# 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 <-

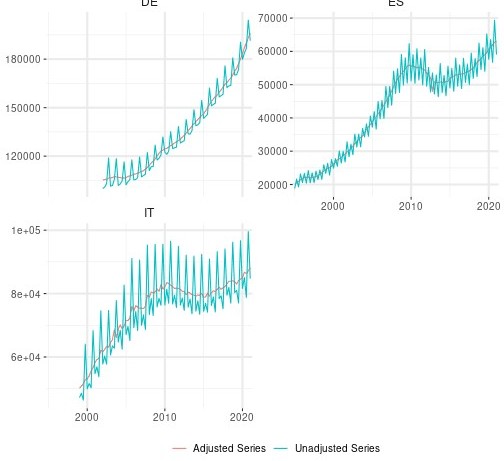
bind\_rows(df\_nsa\_q,deseasoned\_q) %>% na.omit()

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)

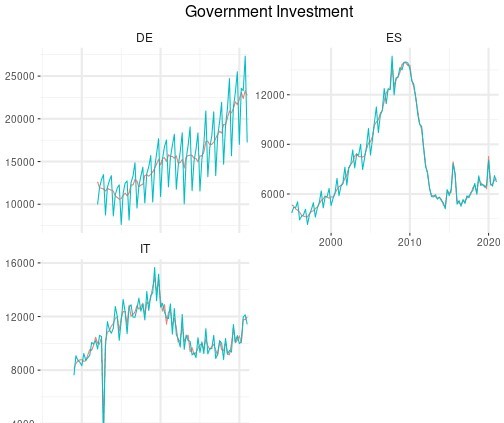
df\_nsa\_q %<>% mutate(Origin = "Unadjusted Series") plot\_df <-

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 <-

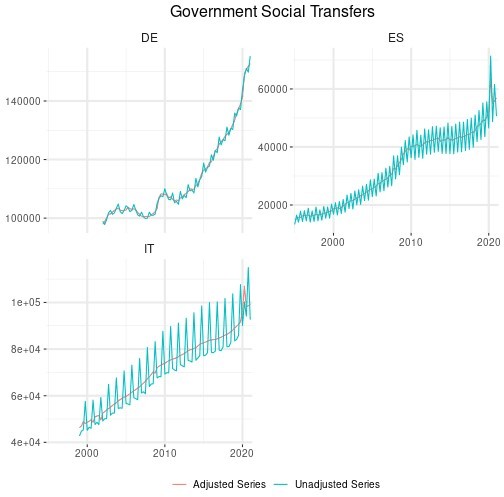
bind\_rows(df\_nsa\_q,deseasoned\_q) %>%

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)

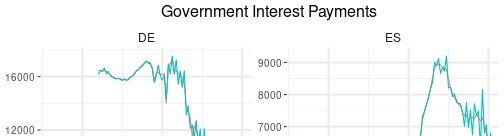
df\_nsa\_q %<>% mutate(Origin = "Unadjusted Series") plot\_df <-

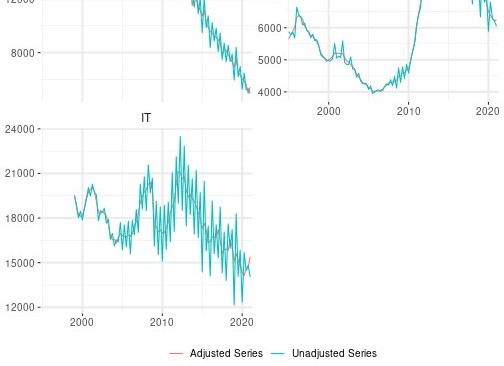
bind\_rows(df\_nsa\_q,deseasoned\_q) %>% na.omit()

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)

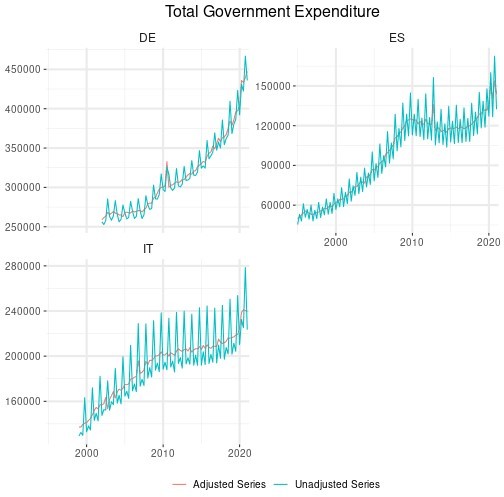
df\_nsa\_q %<>% mutate(Origin = "Unadjusted Series") plot\_df <-

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)

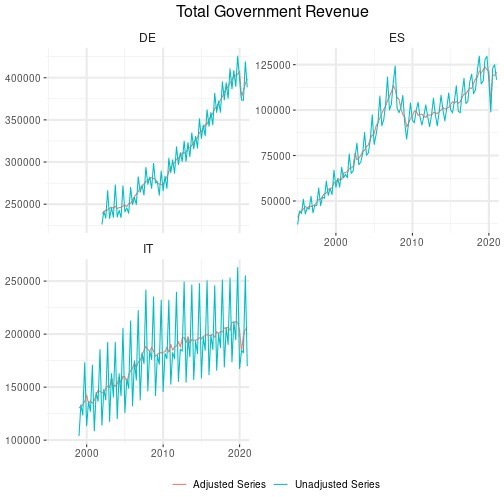
df\_nsa\_q %<>% mutate(Origin = "Unadjusted Series") plot\_df <-

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)

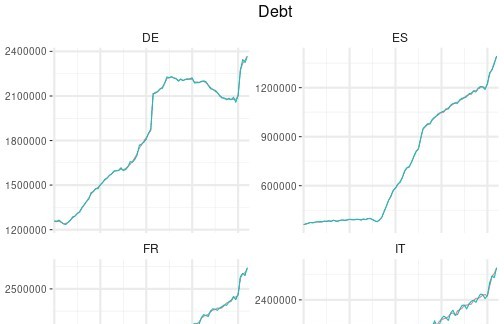
df\_nsa\_q %<>% mutate(Origin = "Unadjusted Series") plot\_df <-

