ENSAE
IP PARIS

Applied macroeconometrics
Les effets d'une hausse de l'Euribor 3-mois

Valentin Giust, Gautier Lenfant et Alain Quartier-la-Tente

Contents

The source code and this repport is also available at https://github.com/AQLT/AppliedMacroEuribor.

Introduction

In this essay, we build on a VAR model to quantify the effects of variations in the Euribor-3M (benchmark short term market rate in Europe) rate on a series of key macroeconomic variables (GDP, inflation, unemployment, investment) both at the Euro Area (EA) level, and in the particular case of France. We map out different scenarios of rise and falls in the Euribor-3M to assess the response of the economy to monetary shocks of different kinds and amplitudes. Our model predicts that a one standard-deviation rise in the Euribor-3M rate is associated with a significant and lingering drift of the EA GDP down its no-shock trend. Further, we find evidence that a rise in the short term monetary rate brings conventional inflation below its no-shock trend for a substantial period of time. We also find that a rising Euribor-3M rate has a detrimental effect on both the EA and the French labor markets. PARLER DES SCENARII

When it comes to the relationship between the short-term rate and the main macroeconomic variables, our VAR-based findings are consistent with the theoretical predictions of the Neo-keynesian framework : a tightening of monetary policy has a negative impact on aggregate demand, which deteriorates employment and brings down inflation in a significant fashion.

In the first part, we present our data set and provide the origin of our empirical resources. We then present our empirical strategy and the features of our specification in the second part. Finally, the last part exhibits the results of our estimations.

Data setting

Data origin and features

ORIGIN

We extensively rely on macroeconomic data provided by official statistical offices... We choose a time frame spanning 1997-2018. Why this time span ?

GDP : real, SA, EURO2005? à vérifier FBCE : real, SA Our data on unemployment originates from the Eurostat Database

FEATURES & STATIONARITY

Before implementing the VAR model, we make sure that our variables feature stationarity. Why ? We thus detrend consumption, unemployment ... The following KPSS tests provides strong evidence for the stationarity of our macroeconomic variables.

Graphs

A few stylized facts

In this subsection, we exhibit a few stylized facts characterizing the macroeconomic trends in the EA between 1997 and 2018. Likewise, we give similar facts in the particular case of France.

ALL GRAPHS BOUND TOGETHER DANS UN FRAME ENTRE 1997 et 2018

Graphs short term rate

Graphs variables of aggregate demand : FBCF, Consumption, GDP consistent with the theory, FBCF varies more than consumption?

Graphs unemployment

Graphs price variables

Comment :

Upon careful observation of the above graphs, it is clear that the macroeconomic history of the EA splits out in two periods, respectively spanning 1997-2008, and 2008-2018. The 2008 financial crisis is a watershed marking the start of a period of weaker macroeconomic performance, whether it be for France or the EA as a whole. It is worth noting that the macroeconomic situation is particularly depressed between 2012 and 2015 amid the austerity policies which

were implemented in the wake of the sovereign debt crisis. How is France wrt. EA? France performs worse that the EA ! Parler justement de ce qu'on observe entre les taux et les macro variables. A noteworthy observation is that the EA economy fell deeper in the wake of the 2011 policy rate rise, which translated into a rise in the Euribor rate.
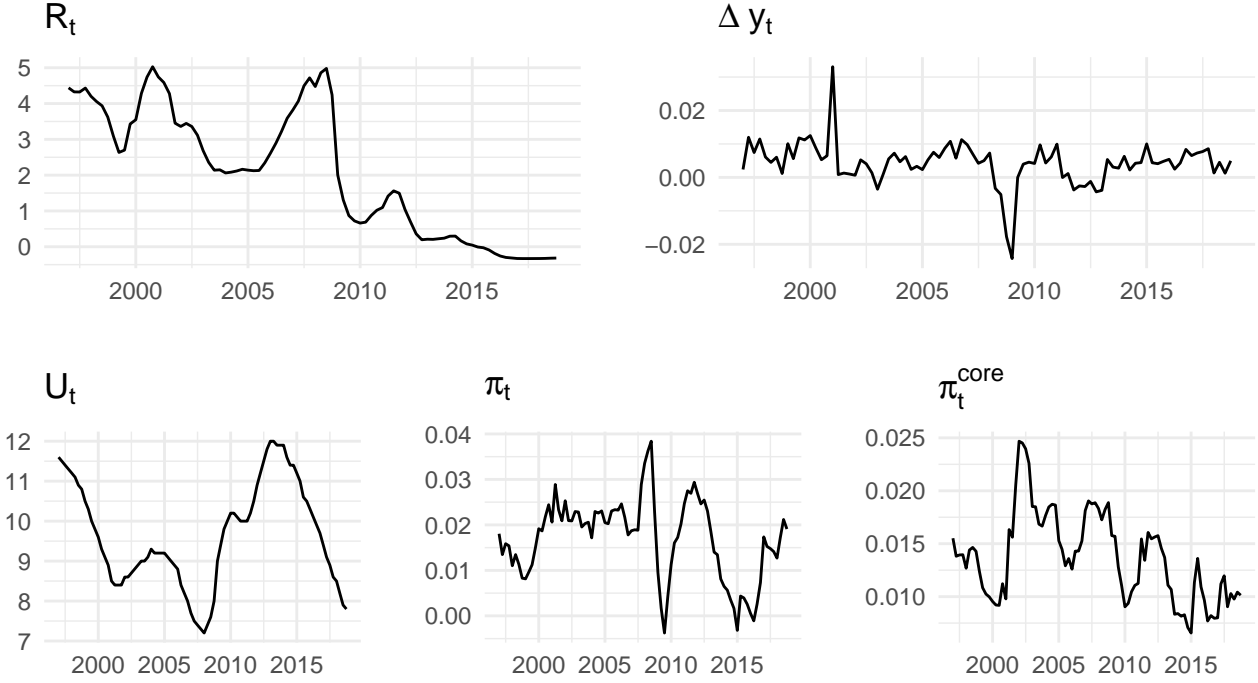

Figure 1: European Area data plots.

1 Empirical strategy

In this section, we describe and explain the features of our VAR specification. In the this report, we will use the following notations:

- y_t for the log of the GDP;
- U_t for the unemployment;
- π_t for the inflation (year on year evolution of the HICP);
- π_t^{core} for the core inflation (year on year evolution);
- R_t for 3 months EURIBOR rate.

The same specification is used for the European Area and the French dataset:

- We estimate a VAR model with a constant and a linear trend.

- The number of lags of the model is determined by minimizing the Schwarz information criterion: we find that 1 lag is enough to have a well specified model (no serial correlation, stability of the estimates and no ARCH effect).

Equations (1) and (2) shows the estimated parameters and the t-sat of the VAR model, respectively for the European Aera and for the French data.

