Temas selectos de econometría y finanzas (modulo de matrices aleatorias)

J. Antonio García Ramirez, Tarea 6: Aplicaciones a datos financieros en el contexto de big data.

12 de noviembre, 2018

```
SMAPE <- function(y.hat, y)
{
  # Calculo de raiz cuadrada de error cuadratico medio
  return( sum( abs(y.hat-y) /(abs(y) + abs(y.hat))   )*(100/length(y)) )
}
ERROR <- function(y.hat, y)
{
  # Calculo de raiz cuadrada de error cuadratico medio
  return( mean( abs(y.hat-y) )/length(y))
}
quita.tendencia.init <- function(data, inicio , frecuencia)
{
  # CLOSURE para quitar seasonality, regresa una funcion
  #  data (vector): valores de la serie de tiempo
  # inicio (vector.longitud2): inicio de la serie de tiempo c(2008,1)
  # frecuencia (numeric): frecuencia de la serie (semanal:54)
  inicio <- inicio
  frecuencia <- frecuencia
  function(data)
  {
    s <- ts(data, start = inicio, frequency = frecuencia) # habra que harcodear estos numeros
    x <- tryCatch(seas(s)$series$s11,
                          error = function(e) data, finally = data )
    x <- as.numeric(x)
    return(x)
  }
}
adf.test.custom <- function(y, option='both')
{
  # funcion para elegir el resago optimo
  # y (numeric): vector con los datos de la serie de tiempo univariada
  # option (chacaracter) : eleccion de la tendencia e intercepto 'none','c','t','both'
  y <- ts(y)
  lag <- floor(log(length(y))) + 1 #acotamos el numero de lags por el que siguiere el
  #texto de Chan Ngai
  datos <- data.frame(y1 = diff(y))
  for (i in 2:lag) #aumentamos las columnas de lags
  {
    datos[, as.character(paste0('y',i))] <- c(diff(y, lag=i), rep(NA, i-1))
  }
  names(datos) <- c('y1', names(datos)[2:lag])
  if (option == 'none')
  {
    #aplicamos el test para cada lag
```

```
  resultado <- mapply(function(x)
  {
    formula <- paste(names(datos)[x], collapse = '+')
    formula <- as.formula(paste0('y1 ~ ', formula, '-1'))
    modelo <- lm(formula , data = datos )
    resumen <- summary(modelo)
    # nos fijamos si todos los coeficientes de la regresion
    # son significativos individualmente
    coeficientes.significativos <- resumen$coefficients[, 'Pr(>|t|)']
    coeficientes.significativos <- coeficientes.significativos <= 0.05
    if(sum(coeficientes.significativos) == 1)
    {
      big <- BIC(modelo)
      # en caso de que todos los coeficientes sean significativos regresamos
      # el BIC de la regresion
      return(big)
    } else {return(Inf)} #si un coeficiente al menos es no significativo
    #regresamos un BIC infinito
  }, 2:lag)
}

if (option == 'c')
{
  datos[, 'c'] <- rep(1, dim(datos)[1] )
  #aplicamos el test para cada lag
  resultado <- mapply(function(x)
  {
    formula <- paste(names(datos)[x], collapse = '+')
    formula <- as.formula(paste0('y1 ~ ', formula))
    modelo <- lm(formula , data = datos )
    resumen <- summary(modelo)
    # nos fijamos si todos los coeficientes de la regresion
    # son significativos individualmente
    coeficientes.significativos <- resumen$coefficients[, 'Pr(>|t|)']
    coeficientes.significativos <- coeficientes.significativos <= 0.05
    if(sum(coeficientes.significativos) == 2)
    {
      big <- BIC(modelo)
      # en caso de que todos los coeficientes sean significativos regresamos
      #el BIC de la regresion
      return(big)
    }else {return(Inf)} #si un coeficiente al menos es no significativo
    #regresamos un BIC infinito
  }, 2:lag)
}
if (option == 't')
{
  datos[, 't'] <- cumsum(1:dim(datos)[1])
  #aplicamos el test para cada lag
  resultado <- mapply(function(x)
  {
    formula <- paste(c(names(datos)[x], 't'), collapse = '+')
    formula <- as.formula(paste0('y1 ~ ', formula, '-1'))
```

```
      modelo <- lm(formula , data = datos )
      resumen <- summary(modelo)
      # nos fijamos si todos los coeficientes de la regresion
      # son significativos individualmente
      coeficientes.significativos <- resumen$coefficients[, 'Pr(>|t|)']
      coeficientes.significativos <- coeficientes.significativos <= 0.05
      if(sum(coeficientes.significativos) == 2)
      {
        big <- BIC(modelo)
        # en caso de que todos los coeficientes sean significativos regresamos
        #el BIC de la regresion
        return(big)
      }else { return(Inf)}  #si un coeficiente al menos es no significativo
      #regresamos un BIC infinito
    }, 2:lag)
  }
  if (option == 'both')
  {
    datos[, 't'] <- cumsum(1:dim(datos)[1])
    #aplicamos el test para cada lag
    resultado <- mapply(function(x)
    {
      formula <- paste(c(names(datos)[2:(x)], 't'), collapse = '+')
      formula <- as.formula(paste0('y1 ~ ', formula))
      modelo <- lm(formula , data = datos )
      resumen <- summary(modelo)
      # nos fijamos si todos los coeficientes de la regresion
      # son significativos individualmente
      coeficientes.significativos <- resumen$coefficients[, 'Pr(>|t|)']
      coeficientes.significativos <- coeficientes.significativos <= 0.05
      if(sum(coeficientes.significativos) == 3)
      {
        big <- BIC(modelo)
        # en caso de que todos los coeficientes sean significativos regresamos
        #el BIC de la regresion
        return(big)
      } else { return(Inf)}  #si un coeficiente al menos es no significativo
      #regresamos un BIC infinito
    }, 2:lag)
  }
  parsimonia <- which.min(resultado)
  names(parsimonia) <- 'Lag optimo'
  return(parsimonia)
}
```

Ejercicio 1

En este ejercicio se busca integrar los conocimientos aprendidos a lo largo del curso. Para ello se solicita realizar lo siguiente:

a. *Completar la derivación de la distribución de Marcenko-Pastur, partiendo de las notas de clase. Sea los más claro posible, sin omitir ningún detalle algebraico (puede escanearlo.)*

b. *Reproduzca la figura 14.1 del libro seguido en este módulo (**Introduction to Random Matrices**, G. Livan et. al.), bajo las mismas condiciones y parámetros (compruebe que $p > 0.05$ en el test de Kolmogorov-Smirnov).*

c. *Descargue las series de tiempo que componen el índice bursátil Standard & Poor's 500. Utilizando una periodicidad semanal durante los últimos 10 años (Enero 2008 a la fecha).*

