Econometrie II : Transports

Rapport

Guewen HESLAN - Bijan VALILOU

Janvier 2021

Contents

1 Introduction

```
library(usethis)
library(readxl)
library(gitcreds)
library(strucchange)
```

```
## Le chargement a nécessité le package : zoo

##
## Attachement du package : 'zoo'

## Les objets suivants sont masqués depuis 'package:base':
##
##     as.Date, as.Date.numeric

## Le chargement a nécessité le package : sandwich
```

```
library(gets)
```

```
## Le chargement a nécessité le package : parallel
```

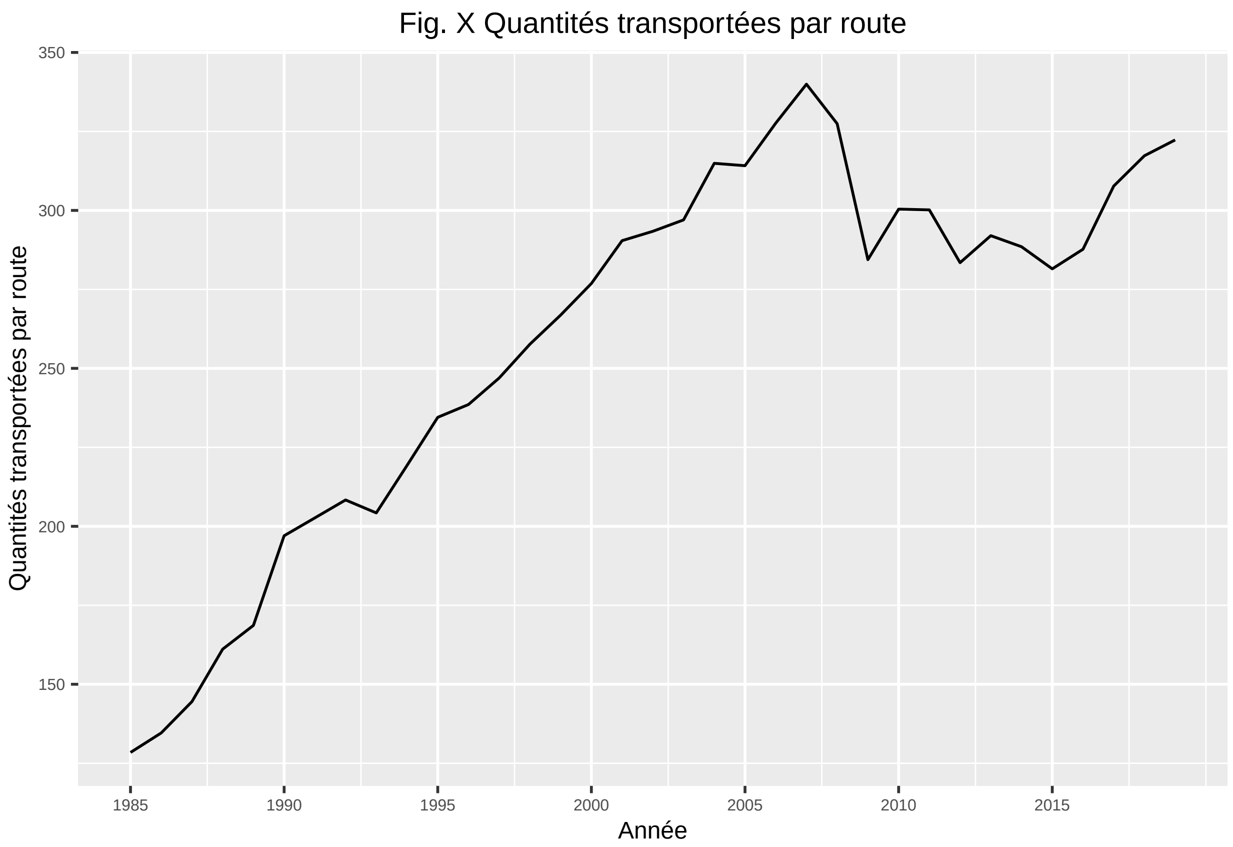
```
library(ggplot2)
library(kableExtra)
library(pander)
```

```
Transport_France2019 <- read_excel("Transport_France2019_v2.xlsx")

##Vecteurs des séries
#Qtt_Trsp_route <- Transport_France2019$Qtt_Trsp_route
#Qtt_Trsp_train <- Transport_France2019$Qtt_Trsp_train
#Pdiesel <- Transport_France2019$Pdiesel
#Qdiesel <- Transport_France2019$QDiesel
#GDP <- Transport_France2019$GDP
#CPI <- Transport_France2019$CPI
#QDieselCamion <- Transport_France2019$Qdieselcamion
```

```
##Séries temporelles
Qtt_Trsp_route.ts <- ts(Transport_France2019$Qtt_Trsp_route, start=c(1985) , end=c(2019), frequency=1)
Qtt_Trsp_train.ts <- ts(Transport_France2019$Qtt_Trsp_train, start=c(1985) , end=c(2019), frequency=1)
Pdiesel.ts <- ts(Transport_France2019$Pdiesel, start=c(1985) , end=c(2019), frequency=1)
Qdiesel.ts <- ts(Transport_France2019$QDiesel, start=c(1985) , end=c(2019), frequency=1)
GDP.ts <- ts(Transport_France2019$"PIB en volume (en milliards d'euros 2014)", start=c(1985) , end=c(20
CPI.ts <- ts(Transport_France2019$CPI, start=c(1985) , end=c(2019), frequency=1)
Qdieselcamion.ts <- ts(Transport_France2019$Qdieselcamion, start=c(1985) , end=c(2019), frequency=1)
```

```
##Graph en niveau
ggplot() +
  geom_line( aes(x = Transport_France2019$Year,y = Transport_France2019$Qtt_Trsp_route))+
labs(x = "Année", y = "Quantités transportées par route", title = "Fig. X Quantités transportées par rou
    theme(axis.text=element_text(size=6),legend.text=element_text(size=7),axis.title=element_text(size=
  scale_x_continuous(breaks=seq(1985,2019,5))
```



```r
ggplot() +
  geom_line( aes(x = Transport_France2019$Year,y = Transport_France2019$Qtt_Trsp_train))+
labs(x = "Année", y = "Quantités transportées par train", title = "Fig. X Quantités transportées par tra
    theme(axis.text=element_text(size=6),legend.text=element_text(size=7),axis.title=element_text(size=
  scale_x_continuous(breaks=seq(1985,2019,5))
```