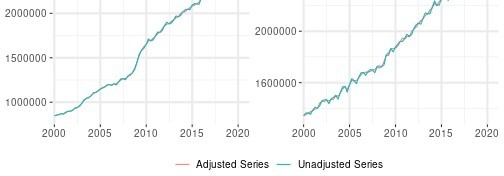
bind\_rows(df\_nsa\_q,deseasoned\_q) %>% na.omit()

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:

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 <-

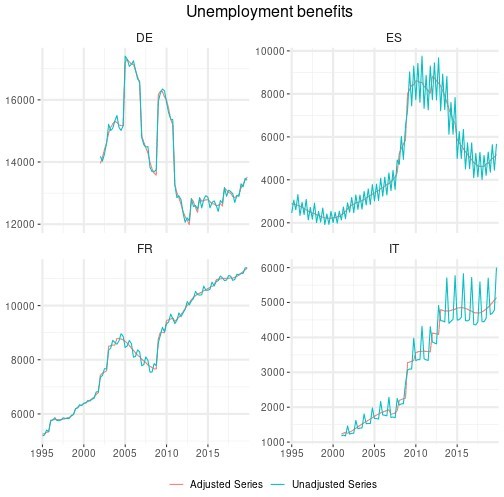
bind\_rows(unempbenef\_q,unempbenef\_q\_deseasoned) %>% na.omit()

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) %>%

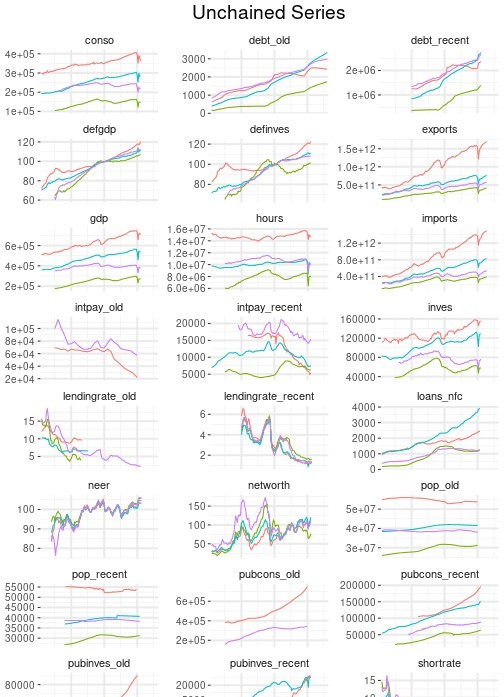
na.omit() %>% filter(year(period)>=1991) %>% arrange(var, period)

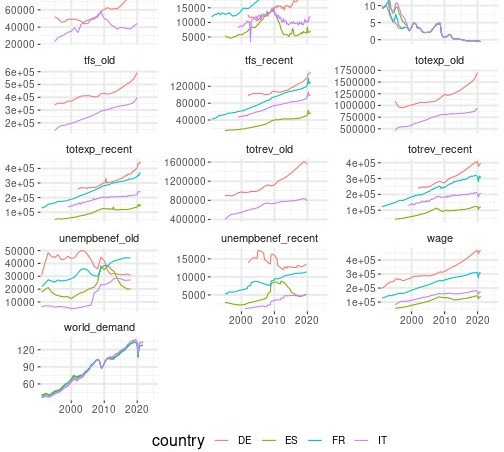
ggplot(df,aes(period,value,colour=country))+ geom\_line()+

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,

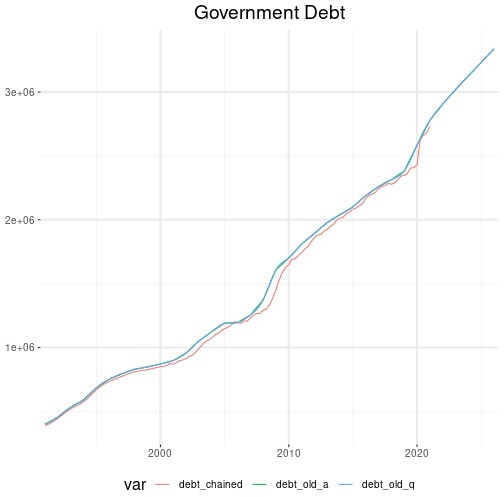
date\_chain=minDateDebtRecent) %>% mutate(var="debt")

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,

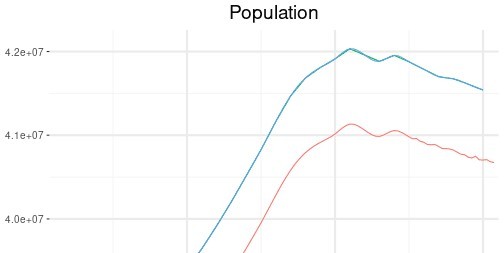
to\_rebase= pop\_old\_q, date\_chain="2015-01-01")

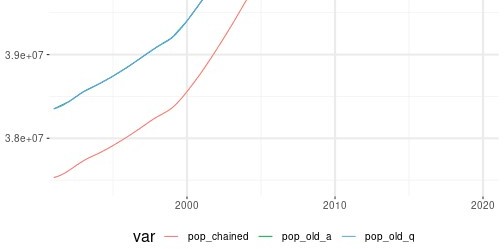
plot\_df <- bind\_rows(mutate(pop\_old\_a,var="pop\_old\_a"),

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")





