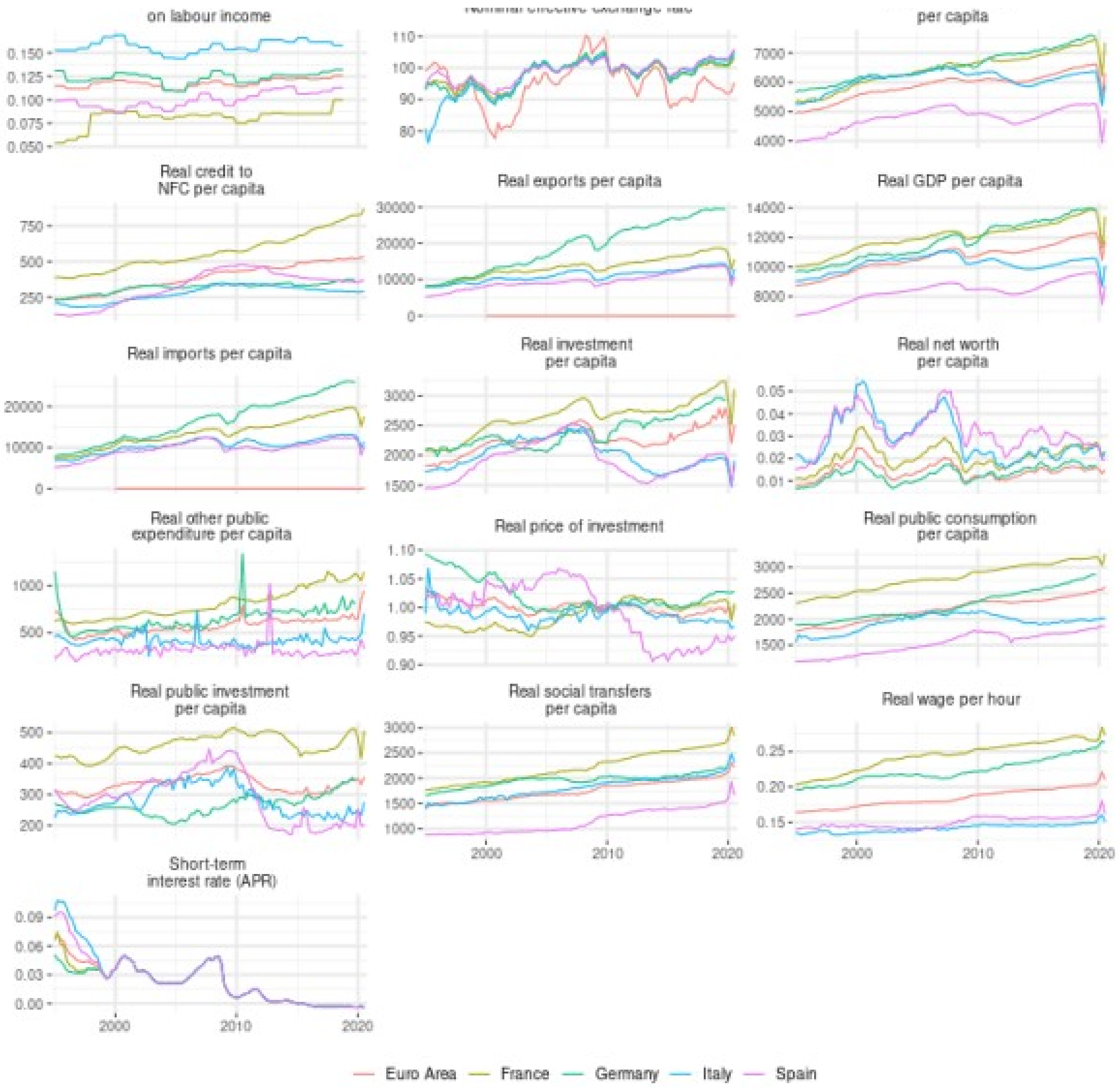
Then we select and normalize the data by population and prices.

```
data_df <-
  rawdata_df %>%
  transmute(period,
            country,
            gdp_rpc=1e+6*gdp/pop,
            conso_rpc=1e+6*conso/pop,
            inves_rpc=1e+6*inves/pop,
            defgdp = defgdp,
            wage_rph=1e+6*wage/defgdp/(hours*1000),
            hours_pc=(hours*1000)/pop,
            pinves_defl=definves/defgdp,
            loans_nfc_rpc=1e+9*loans_nfc/pop/defgdp,
            networth_rpc=1e+6*networth/pop/defgdp,
            re=shortrate/100,
            creditspread=(lendingrate-shortrate)/100,
            pubcons_rpc=100*1e+6*pubcons/(defgdp*pop),
            pubinves_rpc=100*1e+6*pubinves/(defgdp*pop),
            tfs_rpc=100*1e+6*tfs/(defgdp*pop),
            othgov_rpc=100*1e+6*(totexp-pubcons-pubinves-tfs-intpay)/(
defgdp*pop),
            debt_gdp=100*debt/(defgdp*gdp),
            taun,tauwf,tauwh,tauc,
            world_demand,
            oil_prices,
            neer,
            imports_rpc=imports/pop,
            exports_rpc=exports/pop) %>%
  gather(var,value,-period,-country)
```

The figure below shows the final series for all the listed countries.

```
plot_data_df <-
  data_df %>%
  mutate(varname=
           case_when(
             var=="gdp_rpc"          ~ "Real GDP per capita",
             var=="conso_rpc"        ~ "Real consumption \n per
capita",
             var=="inves_rpc"        ~ "Real investment \n per
capita",
             var=="defgdp"           ~ "GDP deflator",
             var=="wage_rph"         ~ "Real wage per hour" ,
             var=="hours_pc"         ~ "Hours worked per capita",
             var=="pinves_defl"      ~ "Real price of investment",
             var=="loans_nfc_rpc"    ~ "Real credit to \n NFC per
capita",
             var=="networth_rpc"     ~ "Real net worth \n per capita",
             var=="re"               ~ "Short-term \n interest rate
(APR)",
             var=="creditspread"     ~ "Credit spread (APP)",
             var=="pubcons_rpc"      ~ "Real public consumption\n per
capita",
             var=="pubinves_rpc"     ~ "Real public investment\n per
capita",
             var=="tfs_rpc"          ~ "Real social transfers\n per
capita",
             var=="othgov_rpc"       ~ "Real other public\n
expenditure per capita",
             var=="debt_gdp"         ~ "Debt-to-GDP ratio",
             var=="taun"             ~ "Implicit Tax Rate \n on labour
income ",
             var=="tauwh"            ~ "Implicit Tax Rate \n on
employees' SSC",
             var=="tauwf"            ~ "Implicit Tax Rate \n on
employers' SSC" ,
             var=="tauc"             ~ "Implicit Tax Rate \n on
consumption" ,
             var=="world_demand"     ~ "Foreign demand",
             var=="oil_prices"       ~ "Crude oil prices",
             var=="neer"             ~ "Nominal effective exchange
rate",
             var=="imports_rpc"      ~ "Real imports per capita",
             var=="exports_rpc"      ~ "Real exports per capita"),
         country_name=
           case_when(
             country=="FR"            ~ "France",
             country=="DE"            ~ "Germany",
             country=="IT"            ~ "Italy",
             country=="ES"            ~ "Spain" ,
             country=="EA"            ~ "Euro Area")) %>%
  na.omit()

tikz('estimated.tex', width=5.2,height=8.4, sanitize=TRUE)

ggplot(plot_data_df,aes(period,value,col=country_name))+
  geom_line()+
  facet_wrap(~varname,ncol=3,scales="free_y")+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  theme(legend.title=element_blank()) +
  theme(strip.text=element_text(size=8),
        axis.text=element_text(size=7))+
  ggtitle("Series for the estimation")+
    theme(plot.title=element_text(size=12))

dev.off()

## png
##   2

ggplot(plot_data_df,aes(period,value,col=country_name))+
  geom_line()+
  facet_wrap(~varname,ncol=3,scales="free_y")+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  theme(legend.title=element_blank()) +
  theme(strip.text=element_text(size=10),
        axis.text=element_text(size=9))+
  ggtitle("Series for the estimation")+
    theme(plot.title=element_text(size=15))
```





```
data <-
  data_df %>%
  unite("var",c("country","var")) %>%
  mutate(period=gsub(" ","",as.yearqtr(period))) %>%
  spread(var,value) %>%
  select(-DE_oil_prices,-FR_oil_prices,-ES_oil_prices,-IT_oil_prices)

colnames(data)[1] <- ""

write.csv(data,"data_DE_EA_ES_FR_IT.csv",row.names = FALSE)
```