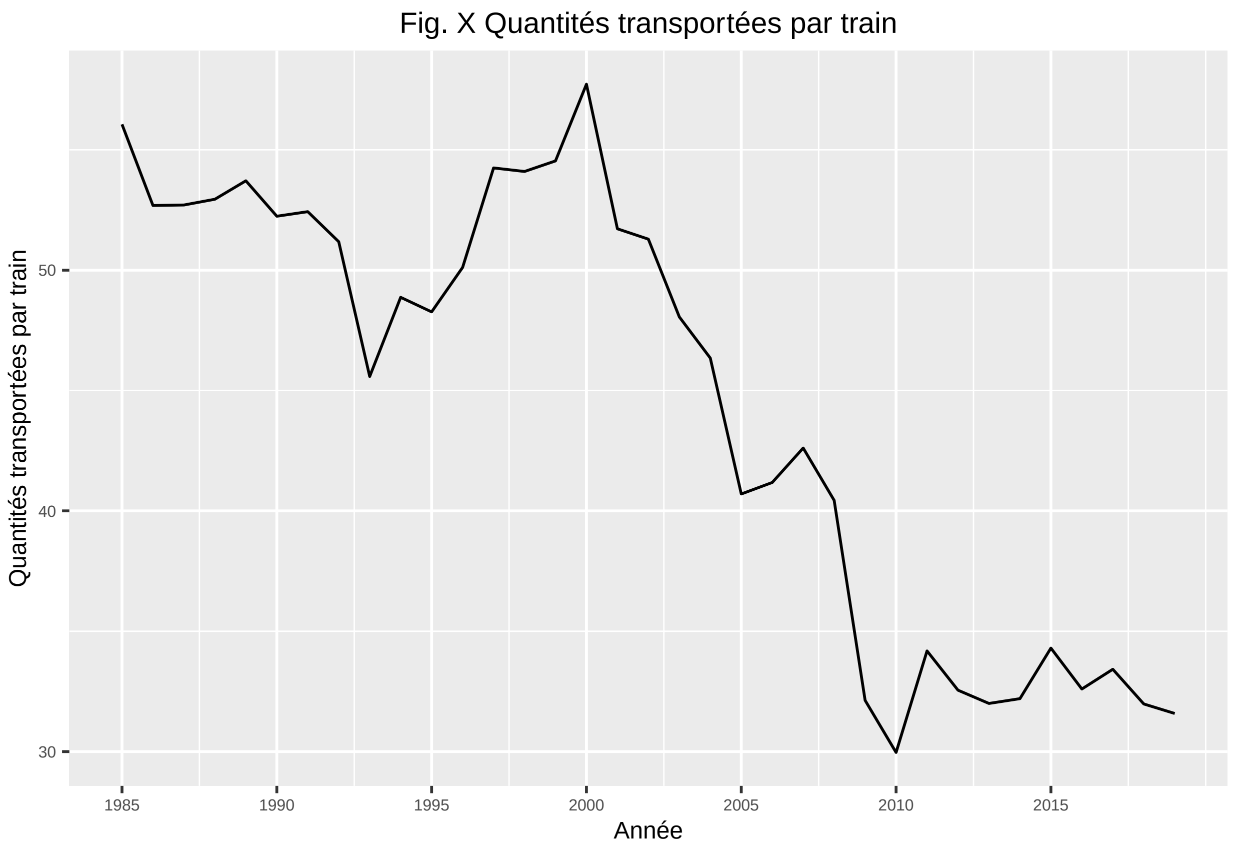
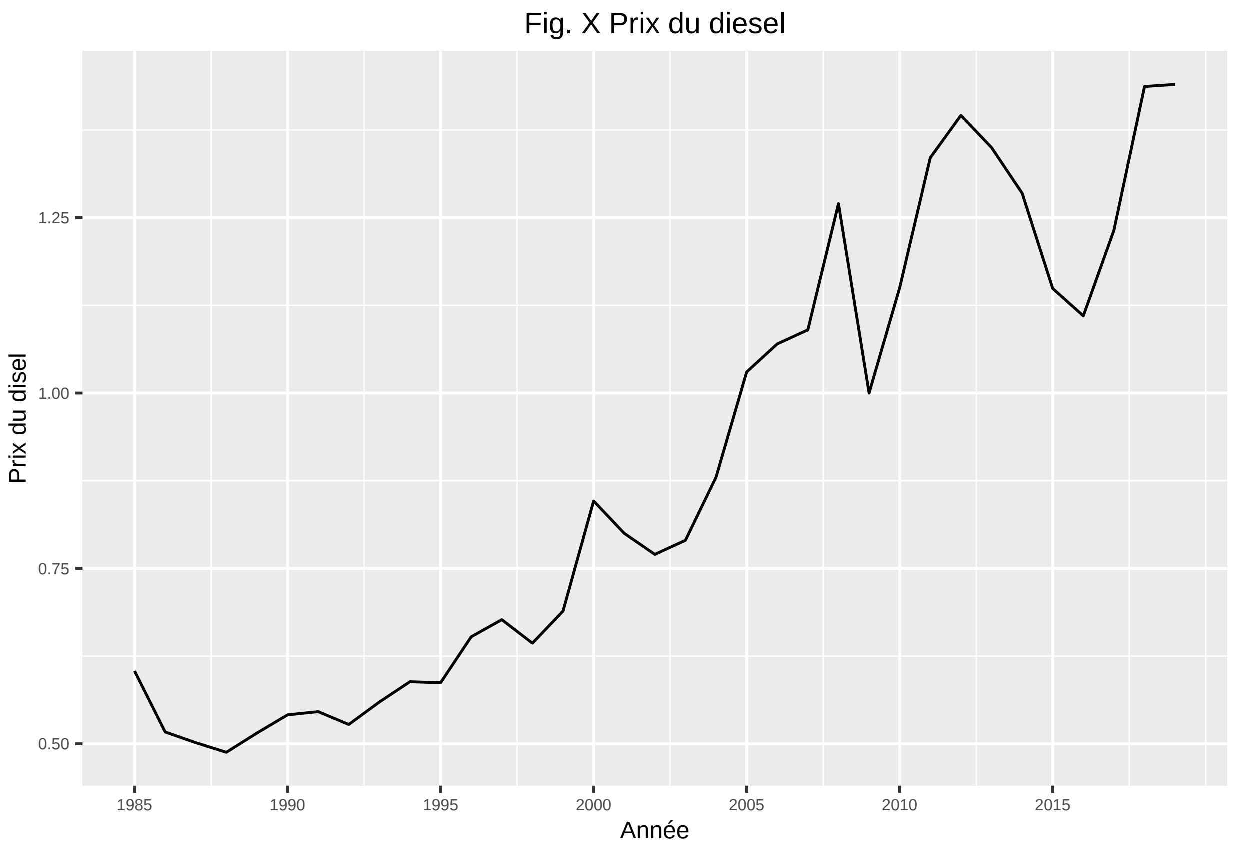
