pkb.R

bartosz

2020-02-11

library(readxl)  
pkb = read\_xlsx("data/data Poland PKB.xlsx")  
head(pkb)

## # A tibble: 6 x 2  
## Date `GDP in mln PLN`  
## <chr> <dbl>  
## 1 1993 01 45366.  
## 2 1993 02 48078.  
## 3 1993 03 47478.  
## 4 1993 04 56081.  
## 5 1994 01 59861.  
## 6 1994 02 69447.

pkb$yy = as.numeric(sapply(pkb$Date, function(x) strsplit(x, " ")[[1]])[1,])  
pkb$mm = as.numeric(sapply(pkb$Date, function(x) strsplit(x, " ")[[1]])[2,])  
head(pkb )

## # A tibble: 6 x 4  
## Date `GDP in mln PLN` yy mm  
## <chr> <dbl> <dbl> <dbl>  
## 1 1993 01 45366. 1993 1  
## 2 1993 02 48078. 1993 2  
## 3 1993 03 47478. 1993 3  
## 4 1993 04 56081. 1993 4  
## 5 1994 01 59861. 1994 1  
## 6 1994 02 69447. 1994 2

library(tidyverse)

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

## ✓ ggplot2 3.2.1 ✓ purrr 0.3.3  
## ✓ tibble 2.1.3 ✓ dplyr 0.8.4  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.4.0

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

pkb = select(pkb, -Date)   
head(pkb)

## # A tibble: 6 x 3  
## `GDP in mln PLN` yy mm  
## <dbl> <dbl> <dbl>  
## 1 45366. 1993 1  
## 2 48078. 1993 2  
## 3 47478. 1993 3  
## 4 56081. 1993 4  
## 5 59861. 1994 1  
## 6 69447. 1994 2

colnames(pkb) = c("gdp", "yy", "mm")  
pkb = select(pkb, "yy", "mm", "gdp")  
saveRDS(pkb, "data/pkb.rds")  
pkb = readRDS("data/pkb.rds")  
  
library(fpp)

## Loading required package: forecast

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

## Loading required package: fma

## Loading required package: expsmooth

## Loading required package: lmtest

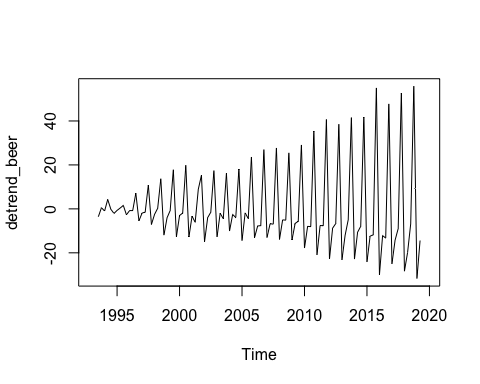
## Loading required package: zoo

##   
## Attaching package: 'zoo'

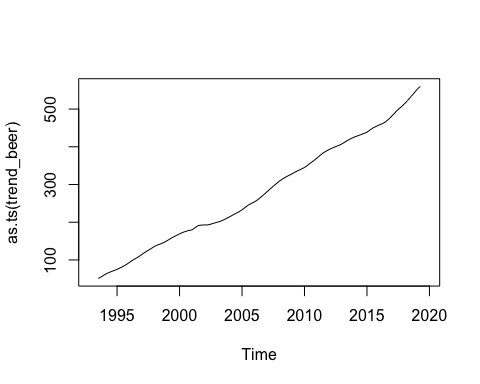
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: tseries

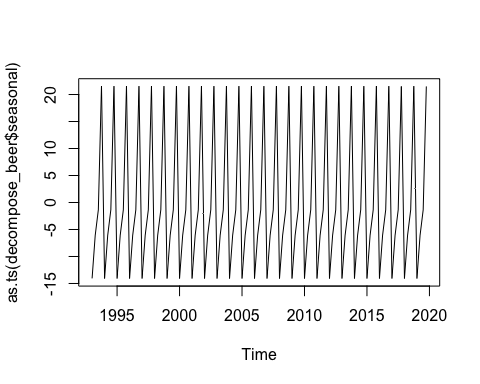
pkb2 = ts(pkb$gdp/1000, frequency = 4, start = c(1993,1), end = c(2019,4))  
  
library(forecast)  
trend\_beer = ma(pkb2, order = 4, centre = T)  
detrend\_beer = pkb2 - trend\_beer  
plot(detrend\_beer)



