consolidation code

Charles Vanderbilt

01/09/2019

## template for installing and loading multiple packages at once  
for (package in c("tidyverse","here","skimr","janitor","magrittr","dplyr","reshape","moments","rsdmx","zoo","xts","Quandl","raustats","tidyquant","hydroTSM")) {  
 if (!package %in% installed.packages()) {  
 install.packages(package)  
 }  
 if (!package %in% .packages()) {  
 library(package, character.only = TRUE)  
 }  
}

## ── Attaching packages ──────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.1 ✔ purrr 0.3.2  
## ✔ tibble 2.1.1 ✔ dplyr 0.8.1  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── Conflicts ─────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

## here() starts at /Users/tampilprimananda/Desktop/STDS/AT2/Charles

##   
## Attaching package: 'skimr'

## The following object is masked from 'package:stats':  
##   
## filter

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

## The following objects are masked from 'package:tidyr':  
##   
## expand, smiths

##   
## Attaching package: 'zoo'

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

##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

## Loading required package: readxl

## Attaching package: 'raustats'

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:reshape':  
##   
## stamp

## The following object is masked from 'package:here':  
##   
## here

## The following object is masked from 'package:base':  
##   
## date

## Loading required package: PerformanceAnalytics

##   
## Attaching package: 'PerformanceAnalytics'

## The following objects are masked from 'package:moments':  
##   
## kurtosis, skewness

## The following object is masked from 'package:graphics':  
##   
## legend

## Loading required package: quantmod

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.

##   
## Attaching package: 'hydroTSM'

## The following object is masked from 'package:magrittr':  
##   
## extract

## The following object is masked from 'package:tidyr':  
##   
## extract

######## ANGUS's Code #######  
# get some data ------  
  
(url <- "https://stats.oecd.org/restsdmx/sdmx.ashx/GetData/MEI\_CLI/LOLITONO.AUS.M/all?startTime=2005-01&endTime=2019-07")

## [1] "https://stats.oecd.org/restsdmx/sdmx.ashx/GetData/MEI\_CLI/LOLITONO.AUS.M/all?startTime=2005-01&endTime=2019-07"

dataset <- readSDMX(url)  
OECDLI <- as.data.frame(dataset)  
#Sort dates in xts  
date = seq(as.Date("2005-01-01"), by = "1 month", length.out = nrow(OECDLI))  
OECDLI <- xts(OECDLI[,-1], order.by = date, frequency = 1)  
#select data and label column  
OECDLI <- setNames(OECDLI[,7], "oecd\_li")  
  
(url <- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/MERCH\_IMP/-.-1.-1.-.M/all?startTime=2005-01&endTime=2019-06")

## [1] "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/MERCH\_IMP/-.-1.-1.-.M/all?startTime=2005-01&endTime=2019-06"

dataset <- readSDMX(url)  
AusImport <- as.data.frame(dataset)  
#Sort dates in xts  
date = seq(as.Date("2005-01-01"), by = "1 month",   
 length.out = nrow(AusImport))  
AusImport <- xts(AusImport[,-1], order.by = date, frequency = 1)  
#select data and label column  
AusImport <- setNames(AusImport[,7], "abs\_imports")  
  
(url <- "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/MERCH\_EXP/-.-1.-1.-.M/all?startTime=2005-01&endTime=2019-06")

## [1] "http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/MERCH\_EXP/-.-1.-1.-.M/all?startTime=2005-01&endTime=2019-06"

dataset <- readSDMX(url)  
AusExport <- as.data.frame(dataset)  
#Sort dates in xts  
date = seq(as.Date("2005-01-01"), by = "1 month",   
 length.out = nrow(AusExport))  
AusExport <- xts(AusExport[,-1], order.by = date, frequency = 1)  
#select data and label column  
AusExport <- setNames(AusExport[,7], "abs\_exports")  
  