$$
\begin{pmatrix} \Delta y_t \\ U_t \\ \pi_t^{core} \\ \pi_t \\ R_t \end{pmatrix} = \begin{pmatrix} 0.03 & -0.00 \\ (0.01) & (0.00) \\ -0.76 & 0.00 \\ (0.32) & (0.00) \\ 0.00 & 0.00 \\ (0.00) & (0.00) \\ 0.01 & -0.00 \\ (0.01) & (0.00) \\ 0.51 & -0.00 \\ (0.56) & (0.00) \end{pmatrix} \begin{pmatrix} 1 \\ t \end{pmatrix} + \begin{pmatrix} 0.24 & -16.20 & 0.08 & 0.32 & 31.29 \\ (0.11) & (3.35) & (0.03) & (0.07) & (5.84) \\ -0.00 & 1.03 & -0.00 & -0.00 & -0.03 \\ (0.00) & (0.02) & (0.00) & (0.00) & (0.03) \\ -0.48 & 18.55 & 0.85 & -0.00 & -11.01 \\ (0.21) & (6.55) & (0.07) & (0.14) & (11.44) \\ -0.11 & 4.25 & 0.04 & 0.90 & 10.51 \\ (0.09) & (2.75) & (0.03) & (0.06) & (4.79) \\ -0.00 & 0.02 & 0.00 & -0.00 & 0.89 \\ (0.00) & (0.03) & (0.00) & (0.00) & (0.05) \end{pmatrix} \begin{pmatrix} \Delta y_{t-1} \\ U_{t-1} \\ \pi_{t-1}^{core} \\ \pi_{t-1} \\ R_{t-1} \end{pmatrix} + \varepsilon_t^{EA} \quad (1)
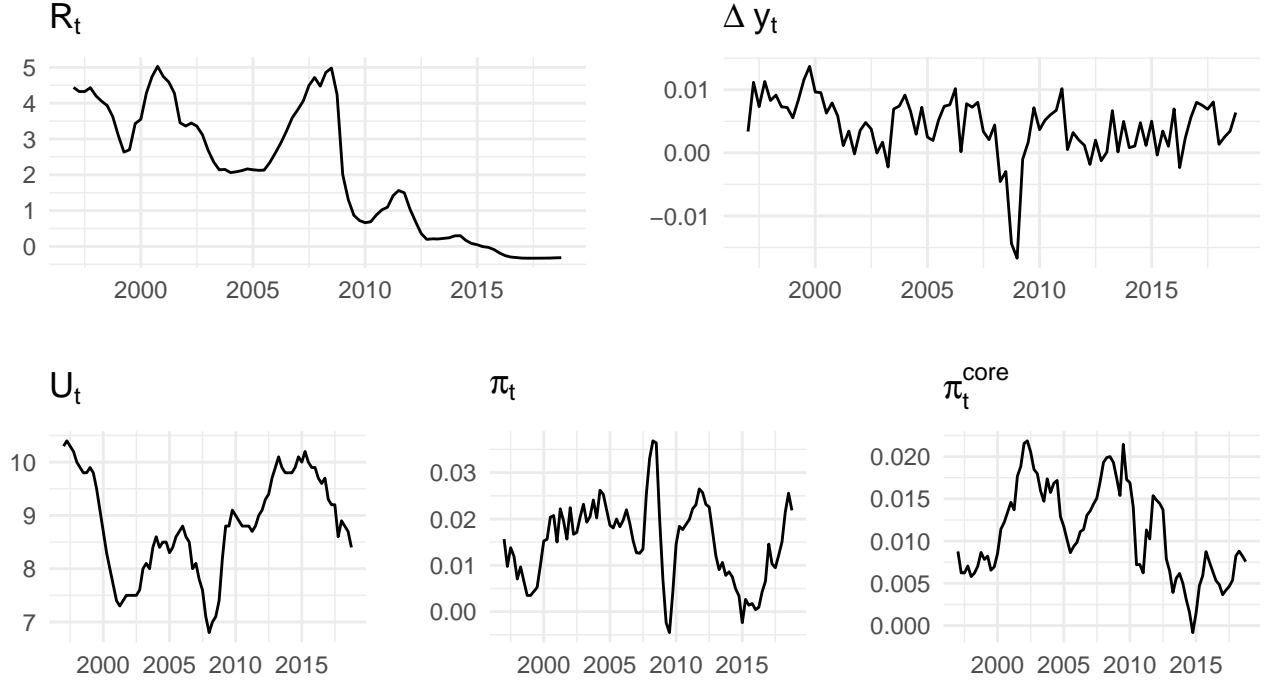$$

Figure 2: French data plots.

$$
\begin{pmatrix} \Delta y_t \\ U_t \\ \pi_t^{core} \\ \pi_t \\ R_t \end{pmatrix} = \begin{pmatrix} 0.02 & -0.00 \\ \scriptstyle(0.01) & \scriptstyle(0.00) \\ 0.18 & -0.00 \\ \scriptstyle(0.60) & \scriptstyle(0.00) \\ 0.02 & 0.00 \\ \scriptstyle(0.01) & \scriptstyle(0.00) \\ 0.02 & 0.00 \\ \scriptstyle(0.01) & \scriptstyle(0.00) \\ 0.91 & -0.00 \\ \scriptstyle(0.88) & \scriptstyle(0.00) \end{pmatrix} \begin{pmatrix} 1 \\ t \end{pmatrix} + \begin{pmatrix} 0.32 & -27.97 & 0.06 & 0.53 & 45.45 \\ \scriptstyle(0.10) & \scriptstyle(5.42) & \scriptstyle(0.05) & \scriptstyle(0.11) & \scriptstyle(7.91) \\ -0.00 & 1.01 & -0.00 & -0.00 & -0.10 \\ \scriptstyle(0.00) & \scriptstyle(0.05) & \scriptstyle(0.00) & \scriptstyle(0.00) & \scriptstyle(0.07) \\ -0.24 & 5.34 & 0.74 & 0.18 & -14.48 \\ \scriptstyle(0.13) & \scriptstyle(7.41) & \scriptstyle(0.07) & \scriptstyle(0.15) & \scriptstyle(10.81) \\ -0.04 & 4.24 & -0.06 & 0.75 & 3.97 \\ \scriptstyle(0.06) & \scriptstyle(3.23) & \scriptstyle(0.03) & \scriptstyle(0.06) & \scriptstyle(4.72) \\ -0.00 & -0.05 & 0.00 & -0.00 & 0.94 \\ \scriptstyle(0.00) & \scriptstyle(0.03) & \scriptstyle(0.00) & \scriptstyle(0.00) & \scriptstyle(0.05) \end{pmatrix} \begin{pmatrix} \Delta y_{t-1} \\ U_{t-1} \\ \pi_{t-1}^{core} \\ \pi_{t-1} \\ R_{t-1} \end{pmatrix} + \varepsilon_t^{FR} \qquad (2)
$$

Equation (3) shows the estimates of the covariance matrix of the residuals. We can notice that those matrices are not diagonal: this means that a shock on a component is correlated to a shock on the other components. It implies that we cannot directly estimate the impact of a shock on the 3 months Euribor rates, R_t. To do so, we need to make some assumptions on the propagation of the shocks that are described in section 2.

$$
\mathbb{V}\varepsilon_t^{EA} = \begin{pmatrix} 1.00 & -0.61 & -0.21 & 0.06 & 0.41 \\ -0.61 & 1.00 & 0.07 & 0.02 & -0.41 \\ -0.21 & 0.07 & 1.00 & 0.20 & -0.06 \\ 0.06 & 0.02 & 0.20 & 1.00 & 0.25 \\ 0.41 & -0.41 & -0.06 & 0.25 & 1.00 \end{pmatrix} \quad \mathbb{V}\varepsilon_t^{FR} = \begin{pmatrix} 1.00 & -0.44 & -0.06 & 0.24 & 0.48 \\ -0.44 & 1.00 & -0.06 & -0.23 & -0.40 \\ -0.06 & -0.06 & 1.00 & 0.13 & 0.00 \\ 0.24 & -0.23 & 0.13 & 1.00 & 0.31 \\ 0.48 & -0.40 & 0.00 & 0.31 & 1.00 \end{pmatrix} \qquad (3)
$$

2 Estimations of a shock on the 3 months Euribor rate

In this section, we depict the results of the our estimation endeavor.

2.1 A demand oriented set of scenarios

In the first scenario, demand oriented, we suppose that Euribor is the primary force at play. GDP reacts contemporaneously to Euribor and impacts all the other variables. Unemployment follows contemporaneously. Both inflation and core inflation follow with delay.

Statistically speaking, this scenario is equivalent to reorder the variables in the order $\begin{pmatrix} R_t & \Delta y_t & U_t & \pi_t^{core} & \pi_t \end{pmatrix}$ and to estimate a matrix B_1, such has $\mathbb{V}\varepsilon_t = B_1 B_1'$ and with the following constraints:

$$B_1 \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ * & 1 & 0 & 0 & 0 \\ * & * & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & * & 1 \end{pmatrix}$$

Using the data, we found the following B matrices[1]:

$$B_1^{EA} = \begin{pmatrix} 0.288 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.002 & 0.005 & 0.000 & 0.000 & 0.000 \\ -0.068 & -0.080 & 0.128 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.002 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.001 & 0.003 \end{pmatrix} \quad B_1^{FR} = \begin{pmatrix} 0.284 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.002 & 0.003 & 0.000 & 0.000 & 0.000 \\ -0.078 & -0.054 & 0.170 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.002 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.001 & 0.004 \end{pmatrix}$$

2.2   A supply oriented set of scenarios

In the second scenario, supply oriented, we will suppose that GDP is the primary force at play: a shock in the GDP contemporaneously affects all the variables. Then unemployment follows contemporaneously but a shock on the unemployment only affects inflation and core inflation with delay. Finally, the Euribor is contemporaneously affected by a shock on the other variables but the opposite is not true (a shock in the Euribor will only affect with delay the other components).