### 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,

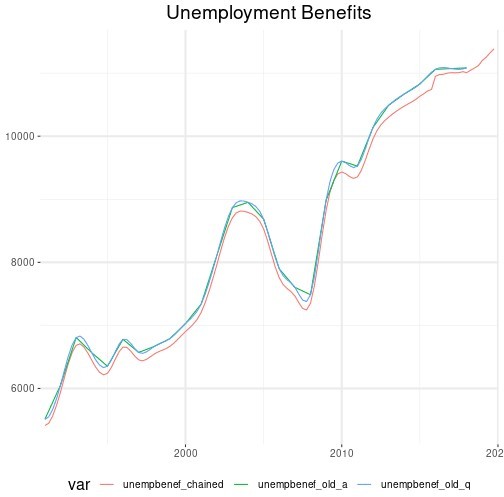
to\_rebase= unempbenef\_old\_q, date\_chain="2015-01-01")

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")

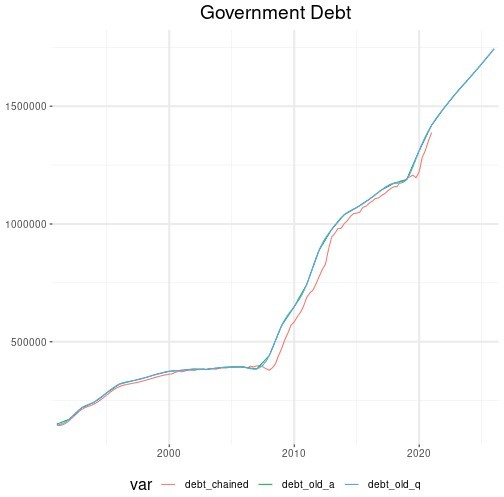
plot\_df <- bind\_rows(add\_column(debt\_old\_a,var="debt\_old\_a"),

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,

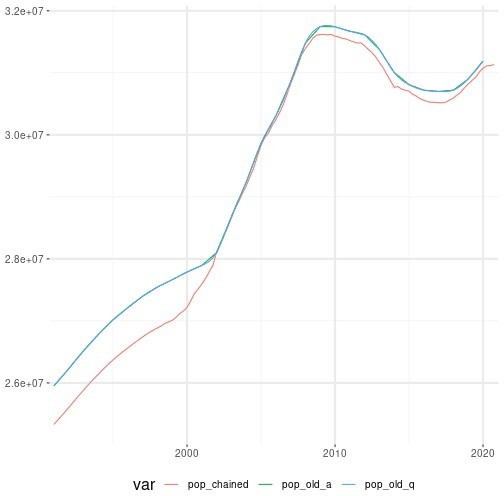
to\_rebase= pop\_old\_q, date\_chain="1998-01-01")

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")

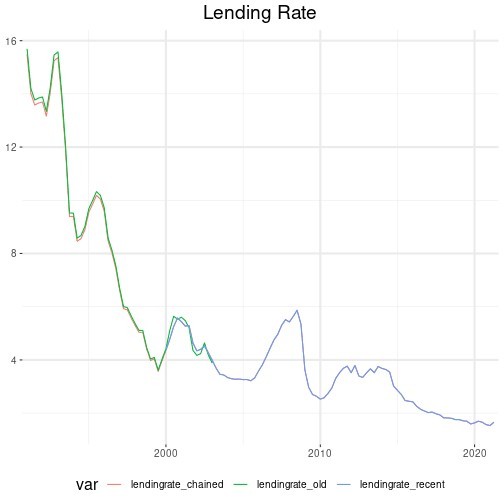
plot\_df <- bind\_rows(mutate(lendingrate\_old,var="lendingrate\_old"),

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,

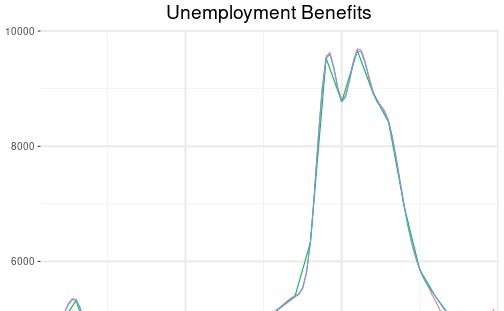
to\_rebase= unempbenef\_old\_q, date\_chain="2015-01-01")

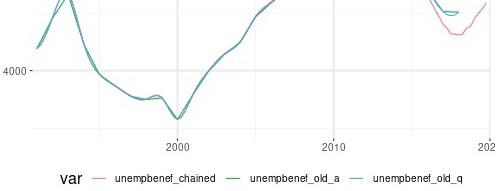
plot\_df <- bind\_rows(mutate(unempbenef\_old\_a,var="unempbenef\_old\_a"),

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,

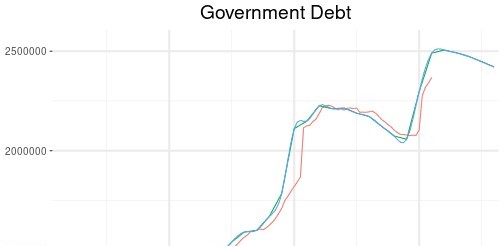
date\_chain=minDateDebtRecent) %>% mutate(var="debt")

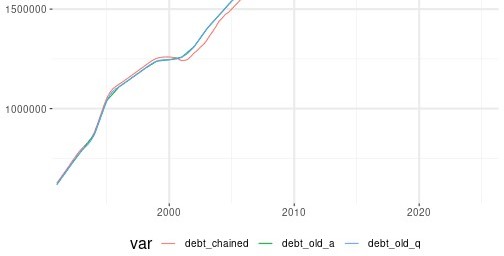
plot\_df <- bind\_rows(add\_column(debt\_old\_a,var="debt\_old\_a"),

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,

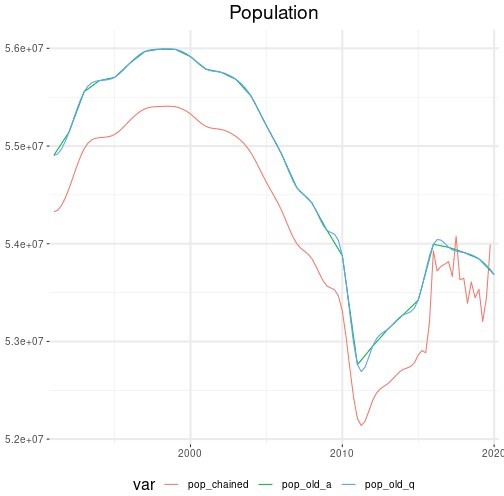
to\_rebase= pop\_old\_q, date\_chain="2015-01-01")

