table_template

2023-10-08

```
#install.packages(c("xts","pdfetch", "ggplot2", "mFilter"))


library(xts)
```

```
## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.1

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(pdfetch) #Library for loading FRED data
```

```
## Warning: package 'pdfetch' was built under R version 4.1.1
```

```
library(ggplot2) #Library for plotting
library(mFilter) #Library for HP filter
library(rollRegres) #Library for Regression
```

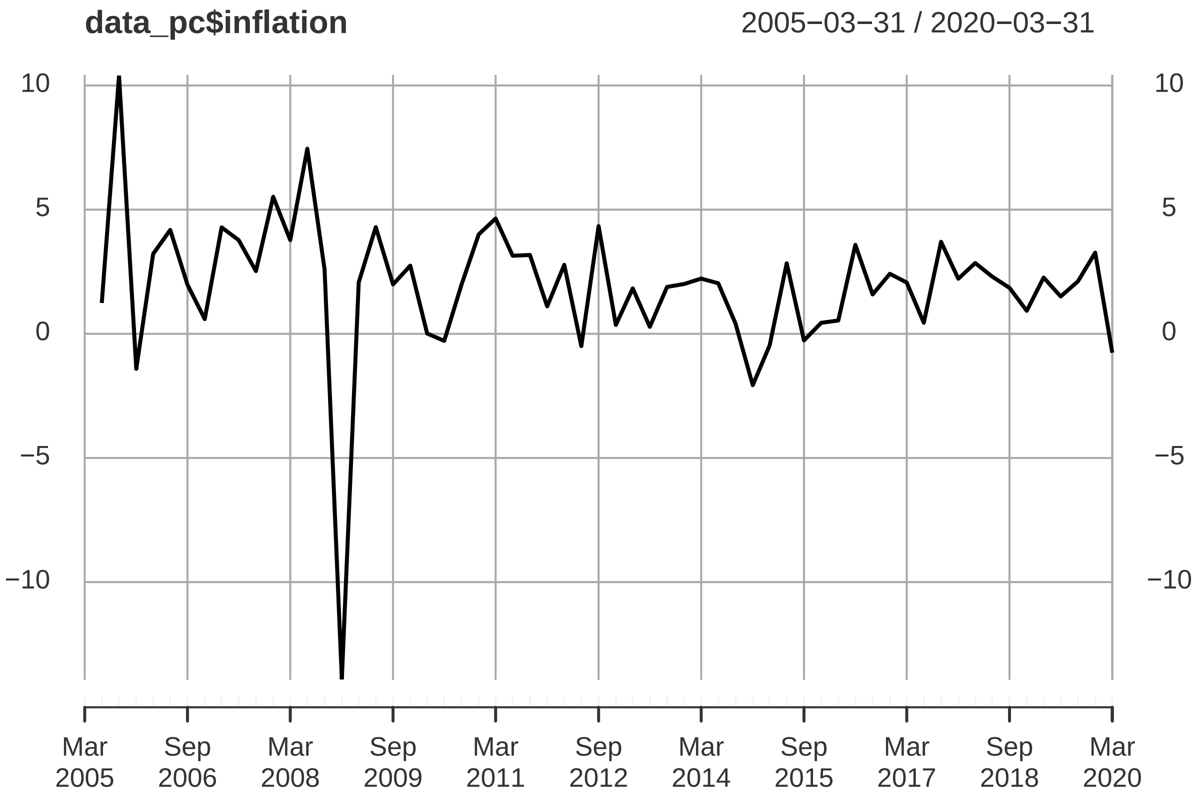
```
## Warning: package 'rollRegres' was built under R version 4.1.1
```

```
data_pc_raw <- pdfetch_FRED(c("GDPC1", "UNRATE", "CPIAUCSL", "CPILFESL"))
data_pc <- data_pc_raw["2005-01-01/2020-04-01"]
# Convert data to quarterly frequency
data_pc <- to.period(data_pc, period = "quarter", OHLC = FALSE)
```

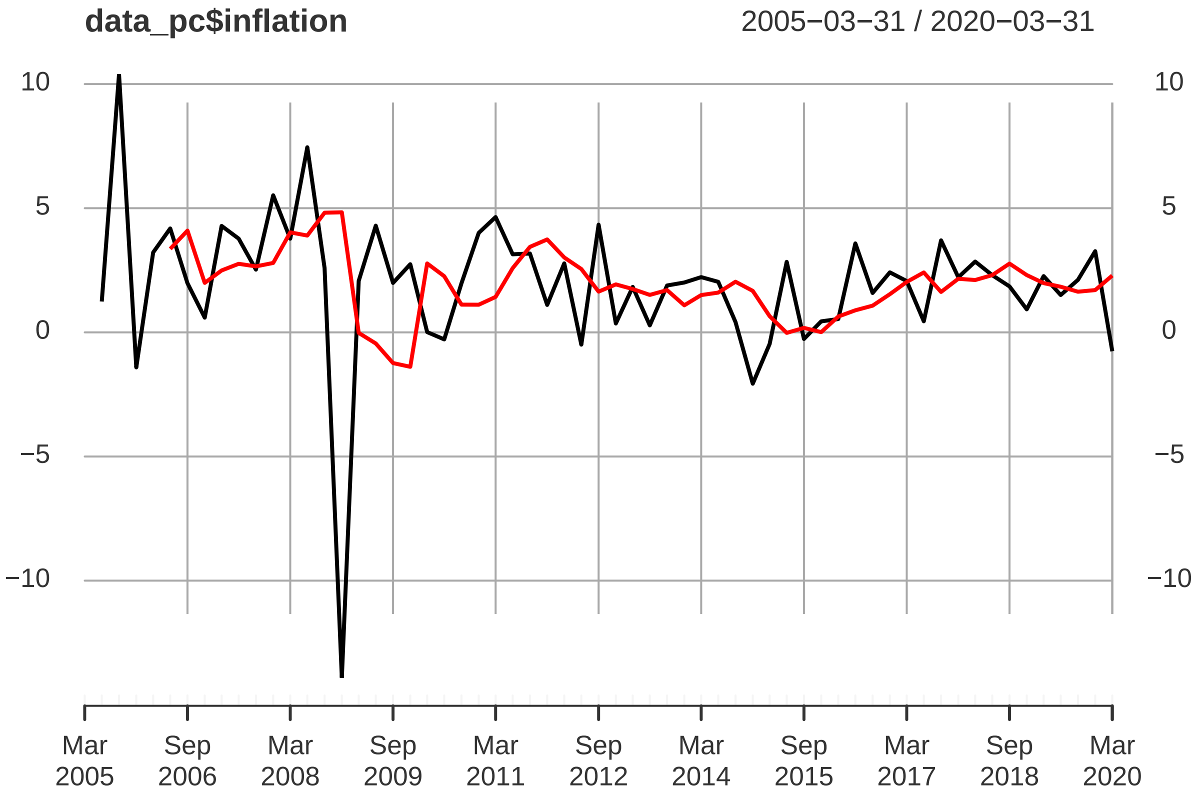
```
## Warning in to.period(data_pc, period = "quarter", OHLC = FALSE): missing values
## removed from data
```

```
#View(data_pc)

#Transformations
data_pc$lgdp <- log(data_pc$GDPC1) # Take logs
hp_gdp <- hpfilter(data_pc$lgdp, freq = 1600, type="lambda")