Statistically speaking, this scenario is equivalent to use the same order of the variables than in equations (1) and (2)and to estimate a matrix B_2, such has $\mathbb{V}\varepsilon_t = B_2 B_2'$ and with the following constraints:

$$B_2 \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ * & 1 & 0 & 0 & 0 \\ * & 0 & 1 & 0 & 0 \\ * & 0 & * & 1 & 0 \\ * & * & * & * & 1 \end{pmatrix}$$

Using the data, we found the following B matrices[2]:

$$B_2^{EA} = \begin{pmatrix} 0.005 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.101 & 0.131 & 0.000 & 0.000 & 0.000 \\ -0.000 & 0.000 & 0.002 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.001 & 0.003 & 0.000 \\ 0.119 & -0.064 & 0.002 & 0.072 & 0.246 \end{pmatrix} \quad B_2^{FR} = \begin{pmatrix} 0.003 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.085 & 0.175 & 0.000 & 0.000 & 0.000 \\ -0.000 & 0.000 & 0.002 & 0.000 & 0.000 \\ 0.001 & 0.000 & 0.001 & 0.004 & 0.000 \\ 0.135 & -0.054 & 0.004 & 0.051 & 0.237 \end{pmatrix}$$

2.3   Impulse response functions

In the demand scenario, upon careful examination of the IRFs (figures 3 and 4), we notice that a unitary shock in the Euribor has a significant effect contemporaneous effect in GDP (positively) and in the unemployment (negatively). It implies a backlash in the GDP that decreases in the first year and then slowly returns to normal. This mechanism involves a decrease in the unemployment in the first quarters and then returns to normal, and an increase in the core inflation and in the inflation during the first year. The impact is higher in the core inflation than in the inflation, the first being more sensitive to the changes in GDP and in unemployment. The Euribor continues to increase in the first quarter after the shock, then begin to decreases and returns to normal after 3 years. The same evolutions are observes for the French economy and for the European Area. However, in the European Area economy, we observe a lower impact of the shock and a faster return to normal.

In the supply scenario, upon careful examination of the IRFs (figures 5 and 6), we notice that a unitary shock in the Euribor has no significant effect on the unemployment and the core inflation. However, this shocks involves a

---

[1]Note that not imposing any coefficient in the lower triangular matrix B_1 to be null (and so to compute a Choleski decomposition) would not change the results. Indeed, in that case the estimates of the associated coefficients are closed to zero.

[2]Note that, as in the previous section, not imposing any coefficient in the lower triangular matrix B_2 to be null (and so to compute a Choleski decomposition) would not change the results. Indeed, in that case the estimates of the associated coefficients are closed to zero.
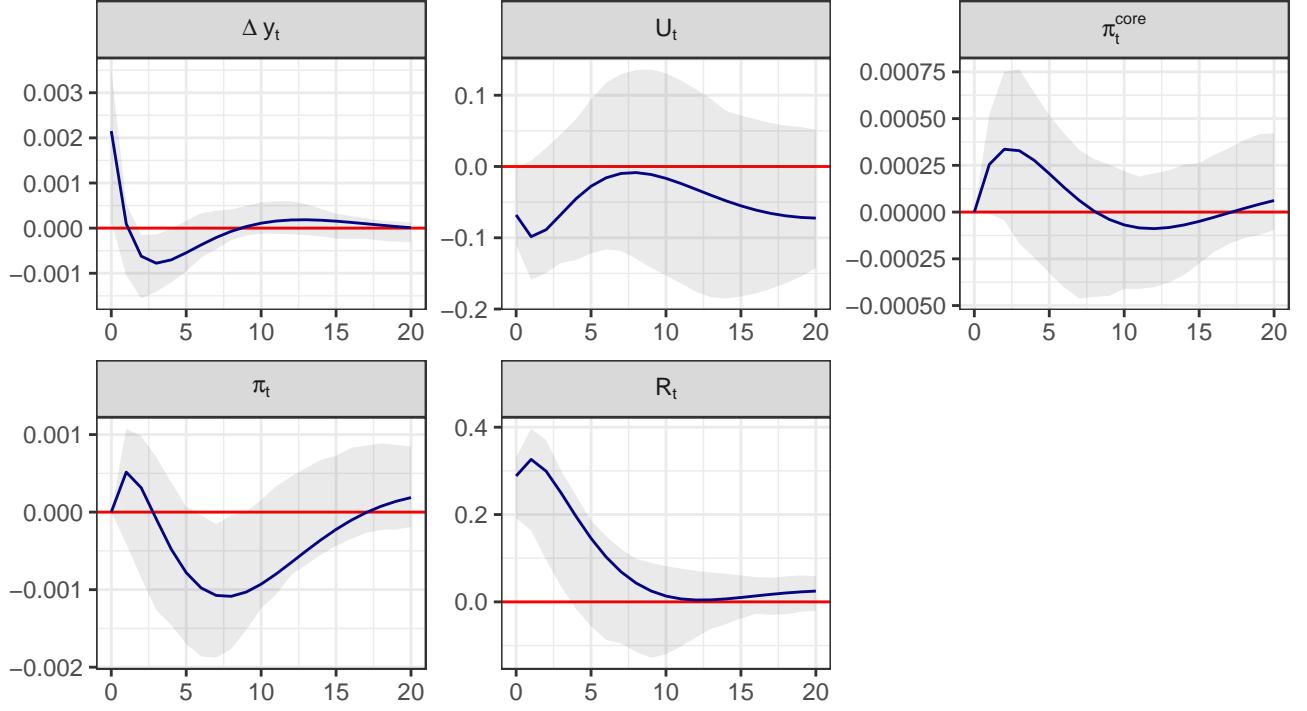
Figure 3: Responses functions on the European Aera economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the demand scenario.
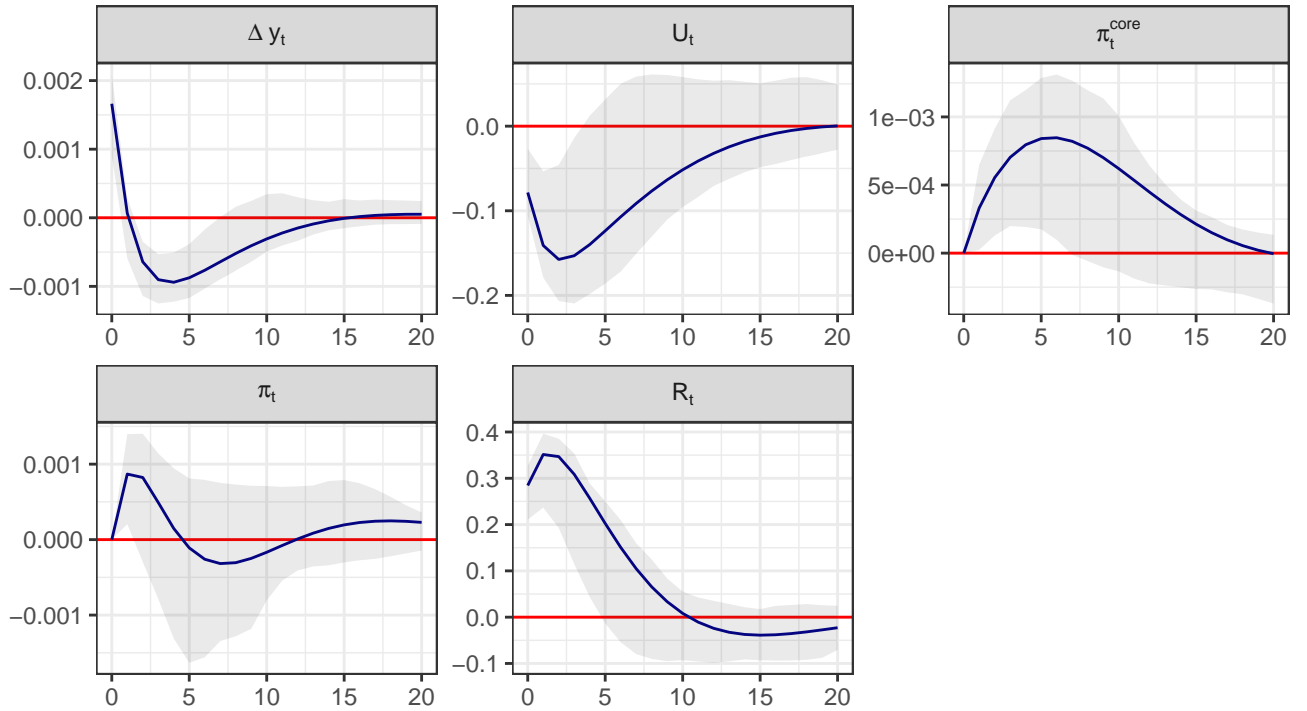

Figure 4: Responses functions on the French economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the demand scenario.

decrease in the GDP that returns to normal after almost 3 years. Unlike what we observe in the demand scenario, the inflation decreases in the first year and then returns to normal. The Euribor slowly returns to normal. The same evolutions are observes for the French economy and for the European Area.

Figure 5: Responses functions on the European Aera economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the supply scenario.

The above observations provide evidence that our findings are consistent with some key theoretical predictions of the Neo Keynesian model:

Figure 6: Responses functions on the French economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the supply scenario.