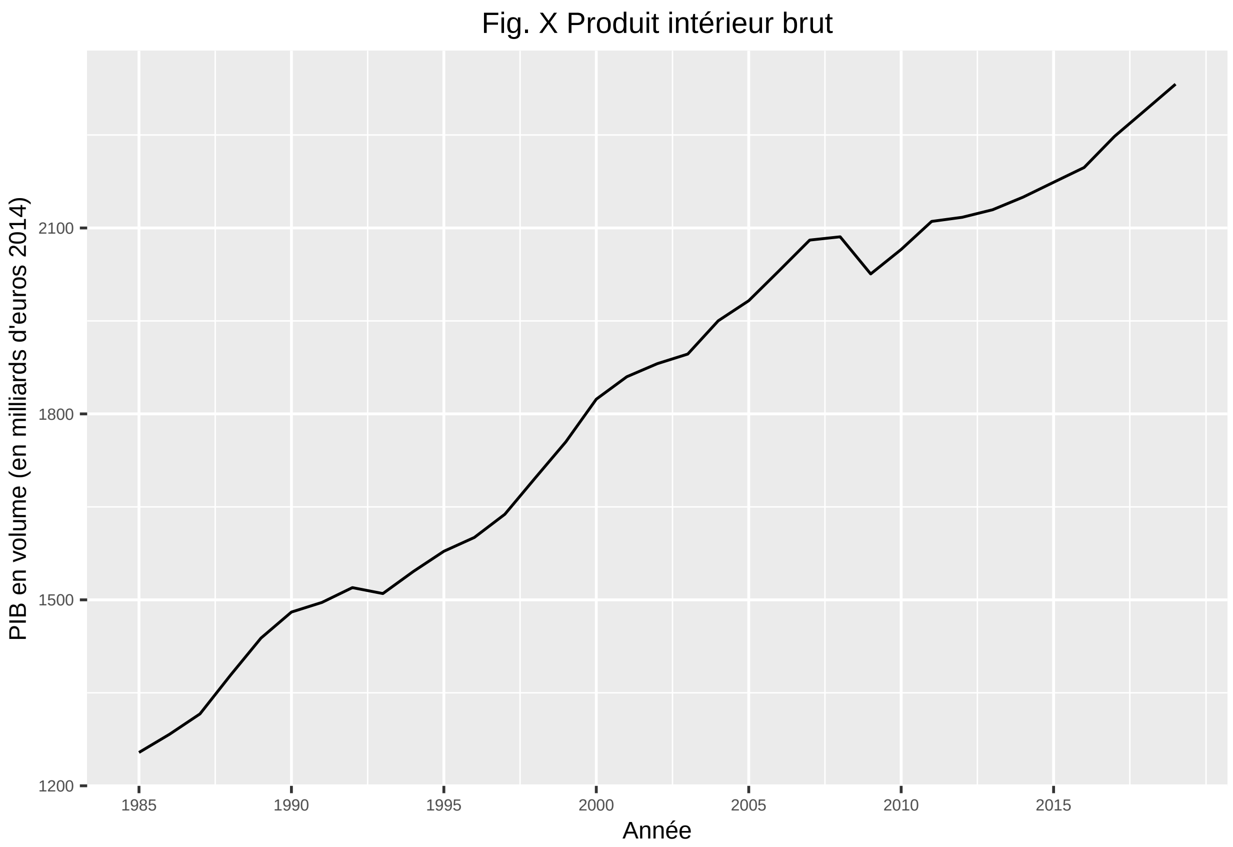