data_pc$gdpgap <- 100*hp_gdp$cycle
```

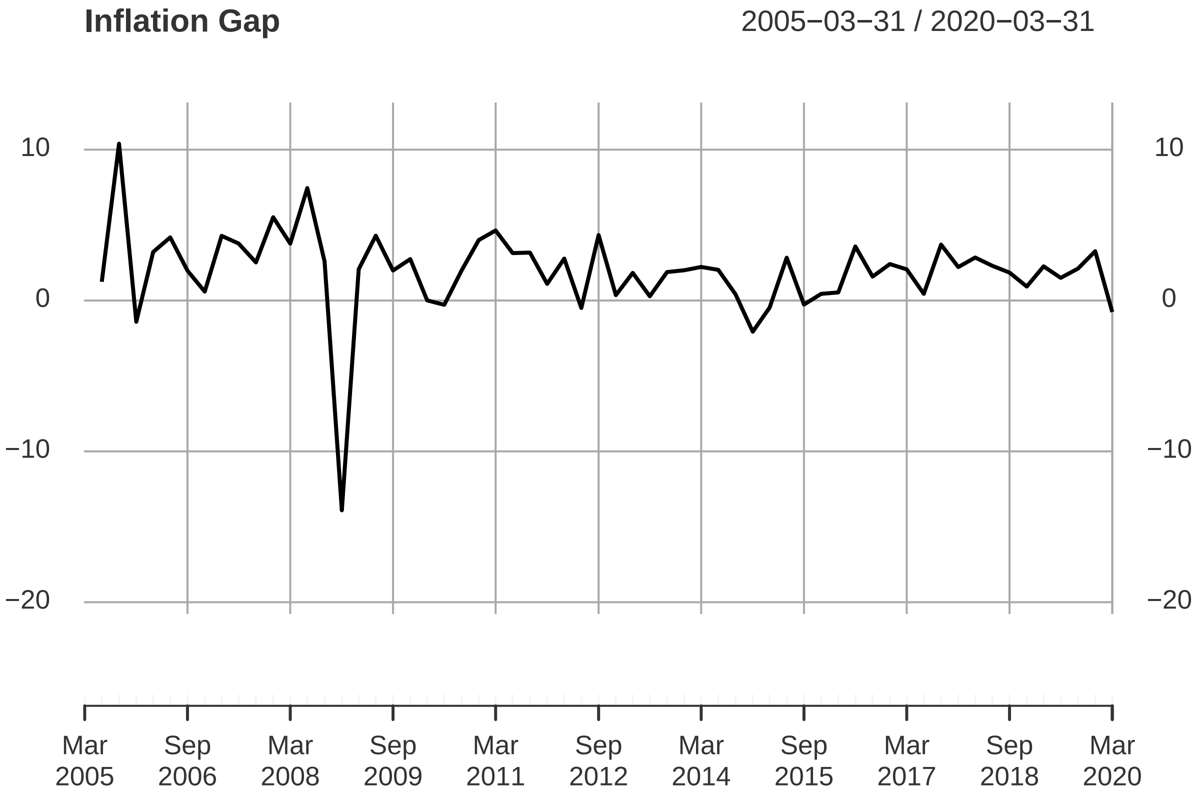
```
data_pc$l_cpi <- log(data_pc$CPIAUCSL)
data_pc$l_cpi_core <- log(data_pc$CPILFESL) # Consumer Price Index of All Items in Japan
data_pc$unemployment_rate <- (data_pc$UNRATE) # seasonally adjusted


#Quarterly inflation, annualized
data_pc$inflation_q = 4*100*diff(data_pc$l_cpi)

#Inflation expectations as an average of 4 past y-o-y inflation rates
data_pc$infexp <- 1/4*(lag(data_pc$inflation, k=1) + lag(data_pc$inflation, k=2) + lag(data_pc$inflation

plot.xts(data_pc$inflation, col = "black", lwd = 2)
```



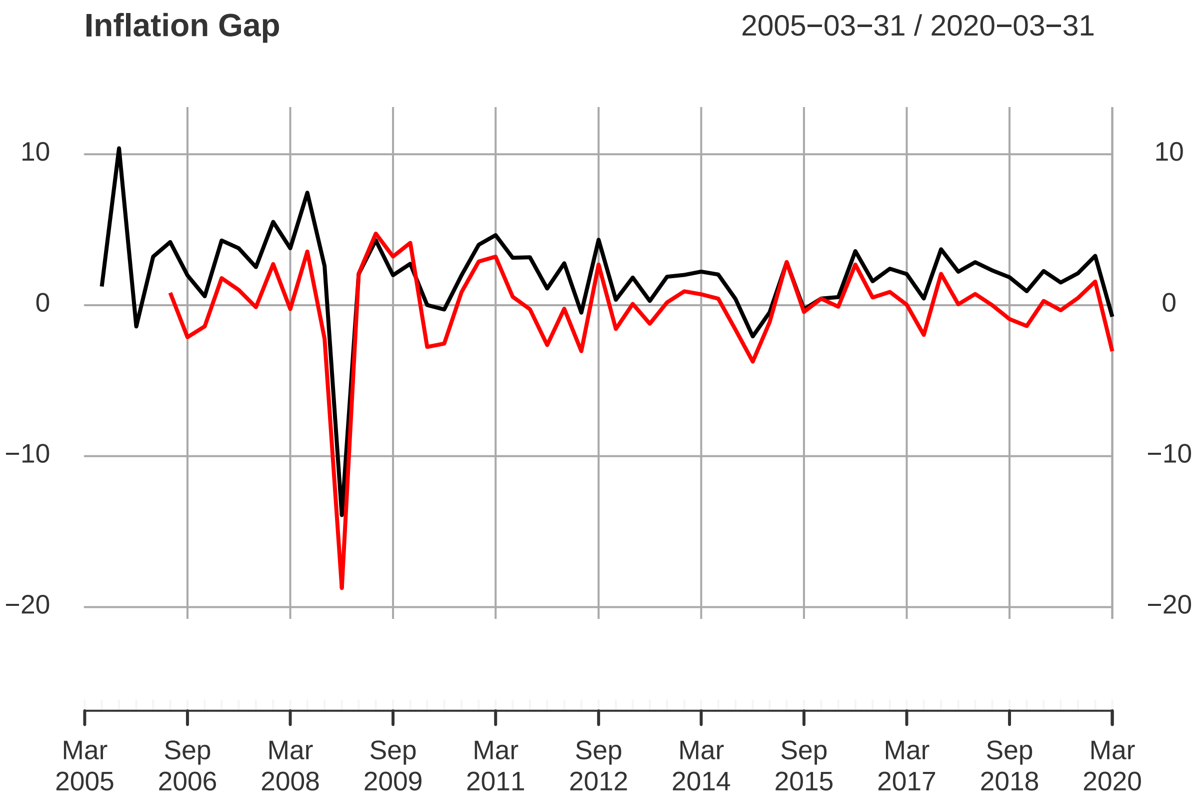
```
addSeries(data_pc$infexp, on = 1, col = "red", lwd = 2)
```



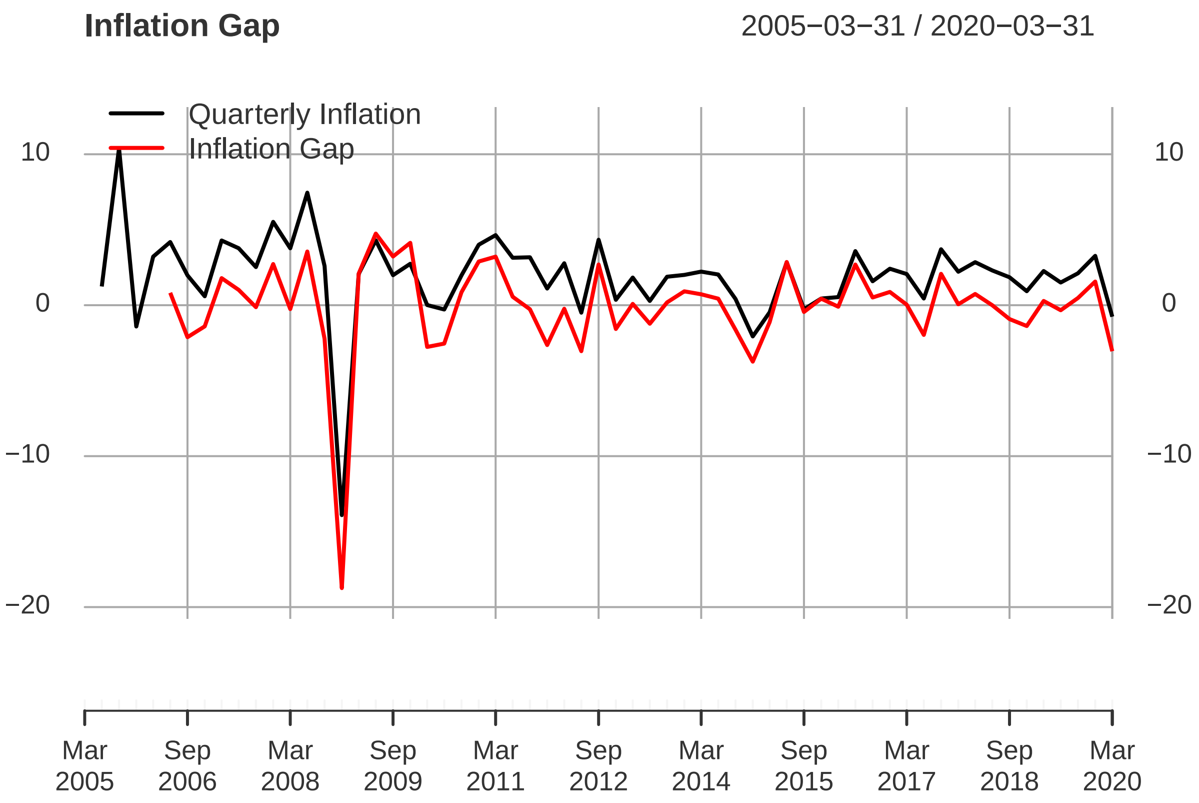
```
#Creating inflation gap
data_pc$infgap <- data_pc$inflation_q-data_pc$infexp
plot.xts(data_pc$inflation_q, main = "Inflation Gap", ylim = c(-25, 15))
```



```
addSeries(data_pc$infgap, on = 1, col = "red", lwd = 2)
```



```
addLegend("topleft", on=1,
          legend.names = c("Quarterly Inflation", "Inflation Gap"),
          lty=c(1, 1), lwd=c(2, 2),