# Merge Data ----  
  
Combi <- merge(OECDLI, AusImport, join="left")  
Combi <- merge(Combi, AusExport, join="left")  
CombiFrame <- as.data.frame(Combi)  
CombiFrame <- mutate\_all(CombiFrame, function(x) as.numeric(as.character(x)))  
  
head(Combi)

## oecd\_li abs\_imports abs\_exports  
## 2005-01-01 "100.21210" "11154780" " 9232638"   
## 2005-02-01 "100.14620" "11123461" " 9503409"   
## 2005-03-01 "100.06930" "12699350" "10451659"   
## 2005-04-01 "100.01640" "12569908" "11566650"   
## 2005-05-01 " 99.97734" "12788296" "12149662"   
## 2005-06-01 " 99.93616" "12806139" "11582633"

######## JOHN's Code ########  
library(Quandl)  
gold\_forward\_offer\_rates <- Quandl("LBMA/GOFO", api\_key="kf3rSrKM5xnKDzHNL74d")  
#Gold forward rates (GOFO), in percentages; London Bullion Market Association (LBMA). LIBOR difference included. The Gold Forward Offered Rate is an international standard rate at which dealers will lend gold on a swap basis against US dollars, providing the foundation for the pricing of gold swaps, forwards and leases.  
  
#Sort dates in xts  
date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(gold\_forward\_offer\_rates))  
gold\_forward\_offer\_rates <- xts(gold\_forward\_offer\_rates[,-1], order.by = date, frequency = 1)   
gold\_forward\_offer\_rates <- gold\_forward\_offer\_rates["2005-01-01/2019-06-01"]  
gold\_forward\_offer\_rates <- gold\_forward\_offer\_rates$`GOFO - 1 Month`  
Combi <- merge(Combi, gold\_forward\_offer\_rates, join="left")  
  
  
gold\_price\_london\_fixing <- Quandl("LBMA/GOLD", api\_key="kf3rSrKM5xnKDzHNL74d")  
#Sort dates in xts  
date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(gold\_price\_london\_fixing))  
gold\_price\_london\_fixing <- xts(gold\_price\_london\_fixing[,-1], order.by = date, frequency = 1)   
gold\_price\_london\_fixing <- gold\_price\_london\_fixing["2005-01-01/2019-06-01"]  
gold\_price\_london\_fixing <- gold\_price\_london\_fixing$`USD (AM`  
Combi <- merge(Combi, gold\_price\_london\_fixing, join="left")  
  
#Gold Price: London Fixings, London Bullion Market Association (LBMA). Fixing levels are set per troy ounce. The London Gold Fixing Companies set the prices for gold that are globally considered as the international standard for pricing of gold. The Gold price in London is set twice a day by five LBMA Market Makers who comprise the London Gold Market Fixing Limited (LGMFL). The process starts with the announcement from the Chairman of the LGMFL to the other members of the LBMA Market Makers, then relayed to the dealing rooms where customers can express their interest as buyers or sellers and also the quantity they wish to trade. The gold fixing price is then set by collating bids and offers until the supply and demand are matched. At this point the price is announced as the 'Fixed' price for gold and all business is conducted on the basis of that price.  
  
aud\_usd <- Quandl("PERTH/AUD\_USD\_D", api\_key="kf3rSrKM5xnKDzHNL74d")  
#Sort dates in xts  
date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(aud\_usd))  
aud\_usd <- xts(aud\_usd[,-1], order.by = date, frequency = 1)   
aud\_usd <- aud\_usd["2005-01-01/2019-06-01"]  
aud\_usd$aud\_usd\_bid\_avg <- aud\_usd$`Bid Average`  
aud\_usd <- aud\_usd$aud\_usd\_bid\_avg  
Combi <- merge(Combi, aud\_usd, join="left")  
  