J'ai rajouté les gaphiques Blanchard Quah au cas où mais je pense que ce n'est pas la peine de les mettre.
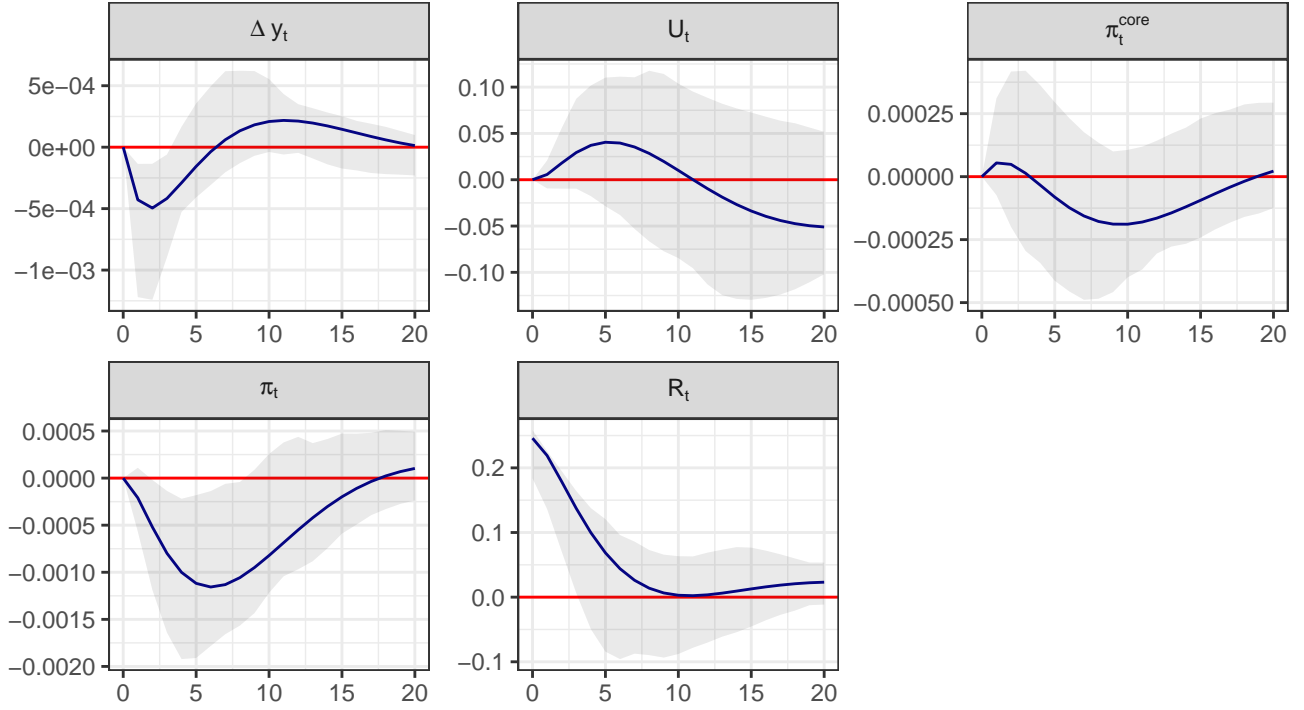
Figure 7: Responses functions on the European Aera economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the demand scenario (Blanchard Quah).

Figure 8: Responses functions on the French economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the demand scenario (Blanchard Quah).