          col=c("black", "red"))
```



```
#Supply shocks
data_pc$supply_shock <- 4*diff(data_pc$l_cpi)*100 - 4*diff(data_pc$l_cpi_core)*100

model1 <- lm(infgap ~ unemployment_rate, data = data_pc)
summary(model1)
```

```
##
## Call:
## lm(formula = infgap ~ unemployment_rate, data = data_pc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7261  -1.1792   0.2388   1.3384   4.6324
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.43779    1.36894  -0.320    0.750
## unemployment_rate  0.05787    0.21083   0.274    0.785
##
## Residual standard error: 3.227 on 54 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.001393,   Adjusted R-squared:  -0.0171
## F-statistic: 0.07535 on 1 and 54 DF,  p-value: 0.7848
```

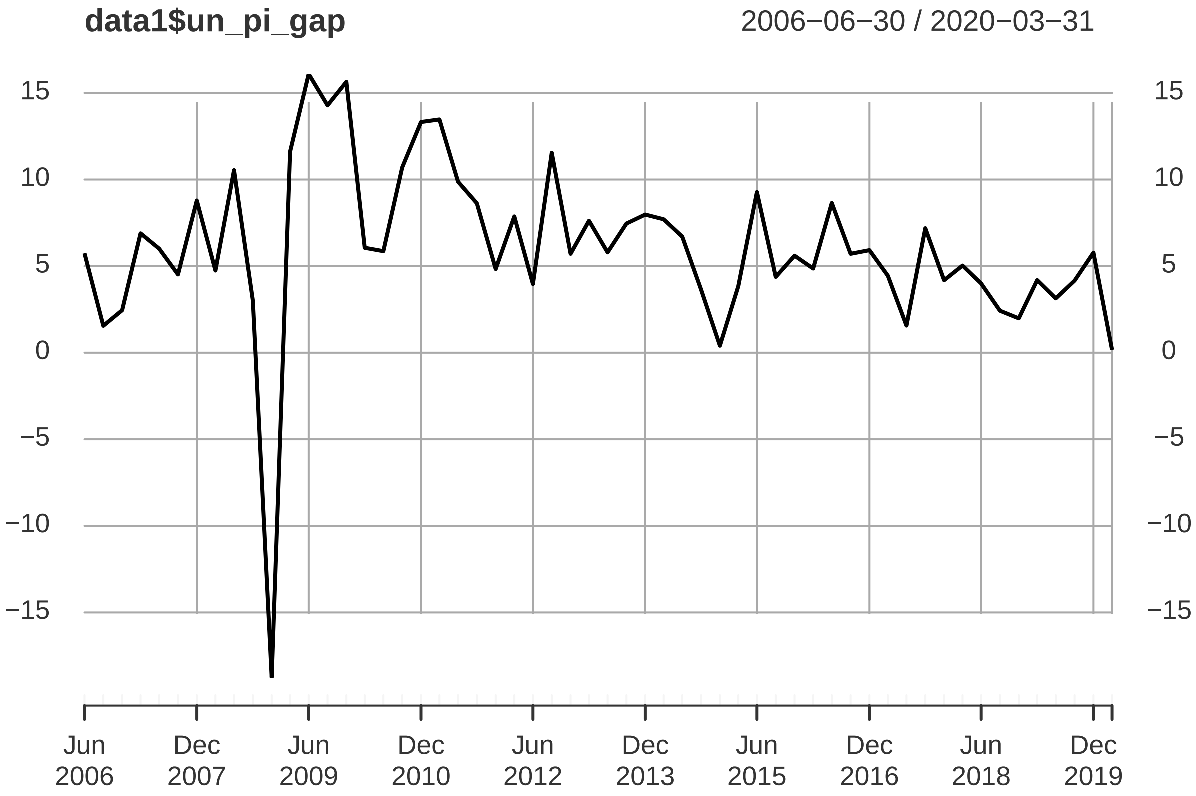
```r
model2 <- lm(infgap ~ 0 + gdpgap, data = data_pc)
summary(model2)
```

```
##
## Call:
## lm(formula = infgap ~ 0 + gdpgap, data = data_pc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.6830  -1.2467   0.1392   1.0632   4.9169
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## gdpgap  0.06793    0.43113   0.158    0.875