#UNEMPLOYMENT  
unemployment <- Quandl("FRED/NROUST", api\_key="kf3rSrKM5xnKDzHNL74d")  
#Sort dates in xts  
date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(unemployment))  
unemployment <- xts(unemployment[,-1], order.by = date, frequency = 1)   
unemployment <- unemployment["2005-01-01/2019-06-01"]  
Combi <- merge(Combi, unemployment, join="left")  
  
head(Combi)

## oecd\_li abs\_imports abs\_exports GOFO...1.Month USD..AM.  
## 2005-01-01 100.21210 11154780 9232638 0.065 1526.55  
## 2005-02-01 100.14620 11123461 9503409 0.090 1536.65  
## 2005-03-01 100.06930 12699350 10451659 0.085 1541.75  
## 2005-04-01 100.01640 12569908 11566650 0.080 1531.85  
## 2005-05-01 99.97734 12788296 12149662 0.080 1495.50  
## 2005-06-01 99.93616 12806139 11582633 0.075 1498.70  
## aud\_usd\_bid\_avg unemployment  
## 2005-01-01 0.6886 4.381  
## 2005-02-01 0.6898 4.386  
## 2005-03-01 0.6907 4.391  
## 2005-04-01 0.6942 4.396  
## 2005-05-01 0.6972 4.401  
## 2005-06-01 0.6983 4.406

######## Charles' Code ########  
 # list functions vailable from raustats package  
 ls("package:raustats")

## [1] "abs\_cachelist" "abs\_cat\_cachelist" "abs\_cat\_download"   
## [4] "abs\_cat\_stats" "abs\_cat\_tables" "abs\_cat\_unzip"   
## [7] "abs\_datasets" "abs\_dimensions" "abs\_metadata"   
## [10] "abs\_read\_tss" "abs\_search" "abs\_stats"   
## [13] "aus\_state\_codes" "rba\_cachelist" "rba\_file\_download"  
## [16] "rba\_read\_tss" "rba\_search" "rba\_stats"   
## [19] "rba\_table\_cache"

# putting the cachelist to an array  
 abslist <- abs\_cat\_cachelist  
   
 # putting the cachelist to an array  
 rbalist <- rba\_cachelist  
   
 ## Download datasets  
 rba\_mon <- rba\_stats("A2")

## Downloading: a02hist.xls  
 |   
 | | 0%  
 |   
 |== | 3%  
 |   
 |==== | 6%  
 |   
 |======= | 10%  
 |   
 |========= | 14%  
 |   
 |=========== | 17%  
 |   
 |============== | 21%  
 |   
 |================ | 25%  
 |   
 |=================== | 29%  
 |   
 |===================== | 32%  
 |   
 |======================= | 36%  
 |   
 |========================== | 40%  
 |   
 |============================ | 43%  
 |   
 |=============================== | 47%  
 |   
 |================================= | 51%  
 |   
 |============================================================ | 92%  
 |   
 |============================================================== | 95%  
 |   
 |================================================================ | 99%  
 |   
 |=================================================================| 100%

## Warning in eval(substitute(list(...)), `\_data`, parent.frame()): NAs  
## introduced by coercion

rba\_infla <- rba\_stats("G1")

## Downloading: g01hist.xls  
 |   
 | | 0%  
 |   
 |= | 1%  
 |   
 |= | 2%  
 |   
 |== | 4%  
 |   
 |=== | 5%  
 |   
 |==== | 6%  
 |   
 |===== | 8%  
 |   
 |====== | 9%  
 |   
 |========== | 16%  
 |   
 |=========== | 17%  
 |   
 |============ | 18%  
 |   
 |============= | 20%  
 |   
 |============== | 21%  
 |   
 |=============== | 23%  
 |   
 |================ | 24%  
 |   
 |========================= | 39%  
 |   
 |=================================== | 54%  
 |   
 |==================================== | 55%  
 |   
 |===================================== | 56%  
 |   
 |===================================== | 58%  
 |   
 |====================================== | 59%  
 |   
 |======================================= | 60%  
 |   
 |=============================================== | 72%  
 |   
 |================================================ | 74%  
 |   
 |================================================= | 75%  
 |   
 |================================================== | 76%  
 |   
 |=================================================== | 78%  
 |   
 |=================================================== | 79%  
 |   
 |==================================================== | 80%  
 |   
 |============================================================== | 95%  
 |   
 |=================================================================| 100%