The data can be downloaded directly here.

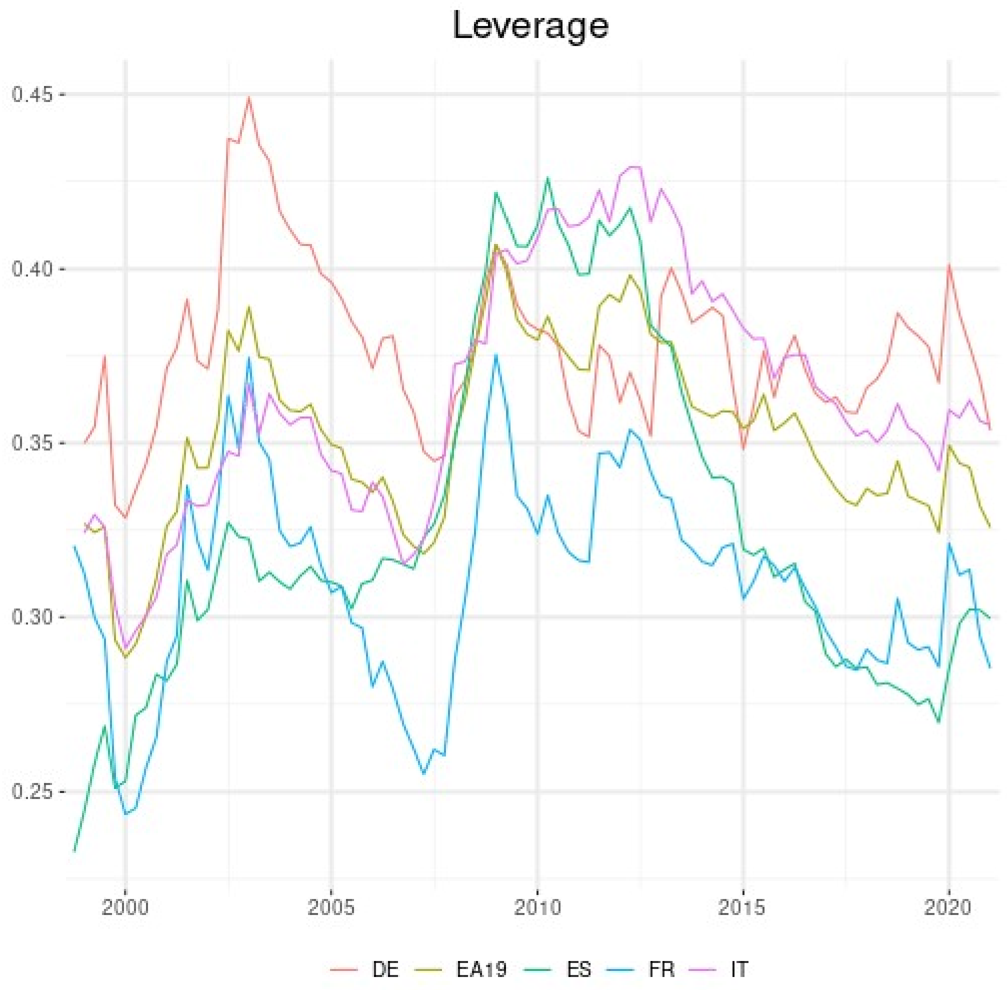
Series for the calibration

For the calibration, we need additional series. The data needed for this purpose is retrieved below by variable.

Leverage of non-financial corporations

Using the `nasq_10_f_bs` database from Eurostat, we retrieve these data. The figure below shows the series for France, Germany, Italy, Spain and the Euro area 19.

```
debt <- rdb("Eurostat","nasq_10_f_bs",mask =
"Q.MIO_EUR.S11.LIAB.F+F3+F4+F6.EA19+IT+DE+FR+ES")
leverage <-
  debt %>%
  select(value,period,country=geo,var=na_item) %>%
  mutate(var=
           case_when(
             var=="F"  ~ "total",
             var=="F3" ~ "debt_securities",
             var=="F4" ~ "loans",
             var=="F6" ~ "pensions_reserves")) %>%
  spread(var,value) %>%
  arrange(country,period) %>%
  transmute(period,
            country,
            value=(debt_securities+loans+pensions_reserves)/total,
            var="leverage") %>%
  na.omit()

ggplot(leverage,aes(period,value,colour=country))+
  geom_line() +
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Leverage")
```



The depreciation rate of the capital stock

We compute the depreciation rate of the capital stock for France, Italy, Spain, Germany and the Euro Area since 1995. We use the data from the Penn World Table Feenstra et al. (2015) available here.

```
listcountry <- list("France"  = "FRA",
                    "Germany"  = "DEU",
                    "Italy" = "ITA",
                    "Spain" = "ESP",
                    "Euro Area" = "EA")

df <- read_dta("https://www.rug.nl/ggdc/docs/pwt100.dta") %>%
  mutate(country=countrycode, period=as.Date(as.yearqtr(year))) %>%
  filter(year(period)>=1995 & currency_unit=="Euro" & !grepl('MNE',
country))
```

Step1: data for Euro Area countries

```
delta <-
  df %>%
  select(country, period, value=delta) %>%
  add_column(var="delta")
```