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")

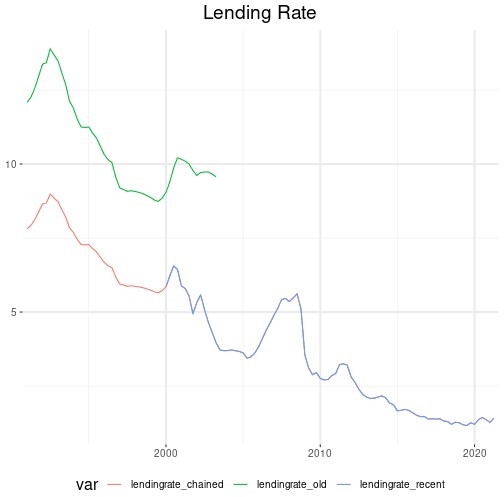
plot\_df <- bind\_rows(mutate(lendingrate\_old,var="lendingrate\_old"),

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,

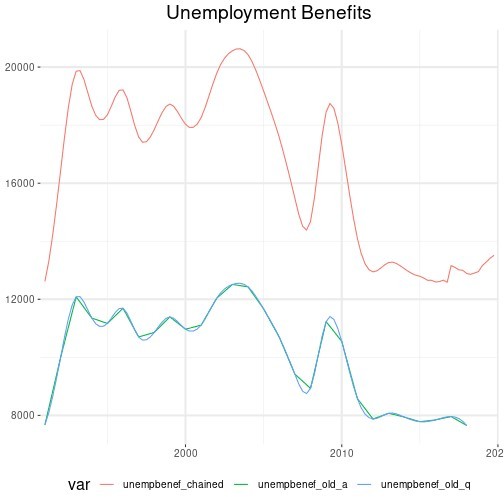
to\_rebase= unempbenef\_old\_q, date\_chain="2015-01-01")

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,

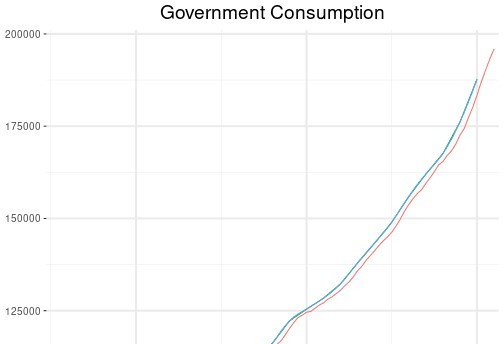
to\_rebase= pubcons\_old\_q, date\_chain=minDatePubConsRecent)

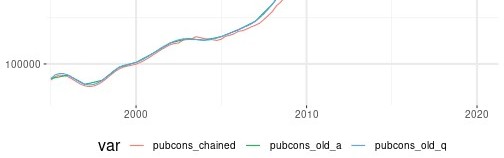
plot\_df <- bind\_rows(mutate(pubcons\_old\_a,var="pubcons\_old\_a"),

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)

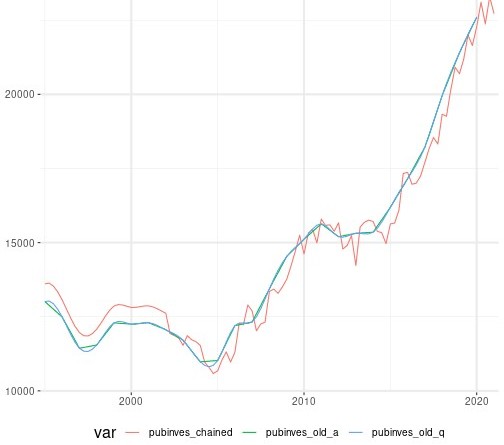
plot\_df <- bind\_rows(mutate(pubinves\_old\_a,var="pubinves\_old\_a"),

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,

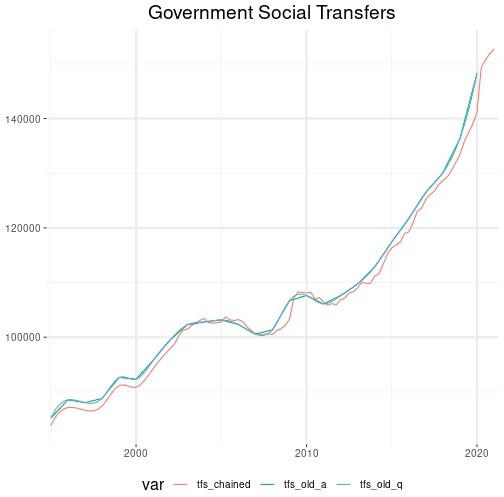
to\_rebase= tfs\_old\_q, date\_chain=minDateTfsRecent)

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,

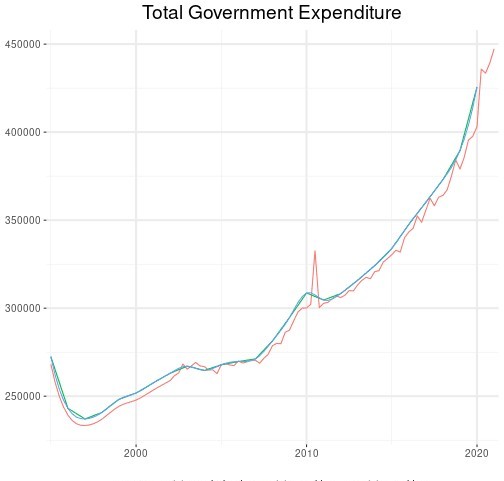
to\_rebase= totexp\_old\_q, date\_chain=minDateTotExpRecent)