Descargamos los datos en tiempo real y gráficamos algunas de las series. Utilizamos los tickers de las empresas que contribuyen al S&P500 previamente de https://mx.investing.com/indices/investing.com-us-500-components.

```
diferencia <- today()- ymd('2008-01-01')
fecha <- as.numeric(diferencia)
today()-days(fecha) #checar fecha de inicio
```

```
## [1] "2008-01-01"
```

```
first.date <- Sys.Date() - fecha #actualización en tiempo real
last.date <- Sys.Date()
freq.data <- 'weekly'  # frecuencia semanal
```

```
# lectura de tickerts
Componentes_Investing_com_United_States_500 <- read_csv("Componentes Investing.com United States 500.csv
       #https://mx.investing.com/indices/investing.com-us-500-components los nombres de las empresas
tickers <- Componentes_Investing_com_United_States_500$Símbolo
companias <- BatchGetSymbols(tickers = tickers,
                         first.date = first.date,
                         last.date = last.date,
                         freq.data = freq.data,
                         do.complete.data = FALSE) #sihay nulos los descartamos
```

```
##
## Running BatchGetSymbols for:
##    tickers = MMM, ABT, ABBV, ACN, ATVI, AYI, ADBE, ADP, AAP, AET, AMG, AFL, A, AIG, APD, AKAM, ALK, A
##    Downloading data for benchmark ticker | Not Cached
## MMM | yahoo (1|496) | Not Cached - OK!
## ABT | yahoo (2|496) | Not Cached - OK!
## ABBV | yahoo (3|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## ACN | yahoo (4|496) | Not Cached - Got it!
## ATVI | yahoo (5|496) | Not Cached - Got it!
## AYI | yahoo (6|496) | Not Cached - You got it!
## ADBE | yahoo (7|496) | Not Cached - Got it!
## ADP | yahoo (8|496) | Not Cached - Got it!
## AAP | yahoo (9|496) | Not Cached - Looking good!
## AET | yahoo (10|496) | Not Cached - Boa!
## AMG | yahoo (11|496) | Not Cached - Got it!
## AFL | yahoo (12|496) | Not Cached - Got it!
## A | yahoo (13|496) | Not Cached - Feels good!
## AIG | yahoo (14|496) | Not Cached - OK!
## APD | yahoo (15|496) | Not Cached - Feels good!
## AKAM | yahoo (16|496) | Not Cached - Looking good!
## ALK | yahoo (17|496) | Not Cached - OK!
## ALB | yahoo (18|496) | Not Cached - Good job!
## ARE | yahoo (19|496) | Not Cached - Got it!
## ALXN | yahoo (20|496) | Not Cached - Youre doing good!
## ALGN | yahoo (21|496) | Not Cached - Good job!
## ALLE | yahoo (22|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
```

```
## AGN | yahoo (23|496) | Not Cached - Looking good!
## ADS | yahoo (24|496) | Not Cached - Good job!
## LNT | yahoo (25|496) | Not Cached - Well done!
## ALL | yahoo (26|496) | Not Cached - Got it!
## GOOGL | yahoo (27|496) | Not Cached - Good job!
## GOOG | yahoo (28|496) | Not Cached - Looking good!
## MO | yahoo (29|496) | Not Cached - Looking good!
## AMZN | yahoo (30|496) | Not Cached - Looking good!
## AMD | yahoo (31|496) | Not Cached - Good job!
## AEE | yahoo (32|496) | Not Cached - Well done!
## AAL | yahoo (33|496) | Not Cached - Well done!
## AEP | yahoo (34|496) | Not Cached - Well done!
## AXP | yahoo (35|496) | Not Cached - Mas bah tche, que coisa linda!
## AMT | yahoo (36|496) | Not Cached - Feels good!
## AWK | yahoo (37|496) | Not Cached - Got it!
## AMP | yahoo (38|496) | Not Cached - Good stuff!
## ABC | yahoo (39|496) | Not Cached - Got it!
## AME | yahoo (40|496) | Not Cached - Got it!
## AMGN | yahoo (41|496) | Not Cached - Nice!
## APH | yahoo (42|496) | Not Cached - OK!
## APC | yahoo (43|496) | Not Cached - Got it!
## ADI | yahoo (44|496) | Not Cached - You got it!
## ANSS | yahoo (45|496) | Not Cached - Looking good!
## ANTM | yahoo (46|496) | Not Cached - Well done!
## AOS | yahoo (47|496) | Not Cached - Feels good!
## AON | yahoo (48|496) | Not Cached - You got it!
## APA | yahoo (49|496) | Not Cached - Well done!
## AIV | yahoo (50|496) | Not Cached - Got it!
## AAPL | yahoo (51|496) | Not Cached - Well done!
## AMAT | yahoo (52|496) | Not Cached - Youre doing good!
## APTV | yahoo (53|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## ADM | yahoo (54|496) | Not Cached - You got it!
## ARNC | yahoo (55|496) | Not Cached - Mais contente que cusco de cozinheira!
## AJG | yahoo (56|496) | Not Cached - Nice!
## AIZ | yahoo (57|496) | Not Cached - Well done!
## T | yahoo (58|496) | Not Cached - Looking good!
## ADSK | yahoo (59|496) | Not Cached - Looking good!
## AZO | yahoo (60|496) | Not Cached - You got it!
## AVB | yahoo (61|496) | Not Cached - Looking good!
## AVY | yahoo (62|496) | Not Cached - You got it!
## BHGE | yahoo (63|496) | Not Cached - Youre doing good!
## BLL | yahoo (64|496) | Not Cached - Youre doing good!
## BAC | yahoo (65|496) | Not Cached - Good stuff!
## BK | yahoo (66|496) | Not Cached - Nice!
## BAX | yahoo (67|496) | Not Cached - OK!
## BBT | yahoo (68|496) | Not Cached - Well done!
## BDX | yahoo (69|496) | Not Cached - Feels good!
## BRKb | yahoo (70|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## BBY | yahoo (71|496) | Not Cached - Got it!
## BIIB | yahoo (72|496) | Not Cached - OK!
## BLK | yahoo (73|496) | Not Cached - Feels good!
## BA | yahoo (74|496) | Not Cached - Well done!
## BKNG | yahoo (75|496) | Not Cached - Boa!
## BWA | yahoo (76|496) | Not Cached - Feels good!
```

```
## BXP | yahoo (77|496) | Not Cached - You got it!
## BSX | yahoo (78|496) | Not Cached - Looking good!
## BHF | yahoo (79|496) | Not Cached - Got it!
## BMY | yahoo (80|496) | Not Cached - Youre doing good!
## AVGO | yahoo (81|496) | Not Cached - Nice!
## BPR | yahoo (82|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## BFb | yahoo (83|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## CA | yahoo (84|496) | Not Cached - Nice!
## COG | yahoo (85|496) | Not Cached - Nice!
## CDNS | yahoo (86|496) | Not Cached - Got it!
## CPB | yahoo (87|496) | Not Cached - Good stuff!
## COF | yahoo (88|496) | Not Cached - Good stuff!
## CAH | yahoo (89|496) | Not Cached - Good job!
## KMX | yahoo (90|496) | Not Cached - Nice!
## CCL | yahoo (91|496) | Not Cached - Good stuff!
## CAT | yahoo (92|496) | Not Cached - Youre doing good!
## CBOE | yahoo (93|496) | Not Cached - Feels good!
## CBRE | yahoo (94|496) | Not Cached - Youre doing good!
## CBS | yahoo (95|496) | Not Cached - Feels good!
## CELG | yahoo (96|496) | Not Cached - OK!
## CNC | yahoo (97|496) | Not Cached - Feels good!
## CNP | yahoo (98|496) | Not Cached - You got it!
## CTL | yahoo (99|496) | Not Cached - Got it!
## CERN | yahoo (100|496) | Not Cached - Good stuff!
## CF | yahoo (101|496) | Not Cached - Good job!
## CHRW | yahoo (102|496) | Not Cached - You got it!
## CHTR | yahoo (103|496) | Not Cached - Youre doing good!
## CHK | yahoo (104|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## CVX | yahoo (105|496) | Not Cached - Mas bah tche, que coisa linda!
## CMG | yahoo (106|496) | Not Cached - OK!
## CB | yahoo (107|496) | Not Cached - Nice!
## CHD | yahoo (108|496) | Not Cached - Good stuff!
## CI | yahoo (109|496) | Not Cached - Looking good!
## XEC | yahoo (110|496) | Not Cached - OK!
## CINF | yahoo (111|496) | Not Cached - Youre doing good!
## CTAS | yahoo (112|496) | Not Cached - Well done!
## CSCO | yahoo (113|496) | Not Cached - Well done!
## C | yahoo (114|496) | Not Cached - Feels good!
## CFG | yahoo (115|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## CTXS | yahoo (116|496) | Not Cached - Good stuff!
## CLX | yahoo (117|496) | Not Cached - OK!
## CME | yahoo (118|496) | Not Cached - Good job!
## CMS | yahoo (119|496) | Not Cached - Looking good!
## KO | yahoo (120|496) | Not Cached - Well done!
## CTSH | yahoo (121|496) | Not Cached - OK!
## CL | yahoo (122|496) | Not Cached - Well done!
## CMCSA | yahoo (123|496) | Not Cached - Looking good!
## CMA | yahoo (124|496) | Not Cached - Well done!
## CAG | yahoo (125|496) | Not Cached - Youre doing good!
## CXO | yahoo (126|496) | Not Cached - Looking good!
## COP | yahoo (127|496) | Not Cached - You got it!
## ED | yahoo (128|496) | Not Cached - Well done!
## STZ | yahoo (129|496) | Not Cached - OK!
## COO | yahoo (130|496) | Not Cached - Good stuff!
```

```