Figure 9: Responses functions on the European Aera economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the supply scenario (Blanchard Quah).
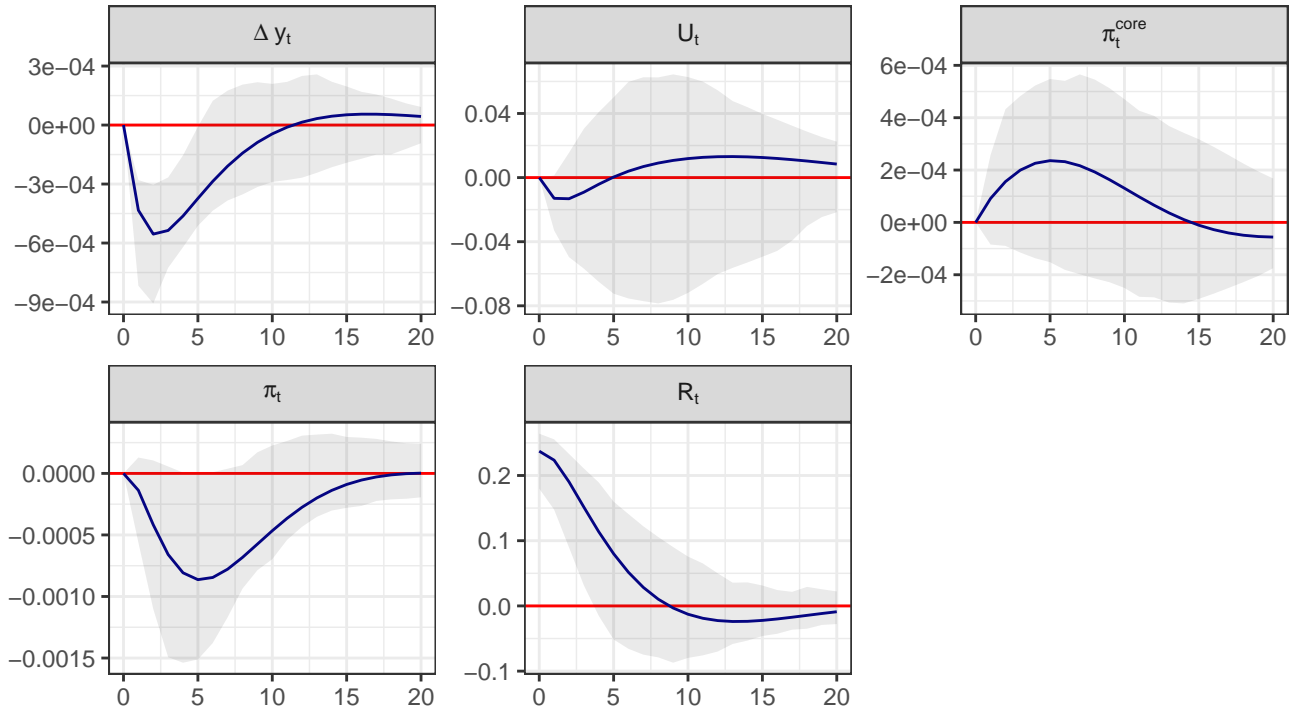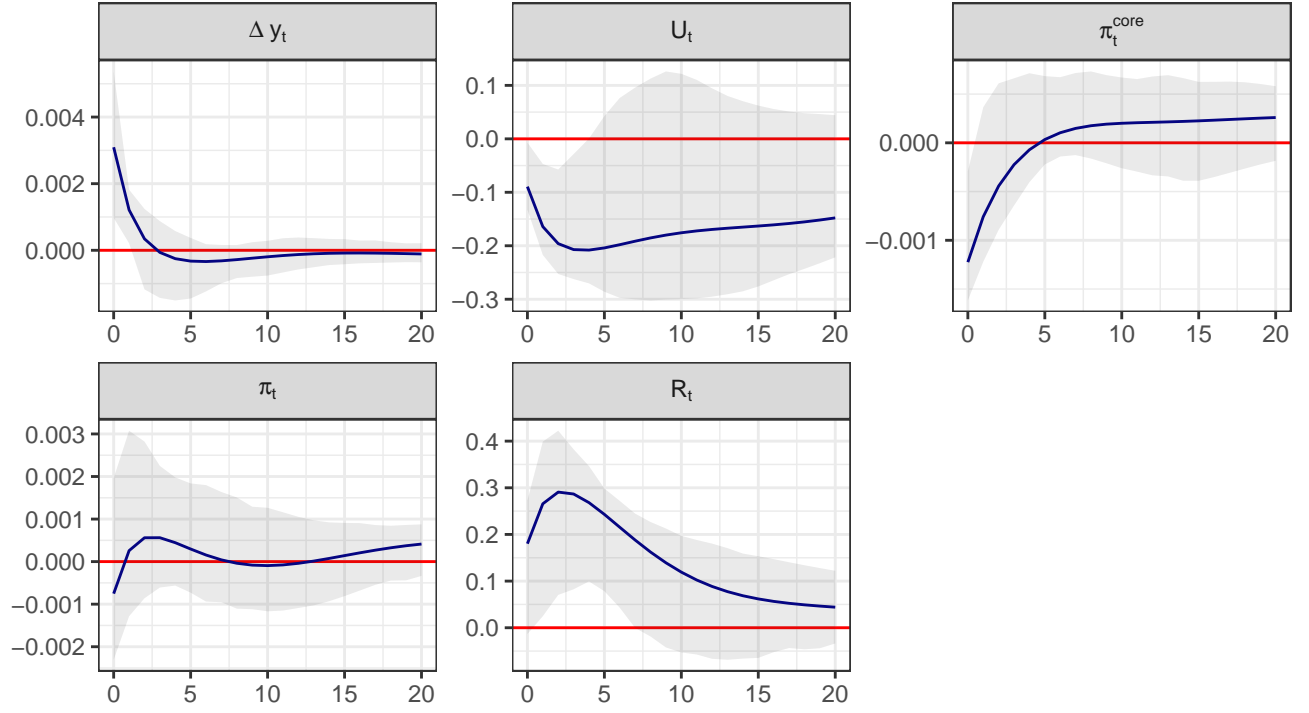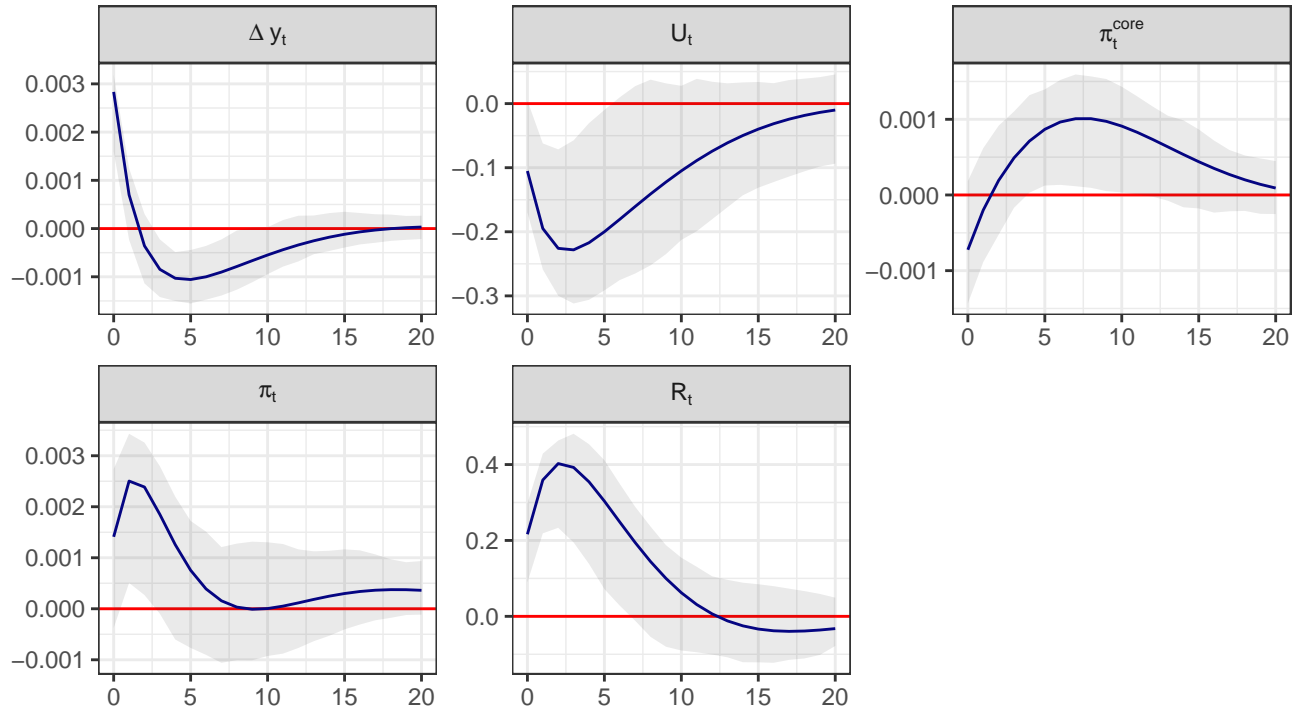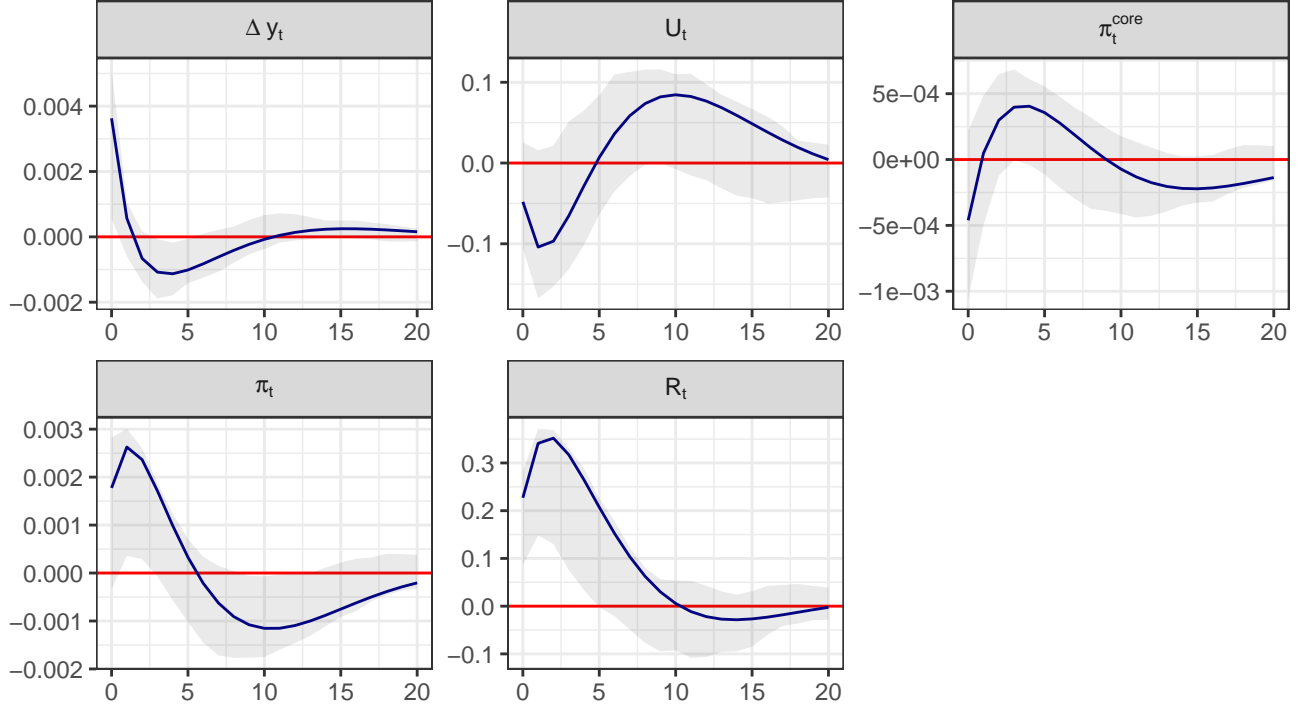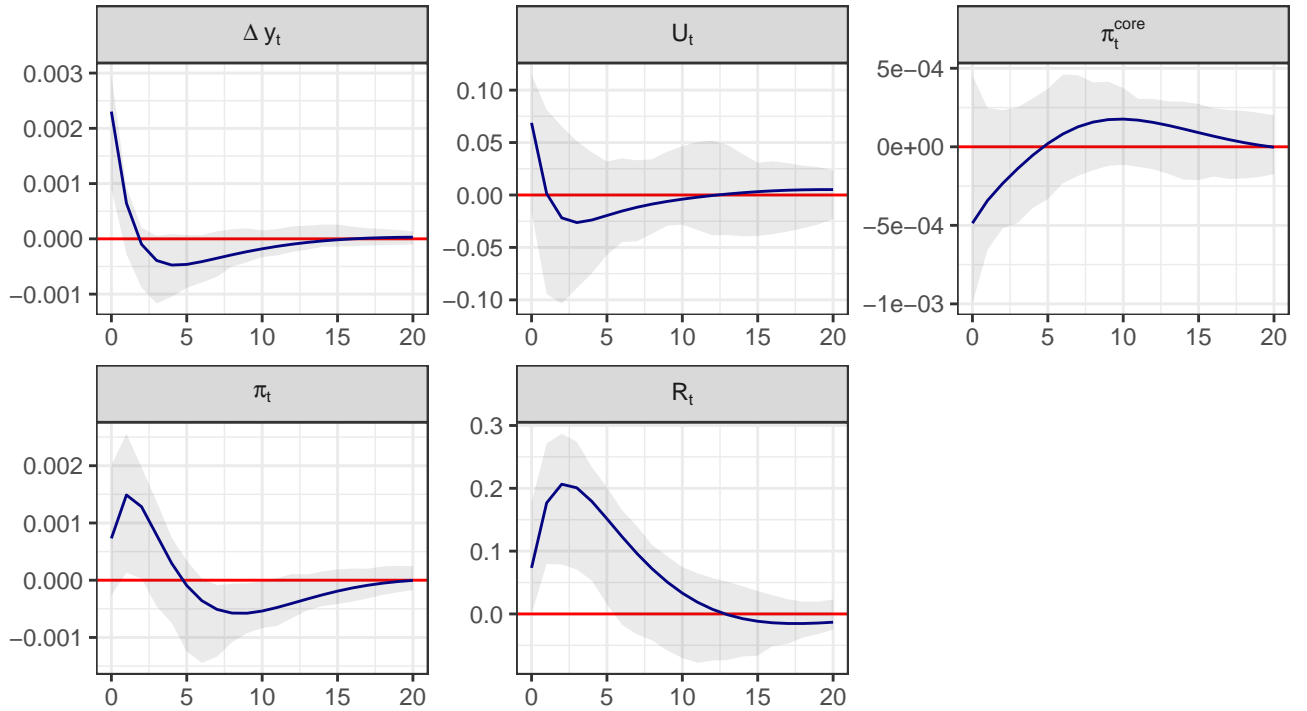

Figure 10: Responses functions on the French economy of a shock on the 3 months Euribor rates, R_t with a 95% bootstrap confidence interval in the supply scenario (Blanchard Quah).