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,

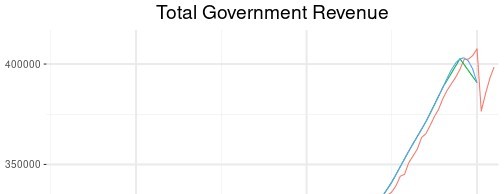
to\_rebase= totrev\_old\_q, date\_chain=minDateTotRevRecent)

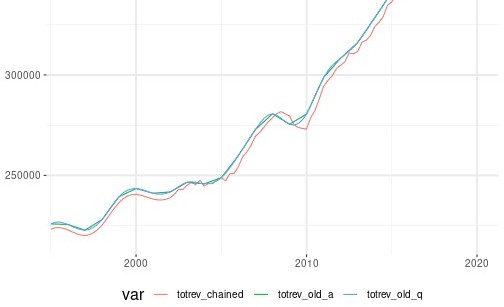
plot\_df <- bind\_rows(mutate(totrev\_old\_a,var="totrev\_old\_a"),

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,

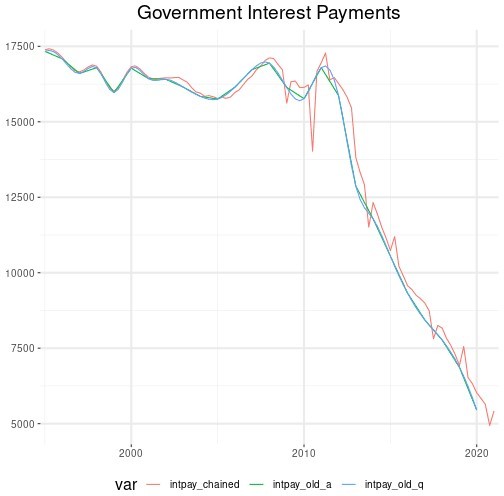
to\_rebase= intpay\_old\_q, date\_chain=minDateIntPayRecent)

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)

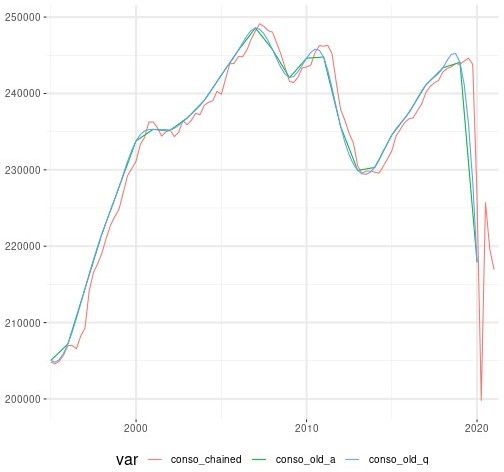
plot\_df <- bind\_rows(mutate(conso\_old\_a,var="conso\_old\_a"),

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,

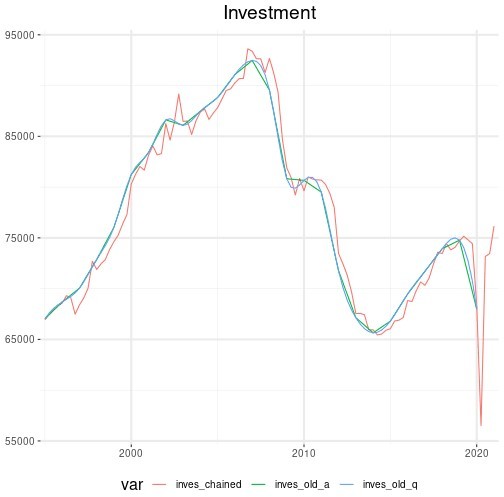
to\_rebase= inves\_old\_q, date\_chain=minDateInvesRecent)

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,

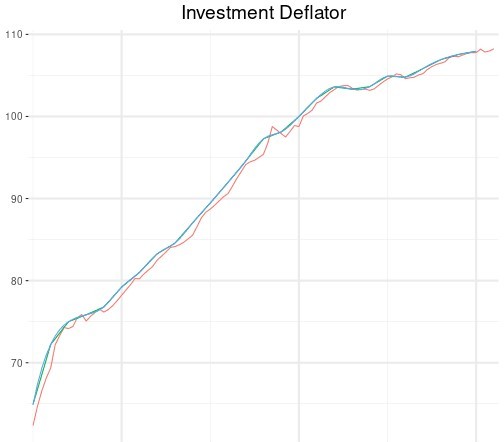
to\_rebase= definves\_old\_q, date\_chain=minDateDefInvesRecent)

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")

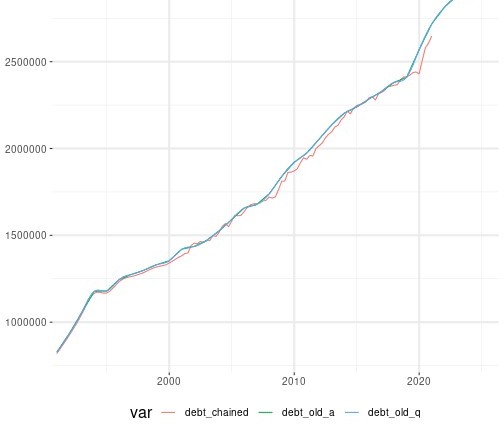
plot\_df <- bind\_rows(add\_column(debt\_old\_a,var="debt\_old\_a"),

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,

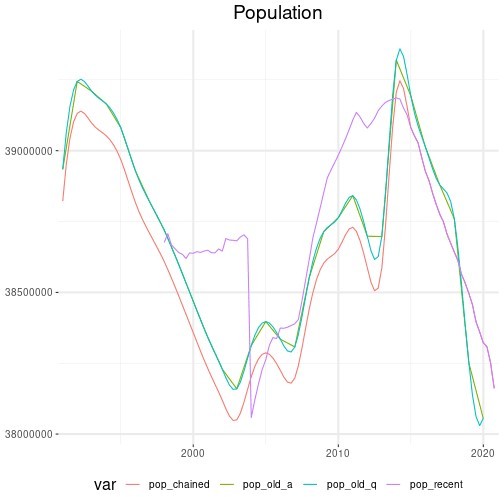
to\_rebase= pop\_old\_q, date\_chain="2015-01-01")