Fig. X Quantités transportées par train

```
ggplot() +
  geom_line( aes(x = Transport_France2019$Year,y = Transport_France2019$Pdiesel))+
labs(x = "Année", y = "Prix du disel", title = "Fig. X Prix du diesel") +
    theme(axis.text=element_text(size=6),legend.text=element_text(size=7),axis.title=element_text(size=9
  scale_x_continuous(breaks=seq(1985,2019,5))
```



```
ggplot() +
  geom_line( aes(x = Transport_France2019$Year,y = Transport_France2019$"PIB en volume (en milliards d'e
labs(x = "Année", y = "PIB en volume (en milliards d'euros 2014)", title = "Fig. X Produit intérieur bru
    theme(axis.text=element_text(size=6),legend.text=element_text(size=7),axis.title=element_text(size=9
  scale_x_continuous(breaks=seq(1985,2019,5))
```



```
ggplot() +
  geom_line( aes(x = Transport_France2019$Year,y = Transport_France2019$"PIB en volume (en milliards d'
labs(x = "Année", y = "PIB en volume (en milliards d'euros 2014)", title = "Fig. X Produit intérieur bru
    theme(axis.text=element_text(size=6),legend.text=element_text(size=7),axis.title=element_text(size=
  scale_x_continuous(breaks=seq(1985,2019,5))
```