## GLW | yahoo (131|496) | Not Cached - OK!
## COST | yahoo (132|496) | Not Cached - You got it!
## COTY | yahoo (133|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## CCI | yahoo (134|496) | Not Cached - OK!
## CSX | yahoo (135|496) | Not Cached - Feels good!
## CMI | yahoo (136|496) | Not Cached - Feels good!
## CVS | yahoo (137|496) | Not Cached - Feliz que nem lambari de sanga!
## DHR | yahoo (138|496) | Not Cached - Well done!
## DRI | yahoo (139|496) | Not Cached - Looking good!
## DVA | yahoo (140|496) | Not Cached - Looking good!
## DE | yahoo (141|496) | Not Cached - Good job!
## DAL | yahoo (142|496) | Not Cached - Looking good!
## XRAY | yahoo (143|496) | Not Cached - Good stuff!
## DVN | yahoo (144|496) | Not Cached - Youre doing good!
## DLR | yahoo (145|496) | Not Cached - Looking good!
## DFS | yahoo (146|496) | Not Cached - Youre doing good!
## DISCA | yahoo (147|496) | Not Cached - You got it!
## DISCK | yahoo (148|496) | Not Cached - OK!
## DISH | yahoo (149|496) | Not Cached - Youre doing good!
## DG | yahoo (150|496) | Not Cached - Got it!
## DLTR | yahoo (151|496) | Not Cached - Looking good!
## D | yahoo (152|496) | Not Cached - Got it!
## DOV | yahoo (153|496) | Not Cached - Well done!
## DHI | yahoo (154|496) | Not Cached - Looking good!
## DTE | yahoo (155|496) | Not Cached - You got it!
## DRE | yahoo (156|496) | Not Cached - Nice!
## DUK | yahoo (157|496) | Not Cached - Youre doing good!
## DWDP | yahoo (158|496) | Not Cached - Looking good!
## DXC | yahoo (159|496) | Not Cached - Feels good!
## ETFC | yahoo (160|496) | Not Cached - Feels good!
## EMN | yahoo (161|496) | Not Cached - Good job!
## ETN | yahoo (162|496) | Not Cached - You got it!
## EBAY | yahoo (163|496) | Not Cached - Good stuff!
## ECL | yahoo (164|496) | Not Cached - Got it!
## EIX | yahoo (165|496) | Not Cached - Nice!
## EW | yahoo (166|496) | Not Cached - Good stuff!
## EA | yahoo (167|496) | Not Cached - OK!
## LLY | yahoo (168|496) | Not Cached - Got it!
## EMR | yahoo (169|496) | Not Cached - OK!
## ETR | yahoo (170|496) | Not Cached - Got it!
## EOG | yahoo (171|496) | Not Cached - Looking good!
## EQT | yahoo (172|496) | Not Cached - Feels good!
## EFX | yahoo (173|496) | Not Cached - Good job!
## EQIX | yahoo (174|496) | Not Cached - OK!
## EQR | yahoo (175|496) | Not Cached - Youre doing good!
## ESS | yahoo (176|496) | Not Cached - Youre doing good!
## EL | yahoo (177|496) | Not Cached - Well done!
## RE | yahoo (178|496) | Not Cached - Well done!
## ES | yahoo (179|496) | Not Cached - Good job!
## EXC | yahoo (180|496) | Not Cached - Good stuff!
## EXPE | yahoo (181|496) | Not Cached - Good job!
## EXPD | yahoo (182|496) | Not Cached - Nice!
## ESRX | yahoo (183|496) | Not Cached - Good job!
## EXR | yahoo (184|496) | Not Cached - Good stuff!
```

```
## XOM | yahoo (185|496) | Not Cached - Good job!
## FFIV | yahoo (186|496) | Not Cached - Well done!
## FB | yahoo (187|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## FAST | yahoo (188|496) | Not Cached - OK!
## FRT | yahoo (189|496) | Not Cached - Youre doing good!
## FDX | yahoo (190|496) | Not Cached - OK!
## FIS | yahoo (191|496) | Not Cached - Youre doing good!
## FITB | yahoo (192|496) | Not Cached - Looking good!
## FE | yahoo (193|496) | Not Cached - Youre doing good!
## FISV | yahoo (194|496) | Not Cached - Mas bah tche, que coisa linda!
## FLIR | yahoo (195|496) | Not Cached - Mais faceiro que guri de bombacha nova!
## FLS | yahoo (196|496) | Not Cached - Nice!
## FLR | yahoo (197|496) | Not Cached - Good stuff!
## FMC | yahoo (198|496) | Not Cached - Feels good!
## FL | yahoo (199|496) | Not Cached - You got it!
## F | yahoo (200|496) | Not Cached - Nice!
## FTV | yahoo (201|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## FBHS | yahoo (202|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## FOX | yahoo (203|496) | Not Cached - You got it!
## BEN | yahoo (204|496) | Not Cached - Youre doing good!
## FCX | yahoo (205|496) | Not Cached - Got it!
## GPS | yahoo (206|496) | Not Cached - Feels good!
## GRMN | yahoo (207|496) | Not Cached - Got it!
## IT | yahoo (208|496) | Not Cached - Feels good!
## GD | yahoo (209|496) | Not Cached - Good job!
## GE | yahoo (210|496) | Not Cached - Good job!
## GIS | yahoo (211|496) | Not Cached - Good stuff!
## GM | yahoo (212|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## GPC | yahoo (213|496) | Not Cached - Good stuff!
## GILD | yahoo (214|496) | Not Cached - Youre doing good!
## GPN | yahoo (215|496) | Not Cached - Well done!
## GS | yahoo (216|496) | Not Cached - Well done!
## HRB | yahoo (217|496) | Not Cached - Good job!
## HAL | yahoo (218|496) | Not Cached - Good job!
## HBI | yahoo (219|496) | Not Cached - OK!
## HOG | yahoo (220|496) | Not Cached - OK!
## HRS | yahoo (221|496) | Not Cached - Well done!
## HIG | yahoo (222|496) | Not Cached - Good job!
## HAS | yahoo (223|496) | Not Cached - Nice!
## HCA | yahoo (224|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## HCP | yahoo (225|496) | Not Cached - Good stuff!
## HP | yahoo (226|496) | Not Cached - Got it!
## HSIC | yahoo (227|496) | Not Cached - You got it!
## HSY | yahoo (228|496) | Not Cached - Looking good!
## HES | yahoo (229|496) | Not Cached - Looking good!
## HPE | yahoo (230|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## HLT | yahoo (231|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## HOLX | yahoo (232|496) | Not Cached - Feels good!
## HD | yahoo (233|496) | Not Cached - Good stuff!
## HON | yahoo (234|496) | Not Cached - Boa!
## HRL | yahoo (235|496) | Not Cached - Well done!
## HST | yahoo (236|496) | Not Cached - Feels good!
## HPQ | yahoo (237|496) | Not Cached - Well done!
## HUM | yahoo (238|496) | Not Cached - Well done!
```

```
## HBAN | yahoo (239|496) | Not Cached - Feels good!
## IBM | yahoo (240|496) | Not Cached - Good job!
## ICE | yahoo (241|496) | Not Cached - Nice!
## IDXX | yahoo (242|496) | Not Cached - Good job!
## IFF | yahoo (243|496) | Not Cached - Youre doing good!
## INFO | yahoo (244|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## ITW | yahoo (245|496) | Not Cached - Well done!
## ILMN | yahoo (246|496) | Not Cached - Mas bah tche, que coisa linda!
## INCY | yahoo (247|496) | Not Cached - Nice!
## IR | yahoo (248|496) | Not Cached - Good stuff!
## INTC | yahoo (249|496) | Not Cached - Looking good!
## IP | yahoo (250|496) | Not Cached - You got it!
## INTU | yahoo (251|496) | Not Cached - Youre doing good!
## ISRG | yahoo (252|496) | Not Cached - You got it!
## IVZ | yahoo (253|496) | Not Cached - Well done!
## IPG | yahoo (254|496) | Not Cached - Got it!
## IPGP | yahoo (255|496) | Not Cached - Well done!
## IRM | yahoo (256|496) | Not Cached - Well done!
## JNJ | yahoo (257|496) | Not Cached - Got it!
## JEC | yahoo (258|496) | Not Cached - Looking good!
## JBHT | yahoo (259|496) | Not Cached - OK!
## JEF | yahoo (260|496) | Not Cached - Good job!
## SJM | yahoo (261|496) | Not Cached - Good job!
## JCI | yahoo (262|496) | Not Cached - Good stuff!
## JPM | yahoo (263|496) | Not Cached - Looking good!
## JNPR | yahoo (264|496) | Not Cached - Feels good!
## KSU | yahoo (265|496) | Not Cached - OK!
## K | yahoo (266|496) | Not Cached - Looking good!
## KDP | yahoo (267|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## KEY | yahoo (268|496) | Not Cached - Looking good!
## KMB | yahoo (269|496) | Not Cached - Got it!
## KIM | yahoo (270|496) | Not Cached - Nice!
## KMI | yahoo (271|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## KR51 | yahoo (272|496) | Not Cached - Error in download..
## KLAC | yahoo (273|496) | Not Cached - Got it!
## KSS | yahoo (274|496) | Not Cached - Feels good!
## KHC | yahoo (275|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## KR | yahoo (276|496) | Not Cached - Well done!
## LB | yahoo (277|496) | Not Cached - OK!
## LLL | yahoo (278|496) | Not Cached - You got it!
## LH | yahoo (279|496) | Not Cached - Boa!
## LRCX | yahoo (280|496) | Not Cached - Nice!
## LEG | yahoo (281|496) | Not Cached - Nice!
## LEN | yahoo (282|496) | Not Cached - Good job!
## LNC | yahoo (283|496) | Not Cached - Feels good!
## LKQ | yahoo (284|496) | Not Cached - Good job!
## LMT | yahoo (285|496) | Not Cached - Feels good!
## L | yahoo (286|496) | Not Cached - Well done!
## LOW | yahoo (287|496) | Not Cached - Well done!
## LYB | yahoo (288|496) | Not Cached - You got it!
## MTB | yahoo (289|496) | Not Cached - OK!
## MAC | yahoo (290|496) | Not Cached - Good stuff!
## M | yahoo (291|496) | Not Cached - Good job!
## MRO | yahoo (292|496) | Not Cached - Well done!
```

```
## MPC | yahoo (293|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## MAR | yahoo (294|496) | Not Cached - Nice!
## MMC | yahoo (295|496) | Not Cached - Youre doing good!
## MLM | yahoo (296|496) | Not Cached - Well done!
## MAS | yahoo (297|496) | Not Cached - OK!
## MA | yahoo (298|496) | Not Cached - Well done!
## MAT | yahoo (299|496) | Not Cached - Good stuff!
## MKC | yahoo (300|496) | Not Cached - Well done!
## MCD | yahoo (301|496) | Not Cached - Nice!
## MCK | yahoo (302|496) | Not Cached - Nice!
## MDT | yahoo (303|496) | Not Cached - Good job!
## MRK | yahoo (304|496) | Not Cached - Good stuff!
## MET | yahoo (305|496) | Not Cached - Well done!
## MTD | yahoo (306|496) | Not Cached - Feliz que nem lambari de sanga!
## MGM | yahoo (307|496) | Not Cached - Good job!
## KORS | yahoo (308|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## MCHP | yahoo (309|496) | Not Cached - Good stuff!
## MU | yahoo (310|496) | Not Cached - Looking good!
## MSFT | yahoo (311|496) | Not Cached - Feels good!
## MAA | yahoo (312|496) | Not Cached - Got it!
## MHK | yahoo (313|496) | Not Cached - Good stuff!
## TAP | yahoo (314|496) | Not Cached - Well done!
## MDLZ | yahoo (315|496) | Not Cached - Feels good!
## MNST | yahoo (316|496) | Not Cached - Looking good!
## MCO | yahoo (317|496) | Not Cached - Good job!
## MS | yahoo (318|496) | Not Cached - OK!
## MOS | yahoo (319|496) | Not Cached - Well done!
## MSI | yahoo (320|496) | Not Cached - Good stuff!
## MYL | yahoo (321|496) | Not Cached - Mais faceiro que guri de bombacha nova!
## NDAQ | yahoo (322|496) | Not Cached - OK!
## NOV | yahoo (323|496) | Not Cached - Well done!
## NAVI | yahoo (324|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## NTAP | yahoo (325|496) | Not Cached - Good job!
## NFLX | yahoo (326|496) | Not Cached - You got it!
## NWL | yahoo (327|496) | Not Cached - Well done!
## NFX | yahoo (328|496) | Not Cached - Got it!
## NEM | yahoo (329|496) | Not Cached - Got it!
## NWS | yahoo (330|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## NWSA | yahoo (331|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## NEE | yahoo (332|496) | Not Cached - Feels good!
## NLSN | yahoo (333|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## NKE | yahoo (334|496) | Not Cached - Feels good!
## NI | yahoo (335|496) | Not Cached - Well done!
## NBL | yahoo (336|496) | Not Cached - Feels good!
## JWN | yahoo (337|496) | Not Cached - Got it!
## NSC | yahoo (338|496) | Not Cached - Well done!
## NTRS | yahoo (339|496) | Not Cached - Well done!
## NOC | yahoo (340|496) | Not Cached - Feels good!
## NCLH | yahoo (341|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## NRG | yahoo (342|496) | Not Cached - Mais contente que cusco de cozinheira!
## NUE | yahoo (343|496) | Not Cached - Good stuff!
## NVDA | yahoo (344|496) | Not Cached - Looking good!
## ORLY | yahoo (345|496) | Not Cached - You got it!
## OXY | yahoo (346|496) | Not Cached - Looking good!
```