#### Data Munging ####  
 #### RBA Interest Rates datasets ####  
 colnames(rba\_mon)

## [1] "date" "series\_id" "value"   
## [4] "title" "description" "frequency"   
## [7] "type" "units" "source"   
## [10] "publication\_date" "table\_no" "table\_name"

unique(rba\_mon$title)

## [1] "Change in Cash Rate Target" "New Cash Rate Target"

# Trim datasets  
 col <- c('date','value','title')  
 rba\_mon <- rba\_mon[,col]  
 colnames(rba\_mon)

## [1] "date" "value" "title"

rba\_mon <- subset(rba\_mon, title == "New Cash Rate Target")  
 col1 <- c('date','value')  
 rba\_mon <- rba\_mon[,col1]  
   
 # complete missing month  
 rba\_mon <- rba\_mon %>% complete(date = seq.Date(min(date), max(date), by="month"))  
   
 # take only data from 2005 onwards  
 rba\_mon <- subset(rba\_mon, date >= '2005-01-01')  
   
 # adding rate into first two months of 2005 becuase rate has not changed since dec 2003 which is 5.25  
 rba\_mon[1:2,2] = 5.25   
   
 # populate the rest of the NA  
 rba\_mon\_fin <- rba\_mon %>% fill('value')  
   
 # check to confirm no na  
 unique(is.na(rba\_mon\_fin))

## date value  
## [1,] FALSE FALSE

# covert to data frame  
 rba\_mon\_fin<-as.data.frame(rba\_mon\_fin)  
   
 # correct colname  
 colnames(rba\_mon\_fin) <- c('date','RBA Interest Rate')  
 colnames(rba\_mon\_fin)

## [1] "date" "RBA Interest Rate"

head(rba\_mon\_fin)

## date RBA Interest Rate  
## 1 2005-01-02 5.25  
## 2 2005-02-02 5.25  
## 3 2005-03-02 5.50  
## 4 2005-04-02 5.50  
## 5 2005-05-02 5.50  
## 6 2005-06-02 5.50

#### RBA Year-end Inflation Datasets ####  
   
 colnames(rba\_infla)

## [1] "date" "series\_id" "value"   
## [4] "title" "description" "frequency"   
## [7] "type" "units" "source"   
## [10] "publication\_date" "table\_no" "table\_name"

unique(rba\_infla$title)

## [1] "Consumer price index"   
## [2] "Year-ended inflation"   
## [3] "Year-ended inflation – excluding interest and tax changes"   
## [4] "Year-ended inflation – excluding volatile items"   
## [5] "Year-ended tradables inflation"   
## [6] "Year-ended tradables inflation – excluding volatile items and tobacco"   
## [7] "Year-ended non-tradables inflation"   
## [8] "Year-ended non-tradable inflation – excluding interest charges and deposit & loan facilities"  
## [9] "Year-ended weighted median inflation"   
## [10] "Year-ended trimmed mean inflation"   
## [11] "Quarterly inflation – original"   
## [12] "Quarterly inflation"   
## [13] "Quarterly inflation – excluding interest and tax changes"   
## [14] "Quarterly inflation – excluding volatile items"   
## [15] "Quarterly tradables inflation"   
## [16] "Quarterly tradables inflation – excluding volatile items and tobacco"   
## [17] "Quarterly non-tradables inflation"   
## [18] "Quarterly non-tradables inflation – excluding deposit and loan facilities"   
## [19] "Quarterly weighted median inflation"   
## [20] "Quarterly trimmed mean inflation"

unique(rba\_infla$frequency)

## [1] "Quarterly"

rba\_infla<- subset(rba\_infla, title == "Year-ended inflation")  
 unique(rba\_infla$title)