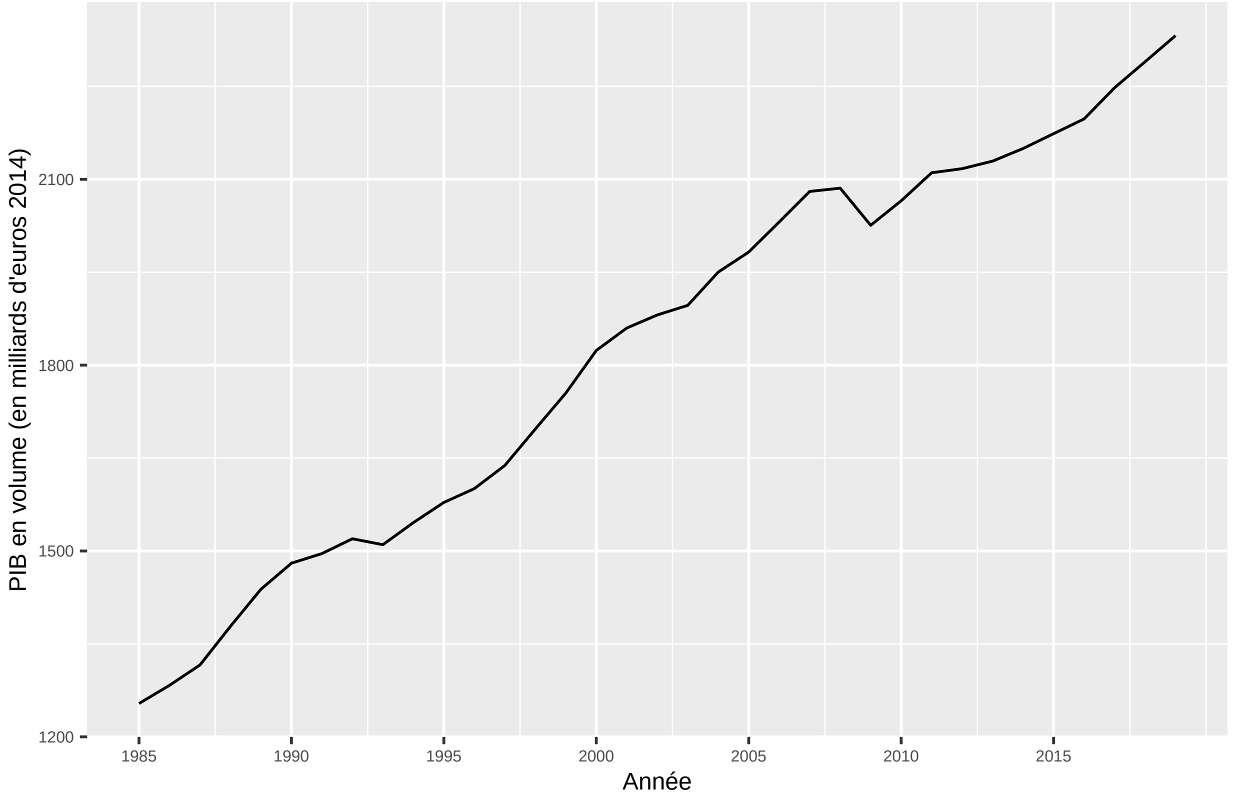
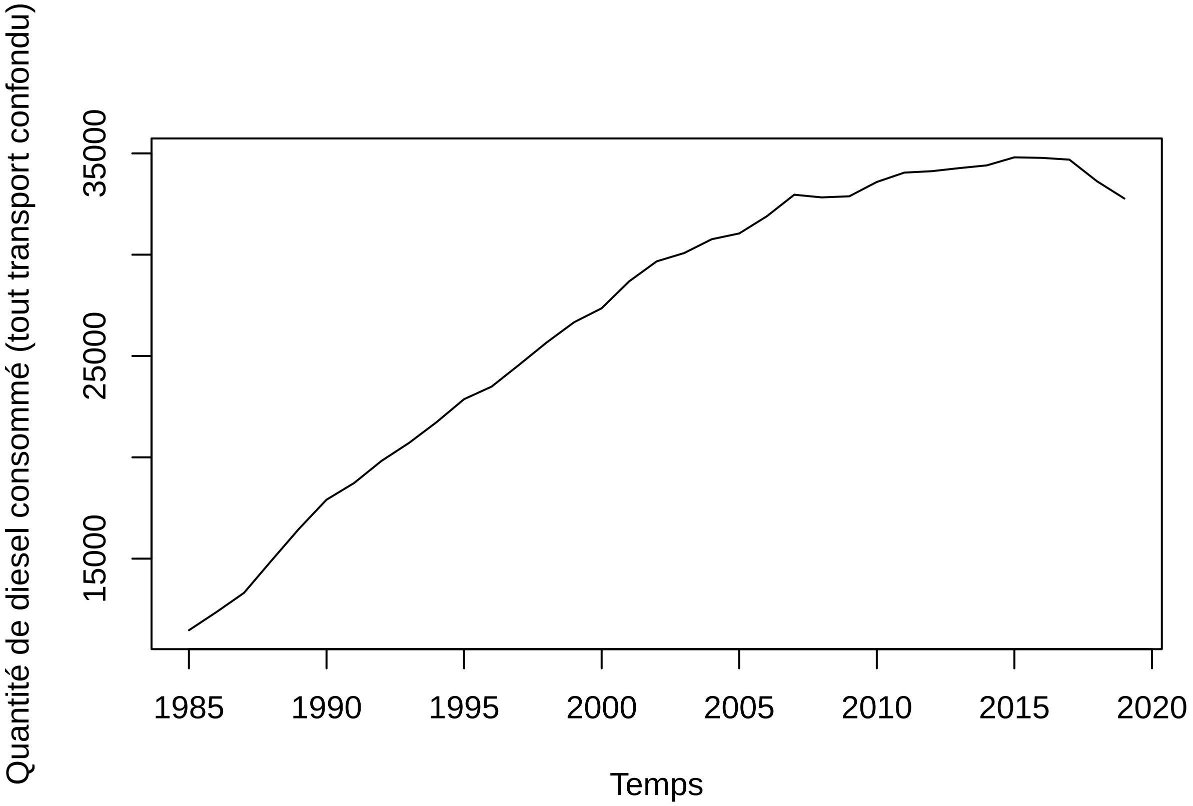
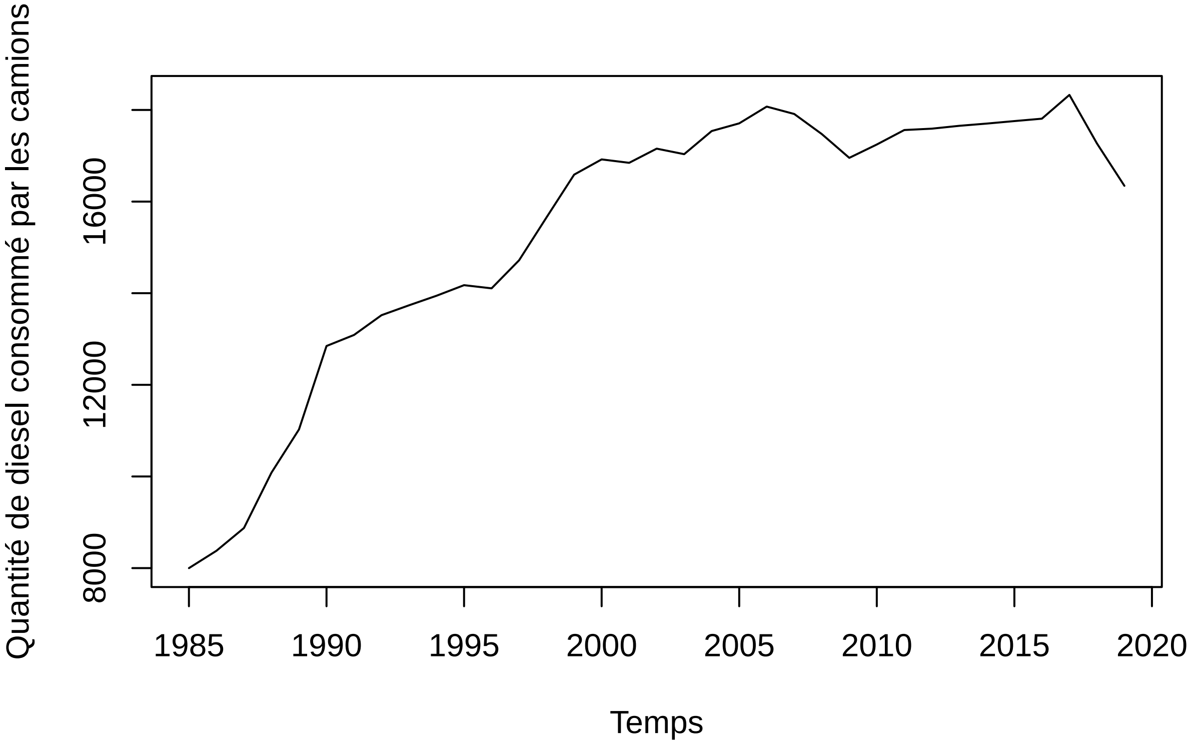