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")

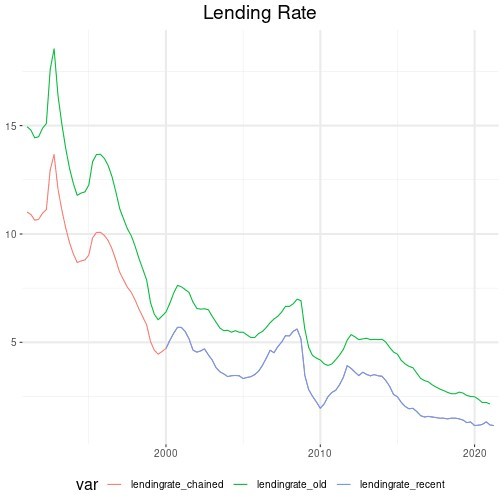
plot\_df <- bind\_rows(mutate(lendingrate\_old,var="lendingrate\_old"),

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,

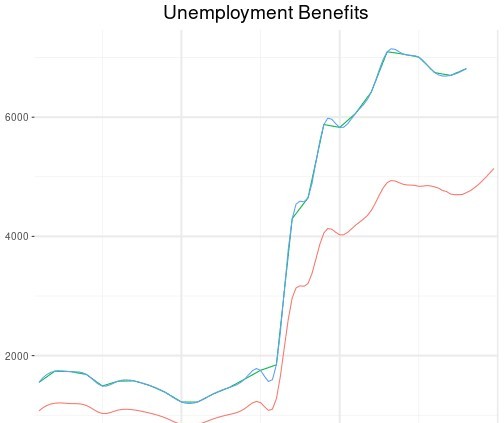
to\_rebase= unempbenef\_old\_q, date\_chain="2015-01-01")

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)

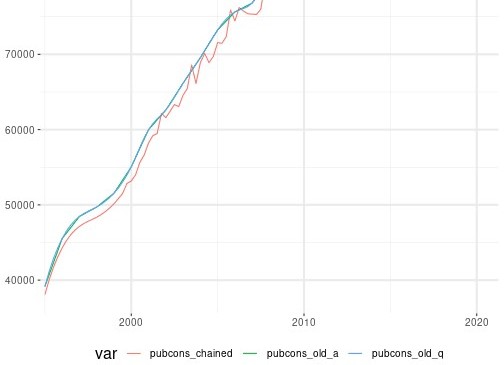
plot\_df <- bind\_rows(mutate(pubcons\_old\_a,var="pubcons\_old\_a"),

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,

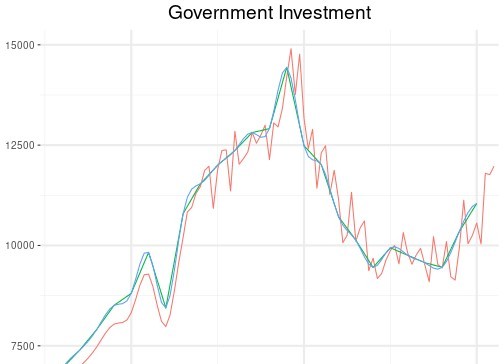
to\_rebase= pubinves\_old\_q, date\_chain="2003-01-01")

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)

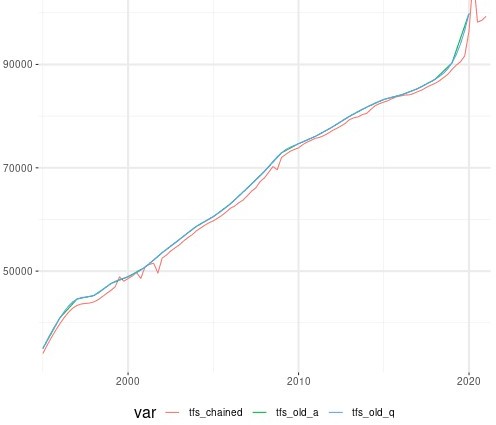
plot\_df <- bind\_rows(mutate(tfs\_old\_a,var="tfs\_old\_a"),

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,

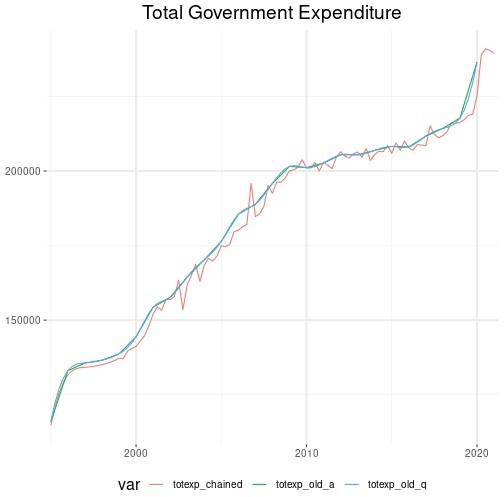
date\_chain=minDateTotExpRecent)

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,

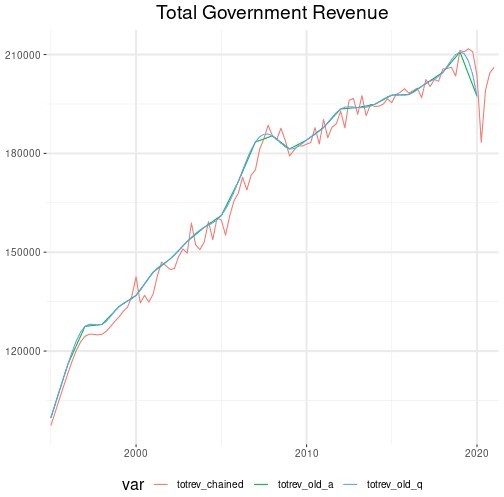
to\_rebase= totrev\_old\_q, date\_chain=minDateTotRevRecent)

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)