## [1] "Year-ended inflation"

# Trim datasets  
 col <- c('date','value','title')  
 rba\_infla <- rba\_infla[,col]  
 colnames(rba\_infla)

## [1] "date" "value" "title"

str(rba\_infla)

## 'data.frame': 385 obs. of 3 variables:  
## $ date : Date, format: "1923-06-01" "1923-09-01" ...  
## $ value: num 2.3 3.8 4.6 1.4 -1.7 -3.7 -3.3 0.8 1.6 2.9 ...  
## $ title: chr "Year-ended inflation" "Year-ended inflation" "Year-ended inflation" "Year-ended inflation" ...

col1 <- c('date','value')  
 rba\_infla <- rba\_infla[,col1]  
   
 nrow(rba\_infla)

## [1] 385

# complete missing month and put it on a new variable  
 rba\_infla\_day <- rba\_infla %>% complete(date = seq.Date(min(date), max(date), by="day"))  
   
 # check to see confirm more rows created  
 nrow(rba\_infla\_day)

## [1] 35094

# populate the rest of the NA  
 rba\_infla\_day <- rba\_infla\_day %>% fill('value')  
   
 # check to confirm no na  
 unique(is.na(rba\_infla\_day))

## date value  
## [1,] FALSE FALSE

# take only data from the last reading before 2005 onwards  
 rba\_infla\_day <- subset(rba\_infla\_day, date >= '2005-01-01')  
   
 # convert to monthly data  
 rba\_infla\_day <- as.data.frame(rba\_infla\_day)  
   
 rba\_infla\_day$date <- as.POSIXct.Date(rba\_infla\_day$date)  
 rba\_infla\_day$date <- strptime(rba\_infla\_day$date,"%Y-%m-%d")  
 rba\_infla\_day <- xts(rba\_infla\_day[,-1], order.by=rba\_infla\_day[,1])  
 rba\_infla\_mon <- apply.monthly(rba\_infla\_day,mean)  
 str(rba\_infla\_mon)

## An 'xts' object on 2005-01-31/2019-06-30 containing:  
## Data: num [1:174, 1] 2.5 2.5 2.5 2.4 2.4 ...  
## Indexed by objects of class: [POSIXlt,POSIXt] TZ:   
## xts Attributes:   
## NULL

# convert to data frame  
 rba\_infla\_mon<-as.data.frame(rba\_infla\_mon)  
   
 nrow(rba\_infla\_mon)

## [1] 174

str(rba\_infla\_mon)

## 'data.frame': 174 obs. of 1 variable:  
## $ V1: num 2.5 2.5 2.5 2.4 2.4 ...

rba\_infla\_mon$V1<- format(rba\_infla\_mon$V1, digits=1, nsmall=1)  
   
 head(rba\_infla\_mon)

## V1  
## 2005-01-31 2.5  
## 2005-02-28 2.5  
## 2005-03-31 2.5  
## 2005-04-30 2.4  
## 2005-05-31 2.4  
## 2005-06-30 2.4

tail(rba\_infla\_mon)

## V1  
## 2019-01-31 1.8  
## 2019-02-28 1.8  
## 2019-03-31 1.8  
## 2019-04-30 1.3  
## 2019-05-31 1.3  
## 2019-06-30 1.3

nrow(rba\_infla\_mon)

## [1] 174

colnames(rba\_infla\_mon) <- c("Year-end Inflation")  
 colnames(rba\_infla\_mon)