Fig. X Produit intérieur brut

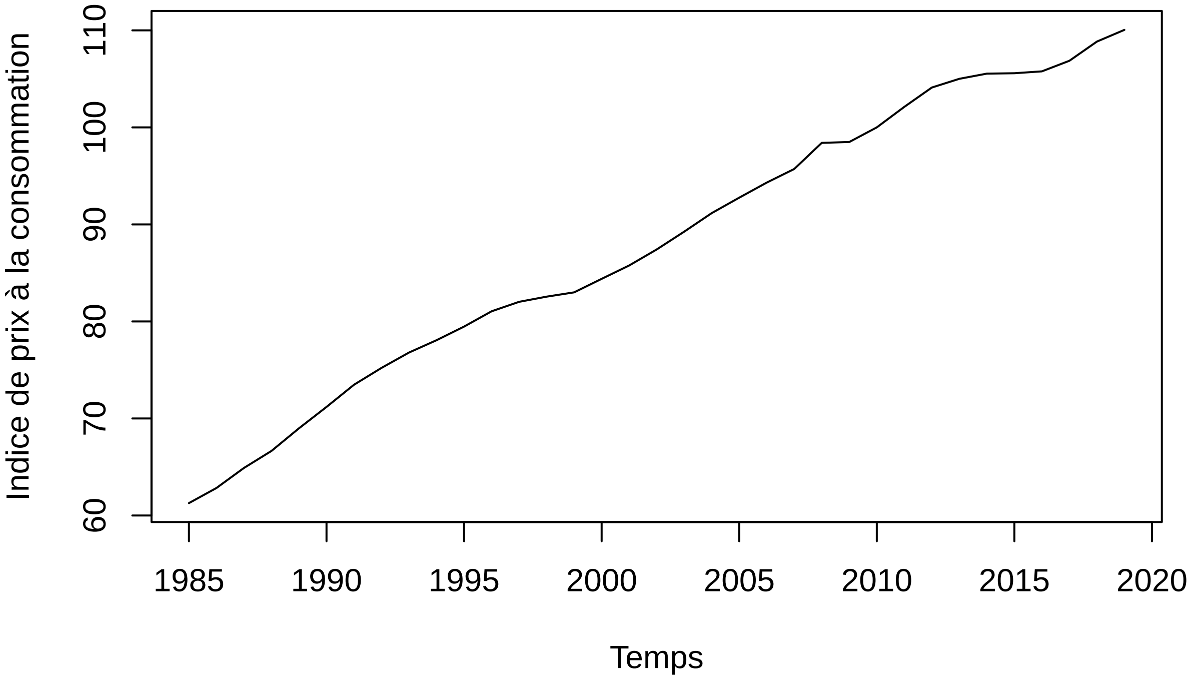
```
plot(Qdiesel.ts, xlab = "Temps", ylab = "Quantité de diesel consommé (tout transport confondu)")
```



```
plot(Qdieselcamion.ts, xlab = "Temps", ylab = "Quantité de diesel consommé par les camions")
```



```
plot(CPI.ts, xlab = "Temps", ylab = "Indice de prix à la consommation")
```



```
n=length(Transport_France2019$Qdieselcamion)

vec <- c(Transport_France2019$Pdiesel/Transport_France2019$CPI,Transport_France2019$"PIB en volume (en
X <- matrix( vec, ncol=3)
Y=matrix(Transport_France2019$Qdieselcamion,n,1)
q=ncol(Y);
k=ncol(X);
K=k+1

 y=Y
 x=X

 nobs=cbind(1:n)

 OLS=lm(formula = y ~ x)

summary(OLS) %>% pander
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1767	790.8	2.235	0.03277
x1	-378013	120267	-3.143	0.003667
x2	4.242	1.423	2.98	0.005559
x3	36.89	5.615	6.57	2.449e-07

Table 2: Fitting linear model: y ~ x

Observations	Residual Std. Error	R^2	Adjusted R^2
35	721.2	0.9475	0.9424

```
kable
```

```
## function (x, format, digits = getOption("digits"), row.names = NA,
##     col.names = NA, align, caption = NULL, label = NULL, format.args = list(),
##     escape = TRUE, ...)
## {
##     format = kable_format(format)
##     if (!missing(align) && length(align) == 1L && !grepl("[^lcr]",
##         align))
##         align = strsplit(align, "")[[1]]
##     if (inherits(x, "list")) {
##         format = kable_format_latex(format)
##         res = lapply(x, kable, format = format, digits = digits,
##             row.names = row.names, col.names = col.names, align = align,
##             caption = NA, format.args = format.args, escape = escape,
##             ...)
##         return(kables(res, format, caption, label))
##     }
##     caption = kable_caption(label, caption, format)
##     if (!is.matrix(x))
##         x = as.data.frame(x)
##     if (identical(col.names, NA))
##         col.names = colnames(x)
##     m = ncol(x)
##     isn = if (is.matrix(x))
##         rep(is.numeric(x), m)
##     else sapply(x, is.numeric)
##     if (missing(align) || (format == "latex" && is.null(align)))
##         align = ifelse(isn, "r", "l")
##     digits = rep(digits, length.out = m)
##     for (j in seq_len(m)) {
##         if (is_numeric(x[, j]))
##             x[, j] = round(x[, j], digits[j])
##     }
##     if (any(isn)) {
##         if (is.matrix(x)) {
##             if (is.table(x) && length(dim(x)) == 2)
##                 class(x) = "matrix"
##             x = format_matrix(x, format.args)
##         }
##         else x[, isn] = format_args(x[, isn], format.args)
##     }
##     if (is.na(row.names))
##         row.names = has_rownames(x)
##     if (!is.null(align))
##         align = rep(align, length.out = m)
##     if (row.names) {
##         x = cbind(` ` = rownames(x), x)
##         if (!is.null(col.names))
##             col.names = c(" ", col.names)
##         if (!is.null(align))
##             align = c("l", align)
##     }
##     n = nrow(x)
##     x = replace_na(to_character(x), is.na(x))
##     if (!is.matrix(x))
##         x = matrix(x, nrow = n)
##     x = trimws(x)
##     colnames(x) = col.names
##     if (format != "latex" && length(align) && !all(align %in%
##         c("l", "r", "c")))
##         stop("'align' must be a character vector of possible values 'l', 'r', and 'c'")
##     attr(x, "align") = align
##     if (format == "simple" && nrow(x) == 0)
##         format = "pipe"
##     res = do.call(paste("kable", format, sep = "_"), list(x = x,
##         caption = caption, escape = escape, ...))
##     structure(res, format = format, class = "knitr_kable")
## }
## <bytecode: 0x000000001fcf99f0>
## <environment: namespace:knitr>
```