Step2: Euro Area GDP-weighted average

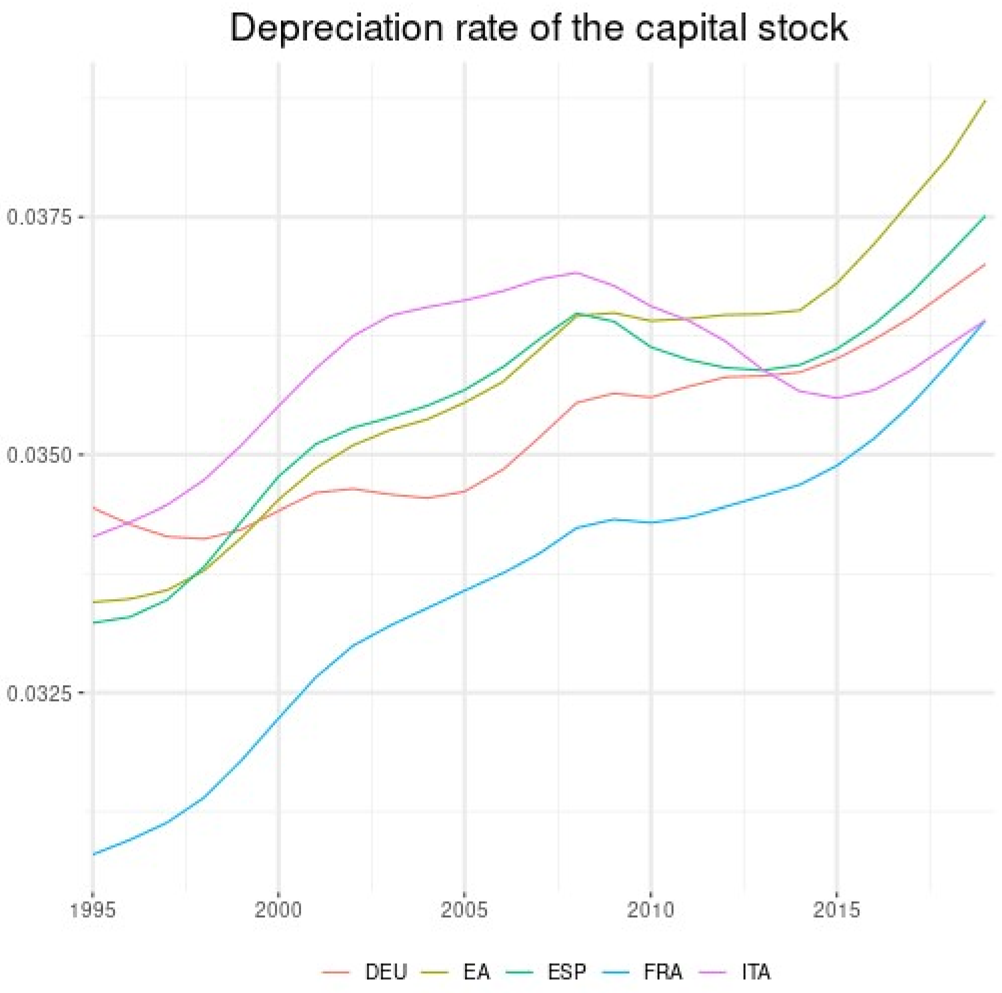
After retrieving the data on the average depreciation rate of the capital stock for the Euro Area countries, it is possible to build the GDP-weighted average for the Eurozone. First, it is necessary to establish the weights that will be used for this purpose, using the output-side real GDP at chained PPPs (in mil. 2011US$) of each country.

```
gdp <-
  df %>%
  select(country, period, value=rgdpo)

EA_gdp <-
  gdp %>%
  group_by(period) %>%
  summarize(value=sum(value)) %>%
  ungroup()

weights <-
  gdp %>%
  left_join(EA_gdp,by="period") %>%
  transmute(country, period, weight=value.x/value.y)
```

Now we apply these weights to our country data in order to build the Euro Area GDP-weighted average. The figure below shows the final series for France, Germany, Italy, Spain and the Euro Area.

```
delta_EA <-
  delta %>%
  left_join(weights,by=c("country","period"))
```

```
delta_EA <-
  delta_EA %>%
  transmute(period,value=value*weight) %>%
  group_by(period) %>%
  summarize(value =sum(value)) %>%
  add_column(country="EA",var="delta")

delta_countries <-
  delta %>%
  filter(grepl('FRA|DEU|ITA|ESP', country))

delta_FIN <-
  bind_rows(delta_countries,delta_EA)

ggplot(delta_FIN,aes(period,value,colour=country))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Depreciation rate of the capital stock")
```



The share of capital revenues in GDP

We proceed in two steps.

Step1: data for Euro Area countries

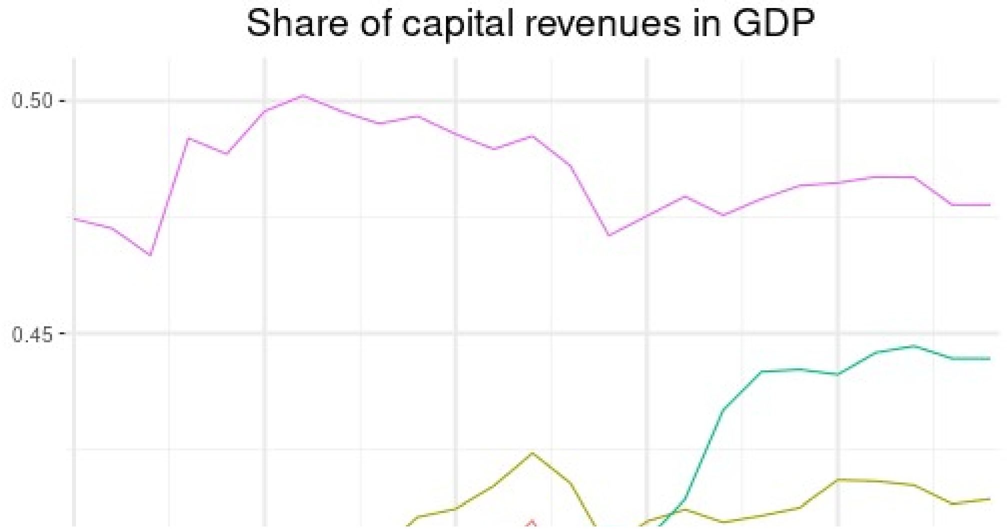
We obtain the share of labour compensation in GDP also from the Penn World Table, for the countries that compose the Euro Area. We deduce then the share of capital revenues in GDP

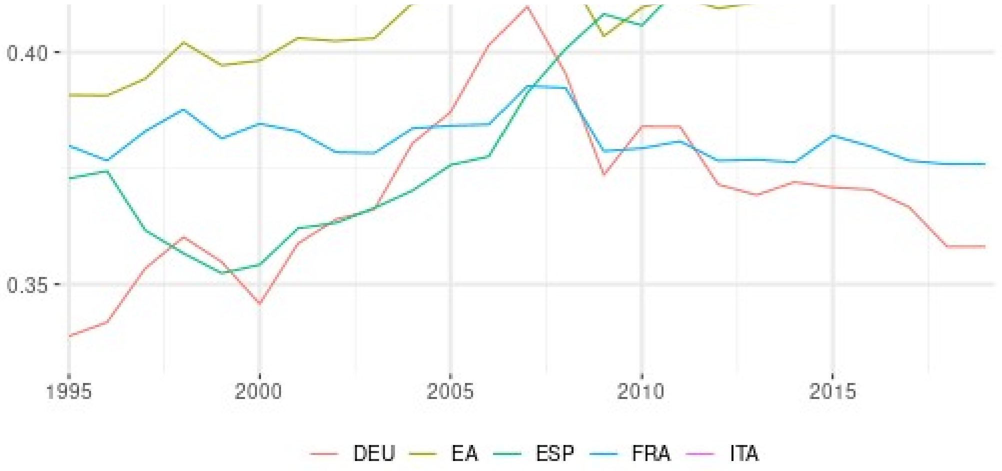
```
alpha <-
  df %>%
  select(country, period, value=labsh) %>%
  mutate(value=1-value) %>%
  add_column(var="alpha")
```