## [1] "Year-end Inflation"

#### RBA Quarterly Inflation Datasets ####  
 # download datasets  
 rba\_infla\_qrt <- rba\_stats("G1")

## Downloading: g01hist.xls  
 |   
 | | 0%  
 |   
 |= | 1%  
 |   
 |= | 2%  
 |   
 |== | 4%  
 |   
 |=== | 5%  
 |   
 |==== | 6%  
 |   
 |===== | 8%  
 |   
 |====== | 9%  
 |   
 |================ | 24%  
 |   
 |=================== | 29%  
 |   
 |==================== | 31%  
 |   
 |===================== | 32%  
 |   
 |====================== | 33%  
 |   
 |======================= | 35%  
 |   
 |======================= | 36%  
 |   
 |================================= | 51%  
 |   
 |======================================== | 62%  
 |   
 |========================================= | 63%  
 |   
 |========================================== | 64%  
 |   
 |=========================================== | 66%  
 |   
 |============================================ | 67%  
 |   
 |============================================ | 68%  
 |   
 |====================================================== | 83%  
 |   
 |============================================================= | 94%  
 |   
 |============================================================== | 95%  
 |   
 |=============================================================== | 97%  
 |   
 |================================================================ | 98%  
 |   
 |=================================================================| 99%  
 |   
 |=================================================================| 100%

colnames(rba\_infla\_qrt)

## [1] "date" "series\_id" "value"   
## [4] "title" "description" "frequency"   
## [7] "type" "units" "source"   
## [10] "publication\_date" "table\_no" "table\_name"

unique(rba\_infla\_qrt$title)

## [1] "Consumer price index"   
## [2] "Year-ended inflation"   
## [3] "Year-ended inflation – excluding interest and tax changes"   
## [4] "Year-ended inflation – excluding volatile items"   
## [5] "Year-ended tradables inflation"   
## [6] "Year-ended tradables inflation – excluding volatile items and tobacco"   
## [7] "Year-ended non-tradables inflation"   
## [8] "Year-ended non-tradable inflation – excluding interest charges and deposit & loan facilities"  
## [9] "Year-ended weighted median inflation"   
## [10] "Year-ended trimmed mean inflation"   
## [11] "Quarterly inflation – original"   
## [12] "Quarterly inflation"   
## [13] "Quarterly inflation – excluding interest and tax changes"   
## [14] "Quarterly inflation – excluding volatile items"   
## [15] "Quarterly tradables inflation"   
## [16] "Quarterly tradables inflation – excluding volatile items and tobacco"   
## [17] "Quarterly non-tradables inflation"   
## [18] "Quarterly non-tradables inflation – excluding deposit and loan facilities"   
## [19] "Quarterly weighted median inflation"   
## [20] "Quarterly trimmed mean inflation"

unique(rba\_infla\_qrt$frequency)

## [1] "Quarterly"

rba\_infla\_qrt<- subset(rba\_infla\_qrt, title == "Quarterly inflation")  
 unique(rba\_infla\_qrt$title)

## [1] "Quarterly inflation"

# Trim datasets  
 col <- c('date','value','title')  
 rba\_infla\_qrt <- rba\_infla\_qrt[,col]  
 colnames(rba\_infla\_qrt)

## [1] "date" "value" "title"

str(rba\_infla)

## 'data.frame': 385 obs. of 2 variables:  
## $ date : Date, format: "1923-06-01" "1923-09-01" ...  
## $ value: num 2.3 3.8 4.6 1.4 -1.7 -3.7 -3.3 0.8 1.6 2.9 ...

col1 <- c('date','value')  
 rba\_infla\_qrt <- rba\_infla\_qrt[,col1]  
   
 nrow(rba\_infla\_qrt)

## [1] 149

# convert to daily readings  
 rba\_infla\_qrt\_day <- rba\_infla\_qrt %>% complete(date = seq.Date(min(date), max(date), by="day"))  
   
 # check to see confirm more rows created  
 nrow(rba\_infla\_qrt\_day)

## [1] 13515

# populate the rest of the NA on daily readings  
 rba\_infla\_qrt\_day <- rba\_infla\_qrt\_day %>% fill('value')  
   
 #confirm no NA  
 unique(is.na(rba\_infla\_day))

## [,1]  
## [1,] FALSE

# take only data from the last reading before 2005 onwards  
 rba\_infla\_qrt\_day <- subset(rba\_infla\_qrt\_day, date >= '2005-01-01')  
   