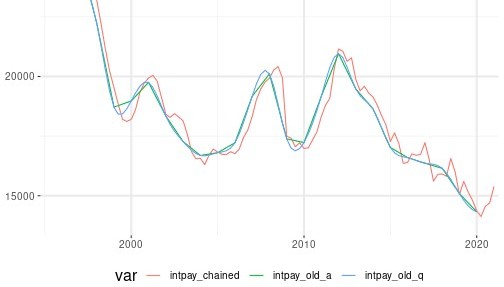
plot\_df <- bind\_rows(mutate(intpay\_old\_a,var="intpay\_old\_a"),

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:

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", capita",

capita",

(APR)",

capita", capita", capita",

var=="inves\_rpc" ~ "Real investment \n per

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

var=="networth\_rpc" ~ "Real net worth \n per capita", var=="re" ~ "Short-term \n interest rate

var=="creditspread" ~ "Credit spread (APP)", var=="pubcons\_rpc" ~ "Real public consumption\n per

var=="pubinves\_rpc" ~ "Real public investment\n per var=="tfs\_rpc" ~ "Real social transfers\n per

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()

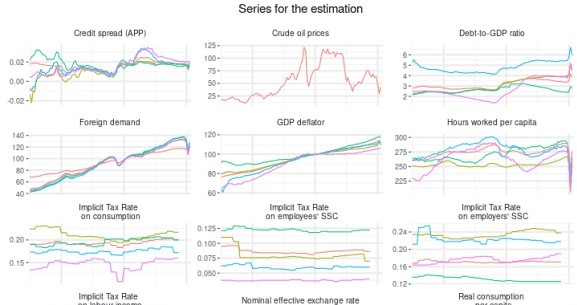
## png ## 2

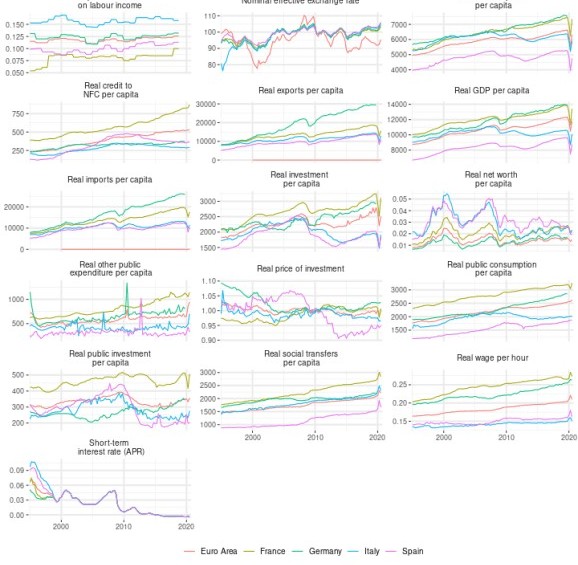
ggplot(plot\_data\_df,aes(period,value,col=country\_name))+ geom\_line()+ facet\_wrap(~varname,ncol=3,scales="free\_y")+ scale\_x\_date(expand = c(0.01,0.01)) +

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)

# 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) %>%

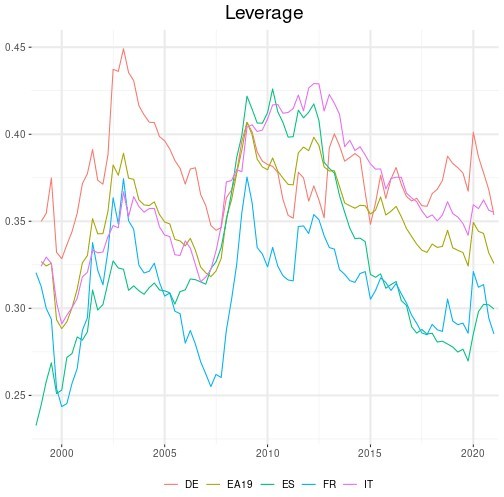
arrange(country,period) %>% transmute(period,

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)

listcountry <- list("France" = "FRA",

"Germany" = "DEU",

"Italy" = "ITA",

"Spain" = "ESP",

"Euro Area" = "EA")

df <- read\_dta("https://[www.rug.nl/ggdc/docs/pwt100.dta](http://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 %>%

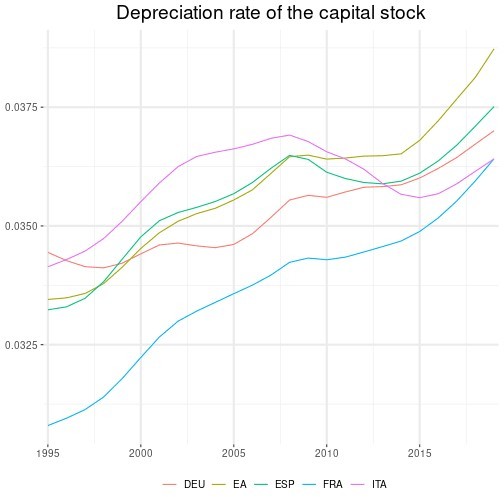
filter(grepl('FRA|DEU|ITA|ESP', country))

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 %>%

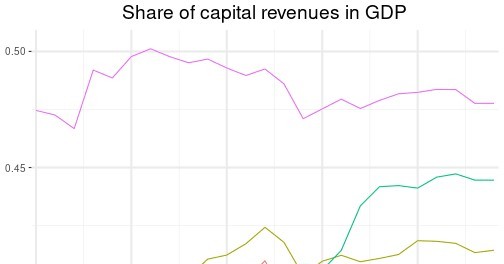
filter(grepl('FRA|DEU|ITA|ESP', country))

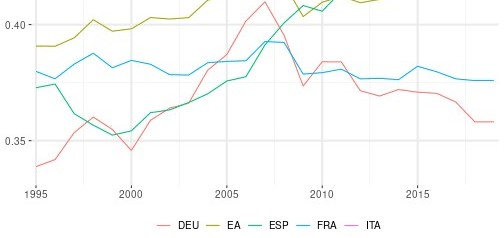
alpha\_FIN <- bind\_rows(alpha\_countries,alpha\_EA)