```
names(OLS)
```

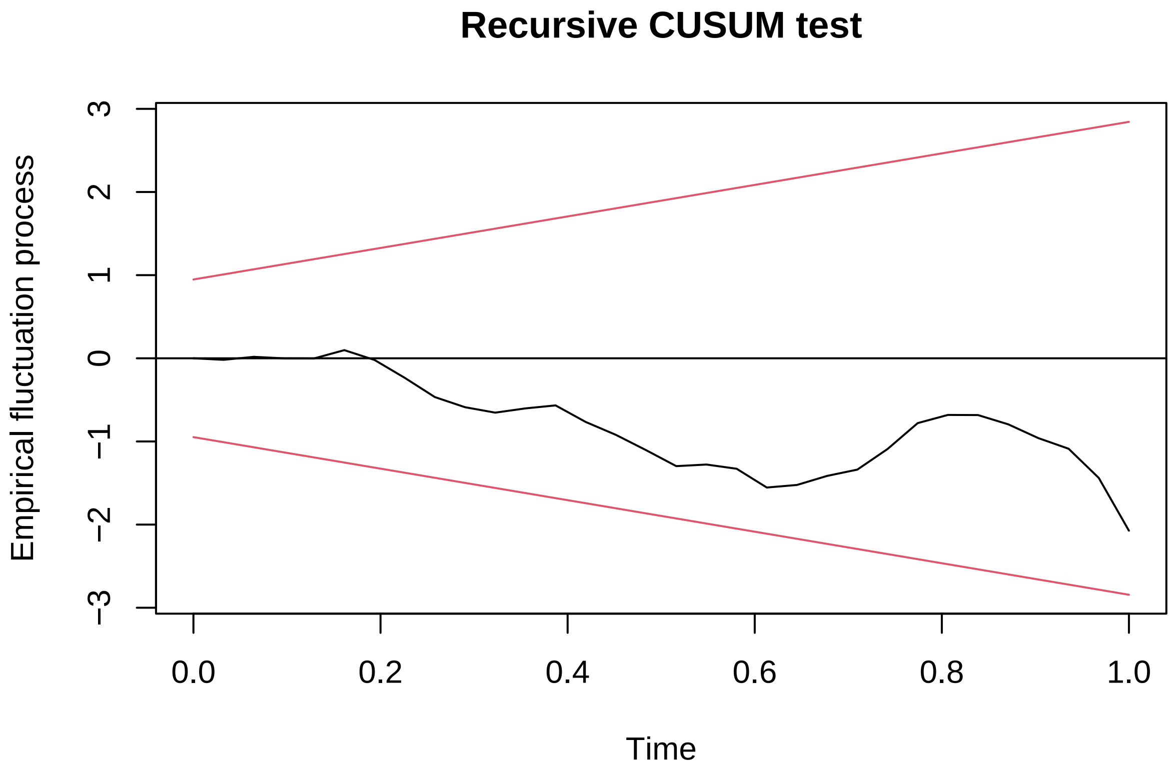
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
xc = cbind(1,x)
bhat = OLS$coefficients
yf = xc %*% bhat
res = y - yf
scr = t(res) %*% res


d1 = t(res) %*% res
d2 =  t(res[2:n]-res[1:n-1]) %*% (res[2:n]-res[1:n-1])
dw = d2/d1
print (dw)
```

```
##           [,1]
## [1,] 0.5667082
```

```
Wr <- efp(y ~ x, type = "Rec-CUSUM")
plot(Wr)
```