 # convert to monthly data  
 rba\_infla\_qrt\_day <- as.data.frame(rba\_infla\_qrt\_day)  
   
 rba\_infla\_qrt\_day$date <- as.POSIXct.Date(rba\_infla\_qrt\_day$date)  
 rba\_infla\_qrt\_day$date <- strptime(rba\_infla\_qrt\_day$date,"%Y-%m-%d")  
 rba\_infla\_qrt\_day <- xts(rba\_infla\_qrt\_day[,-1], order.by=rba\_infla\_qrt\_day[,1])  
 rba\_infla\_qrt\_mon <- apply.monthly(rba\_infla\_qrt\_day,mean)  
 str(rba\_infla\_qrt\_mon)

## An 'xts' object on 2005-01-31/2019-06-30 containing:  
## Data: num [1:174, 1] 1 1 0.984 0.5 0.5 ...  
## Indexed by objects of class: [POSIXlt,POSIXt] TZ:   
## xts Attributes:   
## NULL

rba\_infla\_qrt\_mon<-as.data.frame(rba\_infla\_qrt\_mon)  
   
 nrow(rba\_infla\_qrt\_mon)

## [1] 174

str(rba\_infla\_qrt\_mon)

## 'data.frame': 174 obs. of 1 variable:  
## $ V1: num 1 1 0.984 0.5 0.5 ...

rba\_infla\_qrt\_mon$V1 <- as.numeric(as.character(rba\_infla\_qrt\_mon$V1))  
 str(rba\_infla\_qrt\_mon)

## 'data.frame': 174 obs. of 1 variable:  
## $ V1: num 1 1 0.984 0.5 0.5 ...

summary(rba\_infla\_qrt\_mon)

## V1   
## Min. :-0.1000   
## 1st Qu.: 0.3975   
## Median : 0.6000   
## Mean : 0.5965   
## 3rd Qu.: 0.8000   
## Max. : 1.5000

rba\_infla\_qrt\_mon$V1 <- round(rba\_infla\_qrt\_mon$V1,1)  
   
   
 head(rba\_infla\_qrt\_mon)

## V1  
## 2005-01-31 1.0  
## 2005-02-28 1.0  
## 2005-03-31 1.0  
## 2005-04-30 0.5  
## 2005-05-31 0.5  
## 2005-06-30 0.5

tail(rba\_infla\_qrt\_mon)

## V1  
## 2019-01-31 0.4  
## 2019-02-28 0.4  
## 2019-03-31 0.4  
## 2019-04-30 0.1  
## 2019-05-31 0.1  
## 2019-06-30 0.1

nrow(rba\_infla\_qrt\_mon)

## [1] 174

colnames(rba\_infla\_qrt\_mon) <- c("Quarterly Inflation")  
 colnames(rba\_infla\_qrt\_mon)

## [1] "Quarterly Inflation"

#### Merge the three datasets ####  
 # list all the datasets  
 head(rba\_mon\_fin)

## date RBA Interest Rate  
## 1 2005-01-02 5.25  
## 2 2005-02-02 5.25  
## 3 2005-03-02 5.50  
## 4 2005-04-02 5.50  
## 5 2005-05-02 5.50  
## 6 2005-06-02 5.50

tail(rba\_mon\_fin)

## date RBA Interest Rate  
## 170 2019-02-02 3.75  
## 171 2019-03-02 3.75  
## 172 2019-04-02 3.75  
## 173 2019-05-02 3.75  
## 174 2019-06-02 3.75  
## 175 2019-07-02 3.75

head(rba\_infla\_mon)

## Year-end Inflation  
## 2005-01-31 2.5  
## 2005-02-28 2.5  
## 2005-03-31 2.5  
## 2005-04-30 2.4  
## 2005-05-31 2.4  
## 2005-06-30 2.4

head(rba\_infla\_qrt\_mon)