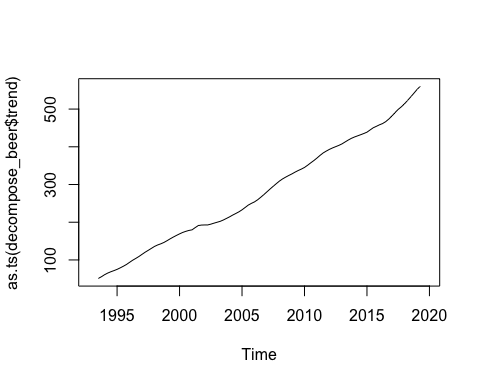
plot(as.ts(trend\_beer))



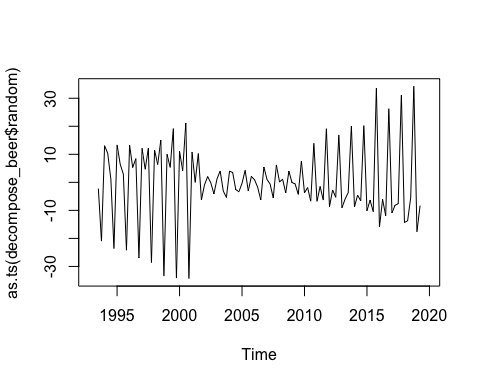
ts\_beer = pkb2  
decompose\_beer = decompose(ts\_beer, "additive")  
  
plot(as.ts(decompose\_beer$seasonal))



plot(as.ts(decompose\_beer$trend))



plot(as.ts(decompose\_beer$random))



svg("dekompozycja.svg")  
plot(decompose\_beer)  
dev.off()

## quartz\_off\_screen   
## 2

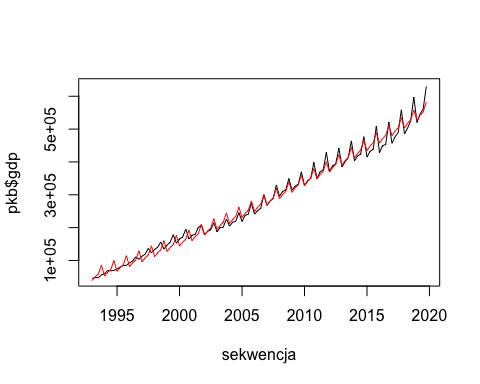
#remotes::install\_github("statisticspoland/R\_Package\_to\_API\_BDL", upgrade = "always", type = "binary")  
#library(bdl)  
decompose\_beer$random

## Qtr1 Qtr2 Qtr3 Qtr4  
## 1993 NA NA -2.22629543 -20.96757476  
## 1994 13.07341226 10.35901705 1.21375981 -23.58377722  
## 1995 13.35568215 6.48059387 2.90599998 -24.20094465  
## 1996 13.27841331 5.29171887 8.46159998 -26.91615715  
## 1997 12.10911331 4.66091887 12.17502498 -28.57569465  
## 1998 11.54681331 6.36668137 15.05821248 -33.44484465  
## 1999 9.99655081 5.33731887 19.15362498 -34.15188215  
## 2000 11.01861331 4.18723137 21.12569998 -34.27200715  
## 2001 10.74097581 -0.01059363 10.26511248 -6.24353215  
## 2002 -0.84987419 2.06258137 -0.15192502 -4.12171965  
## 2003 1.25521331 4.06280637 -3.09851252 -5.38356965  
## 2004 3.97262581 3.52218137 -2.56711252 -3.36781965  
## 2005 -0.32178669 4.26050637 -3.05486252 2.10718035  
## 2006 0.93387581 -1.71315613 -6.25361252 5.43531785  
## 2007 1.05162581 -0.64430613 -5.58520002 6.21244285  
## 2008 0.18073831 1.07701887 -3.77286252 3.98131785  
## 2009 -0.07122419 -0.55013113 -4.28167502 7.58106785  
## 2010 -3.75798669 -1.87660613 -6.72362502 13.86981785  
## 2011 -6.77176169 -1.47189363 -6.32156252 19.12836785  
## 2012 -8.74238669 -2.70855613 -5.33358752 16.92098035  
## 2013 -9.05513669 -5.97564363 -3.59965002 20.00140535  
## 2014 -8.78172419 -4.55278113 -6.58351252 20.21304285  
## 2015 -10.12343669 -6.35421863 -10.42495002 33.52890535  
## 2016 -15.89714919 -6.09675613 -11.90838752 26.26325535  
## 2017 -10.86741169 -8.19930613 -7.58955002 31.16330535  
## 2018 -14.28934919 -13.80090613 -5.47860002 34.22666785  
## 2019 -17.58087419 -8.31016863 NA NA