```
## OMC | yahoo (347|496) | Not Cached - OK!
## OKE | yahoo (348|496) | Not Cached - Looking good!
## ORCL | yahoo (349|496) | Not Cached - Well done!
## PCAR | yahoo (350|496) | Not Cached - Feels good!
## PCG | yahoo (351|496) | Not Cached - You got it!
## PKG | yahoo (352|496) | Not Cached - Got it!
## PH | yahoo (353|496) | Not Cached - Good stuff!
## PDCO | yahoo (354|496) | Not Cached - Nice!
## PAYX | yahoo (355|496) | Not Cached - Good stuff!
## PYPL | yahoo (356|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## PNR | yahoo (357|496) | Not Cached - Good job!
## PBCT | yahoo (358|496) | Not Cached - Got it!
## PEP | yahoo (359|496) | Not Cached - Well done!
## PKI | yahoo (360|496) | Not Cached - Got it!
## PRGO | yahoo (361|496) | Not Cached - Good job!
## PFE | yahoo (362|496) | Not Cached - Youre doing good!
## PSX | yahoo (363|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## PNW | yahoo (364|496) | Not Cached - Nice!
## PXD | yahoo (365|496) | Not Cached - OK!
## PNC | yahoo (366|496) | Not Cached - You got it!
## PPG | yahoo (367|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## PPL | yahoo (368|496) | Not Cached - You got it!
## PFG | yahoo (369|496) | Not Cached - Well done!
## PG | yahoo (370|496) | Not Cached - Looking good!
## PGR | yahoo (371|496) | Not Cached - Feels good!
## PLD | yahoo (372|496) | Not Cached - Looking good!
## PRU | yahoo (373|496) | Not Cached - Youre doing good!
## PEG | yahoo (374|496) | Not Cached - Good job!
## PSA | yahoo (375|496) | Not Cached - Nice!
## PHM | yahoo (376|496) | Not Cached - OK!
## PVH | yahoo (377|496) | Not Cached - Looking good!
## QRVO | yahoo (378|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## QCOM | yahoo (379|496) | Not Cached - OK!
## PWR | yahoo (380|496) | Not Cached - You got it!
## DGX | yahoo (381|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## RL | yahoo (382|496) | Not Cached - Nice!
## RRC | yahoo (383|496) | Not Cached - Youre doing good!
## RJF | yahoo (384|496) | Not Cached - Youre doing good!
## RTN | yahoo (385|496) | Not Cached - Feels good!
## O | yahoo (386|496) | Not Cached - Youre doing good!
## RHT | yahoo (387|496) | Not Cached - Good job!
## REG | yahoo (388|496) | Not Cached - Good stuff!
## REGN | yahoo (389|496) | Not Cached - Nice!
## RF | yahoo (390|496) | Not Cached - Good stuff!
## RSG | yahoo (391|496) | Not Cached - Feels good!
## RMD | yahoo (392|496) | Not Cached - Boa!
## RHI | yahoo (393|496) | Not Cached - OK!
## ROK | yahoo (394|496) | Not Cached - You got it!
## COL | yahoo (395|496) | Not Cached - Nice!
## ROP | yahoo (396|496) | Not Cached - Got it!
## ROST | yahoo (397|496) | Not Cached - Mais contente que cusco de cozinheira!
## RCL | yahoo (398|496) | Not Cached - Feels good!
## SPGI | yahoo (399|496) | Not Cached - Nice!
## CRM | yahoo (400|496) | Not Cached - Got it!
```

```
## SBAC | yahoo (401|496) | Not Cached - Youre doing good!
## SCG | yahoo (402|496) | Not Cached - Good job!
## SLB | yahoo (403|496) | Not Cached - Got it!
## STX | yahoo (404|496) | Not Cached - Looking good!
## SEE | yahoo (405|496) | Not Cached - You got it!
## SRE | yahoo (406|496) | Not Cached - Feels good!
## SHW | yahoo (407|496) | Not Cached - Good stuff!
## SIG | yahoo (408|496) | Not Cached - Good job!
## SPG | yahoo (409|496) | Not Cached - Looking good!
## SWKS | yahoo (410|496) | Not Cached - You got it!
## SLG | yahoo (411|496) | Not Cached - Feels good!
## SNA | yahoo (412|496) | Not Cached - Youre doing good!
## SO | yahoo (413|496) | Not Cached - Youre doing good!
## LUV | yahoo (414|496) | Not Cached - Got it!
## SWK | yahoo (415|496) | Not Cached - Mas bah tche, que coisa linda!
## SBUX | yahoo (416|496) | Not Cached - Feels good!
## STT | yahoo (417|496) | Not Cached - Got it!
## SRCL | yahoo (418|496) | Not Cached - Got it!
## SYK | yahoo (419|496) | Not Cached - Feels good!
## STI | yahoo (420|496) | Not Cached - Good job!
## SYMC | yahoo (421|496) | Not Cached - Feels good!
## SYF | yahoo (422|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## SNPS | yahoo (423|496) | Not Cached - OK!
## SYY | yahoo (424|496) | Not Cached - Nice!
## TROW | yahoo (425|496) | Not Cached - Looking good!
## TPR | yahoo (426|496) | Not Cached - Youre doing good!
## TGT | yahoo (427|496) | Not Cached - Good stuff!
## TEL | yahoo (428|496) | Not Cached - OK!
## FTI | yahoo (429|496) | Not Cached - Looking good!
## TXN | yahoo (430|496) | Not Cached - Good job!
## TXT | yahoo (431|496) | Not Cached - Feels good!
## AES | yahoo (432|496) | Not Cached - Good stuff!
## SCHW | yahoo (433|496) | Not Cached - Good job!
## GT | yahoo (434|496) | Not Cached - Well done!
## TRV | yahoo (435|496) | Not Cached - Looking good!
## TMO | yahoo (436|496) | Not Cached - You got it!
## TIF | yahoo (437|496) | Not Cached - Youre doing good!
## TJX | yahoo (438|496) | Not Cached - Youre doing good!
## TMK | yahoo (439|496) | Not Cached - Well done!
## TSS | yahoo (440|496) | Not Cached - Well done!
## TSCO | yahoo (441|496) | Not Cached - Got it!
## TDG | yahoo (442|496) | Not Cached - You got it!
## TRIP | yahoo (443|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## FOXA | yahoo (444|496) | Not Cached - Mas bah tche, que coisa linda!
## TSN | yahoo (445|496) | Not Cached - You got it!
## ULTA | yahoo (446|496) | Not Cached - Good job!
## UAA | yahoo (447|496) | Not Cached - Good stuff!
## UA | yahoo (448|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## UNP | yahoo (449|496) | Not Cached - Nice!
## UAL | yahoo (450|496) | Not Cached - Feels good!
## UDR | yahoo (451|496) | Not Cached - Youre doing good!
## UPS | yahoo (452|496) | Not Cached - OK!
## URI | yahoo (453|496) | Not Cached - Youre doing good!
## UTX | yahoo (454|496) | Not Cached - Well done!
```

```
## UNH | yahoo (455|496) | Not Cached - Good stuff!
## UHS | yahoo (456|496) | Not Cached - Good stuff!
## UNM | yahoo (457|496) | Not Cached - You got it!
## USB | yahoo (458|496) | Not Cached - Got it!
## VLO | yahoo (459|496) | Not Cached - Got it!
## VAR | yahoo (460|496) | Not Cached - Well done!
## VTR | yahoo (461|496) | Not Cached - Feels good!
## VRSN | yahoo (462|496) | Not Cached - You got it!
## VRSK | yahoo (463|496) | Not Cached - Got it!
## VZ | yahoo (464|496) | Not Cached - Youre doing good!
## VRTX | yahoo (465|496) | Not Cached - You got it!
## VFC | yahoo (466|496) | Not Cached - Youre doing good!
## VIAB | yahoo (467|496) | Not Cached - Well done!
## V | yahoo (468|496) | Not Cached - Youre doing good!
## VNO | yahoo (469|496) | Not Cached - Mas bah tche, que coisa linda!
## VMC | yahoo (470|496) | Not Cached - Youre doing good!
## WBA | yahoo (471|496) | Not Cached - Feels good!
## WMT | yahoo (472|496) | Not Cached - Good stuff!
## DIS | yahoo (473|496) | Not Cached - Nice!
## WM | yahoo (474|496) | Not Cached - Looking good!
## WAT | yahoo (475|496) | Not Cached - Good job!
## WEC | yahoo (476|496) | Not Cached - Nice!
## WFC | yahoo (477|496) | Not Cached - Got it!
## WELL | yahoo (478|496) | Not Cached - Well done!
## WDC | yahoo (479|496) | Not Cached - Youre doing good!
## WU | yahoo (480|496) | Not Cached - You got it!
## WRK | yahoo (481|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## WY | yahoo (482|496) | Not Cached - Looking good!
## WHR | yahoo (483|496) | Not Cached - OK!
## WMB | yahoo (484|496) | Not Cached - OK!
## WLTW | yahoo (485|496) | Not Cached - OK!
## GWW | yahoo (486|496) | Not Cached - Got it!
## WYND | yahoo (487|496) | Not Cached - Looking good!
## WYNN | yahoo (488|496) | Not Cached - Good stuff!
## XEL | yahoo (489|496) | Not Cached - Got it!
## XRX | yahoo (490|496) | Not Cached - You got it!
## XLNX | yahoo (491|496) | Not Cached - You got it!
## XYL | yahoo (492|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
## YUM | yahoo (493|496) | Not Cached - Good job!
## ZBH | yahoo (494|496) | Not Cached - Looking good!
## ZION | yahoo (495|496) | Not Cached - Well done!
## ZTS | yahoo (496|496) | Not Cached - OUT: not enough obs (see arg thresh.bad.data)
```

```
# comprobamos que variable es la que se registra 'price.close '
#a <- companias$df.tickers
#a <- subset(a, ticker=='A')
#sapply(a,class )
#a <- a[ a$price.open !=a$price.high,  ]
#a <- a[ a$price.low !=a$price.high,  ]
#a <- a[ a$price.low !=a$price.close ,  ]
#a <- a[ a$price.adjusted !=a$price.close ,  ]
#a <- unique(as.data.frame(a)) #identificamos la variable de interes
serie <- companias$df.tickers
class(serie) <- 'data.frame'
```

```
serie %>% select(ticker, ref.date, price.open ) -> serie#  era open o close ?
```