##
## Residual standard error: 3.2 on 55 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.0004512,  Adjusted R-squared:  -0.01772
## F-statistic: 0.02483 on 1 and 55 DF,  p-value: 0.8754
```

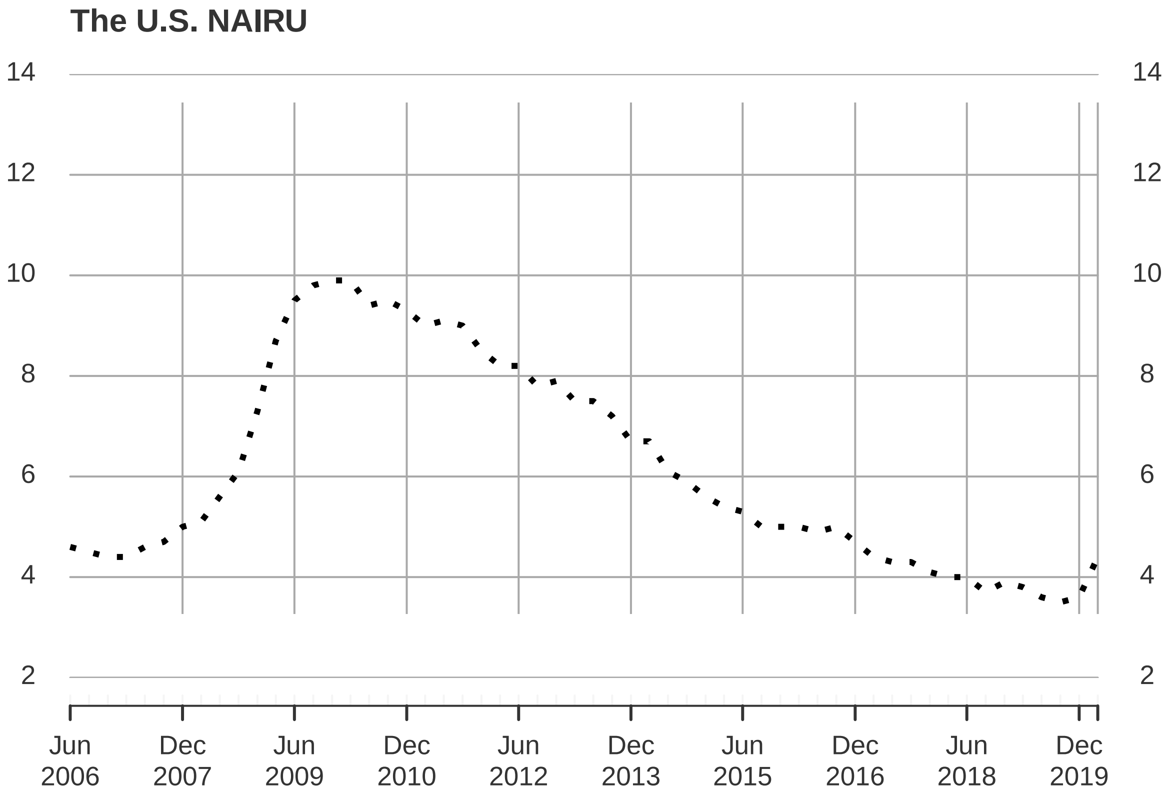
```r
model3 <- lm(infgap ~ unemployment_rate + supply_shock, data = data_pc)
summary(model3)
```

```
##
## Call:
## lm(formula = infgap ~ unemployment_rate + supply_shock, data = data_pc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9684 -0.5616 -0.1596  0.6679  2.9882
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.057197   0.581284   0.098    0.922
## unemployment_rate -0.007194   0.089487  -0.080    0.936
## supply_shock       1.157393   0.073588  15.728   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.368 on 53 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8238, Adjusted R-squared:  0.8171
## F-statistic: 123.9 on 2 and 53 DF,  p-value: < 2.2e-16
```

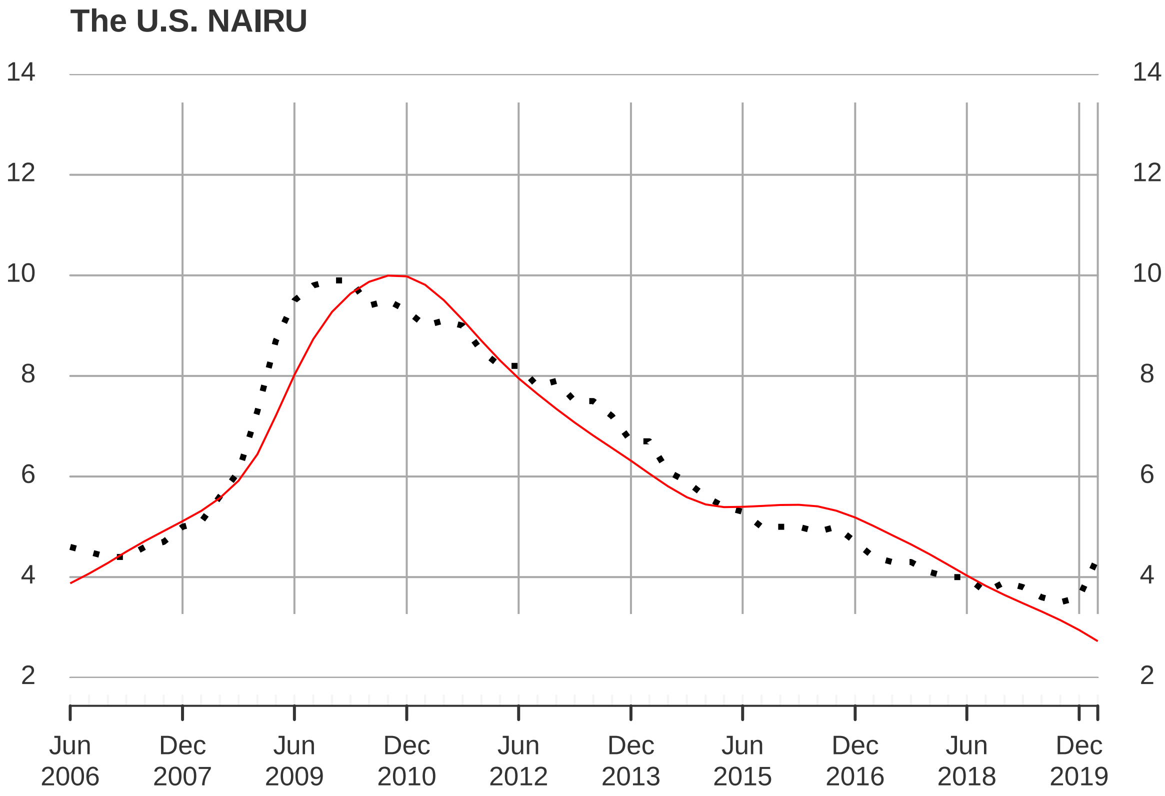
```r
data1 <- na.omit(data_pc)

pc_rolling <- roll_regres(data1$infgap ~ data1$unemployment_rate + data1$supply_shock, width = 40, do_de