## Quarterly Inflation  
## 2005-01-31 1.0  
## 2005-02-28 1.0  
## 2005-03-31 1.0  
## 2005-04-30 0.5  
## 2005-05-31 0.5  
## 2005-06-30 0.5

# check row numbers for all the datasets  
 nrow(rba\_mon\_fin)

## [1] 175

nrow(rba\_infla\_mon)

## [1] 174

nrow(rba\_infla\_qrt\_mon)

## [1] 174

# summary & str  
 str(rba\_infla\_mon)

## 'data.frame': 174 obs. of 1 variable:  
## $ Year-end Inflation: chr "2.5" "2.5" "2.5" "2.4" ...

# sort date in xts for rba\_mon\_fin  
 date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(rba\_mon\_fin))  
 rba\_mon\_fin <- xts(rba\_mon\_fin, order.by = date, frequency = 1)   
   
 # cut off excesses date range and put in the correct date range  
 rba\_mon\_fin <- rba\_mon\_fin["2005-01-01/2019-06-01"]  
   
 # take out the date value  
 rba\_mon\_fin <- rba\_mon\_fin[,2]  
   
 # sort date in xts for rba\_infla\_mon  
 date <- seq(as.Date("2005-01-01"), by = "1 month",length.out = nrow(rba\_mon\_fin))  
 rba\_infla\_mon <- xts( x = rba\_infla\_mon, order.by = date)  
 rba\_infla\_mon <- as.xts(rba\_infla\_mon)  
 # rba\_infla\_mon <- xts(rba\_infla\_mon[,-1], order.by = date, frequency = 1)   
   
 # sort date in xts for rba\_infla\_qrt\_mon  
 date <- seq(as.Date("2005-01-01/2019-06-01"), by = "1 month",   
 length.out = nrow(rba\_infla\_qrt\_mon))  
 rba\_infla\_qrt\_mon <- xts(rba\_infla\_qrt\_mon, order.by = date, frequency = 1)   
   
 # merge with the consolidated datasets  
 Combi <- merge(Combi, rba\_mon\_fin, join="left")  
 Combi <- merge(Combi, rba\_infla\_mon, join="left")  
 Combi <- merge(Combi, rba\_infla\_qrt\_mon, join="left")  
   
 colnames(Combi)

## [1] "oecd\_li" "abs\_imports" "abs\_exports"   
## [4] "GOFO...1.Month" "USD..AM." "aud\_usd\_bid\_avg"   
## [7] "unemployment" "RBA.Interest.Rate" "Year.end.Inflation"   
## [10] "Quarterly.Inflation"

head(Combi)

## oecd\_li abs\_imports abs\_exports GOFO...1.Month USD..AM.  
## 2005-01-01 100.21210 11154780 9232638 0.065 1526.55  
## 2005-02-01 100.14620 11123461 9503409 0.090 1536.65  
## 2005-03-01 100.06930 12699350 10451659 0.085 1541.75  
## 2005-04-01 100.01640 12569908 11566650 0.080 1531.85  
## 2005-05-01 99.97734 12788296 12149662 0.080 1495.50  
## 2005-06-01 99.93616 12806139 11582633 0.075 1498.70  
## aud\_usd\_bid\_avg unemployment RBA.Interest.Rate  
## 2005-01-01 0.6886 4.381 5.25  
## 2005-02-01 0.6898 4.386 5.25  
## 2005-03-01 0.6907 4.391 5.50  
## 2005-04-01 0.6942 4.396 5.50  
## 2005-05-01 0.6972 4.401 5.50  
## 2005-06-01 0.6983 4.406 5.50  
## Year.end.Inflation Quarterly.Inflation  
## 2005-01-01 2.5 1.0  
## 2005-02-01 2.5 1.0  
## 2005-03-01 2.5 1.0  
## 2005-04-01 2.4 0.5  
## 2005-05-01 2.4 0.5  
## 2005-06-01 2.4 0.5