A Code

A.1 Functions

`Z - Fonctions.R` file.

```r
if (!require(ggplot2)){
    install.packages(ggplot2)
    require(ggplot2)
}
if (!require(forcats)){
    install.packages(forcats)
    require(forcats)
}
if (!require(ggfortify)){
    install.packages(forcats)
    require(forcats)
}
if (!require(patchwork)){
    install.packages(forcats)
    require(forcats)
}
if (!require(reshape2)){
    install.packages(reshape2)
}
plot_irf <- function(oir,
                     labeller = "label_parsed",
                     recode = c("Delta~y[t]"= "dlGDP",
                                "U[t]" = "U",
                                "pi[t]^core"="underinf",
                                "pi[t]"="HICP",
                                "R[t]" = "EURIBOR_3M")){
    data_plot <- Reduce(function(x,y) merge(x,y,by = c("Var1","Var2")),
                        list(reshape2::melt(oir$irf$EURIBOR_3M),
                             reshape2::melt(oir$Lower$EURIBOR_3M),
                             reshape2::melt(oir$Upper$EURIBOR_3M)))
    data_plot[,"Var1"] <- data_plot[,"Var1"] -1
    colnames(data_plot) <- c("x","variable","y","lower","upper")
    data_plot$variable <- factor(data_plot$variable,
                                 levels = c("dlGDP", "U", "underinf", "HICP", "EURIBOR_3M"),
                                 ordered = TRUE)
    data_plot$variable <- fct_recode(data_plot$variable, !!!recode)
    ggplot(data = data_plot,aes(x = x,y=y,ymin=lower,ymax=upper)) +
        geom_hline(yintercept=0,color="red")+
        geom_line(color = "darkblue")+
        geom_ribbon(alpha=0.1) +
        facet_wrap(vars(variable),
                   scales = "free", nrow = 2, strip.position = "top",
                   labeller = labeller)+
        theme_bw()+xlab("") + ylab("")
}
plot_fevd <- function(fevd,
                      labeller = "label_parsed",
                      recode = c("Delta~y[t]"= "dlGDP",
                                 "U[t]" = "U",
                                 "pi[t]^core"="underinf",
                                 "pi[t]"="HICP",
```

```r
                                           "R[t]" = "EURIBOR_3M")){
    data_plot <- Reduce(rbind,
                        lapply(names(fevd), function(x){
                            d <- reshape2::melt(fevd[[x]])
                            d$var_comp = x
                            d
                        }))
    colnames(data_plot) <- c("horizon","variable","y","decomp")
    data_plot$y <- 100 *data_plot$y
    data_plot$decomp <- factor(data_plot$decomp,
                               levels = c("dlGDP", "U", "underinf", "HICP", "EURIBOR_3M"),
                               ordered = TRUE)

    data_plot$variable = fct_recode(data_plot$variable, !!!recode)
    data_plot$decomp = fct_recode(data_plot$decomp, !!!recode)

    ggplot(data = data_plot, mapping = aes(x = horizon, y = y,
                                           fill = variable)) +
        geom_bar(stat = "identity")+
        facet_wrap(vars(decomp),
                   scales = "free", strip.position = "top",
                   labeller = labeller)+
        theme_bw()+xlab("Horizon") + ylab("Percentage") +
        scale_fill_viridis_d(
            breaks = levels(data_plot$variable),
            labels = unlist(label_parsed(levels(data_plot$variable))))
}
plot_ts <- function(data, x){
    rename_fun <- function(x){
        x <- sub("dlGDP", "Delta~y[t]" , x)
        x <- sub("underinf", "pi[t]^{core}" , x)
        x <- sub("HICP", "pi[t]" , x)
        x <- sub("EURIBOR_3M", "R[t]" , x)
        x <- sub("^U$", "U[t]" , x)
        x
    }
    autoplot(data[,x]) +
        labs(title = label_parsed(rename_fun(x))[[1]][[1]])
}
latexify_var <- function(model, nb_dec = 1, align = FALSE,
                         se = FALSE) {
    rename_fun <- function(x){
        x <- sub("dlGDP", "\\Delta y" , x, fixed = TRUE)
        x <- sub("underinf", "\\pi^{core}" , x, fixed = TRUE)
        x <- sub("HICP", "\\pi" , x, fixed = TRUE)
        x <- sub("EURIBOR_3M", "R" , x, fixed = TRUE)
        x
    }
    # rename_fun = function(x) {gsub("_", "\\_", x,fixed = TRUE)}
    coefficients <- coef(model)
    names_var <- rename_fun(names(coefficients))

    coef_mats <- lapply(seq_len(model$p),function(lag){
        sapply(coefficients,function(x){
            x[grep(sprintf("\\.l%i$",lag),rownames(x)), "Estimate"]
        })
```

```r
    })

    coef_mats_se <- lapply(seq_len(model$p),function(lag){
        if(se){
            sapply(coefficients,function(x){
                x[grep(sprintf("\\.l%i$",lag),rownames(x)), "Std. Error"]
            })
        }else{
            NULL
        }
    })
    X_tex = c(latexify_const_trend(model, se = se, nb_dec = nb_dec),
             sapply(seq_along(coef_mats), function(lag){
                 paste(latexify_mat(coef_mats[[lag]],
                                     se_mat = coef_mats_se[[lag]],
                                     nb_dec = nb_dec),
                       latexify_y(names_var, lag = lag)
                 )
             }))

    Y_tex = latexify_y(names_var, lag = 0)
    if (align){
        # Concaténation de la constante
        X_tex = c(paste(X_tex[1], X_tex[2], sep = " + "),
                  X_tex[-(1:2)])
        X_tex = paste(X_tex,
                      collapse = "\\nonumber \\\\ \n &+")
        res = sprintf("\\begin{align} \n%s &= %s + \\varepsilon_t \n\\end{align}",
                      Y_tex, X_tex)
    }else{
        X_tex = paste(X_tex,
                      collapse = "+")
        res = sprintf("%s = %s+ \\varepsilon_t\n",
                      Y_tex, X_tex)
    }
    return (res)
}

latexify_mat <- function(mat, se_mat = NULL, nb_dec = 1){
    mat = formatC(mat, digits = nb_dec, format = "f")
    if(!is.null(se_mat)){
        se_mat = formatC(se_mat, digits = nb_dec, format = "f")
        se_mat = apply(se_mat,2,
               function(x) paste0("\\underset{(",x,")}"))
        mat = apply(mat,2,
                    function(x) paste0("{",x,"}"))
        mat[]= paste0(se_mat,mat)
    }

    mat = paste(apply(mat,1, paste, collapse = " & "), collapse = " \\\\\n")
    mat <- paste("\\begin{pmatrix}\n",mat,
                 "\n\\end{pmatrix}")
    mat
}
latexify_const_trend <- function(model, nb_dec = 1, se = FALSE){
    coefficients <- coef(model)
```