Step2: Euro Area GDP-weighted average

Now we apply the GDP-weights to our country data in order to build the Euro Area GDP-weighted average. The figure below shows the final series for France, Germany, Italy, Spain and the Euro Area.

```
alpha_EA <-
  alpha %>%
  left_join(weights,by=c("country","period"))

alpha_EA <-
  alpha_EA %>%
  transmute(period,value=value*weight) %>%
  group_by(period) %>%
  summarize(value =sum(value)) %>%
  add_column(country="EA",var="alpha")

alpha_countries <-
  alpha %>%
  filter(grepl('FRA|DEU|ITA|ESP', country))

alpha_FIN <-
  bind_rows(alpha_countries,alpha_EA)

ggplot(alpha_FIN,aes(period,value,colour=country)) +
  geom_line() +
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Share of capital revenues in GDP")
```





The share of capital in GDP

We proceed in two steps.

Step1: data for Euro Area countries

We obtain the stock of capital in GDP also from the Penn World Table, for the countries that compose the Euro Area.

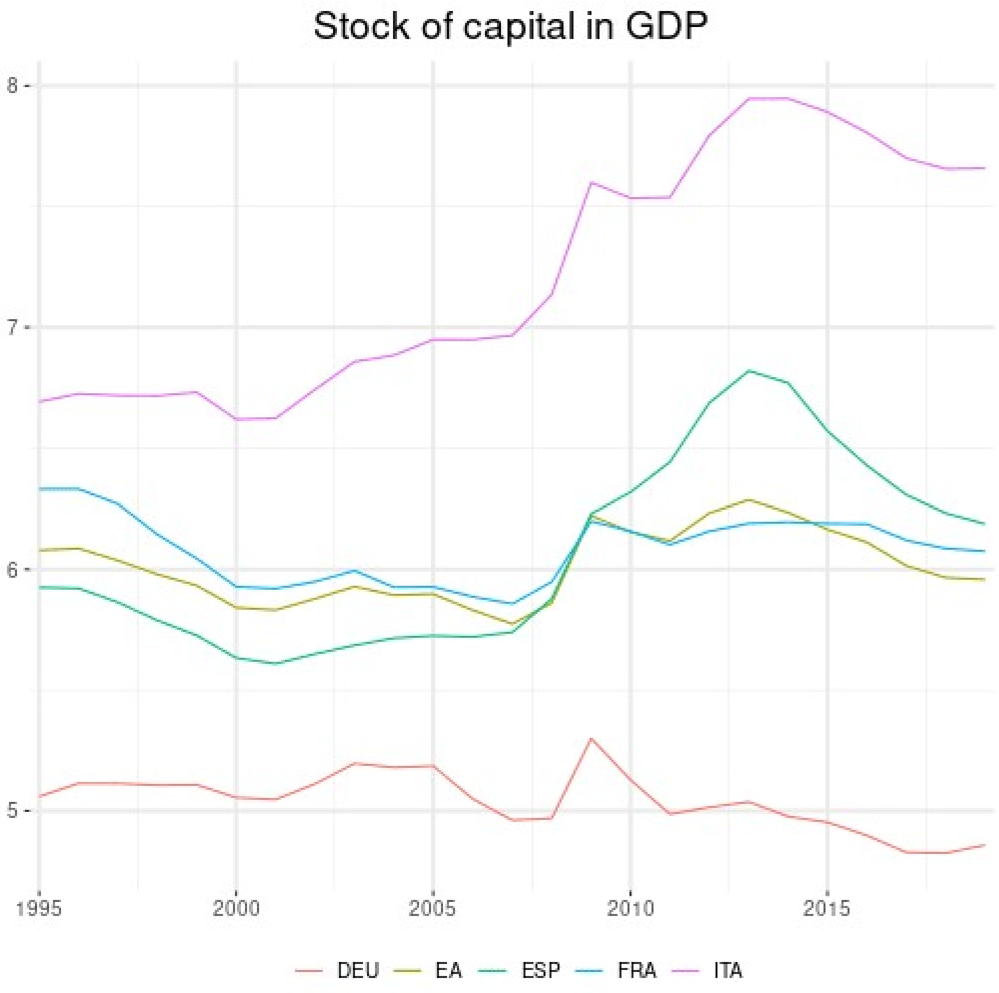
```
capital <-
  df %>%
  select(country,period,gdp=rgdpna,capital=rnna) %>%
  mutate(value=capital/gdp) %>%
  select(-capital,-gdp) %>%
  add_column(var="capital_gdp")
```

Step2: Euro Area GDP-weighted average

Now we apply the GDP-weights to our country data in order to build the Euro Area GDP-weighted average. The figure below shows the final series for France, Germany, Italy, Spain and the Euro Area.