# chyba lepiej walnąć jakiś model regresyjny  
head(pkb)

## # A tibble: 6 x 3  
## yy mm gdp  
## <dbl> <dbl> <dbl>  
## 1 1993 1 45366.  
## 2 1993 2 48078.  
## 3 1993 3 47478.  
## 4 1993 4 56081.  
## 5 1994 1 59861.  
## 6 1994 2 69447.

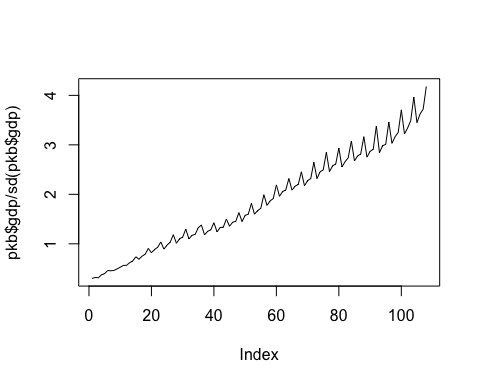
# skopiujmy dane do pkb2  
pkb2 = pkb  
pkb2$mm = pkb2$yy + (1/4\*1:4-0.15)  
pkb2$mm2 = 1:4  
model = lm(gdp~mm+I(mm^2)+as.factor(mm2), data=pkb2)  
  
sekwencja = seq.Date(as.Date("1993-01-01"), as.Date("2019-12-31"), by = "quarter")  
  
plot(sekwencja, pkb$gdp, type= 'l')  
lines(sekwencja, predict(model), col='red')



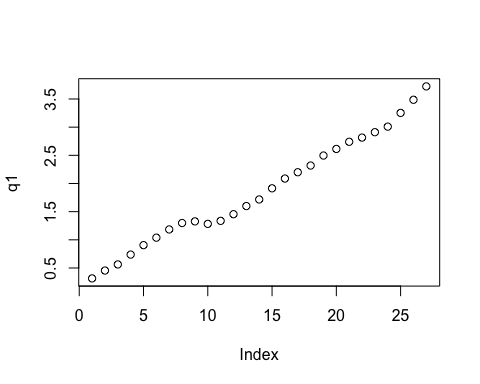
# generalnie rośnie nam niesamowicie odchylenie standardowe w kolejnych latach   
# zatem:  
head(pkb)

## # A tibble: 6 x 3  
## yy mm gdp  
## <dbl> <dbl> <dbl>  
## 1 1993 1 45366.  
## 2 1993 2 48078.  
## 3 1993 3 47478.  
## 4 1993 4 56081.  
## 5 1994 1 59861.  
## 6 1994 2 69447.

test = filter(pkb, mm == 1)  
  
pkb$sd = pkb$gdp/sd(pkb$gdp)  
plot(pkb$gdp/sd(pkb$gdp), type='l')



q1 = filter(pkb, mm == 3) %>% select(sd) %>% unlist()  
plot(q1)



dane = data.frame(data = 1993:2019, q1 = q1)  
plot(dane)  
model1 = lm(q1 ~ data, dane)  
abline(model1)