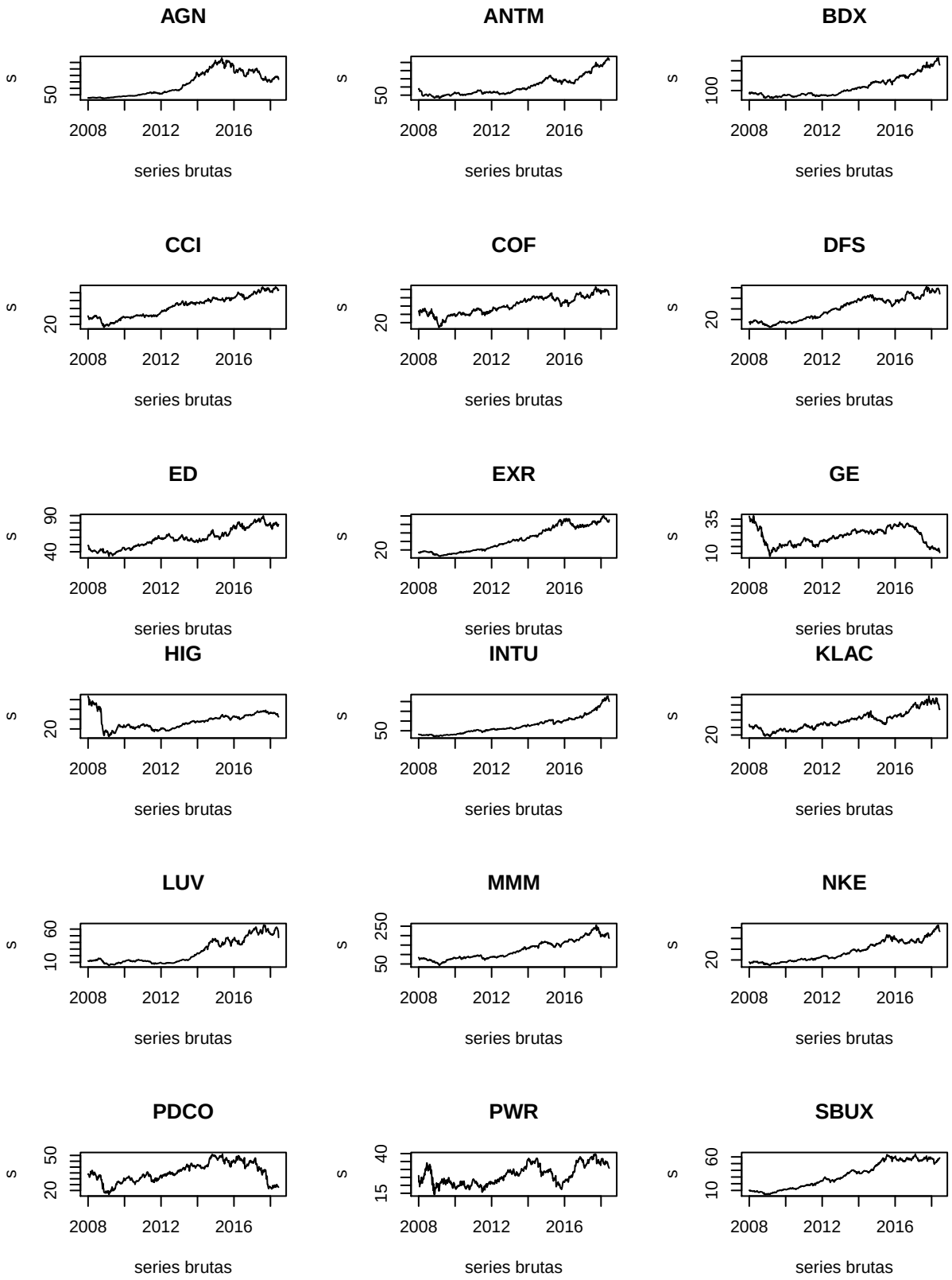
d. *Aplique las transformaciones necesarias (aprendidas en el módulo de series de tiempo) para trabajar las series de tiempo desde el punto de vista estacionario. Deseche las series de los mercados que presentan problemas.*