data1$un_pi_gap <- data1$unemployment_rate + data1$infgap/(0.007194*100)
#Note that 0.007194 was the estimated coefficient of unemployment rate in model 3.
plot.xts(data1$un_pi_gap)
```



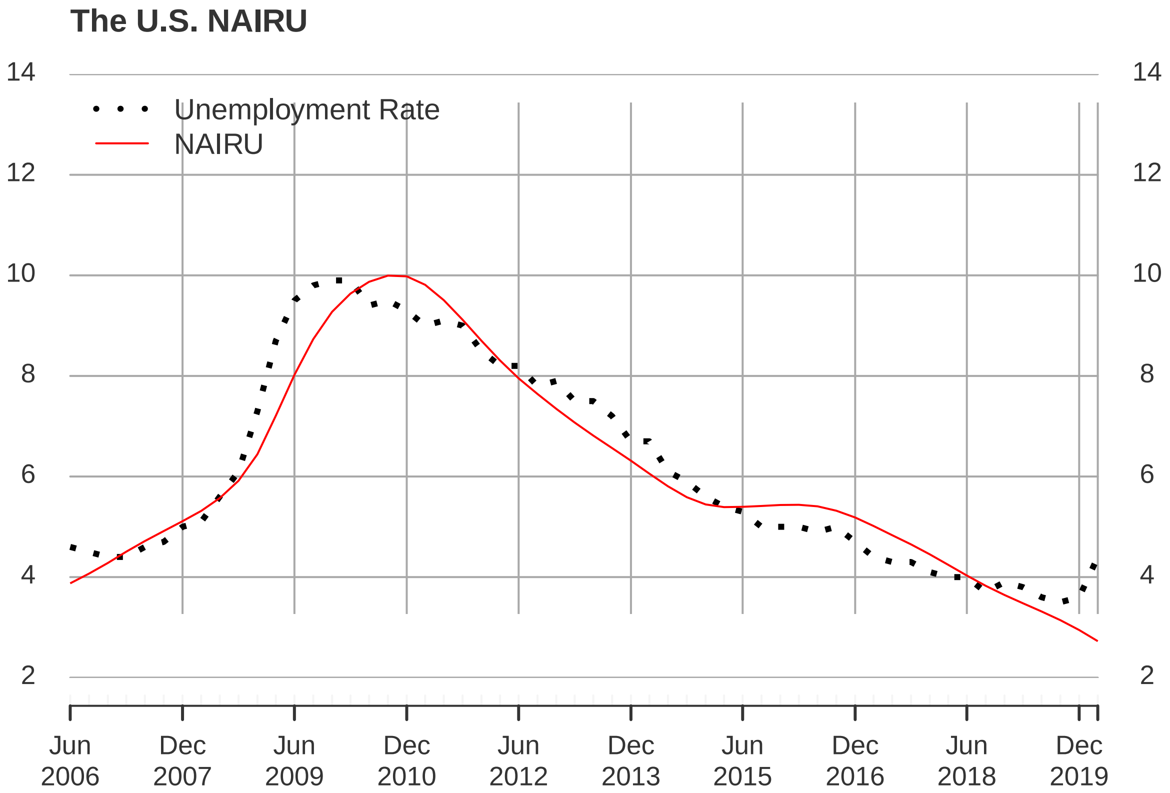
```r
#Get trend using the HP filter with high lambda (much higner than for business cycles frequencies)
data1_1 <- na.omit(data1)

hp_un_pi_gap <- hpfilter(data1_1$un_pi_gap, freq = 100, type="lambda") # lambda at 100
hp_un_pi_gap_1000 <- hpfilter(data1_1$un_pi_gap, freq = 1000, type="lambda") # lambda at 1000


hpgap_dat <- data.frame(hp_un_pi_gap$trend) %>%
  tibble::rownames_to_column("date") %>%
  dplyr::rename(nairu = un_pi_gap)


data2 <- data.frame(data1) %>%
  tibble::rownames_to_column("date")

data3 <- merge(hpgap_dat, data2, by ="date") %>%
  tibble::column_to_rownames("date")

data4 <- as.xts(data3)

data5 <- na.omit(data4)
plot.xts(data5$unemployment_rate, col = "black", lwd = 3, main = "The U.S. NAIRU", main.timespan = FALS
```



```r
addSeries(data5$nairu, on = 1, col = "red", lwd = 1) # NAIRU
```



```r
addLegend("topleft", on=1,