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")

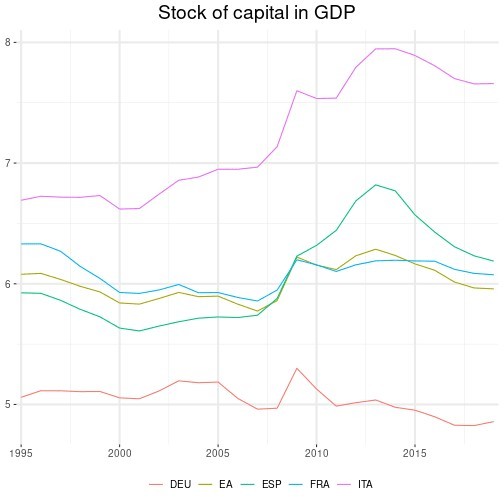
capital\_countries <- capital %>%

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))

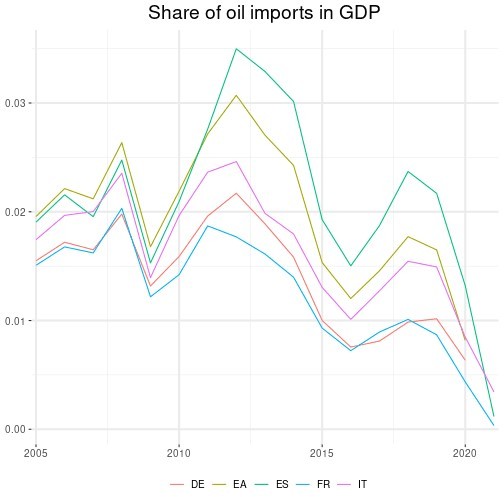
oil <-

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 %>%

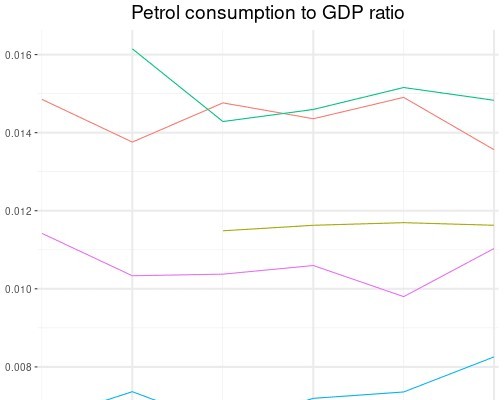
left\_join(petrol\_weight,by=c("country","period")) %>% na.omit() %>%

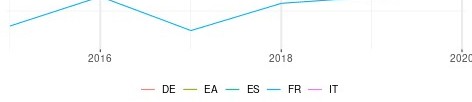
transmute(period,

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)) %>%

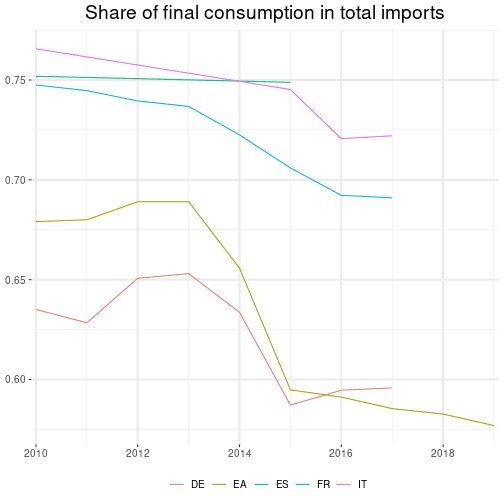
select(-P3,-P51G) %>% filter(year(period)>=2010) %>% mutate(country=case\_when(country=="EA19" ~ "EA",

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",

rate",

var=="defgdp\_growth" ~ "GDP deflator \n growth rate", var=="definves\_growth" ~ "Price of investment \n growth

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" , index",

var=="hours\_pc\_index" ~ "Hours worked per \n capita

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", ratio", (APR)",

income " ,

var=="pubinves\_gdp" ~ "Government investment \n to GDP var=="shortrate" ~ "Short-term \n interest rate var=="taun" ~ "Implicit Tax Rate \n on labour

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()

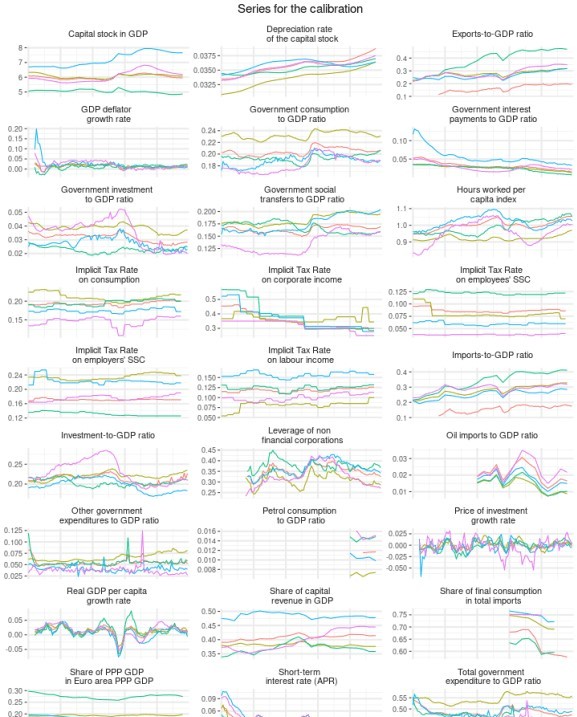
## png ## 2

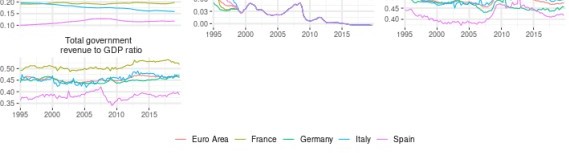
ggplot(rawdata\_growth\_ratio,aes(period,value,color=country))+ geom\_line()+

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")…