Se procedió a aplicar la transformación logaritmo para disminuir la varianza de las series originales, también se aplicaron técnicas para eliminar la estacionalidad en las pocas series que la presentan. El resultado es un conjunto de datos (de dimensiones $565, 449$) donde las observaciones son semanas registradas y las columnas las empresas que reportan su indicador de cierre. También se determino un resago de 8 para todas las series para estacionalizar las series, la prueba de Anderson Darling descarta que tengamos raices unitarias.

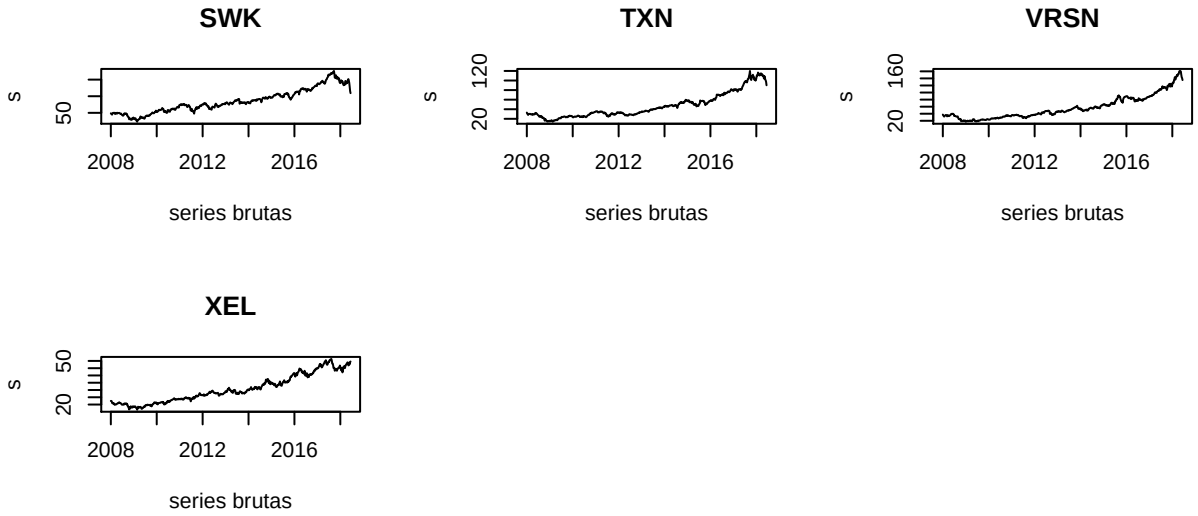
```
SP500 <- BatchGetSymbols(tickers = "^GSPC",
                         first.date = first.date,
                         last.date = last.date,
                         freq.data = freq.data,
                         do.complete.data = FALSE, #sihay nulos los descartamos
                         cache.folder = file.path(tempdir(),'BGS_Cache'))
```

```
##
## Running BatchGetSymbols for:
##    tickers = ^GSPC
##    Downloading data for benchmark ticker | Not Cached
## ^GSPC | yahoo (1|1) | Found cache file - Looking good!
```

```
#names(SP500)
SP500 <- as.data.frame(SP500$df.tickers)[, c('ref.date', 'price.close')]
sp.500 <- na.omit(SP500)
serie2 <- dcast(serie, ref.date ~ ticker, value.var = 'price.open' )
#write_csv(serie2, path='serie2.csv')
#serie2 <- read_csv( file='serie2.csv')
serie3 <- apply(serie2, 2, function(x) sum(is.na(x))) # identificamos series problematicas
#table(serie3)
malas <- which(serie3 > 8 )
serie4 <- serie2[,  !(colnames(serie2) %in% names(malas)) ]
serie5 <- na.omit(serie4)
class(serie5) <- 'data.frame'
serie5$ref.date <- as.Date(serie5$ref.date)
serie.cruda <- serie5 # para comparacion sin estacionalizar
par(mfrow=c(3,3))
#inspeccion visual
for(i in 1:(dim(serie5)[2]-1))
{
  s <- ts(serie5[, i], start = c(2008,1), frequency = 54)
  class(s)
  if(i %% 20==0) plot(s, main=as.character(names(serie5)[i]), xlab='series brutas')
}
```