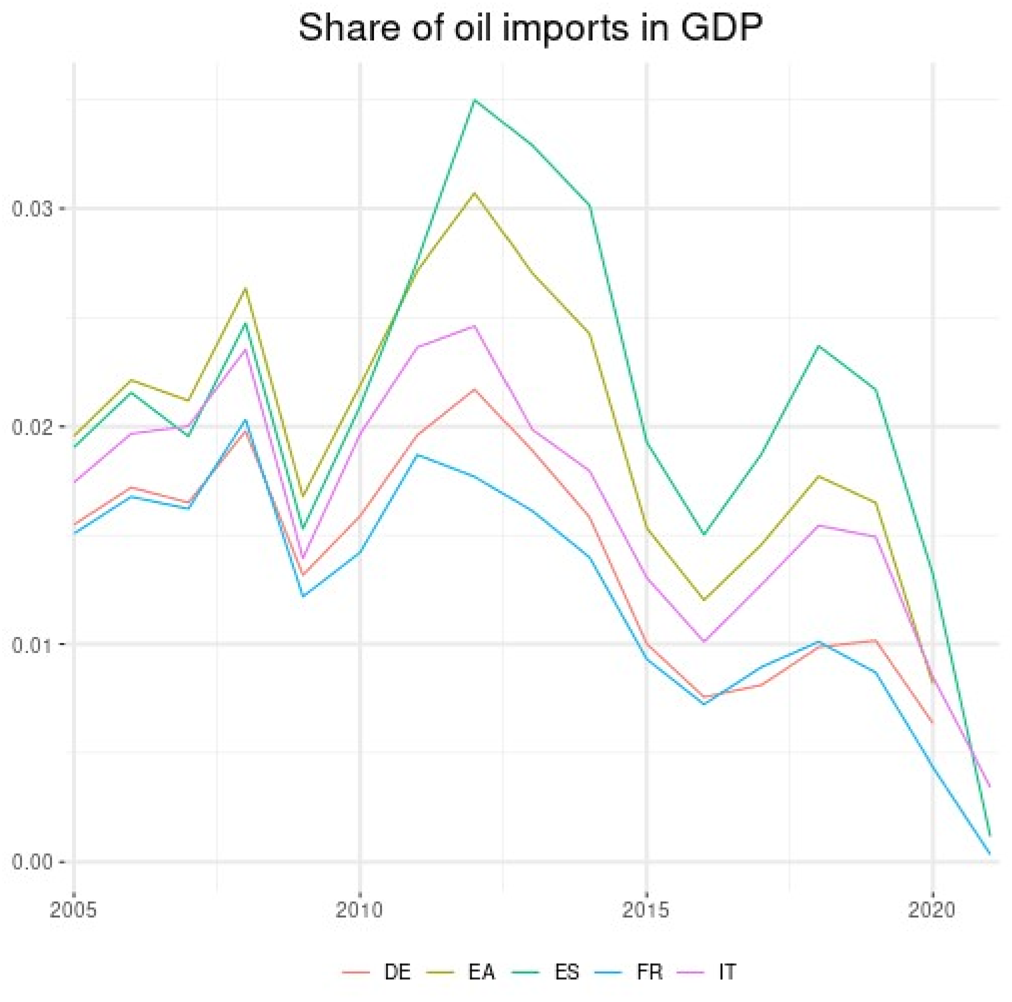
```
capital_EA <-
  capital %>%
  left_join(weights,by=c("country","period"))

capital_EA <-
  capital_EA %>%
  transmute(period,value=value*weight) %>%
  group_by(period) %>%
  summarize(value =sum(value)) %>%
  add_column(country="EA",var="capital_gdp")

capital_countries <-
  capital %>%
  filter(grepl('FRA|DEU|ITA|ESP', country))

capital_FIN <-
  bind_rows(capital_countries,capital_EA)
```

```
ggplot(capital_FIN,aes(period,value,colour=country))+
  geom_line()+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Stock of capital in GDP")
```



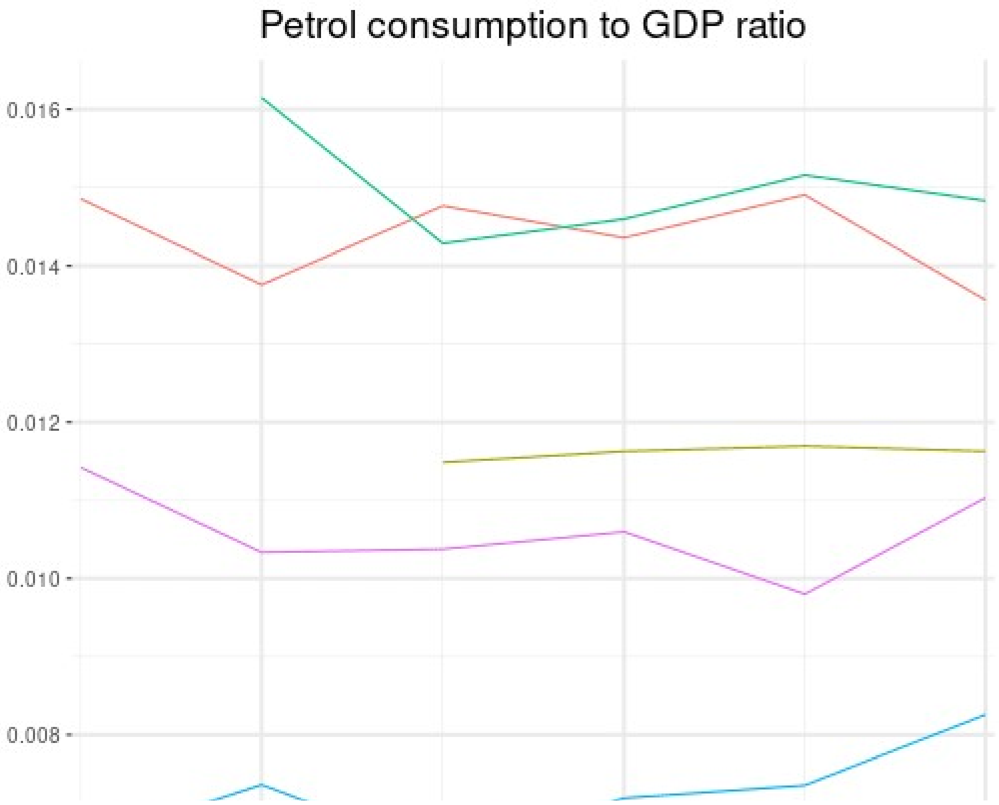
The share of crude oil imports in GDP

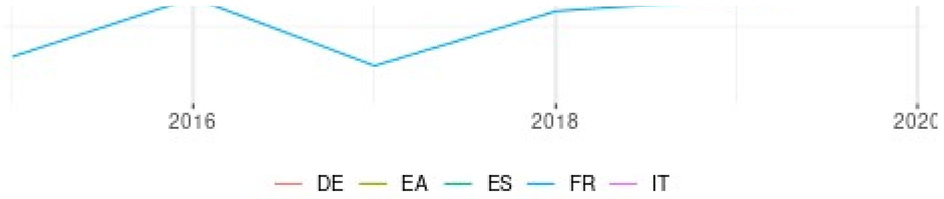
```
oil_import_value <-
  rdb("Eurostat","nrg_ti_coifpm",mask="M.TOTAL.VAL_THS_USD.DE+
FR+IT+ES+EU_V") %>%
  select(period,oil_import=value,country=geo) %>%
  mutate(year=year(period)) %>%
  group_by(country,year) %>%
  summarise(oil_import=sum(oil_import)) %>%
  mutate(period=as.Date(paste0(year,"-01-01"))) %>%
  select(-year) %>%
  mutate(country=case_when(
    country=="EU_V" ~ "EA",
    TRUE ~ country))
# in US $

ea_gdp_usd <-
  rdb("IMF","WEOAGG:latest",mask = "998.NGDPD.us_dollars") %>%
  select(period,gdp=value,country=`weo-countries-group`)
gdp_usd <-
  rdb("IMF","WEO:latest",mask = "FRA+DEU+ITA+ESP.NGDPD.us_dollars") %>%
  select(period,gdp=value,country=`weo-country`) %>%
  bind_rows(ea_gdp_usd) %>%
  mutate(country=case_when(
    country=="FRA" ~ "FR",
    country=="DEU" ~ "DE",
    country=="ITA" ~ "IT",
    country=="ESP" ~ "ES",
    country=="998" ~ "EA",
    TRUE ~ country))

oil <-
  oil_import_value %>%
  left_join(gdp_usd) %>%
  transmute(period,
            value=oil_import/(gdp*1000000),
            var="oil_imports_gdp")

ggplot(oil,aes(period,value,color=country))+
  geom_line() +
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Share of oil imports in GDP")
```