          legend.names = c("Unemployment Rate", "NAIRU"),
          lty=c(3, 1), lwd=c(3, 1),
          col=c("black", "red"))
```



```
summary(model3)
```

```
##
## Call:
## lm(formula = infgap ~ unemployment_rate + supply_shock, data = data_pc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9684 -0.5616 -0.1596  0.6679  2.9882
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.057197   0.581284   0.098    0.922
## unemployment_rate -0.007194   0.089487  -0.080    0.936
## supply_shock       1.157393   0.073588  15.728   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.368 on 53 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8238, Adjusted R-squared:  0.8171
## F-statistic: 123.9 on 2 and 53 DF,  p-value: < 2.2e-16
```

```
# Load the gt and flextable packages
library(gt)
library(flextable)

# Create a data frame with the table data
survey_data <- data.frame(
  Item = c("Frequency of the survey", "Households or persons", "Age limit", "Reference for job search pe
  U.S. = c("Monthly", "Households", "16 years and older civilians", "4 weeks", 60000),
  Japan = c("Monthly", "Households", "15 years and older civilians", "4 weeks", 40000),
  Canada = c("Monthly", "Households", "15 years and older civilians", "4 weeks", 56000),
  E.U. = c("Monthly", "Households or persons", "15 years and older, includes military personnel and civi
)


# Print the gt table as a flextable
ft <- flextable(survey_data)
ft <- set_table_properties(ft, layout = "autofit")

# Print the table
ft
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Item	U.S.	Japan	Canada	E.U
Frequency of the survey	Monthly	Monthly	Monthly	Mo
Households or persons	Households	Households	Households	Ho
Age limit	16 years and older civilians	15 years and older civilians	15 years and older civilians	15
Reference for job search period	4 weeks	4 weeks	4 weeks	4 w
Sample Size	60000	40000	56000	410

```
# Load the gt package
library(gt)

# Create a data frame with the table data
data <- data.frame(
  Measure = c("Inflation measures for projections", "Monitored core inflation measures", "Others"),
  `Federal Reserve Bank (U.S.)` = c("All items, Excl. food and energy", "Exclusion measures", "N/A"),
  `Bank of Japan` = c("Excl. fresh food", "Excl. food and energy", "N/A"),
  `European Central Bank` = c("All items, Excl. energy", "Excl. fresh food, Excl. food and energy, Excl
  `Bank of England` = c("Excl. energy", "Excl. unprocessed food and energy, Excl. food and energy", "Vol
  `Bank of Canada` = c("All items", "Excl. food and energy, Excl. food, energy, and non-alcoholic bevera
)


ft <- flextable(data)

# Print the table
ft
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Measure	Federal.Reserve.Bank.of.U.S.	Bank.of.Japan	European.Central.Bank	Bank.of.England	Bank.of.Canada
Inflation measures for projections	All items, Excl. food and energy	Excl. fresh food	All items, Excl. energy	Excl. energy	All items
Monitored core inflation measures	Exclusion measures	Excl. food and energy	Excl. fresh food, Excl. food and energy, Excl. food and energy	Excl. unprocessed food and energy, Excl. food and energy	Excl. food and energy, Excl. food, energy, and non-alcoholic beverage, Excl. eight of the most volatile component excluding the effect of change s indirect tax
Others	N/A	N/A	Estimates from dynamic factor models	Volatility adjusted CPI	N/A