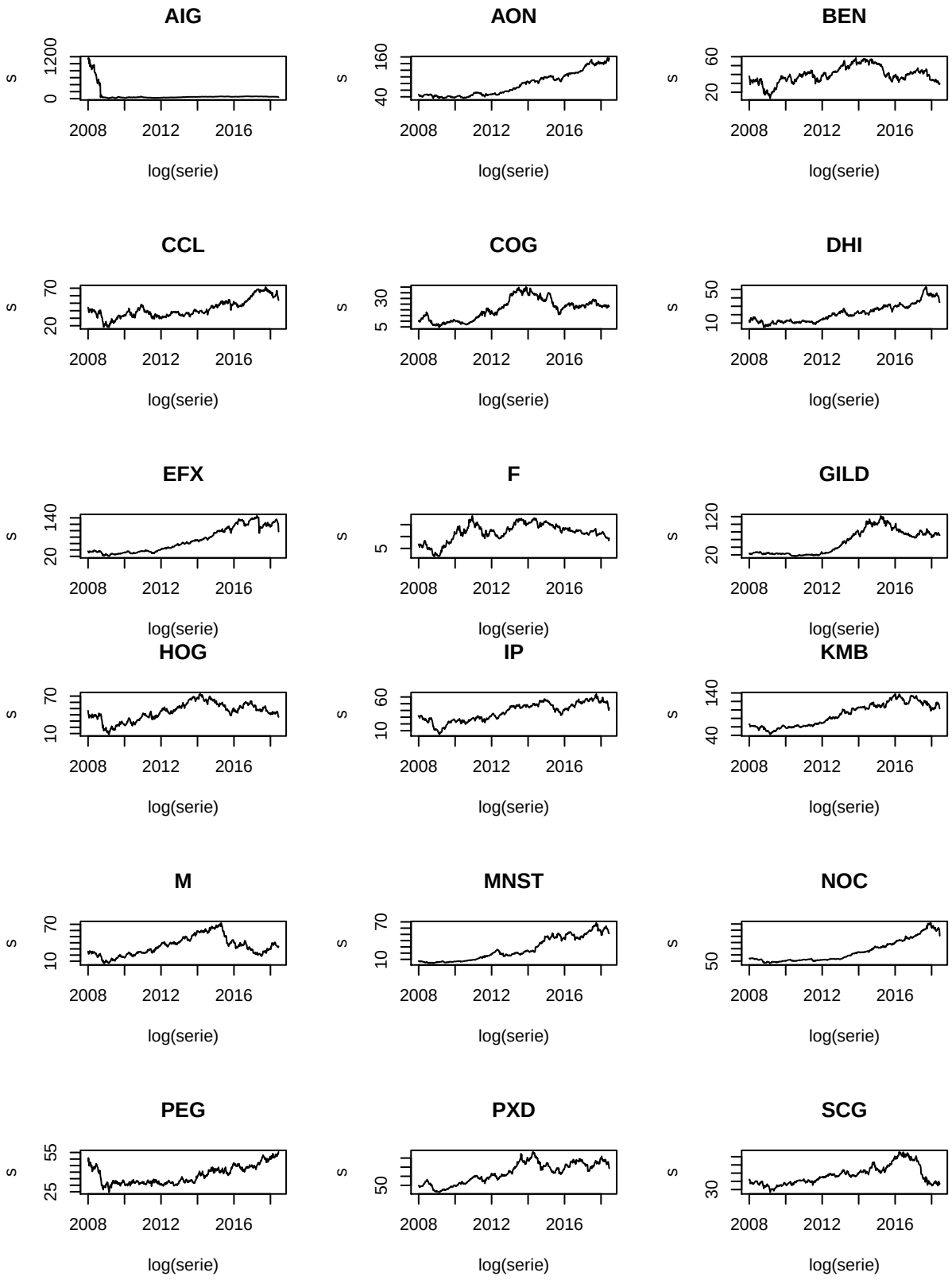



```
par(mfrow=c(1,1))
```

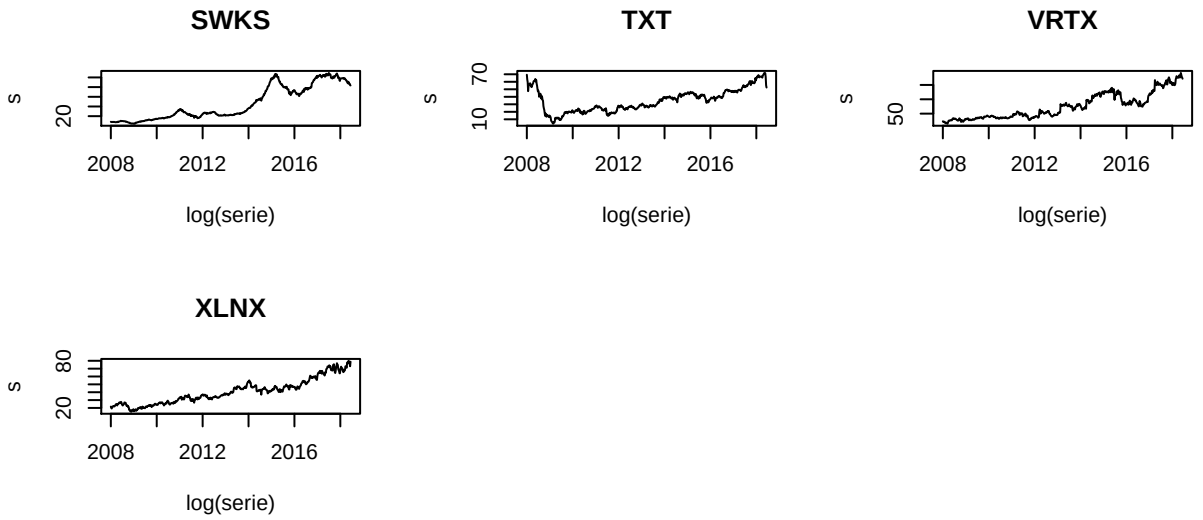



```
# quitamos tendencia
quita.tendencia <- quita.tendencia.init(inicio= c(2008, 1),frecuencia = 12 )
series <- mclapply(FUN=quita.tendencia, serie5[,2:dim(serie5)[2]], mc.cores = 6)
series <- as.data.frame(series)
```

```
series$time <- serie5$ref.date
series.ex <- (series[, -dim(series)[2]]) # aplicamos logaritmo para estabilizar la varianza
series.ex$time <- series$time
par(mfrow=c(3,3))
#inspeccion visual sin seasonality
for(i in 1:(dim(series)[2]-1))
{
  s <- ts(series.ex[, i], start = c(2008,1), frequency = 54)
  if(i %% 20==0) plot(s, main=as.character(names(series.ex)[i]), xlab='log(serie)')
}
```




```
par(mfrow=c(1,1))
```




```
# resagos
resagos <- matrix(rep(0, (dim(series.ex)[2]-1)*2), ncol=2)
dim(resagos)
```