The share of petrol in private consumption

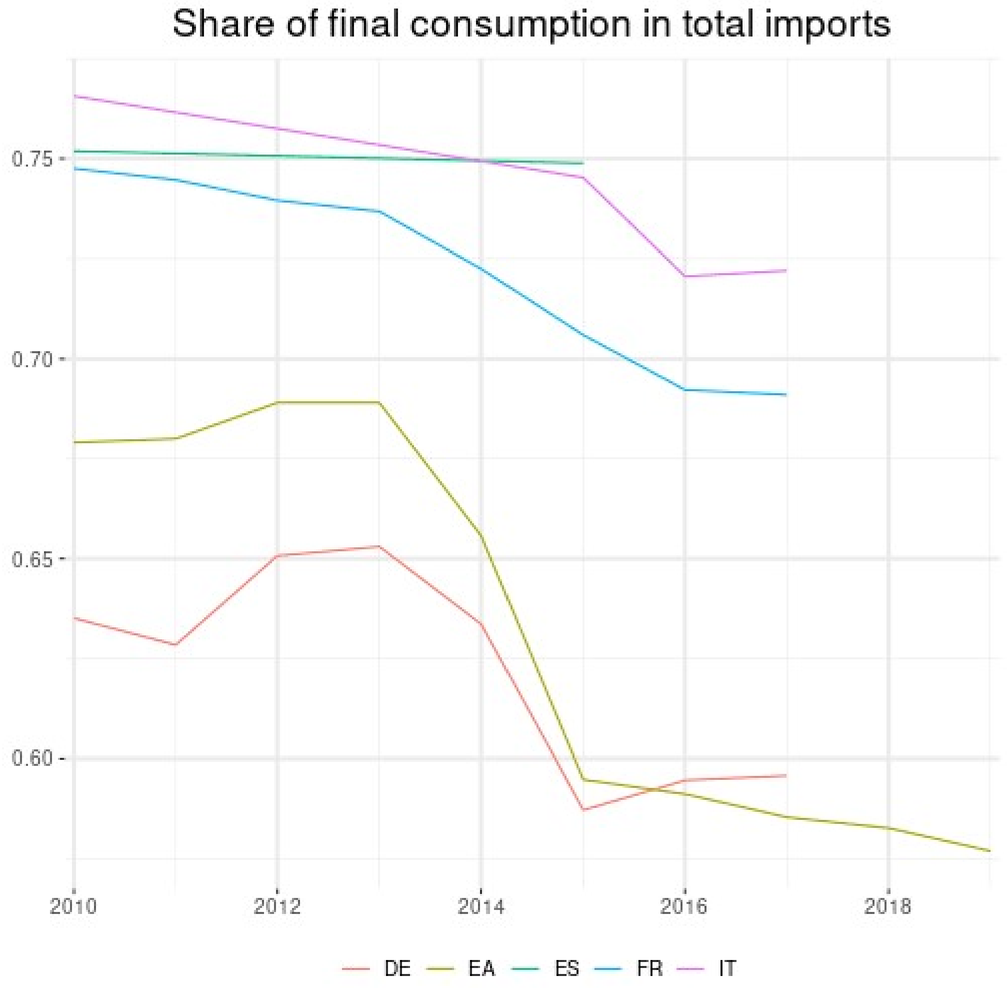
```
petrol_weight <-
  rdb("Eurostat","prc_hicp_inw",mask="A.CP07222.FR+DE+IT+ES+EA") %>%
  select(value,country=geo,period) %>%
  mutate(value=value/1000) %>%
  rename(petrol_weight=value) %>%
  filter(year(period)>=2015)

hhconso <-
  rdb("Eurostat","nama_10_gdp",mask="A.PC_GDP.P31_S14.FR+DE+IT+ES+EA")
%>%
  select(period,conso_gdp=value,country=geo) %>%
  filter(year(period)>=2015)

petrol_conso <-
  hhconso %>%
  left_join(petrol_weight,by=c("country","period")) %>%
  na.omit() %>%
  transmute(period,
            country,
            value=conso_gdp*petrol_weight/100,
            var="petrol_conso")

ggplot(petrol_conso,aes(period,value,color=country))+
  geom_line() +
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Petrol consumption to GDP ratio")
```





Share of final consumption in imports

```
imported_conso <-
  rdb("Eurostat","naio_10_cp1700",mask="A.MIO_EUR.IMP.
P3+P51G.TOTAL.DE+FR+IT+ES+EA19") %>%
  select(period,value,country=geo,var=induse) %>%
  spread(var,value) %>%
  mutate(value=P3/(P3+P51G)) %>%
  select(-P3,-P51G) %>%
  filter(year(period)>=2010) %>%
  mutate(country=case_when(country=="EA19" ~ "EA",
                           TRUE ~ country),
         var="imported_conso")

ggplot(imported_conso,aes(period,value,color=country))+
  geom_line() +
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL) +
  theme(legend.title=element_blank()) +
  ggtitle("Share of final consumption in total imports")
```



Miscellaneous

```
ea_gdp <-
  rdb("Eurostat/namq_10_gdp/Q.CP_MEUR.SCA.B1GQ.EA19") %>%
  select(period,gdp=value)

df <- rdb("ECB","TRD",mask = 'M.I8.Y.M+X.TTT.J8.4.VAL')

ea_trade <-
  df %>%
  transmute(period=paste(year(period),quarter(period),sep="-"),
            value,
            var=ifelse(grepl("Import",series_name),"imports","
exports")) %>%
  group_by(var,period) %>%
  summarize(value = sum(value)) %>%
  ungroup() %>%
  mutate(period=yq(period)) %>%
  spread(var,value) %>%
  left_join(ea_gdp) %>%
  transmute(period,
            imports_gdp=imports/(gdp*1000),
            exports_gdp=exports/(gdp*1000)) %>%
  gather(var,value,-period) %>%
  add_column(country="EA")

df <- rdb("OECD","EO",mask = "FRA+DEU+ITA+ESP.XGS+MGS+GDP.A")

imports_exports <-
  df %>%
  select(var=VARIABLE,period,value, country=LOCATION) %>%
  spread(var,value) %>%
  transmute(period, country,
            imports_gdp=MGS/GDP,
            exports_gdp=XGS/GDP) %>%
  gather(var,value,-period,-country) %>%
  mutate(country=case_when(
    country=="FRA" ~ "FR",
    country=="DEU" ~ "DE",
    country=="ITA" ~ "IT",
    country=="ESP" ~ "ES",
    TRUE ~ country)) %>%
  bind_rows(ea_trade)

alpha_delta_capital <-
  bind_rows(alpha_FIN,delta_FIN,capital_FIN) %>%
  mutate(country=case_when(
    country=="FRA" ~ "FR",
    country=="DEU" ~ "DE",
    country=="ITA" ~ "IT",
    country=="ESP" ~ "ES",
    TRUE ~ country))

leverage2 <-
  leverage %>%
  mutate(country=case_when(
    country=="EA19" ~ "EA",
    TRUE ~ country))

share <- rdb("Eurostat","nama_10_gdp",mask="A.CP_MPPS.B1GQ.DE+FR+IT+
ES+EA19")

share2 <-
  share %>%
  select(value,country=geo,period) %>%
  spread(country,value) %>%
  transmute(period,
            FR=FR/EA19,
            DE=DE/EA19,
            IT=IT/EA19,
            ES=ES/EA19) %>%
  gather(country,value,-period) %>%
  mutate(var="share")
```