13

```r
    const_trend <- t(sapply(coefficients,function(x){
        x[grep("(^const$)|(^trend$)",rownames(x)), "Estimate"]
    }))
    if(se){
        const_trend_se <- t(sapply(coefficients,function(x){
            x[grep("(^const$)|(^trend$)",rownames(x)), "Std. Error"]
        }))
    }else{
        const_trend_se <- NULL
    }

    if(model$type == "both"){
        # constante + tendance
        res <- latexify_mat(const_trend,
                            se_mat = const_trend_se,
                            nb_dec = nb_dec)
        res <- paste(res,
                     "\\begin{pmatrix} 1 \\\\ t \\end{pmatrix}")
    }
    if(model$type == "trend"){
        res <- latexify_mat(const_trend,
                            se_mat = const_trend_se,
                            nb_dec = nb_dec)
        res <- paste(res,
                     "\\begin{pmatrix} 1 \\end{pmatrix}")
    }
    if(model$type == "const"){
        res <- latexify_mat(const_trend,
                            se_mat = const_trend_se,
                            nb_dec = nb_dec)
        res <- paste(res,
                     "\\begin{pmatrix} t  \\end{pmatrix}")
    }
    if(model$type == "none"){
        res <- NULL
    }
    res
}
latexify_y <- function(vec, lag = 0){
    vec = sprintf("%s_{t%s}",vec,
                  ifelse(lag == 0,"",
                         sprintf(" - %i", lag)))
    paste("\\begin{pmatrix}\n",
          paste(vec,collapse = " \\\\\n"),
          "\n\\end{pmatrix}")
}


# Téléchargement données depuis BDM
lectureBDM <- function(idbank, ...)
{
    #On récupère les idbank et on supprime les éventuels espaces
    idbank<-gsub(" ","",c(idbank,unlist(list(...))))

    #Les url pour télécharger le(s) série(s)
    UrlData <- paste0("https://bdm.insee.fr/series/sdmx/data/SERIES_BDM/",
```

```r
                                paste(idbank,collapse = "+"))

    tryCatch({
        dataBDM <- as.data.frame(rsdmx::readSDMX(UrlData,isURL = T),
                                  stringsAsFactors=TRUE)
    },error=function(e){
        stop(paste0("Il y a une erreur dans le téléchargement des données. Vérifier le lien\n",
                    UrlData),
             call. = FALSE)
    })

    FREQ <- levels(factor(dataBDM$FREQ))

    if (length(FREQ)!=1)
        stop("Les séries ne sont pas de la même périodicité !")

    freq<-switch(FREQ
                 ,M=12
                 ,B=6
                 ,T=4
                 ,S=2
                 ,A=1)
    #On détermine le format de la colonne qui contient les dates en fonction de la fréquence
    sepDate<-switch(FREQ
                    ,M="-"
                    ,B="-B"
                    ,T="-Q"
                    ,S="-S"
                    ,A=" ")
    dataBDM <- reshape2::dcast(dataBDM,"TIME_PERIOD ~ IDBANK",value.var = "OBS_VALUE")
    dataBDM <- dataBDM[order(dataBDM$TIME_PERIOD),]

    #On récupère la première date
    dateDeb <- dataBDM$TIME_PERIOD[1]
    dateDeb <- regmatches(dateDeb,gregexpr(sepDate,dateDeb),invert=T)[[1]]
    dateDeb <- as.numeric(dateDeb)

    #On supprime la colonne des dates et on convertit les séries en numérique
    dataBDM$TIME_PERIOD <- NULL
    dataBDM <- apply(dataBDM,2,as.numeric)

    if(ncol(dataBDM) != length(idbank))
        warning(paste("Le ou les idbank suivant n'existent pas :",
                      paste(grep(paste(colnames(dataBDM),collapse="|"),idbank,value=T,invert = T),
                            collapse=", ")))
    if(ncol(dataBDM) > 1){
        # On a au moins 2 colonnes : on replace les colonnes dans le même ordre
        # que les séries en entrée

        # On ne garde que les idbank présents dans la base
        idbank <- idbank[idbank %in% colnames(dataBDM)]
        dataBDM <- dataBDM[,idbank]
    }
    dataBDM <- ts(dataBDM,start=dateDeb,freq=freq)
    return(dataBDM)
}
```

A.2 Downloading of the data

`1-extraction_donnees.R` file.

```r
library(ecb)
library(eurostat)
library(zoo)

source("R/Z - Fonctions.R",encoding = "UTF-8")

############################## DATA GATHERING & CLEANING ##############################

##################################
######## DONNEES UE ############
##################################
##EURIBOR_3M
euri<-get_eurostat(
    "irt_st_q",
    time_format="date",
    filters=list(geo="EA",int_rt="IRT_M3")
)
euribor <- ts(euri$values,start=c(substr(euri$time[1],1,4),1), freq=4)

##GDP

gdp<-get_eurostat(
    "namq_10_gdp",
    time_format="date",
    filters=list(geo="EA", s_adj="SCA", na_item="B1GQ", unit="CLV10_MEUR")
)  # La série désaisonnalisée n'est pas disponible au niveau agrégé
gdp <- log(ts(gdp$values,start=c(substr(gdp$time[1],1,4),1), freq=4))
dlgdp <- diff(gdp)


##unemployment
# Extraction du taux de chômage harmonisé pour les personnes de 15 à 74 ans,
# En ne faisant pas de distinction par sexe et en prenant le pourcentage dans la population active
unem<-get_eurostat(
    "une_rt_q",
    time_format="date",
    filters=list(age = "Y15-74", geo="EA19",sex="T",s_adj = "SA", unit="PC_ACT")
)
unemp<-ts(unem$values,start=c(substr(unem$time[1],1,4),1), freq=4)

## inflation and underlying inflation (From ECB database)

hicp <-get_data("ICP.M.U2.N.000000.4.INX"
)
hicp<-ts(hicp$obsvalue,
        start=as.numeric(c(substr(hicp$obstime[1],1,4),
                substr(hicp$obstime[1],6,7))),
        freq=12)
hicpq <- aggregate(as.zoo(hicp), yearqtr, mean)
hicpq <- as.ts(hicpq)
hicpq <- hicpq/lag(hicpq,-4)-1 # glissement annuel

infex <-get_data("ICP.M.U2.N.XEF000.4.INX"
```

```
)
infex<-ts(infex$obsvalue,
          start=as.numeric(c(substr(infex$obstime[1],1,4),
                             substr(infex$obstime[1],6,7))),
          freq=12)
infexq <- aggregate(as.zoo(infex), yearqtr, mean)
infexq <- as.ts(infexq)
infexq <- infexq/lag(infexq,-4)-1 # glissement annuel

data <- ts.union(euribor, dlgdp,gdp, unemp, hicpq, infexq)
colnames(data)<-cbind("EURIBOR_3M", "dlGDP","lGDP",
                      "U","HICP","underinf")
saveRDS(data, file="data/data_UE.RDS")


#################################
######### DONNEES FR ############
#################################

# Indice d'inflation sous-jacente - Base 2015 - Ensemble des ménages -
# France métropolitaine - Ensemble
infex <- lectureBDM("001769686")
infexq <- aggregate(as.zoo(infex), yearqtr, mean)
infexq <- as.ts(infexq)
infexq <- infexq/lag(infexq,-4)-1 # glissement annuel

# Produit intérieur brut total - Volume aux prix de l'année précédente chaînés -
# Série CVS-CJO aux prix de l'année précédente chaînés - Série CVS-CJO
gdp <- log(lectureBDM("010565708"))
dlgdp <- diff(gdp)

# Taux de chômage au sens du BIT - Ensemble - France métropolitaine -
# Données CVS
unemp <- lectureBDM("001688526")

hicp <- lectureBDM("001759971")
hicpq <- aggregate(as.zoo(hicp), yearqtr, mean)
hicpq <- as.ts(hicpq)
hicpq <- hicpq/lag(hicpq,-4)-1 # glissement annuel

data <- ts.union(euribor, dlgdp,gdp, unemp, hicpq, infexq)
colnames(data)<-cbind("EURIBOR_3M", "dlGDP","lGDP",
                      "U","HICP","underinf")
saveRDS(data, file="data/data_FR.RDS")
```