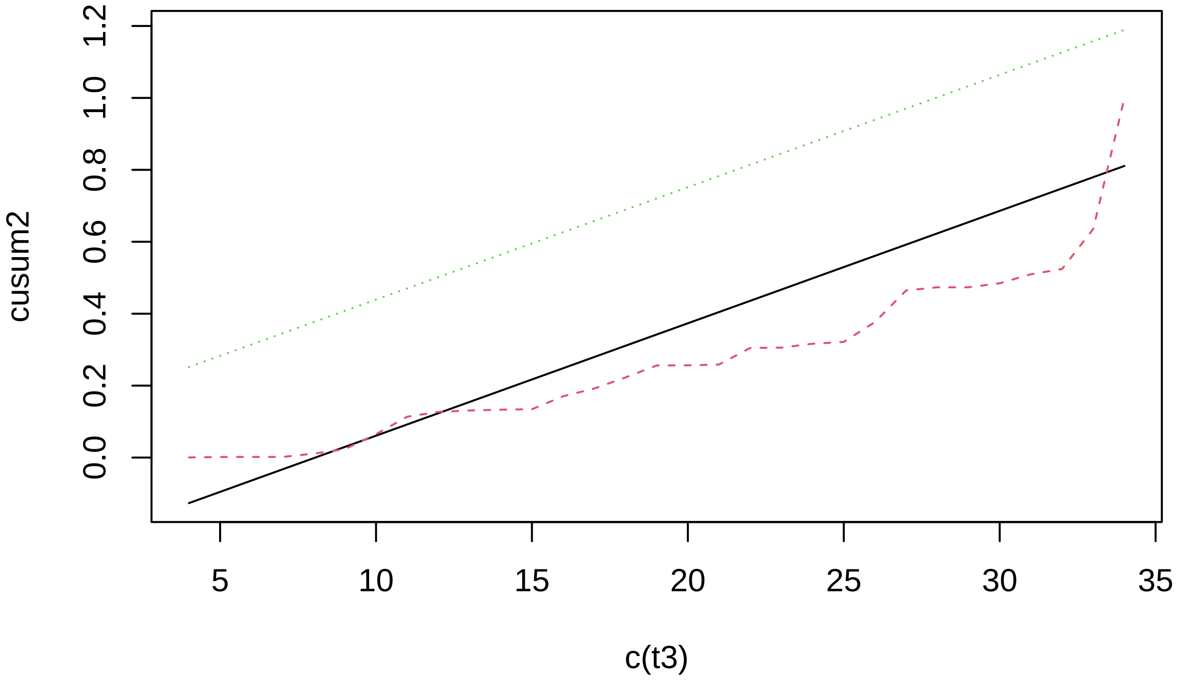
```
#
# Test Cusum Square
#
rr <- (recresid(y ~ x))
rr <- rr^2
cumrr <- cumsum(rr)/scr
```

```
## Warning in cumsum(rr)/scr: Le recyclage d'un tableau (array) de longueur 1 dans un calcul arithmétiq
tableau est obsolète.
##   Utilisez c() ou as.vector() à la place.
```

```
#
# Valeurs seuil de la distribution Cusum
#
c0 = 0.18915

kp2=K+1
c0 = 0.18915 # valeur critique de c0

t2 <- ts(kp2:n)
t3=t2-1
smin <-((t2-k)/(n-k))-c0
smax <- ((t2-k)/(n-k))+c0
#
vec2 <- c(smin, cumrr, smax)
cusum2 <- matrix(vec2, ncol = 3);
matplot(c(t3), cusum2, type ="l")
```



```
#sctest(y ~ x, type = "Chow", point =)
# pour R, la rupture est test?e non pas sur 1979-1980 mais sur 1979.
#Sur Eviews, la rupture est test?e sur 1980/
for(i in 9:30) {

print(sctest(y ~ x, type = "Chow", point = i) )
}
```

```
##
##  Chow test
##
## data:  y ~ x
## F = 4.6272, p-value = 0.005642
##
##
##  Chow test
##
## data:  y ~ x
## F = 4.6428, p-value = 0.005547
##
##
##  Chow test
##
## data:  y ~ x
## F = 4.6513, p-value = 0.005497
##
```

```
##
##  Chow test
##
## data:  y ~ x
## F = 5.796, p-value = 0.00168
##
##
##  Chow test
##
## data:  y ~ x
## F = 8.5115, p-value = 0.0001406
##
##
##  Chow test
##
## data:  y ~ x
## F = 10.764, p-value = 2.389e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 11.173, p-value = 1.772e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 11.398, p-value = 1.507e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 10.585, p-value = 2.728e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 10.003, p-value = 4.242e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 9.932, p-value = 4.48e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 8.8977, p-value = 0.0001021
##
```

```
##
##  Chow test
##
## data:  y ~ x
## F = 8.902, p-value = 0.0001017
##
##
##  Chow test
##
## data:  y ~ x
## F = 9.1596, p-value = 8.247e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 8.2241, p-value = 0.0001793
##
##
##  Chow test
##
## data:  y ~ x
## F = 8.8558, p-value = 0.0001056
##
##
##  Chow test
##
## data:  y ~ x
## F = 9.8364, p-value = 4.825e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 11.003, p-value = 2.004e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 9.5052, p-value = 6.256e-05
##
##
##  Chow test
##
## data:  y ~ x
## F = 7.0616, p-value = 0.0005013
##
##
##  Chow test
##
## data:  y ~ x
## F = 6.9945, p-value = 0.0005331
##
```

```
##
##  Chow test
##
## data:  y ~ x
## F = 7.5019, p-value = 0.0003366
```

```
n=length(Transport_France2019$Qdieselcamion)

P1 <- replicate(35, 0)
P1[1:16] <-  Transport_France2019$Pdiesel[1:16]/Transport_France2019$CPI[1:16]

P2 <- replicate(35, 0)
P2[17:35] <-  Transport_France2019$Pdiesel[17:35]/Transport_France2019$CPI[17:35]

vec <- c(P1,P2, Transport_France2019$"PIB en volume (en milliards d'euros 2014)",Transport_France2019$Q
X <- matrix( vec, ncol=4)
Y=matrix(Transport_France2019$Qdieselcamion,n,1)
q=ncol(Y);
k=ncol(X);
K=k+1

 y=Y
 x=X

 nobs=cbind(1:n)

 OLS=lm(formula = y ~ x)

 summary(OLS)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2393.64  -264.17    46.23   364.36  1468.79
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.174e+02  1.670e+03   0.070  0.94445
## x1          -2.946e+05  1.410e+05  -2.089  0.04528 *
## x2          -3.651e+05  1.203e+05  -3.034  0.00495 **
## x3           4.871e+00  1.525e+00   3.195  0.00328 **
## x4           3.735e+01  5.607e+00   6.661 2.23e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 718.3 on 30 degrees of freedom
## Multiple R-squared:  0.9496, Adjusted R-squared:  0.9429
## F-statistic: 141.4 on 4 and 30 DF,  p-value: < 2.2e-16
```

```
 names(OLS)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
xc = cbind(1,x)
bhat = OLS$coefficients
yf = xc %*% bhat
res = y - yf
scr = t(res) %*% res


d1 = t(res) %*% res
d2 =  t(res[2:n]-res[1:n-1]) %*% (res[2:n]-res[1:n-1])
dw = d2/d1
print (dw)
```

```
##             [,1]
## [1,] 0.5232029
```