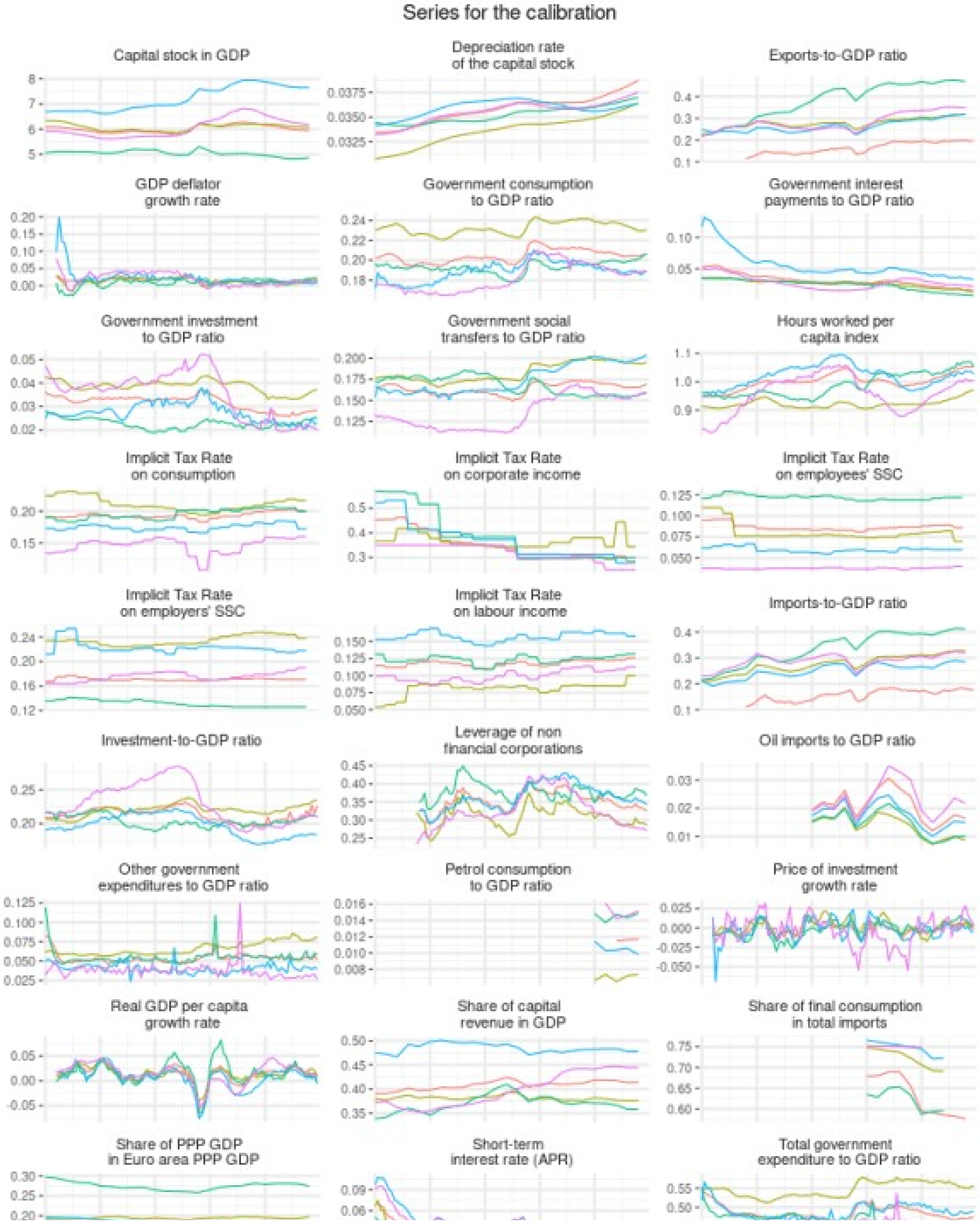
Final series for the calibration, and steady state values by country