A.3 Estimation of the models

A.3.1 European area

2.1-estimation_modeles EA.R file.

```
library(vars)

source("R/Z - Fonctions.R",encoding = "UTF-8")

dataUE <- readRDS("data/data_UE.RDS")
dataUE <- na.omit(dataUE[,c("EURIBOR_3M", "lGDP","dlGDP",
                            "U", "HICP", "underinf")])
```

```r
dataUE <- window(dataUE, end = c(2018,4))

p <- (plot_ts(dataUE, "EURIBOR_3M") +
         plot_ts(dataUE, "dlGDP"))/(
             plot_ts(dataUE, "U") +
                 plot_ts(dataUE, "HICP") +
                 plot_ts(dataUE, "underinf")
             )
(p & theme_minimal() )+
    plot_annotation(title = 'European Area')

var_ordering1 = c("dlGDP",
                  "U", "underinf","HICP", "EURIBOR_3M")
var_ordering2 = c("EURIBOR_3M", "dlGDP",
                  "U", "underinf","HICP")

#Select AIC-suggested lag
lagselect <-VARselect(dataUE[,var_ordering1],
                      lag.max=6,type="const")
# Le critère de Schwartz suggère de ne retenir qu'un lag
lagselect
p_retenu = 1

# L'ordre des variables n'aura pas d'impact sur les analyses du VAR
# mais uniquement dans la spécification du sVAR
model <- VAR(dataUE[,var_ordering1],
             p = p_retenu,type = "both")
model2 <- VAR(dataUE[,var_ordering2],
              p = p_retenu,type = "both")
# Pas d'autocorrélation dans les résidus
serial.test(model)

# Pas d'hétéroscédasticité dans les résidus
arch.test(model)
# Ni de problème de stabilité dans les coefficients
plot(stability(model))

# Pour récupérer le code latex du VAR :
cat(latexify_var(model,align = FALSE, nb_dec = 2, se=TRUE))
cat(latexify_mat(cor(residuals(model)), nb_dec = 2))

Bmat_chol <- diag(nrow = 5)
Bmat_chol[2,1] <- Bmat_chol[3,1:2] <-
    Bmat_chol[4,(1:3)] <- Bmat_chol[5,1:4] <-
    diag(Bmat_chol) <- NA
Bmat_chol # Choleski

Bmat <- diag(nrow = 5)
Bmat[2,1] <- Bmat[3,c(1)] <-
    Bmat[4,c(1, 3)] <- Bmat[5,1:4] <-
    diag(Bmat) <- NA
Bmat # affiné

Bmat2 <- diag(nrow = 5)
Bmat2[2,1] <- Bmat2[3,c(1:2)] <-
    Bmat2[5,4] <-
```

```
     diag(Bmat2) <- NA
Bmat2 # affiné

smodel1_chol <- SVAR(model,Bmat = Bmat_chol)
smodel1 <- SVAR(model,Bmat = Bmat)
smodel1_chol$B # coefficients imposés à 0 proches de 0
smodel1$B
cat(latexify_mat(smodel1$B, nb_dec = 3))

smodel2 <- SVAR(model2,Bmat = Bmat_chol)
smodel2
# les coefficients que l'on va imposer à 0 sont estimés à 0
# en utilisant la décomposition de choleski
# Les mêmes résultats sont donc trouvés entre les deux méthodes
smodel2 <- SVAR(model2,Bmat = Bmat2)
smodel2
cat(latexify_mat(smodel2$B, nb_dec = 3))


smodel1_bq <- BQ(model)
smodel2_bq <- BQ(model2)
irf_1 <- irf(smodel1, impulse = "EURIBOR_3M",
          n.ahead = 20)
irf_1_chol <- irf(smodel1_chol, impulse = "EURIBOR_3M",
              n.ahead = 20)
irf_1_bq <- irf(smodel1_bq, impulse = "EURIBOR_3M",
                 n.ahead = 20)

irf_2 <- irf(smodel2, impulse = "EURIBOR_3M",
             n.ahead = 20)
irf_2_bq <- irf(smodel2_bq, impulse = "EURIBOR_3M",
                 n.ahead = 20)

plot_irf(irf_1_chol) + ggtitle("Choleski - supply")
plot_irf(irf_1) + ggtitle("Matrice affinée - supply") # Quasiment même résultat

plot_irf(irf_1_bq) + ggtitle("Blanchard Quah decomposition - supply")

plot_irf(irf_2) + ggtitle("Matrice affinée - demand")
plot_irf(irf_2_bq) + ggtitle("Blanchard Quah decomposition - demand")

# decomposition de la variance
fevd <- fevd(smodel1, n.ahead = 20)
plot_fevd(fevd)


saveRDS(model, "data/models_EA/var_model.RDS")
saveRDS(irf_1, "data/models_EA/irf_1")
saveRDS(irf_1_chol, "data/models_EA/irf_1_chol")
saveRDS(irf_1_bq, "data/models_EA/irf_1_bq")
saveRDS(irf_2, "data/models_EA/irf_2")
saveRDS(irf_2_bq, "data/models_EA/irf_2_bq")
```

A.3.2 France

`2.2-estimation_modeles FR.R` file.

```r
library(vars)

source("R/Z - Fonctions.R",encoding = "UTF-8")

dataFR <- readRDS("data/data_FR.RDS")
dataFR <- na.omit(dataFR[,c("EURIBOR_3M", "lGDP","dlGDP",
                            "U", "HICP", "underinf")])
dataFR <- window(dataFR, end = c(2018,4))

p <- (plot_ts(dataFR, "EURIBOR_3M") +
        plot_ts(dataFR, "dlGDP"))/(
            plot_ts(dataFR, "U") +
                plot_ts(dataFR, "HICP") +
                plot_ts(dataFR, "underinf")
        )
(p & theme_minimal() )+
    plot_annotation(title = 'France')

var_ordering1 = c("dlGDP",
                  "U", "underinf","HICP", "EURIBOR_3M")
var_ordering2 = c("EURIBOR_3M", "dlGDP",
                  "U", "underinf","HICP")

#Select AIC-suggested lag
lagselect <-VARselect(dataFR[,var_ordering1],
                      lag.max=6,type="const")
# Le critère du BIC indique 1 lag
lagselect
p_retenu = 1

# L'ordre des variables n'aura pas d'impact sur les analyses du VAR
# mais uniquement dans la spécification du sVAR
model <- VAR(dataFR[,var_ordering1],
            p = p_retenu,type = "both")
model2 <- VAR(dataFR[,var_ordering2],
             p = p_retenu,type = "both")
# Pas d'autocorrélation dans les résidus
serial.test(model)

# Pas d'hétéroscédasticité dans les résidus
arch.test(model)
# Ni de problème de stabilité dans les coefficients
plot(stability(model))

# Pour récupérer le code latex du VAR :
cat(latexify_var(model,align = FALSE, nb_dec = 2, se = TRUE))
cat(latexify_mat(cor(residuals(model)), nb_dec = 2))


Bmat_chol <- diag(nrow = 5)
Bmat_chol[2,1] <- Bmat_chol[3,1:2] <-
    Bmat_chol[4,(1:3)] <- Bmat_chol[5,1:4] <-
    diag(Bmat_chol) <- NA
Bmat_chol # Choleski

Bmat <- diag(nrow = 5)
```

```r
Bmat[2,1] <- Bmat[3,c(1)] <-
    Bmat[4,c(1, 3)] <- Bmat[5,1:4] <-
    diag(Bmat) <- NA
Bmat # affiné

Bmat2 <- diag(nrow = 5)
Bmat2[2,1] <- Bmat2[3,c(1:2)] <-
    Bmat2[5,4] <-
    diag(Bmat2) <- NA
Bmat2 # affiné

smodel1_chol <- SVAR(model,Bmat = Bmat_chol)
smodel1 <- SVAR(model,Bmat = Bmat)
smodel1_chol$B # coefficients imposés à 0 proches de 0
cat(latexify_mat(smodel1$B, nb_dec = 3))

smodel2 <- SVAR(model2,Bmat = Bmat_chol)
smodel2 <- SVAR(model2,Bmat = Bmat_chol)
smodel2
# les coefficients que l'on va imposer à 0 sont estimés à 0
# en utilisant la décomposition de choleski
# Les mêmes résultats sont donc trouvés entre les deux méthodes
smodel2 <- SVAR(model2,Bmat = Bmat2)
smodel2
cat(latexify_mat(smodel2$B, nb_dec = 3))

smodel1_bq <- BQ(model)
smodel2_bq <- BQ(model2)
irf_1 <- irf(smodel1, impulse = "EURIBOR_3M",
            n.ahead = 20)
irf_1_chol <- irf(smodel1_chol, impulse = "EURIBOR_3M",
                n.ahead = 20)
irf_1_bq <- irf(smodel1_bq, impulse = "EURIBOR_3M",
                    n.ahead = 20)

irf_2 <- irf(smodel2, impulse = "EURIBOR_3M",
            n.ahead = 20)
irf_2_bq <- irf(smodel2_bq, impulse = "EURIBOR_3M",
                    n.ahead = 20)

plot_irf(irf_1_chol) + ggtitle("Choleski - supply")
plot_irf(irf_1) + ggtitle("Matrice affinée - supply") # Quasiment même résultat

plot_irf(irf_1_bq) + ggtitle("Blanchard Quah decomposition - supply")

plot_irf(irf_2) + ggtitle("Matrice affinée - demand")
plot_irf(irf_2_bq) + ggtitle("Blanchard Quah decomposition - demand")

# decomposition de la variance
fevd <- fevd(smodel1, n.ahead = 20)
plot_fevd(fevd)


saveRDS(model, "data/models_FR/var_model.RDS")
saveRDS(irf_1, "data/models_FR/irf_1")
saveRDS(irf_1_chol, "data/models_FR/irf_1_chol")
```

```
saveRDS(irf_1_bq, "data/models_FR/irf_1_bq")
saveRDS(irf_2, "data/models_FR/irf_2")
saveRDS(irf_2_bq, "data/models_FR/irf_2_bq")
```