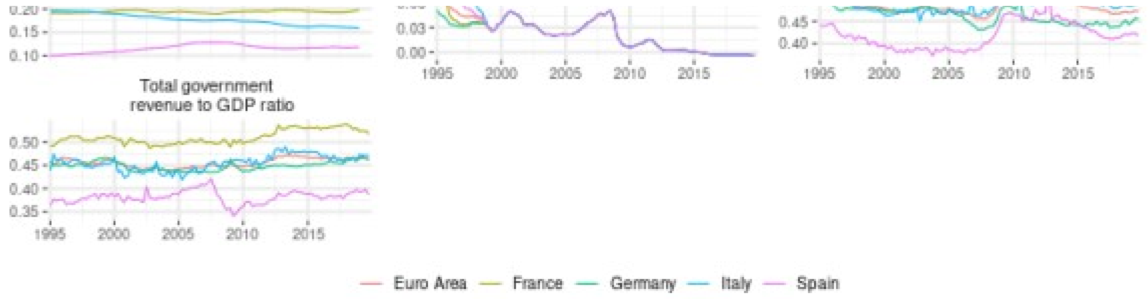
```
rawdata <-
  bind_rows(EA_rawdata_short,
            FR_rawdata,
            ES_rawdata,
            IT_rawdata,
            DE_rawdata) %>%
  left_join(itrq,by=c("country","period")) %>%
  select(period,country,pubcons,pubinves,tfs,totexp,totrev,
intpay,gdp,inves,tauk) %>%
  filter(period<=max(rawdata_df$period)) %>%
  gather(var,value,-country,-period) %>%
  bind_rows(plot_data_df) %>%
  select(-varname,-country_name)

hours_pc_meanEA <-
  rawdata %>%
  filter(var=="hours_pc",
         country=="EA") %>%
  summarise(value=mean(value,na.rm = T)) %>%
  first()

rawdata_growth_ratio <-
  rawdata %>%
  spread(var,value) %>%
  arrange(country) %>%
  transmute(period,
            country,
            defgdp_growth=defgdp/lag(defgdp,4)-1,
            gdp_rpc_growth=gdp_rpc/lag(gdp_rpc,4)-1,
            definves_growth=pinves_defl/lag(pinves_defl,4)-1,
            hours_pc_index=hours_pc/hours_pc_meanEA,
            tfs_gdp=tfs/(defgdp/100*gdp),
            pubcons_gdp=pubcons/(defgdp/100*gdp),
            pubinves_gdp=pubinves/(defgdp/100*gdp),
            totexp_gdp=totexp/(defgdp/100*gdp),
            otherexp_gdp=(totexp-tfs-pubcons-pubinves-intpay)/(
defgdp/100*gdp),
            intpay_gdp=intpay/(defgdp/100*gdp),
            totrev_gdp=totrev/(defgdp/100*gdp),
            inves_gdp=inves/gdp,
            shortrate=re,
            tauk,taun,tauwh,tauwf,tauc) %>%
  gather(var,value,-period,-country) %>%
  bind_rows(alpha_delta_capital,
            leverage2,
            imports_exports,
            imported_conso,
            share2,
            oil,
            petrol_conso) %>%
  filter(year(period) >= 1995,
         year(period) <= 2019) %>%
  mutate(varname=
           case_when(
             var=="alpha"            ~ "Share of capital \n revenue in
GDP",
             var=="defgdp_growth"    ~ "GDP deflator \n growth rate",
             var=="definves_growth"  ~ "Price of investment \n growth
rate",
             var=="delta"            ~ "Depreciation rate \n of the
capital stock" ,
             var=="capital_gdp"      ~ "Capital stock in GDP",
             var=="exports_gdp"      ~ "Exports-to-GDP ratio" ,
             var=="gdp_rpc_growth"   ~ "Real GDP per capita \n growth
rate" ,
             var=="hours_pc_index"   ~ "Hours worked per \n capita
index",
             var=="imports_gdp"      ~ "Imports-to-GDP ratio" ,
             var=="imported_conso"   ~ "Share of final consumption \n
in total imports",
             var=="oil_imports_gdp"  ~ "Oil imports to GDP ratio",
             var=="petrol_conso"     ~ "Petrol consumption \n to GDP
ratio ",
             var=="intpay_gdp"       ~ "Government interest \n payments
to GDP ratio",
             var=="inves_gdp"        ~ "Investment-to-GDP ratio",
             var=="leverage"         ~ "Leverage of non \n financial
corporations",
             var=="share"            ~ "Share of PPP GDP \n in Euro
area PPP GDP",
             var=="otherexp_gdp"     ~ "Other government \n
expenditures to GDP ratio",
             var=="pubcons_gdp"      ~ "Government consumption \n to
GDP ratio",
             var=="pubinves_gdp"     ~ "Government investment \n to GDP
ratio",
             var=="shortrate"        ~ "Short-term \n interest rate
(APR)",
             var=="taun"             ~ "Implicit Tax Rate \n on labour
income " ,
             var=="tauwh"            ~ "Implicit Tax Rate \n on
employees' SSC",
             var=="tauwf"            ~ "Implicit Tax Rate \n on
employers' SSC",
             var=="tauc"             ~ "Implicit Tax Rate \n on
consumption",
             var=="tauk"             ~ "Implicit Tax Rate \n on
corporate income",
             var=="tfs_gdp"          ~ "Government social \n transfers
to GDP ratio",
             var=="totexp_gdp"       ~ "Total government \n expenditure
to GDP ratio",
             var=="totrev_gdp"       ~ "Total government \n revenue to
GDP ratio"),
         country=
           case_when(
             country=="FR"           ~ "France",
             country=="DE"           ~ "Germany",
             country=="IT"           ~ "Italy",
             country=="ES"           ~ "Spain",
             country=="EA"           ~ "Euro Area"))
```

```
## Error in melt_dataframe(data, id_idx - 1L, gather_idx - 1L,
as.character(key_var), : All columns must be atomic vectors or lists.
Problem with column 6.
```

```
tikz('calibrated.tex', width=5.2,height=8.4, sanitize=TRUE)

ggplot(rawdata_growth_ratio,aes(period,value,color=country))+
  geom_line()+
  facet_wrap(~varname,ncol=3,scales = "free_y")+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  theme(legend.title=element_blank()) +
  theme(strip.text=element_text(size=8),
        axis.text=element_text(size=7))+
  ggtitle("Series for the calibration")+
  theme(plot.title=element_text(size=12))
```

```
dev.off()

## png
##   2

ggplot(rawdata_growth_ratio,aes(period,value,color=country))+
  geom_line()+
  facet_wrap(~varname,ncol=3,scales = "free_y")+
  scale_x_date(expand = c(0.01,0.01)) +
  theme + xlab(NULL) + ylab(NULL)+
  theme(legend.title=element_blank()) +
  theme(strip.text=element_text(size=11),
        axis.text=element_text(size=9))+
  ggtitle("Series for the calibration")+
  theme(plot.title=element_text(size=15))
```





```
calibration <-
  rawdata_growth_ratio %>%
  mutate(Parameter=varname) %>%
  group_by(country,Parameter) %>%
  summarise(mean=round(mean(value,na.rm = T),3)) %>%
  ungroup() %>%
  spread(country,mean)

kable(calibration, "html", caption = "Calibration") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
position = "center",full_width = FALSE)
```

Calibration

Parameter	Euro Area	France	Germany	Italy	Spain
Capital stock in GDP	6.012	6.084	5.042	7.215	6.063
Depreciation rate of the capital stock	0.036	0.034	0.035	0.036	0.036
Exports-to-GDP ratio	0.165	0.278	0.381	0.264	0.283
GDP deflator growth rate	0.016	0.012	0.010	0.023	0.020
Government consumption to GDP ratio	0.204	0.232	0.193	0.189	0.183
Government interest payments to GDP ratio	0.032	0.026	0.024	0.056	0.028
Government investment to GDP ratio	0.032	0.038	0.023	0.028	0.036
Government social transfers to GDP ratio	0.164	0.184	0.168	0.176	0.135
Hours worked per capita index	1.000	0.920	0.987	1.018	0.957
Implicit Tax Rate on consumption	0.193	0.212	0.196	0.175	0.146
Implicit Tax Rate on corporate income	0.350	0.368	0.386	0.370	0.319
Implicit Tax Rate on employees’ SSC	0.087	0.081	0.122	0.060	0.038
Implicit Tax Rate on employers’ SSC	0.171	0.235	0.132	0.223	0.176
Implicit Tax Rate on labour income	0.118	0.081	0.124	0.157	0.100
Imports-to-GDP ratio	0.158	0.276	0.337	0.250	0.289
Investment-to-GDP ratio	0.213	0.220	0.204	0.197	0.230
Leverage of non financial corporations	0.353	0.310	0.364	0.364	0.328
Oil imports to GDP ratio	0.021	0.014	0.015	0.018	0.023
Other government expenditures to GDP ratio	0.054	0.066	0.055	0.040	0.038
Petrol consumption to GDP ratio	0.012	0.007	0.015	0.011	0.015
Price of investment growth rate	-0.001	0.002	-0.002	-0.002	-0.003
Real GDP per capita growth rate	0.014	0.013	0.015	0.007	0.015
Share of capital revenue in GDP	0.408	0.381	0.369	0.484	0.398
Share of final consumption in total imports	0.632	0.723	0.622	0.738	0.750

Parameter	Euro Area	France	Germany	Italy	Spain
Share of PPP GDP in Euro area PPP GDP	NA	0.194	0.275	0.177	0.115
Short-term interest rate (APR)	0.023	0.022	0.021	0.027	0.025
Total government expenditure to GDP ratio	0.485	0.547	0.464	0.489	0.419
Total government revenue to GDP ratio	0.453	0.511	0.448	0.456	0.382

```
print(xtable(calibration, caption=c("Calibration"), type = "latex",
digits=c(0,0,3,3,3,3,3)), include.rownames = FALSE, file =
"calibration.tex")…
```