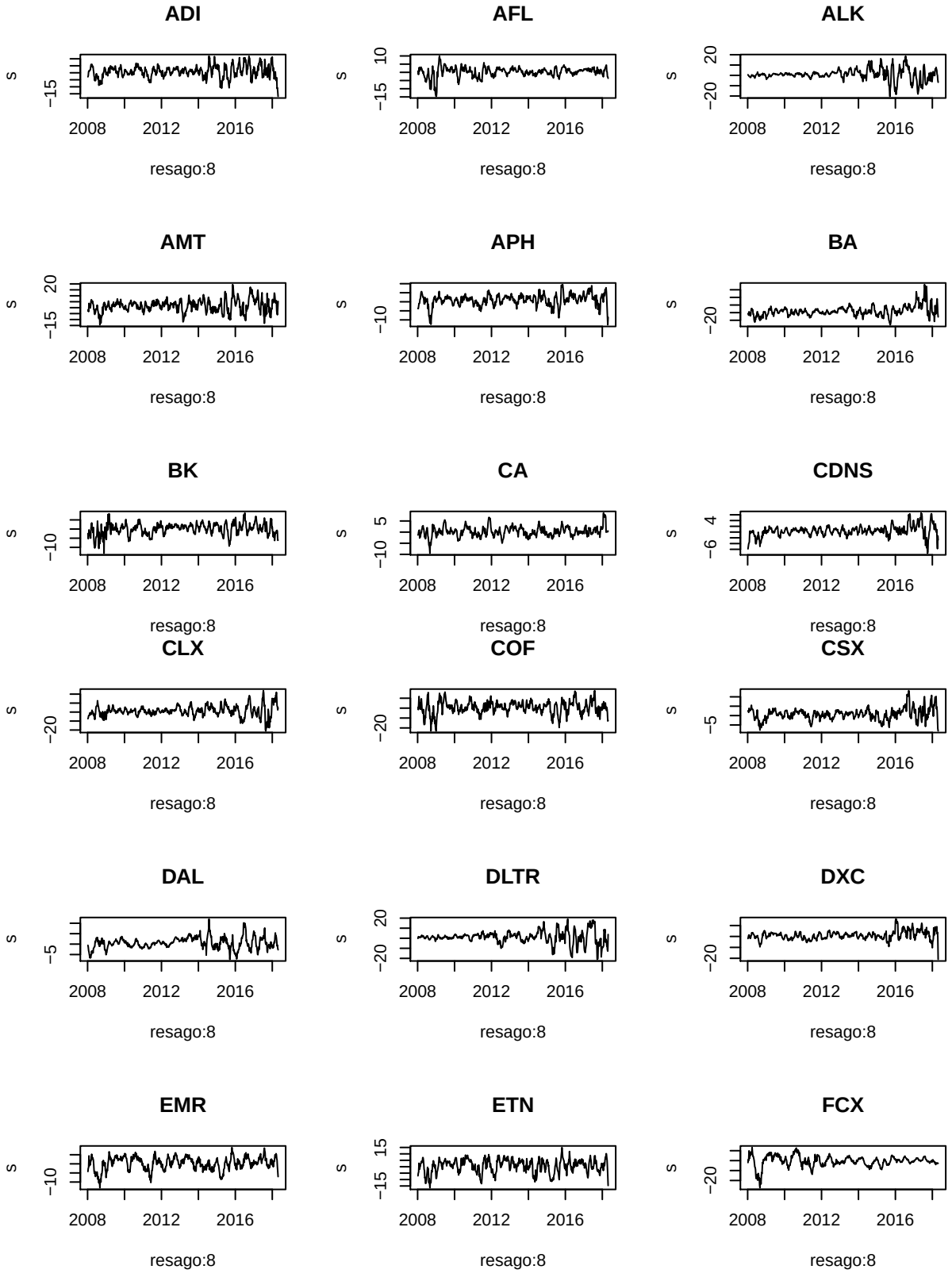
```
## [1] 446   2
```

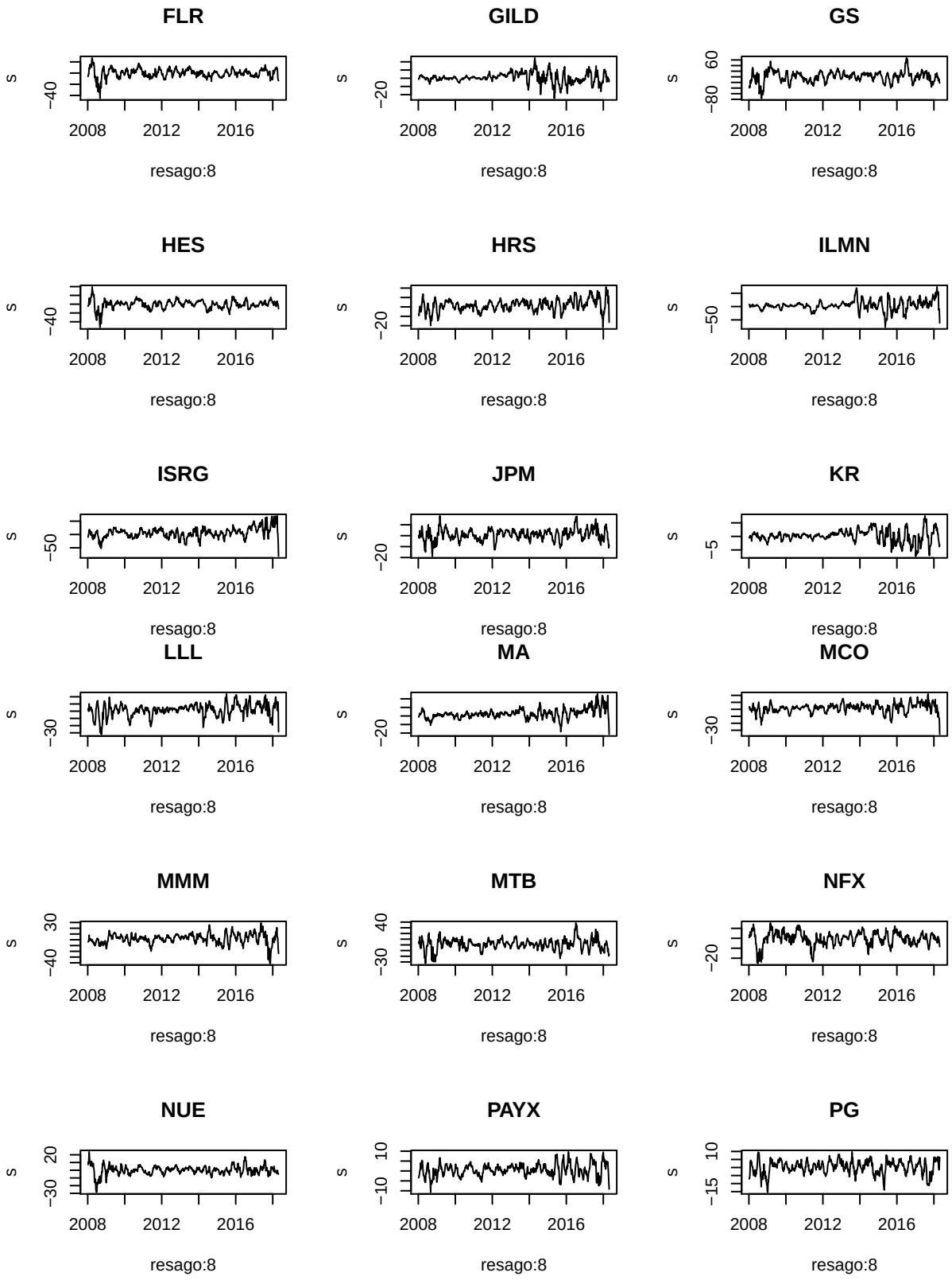
```
colnames(resagos) <- c('adf', 'adf.propio')
for(i in 1:(dim(series.ex)[2]-1))
{
  resagos[i, 'adf'] <- adf.test(ts(series.ex[,i]))$parameter
  resagos[i, 'adf.propio'] <- adf.test.custom(ts(series.ex[,i]))
}
#resagos
apply(resagos, 2, mean)
```

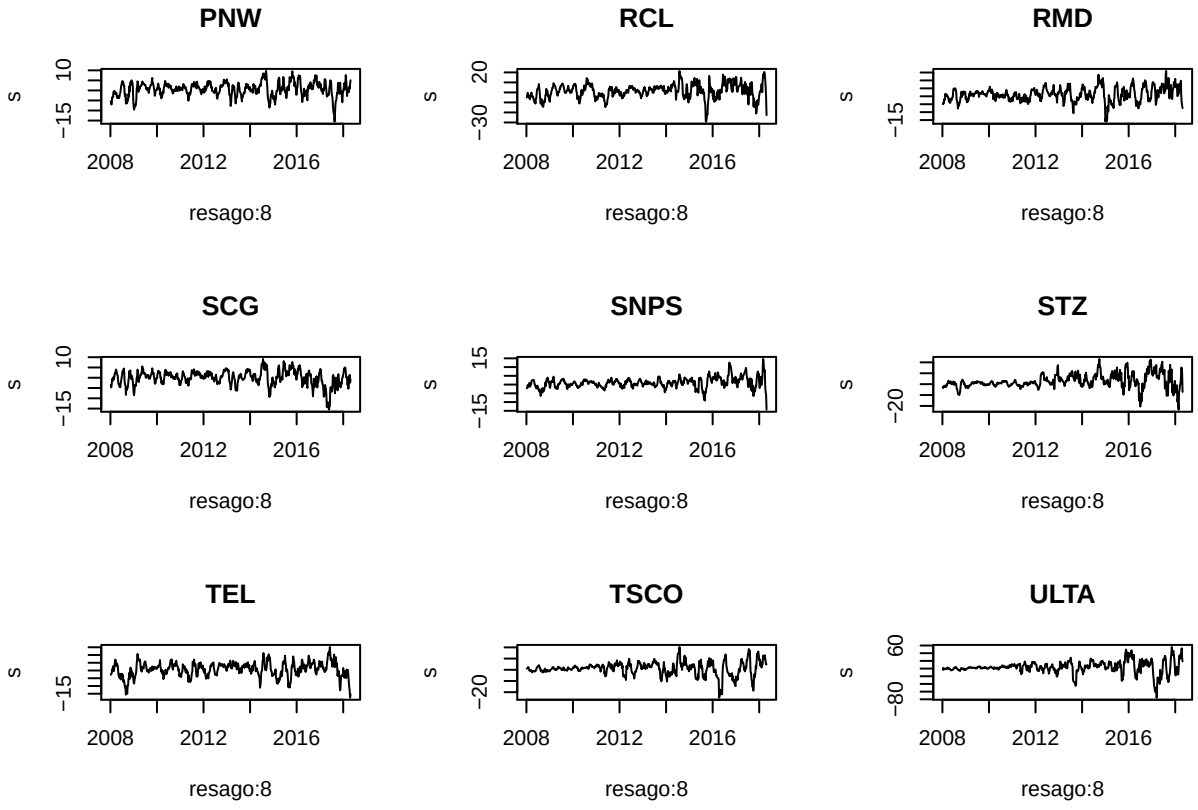
```
##        adf adf.propio
##   8.000000   2.165919
```

```
resago <- 8
# diferenciar
serie.chida <- apply(series.ex[,-dim(series.ex)[2]], 2, function(x) diff(x, lag=resago))
serie.chida <- as.data.frame(serie.chida)
serie.chida$time <- serie5$ref.date[-(1:resago)]
# test de raices
p.value <- rep(5, dim(serie.chida)[2]-1)
for(i in 1:(dim(serie.chida)[2]-1))
{
  p.value[i] <- adf.test(serie.chida[,i ])$p.value
}
#hist(p.value) # ya son estacionales
#p.value # porque son menores a 0.05
# visualizar
par(mfrow=c(3,3))
#inspeccion visual sin tendencia
for(i in 1:(dim(serie.chida)[2]-1))
{
  s <- ts(serie.chida[, i], start = c(2008,2), frequency = 54) #manual
  if(i %% 9==0) plot(s, main=as.character(names(serie.chida)[i]), xlab='resago:8')
  #if(i %% 9==0) a <- scan()
```

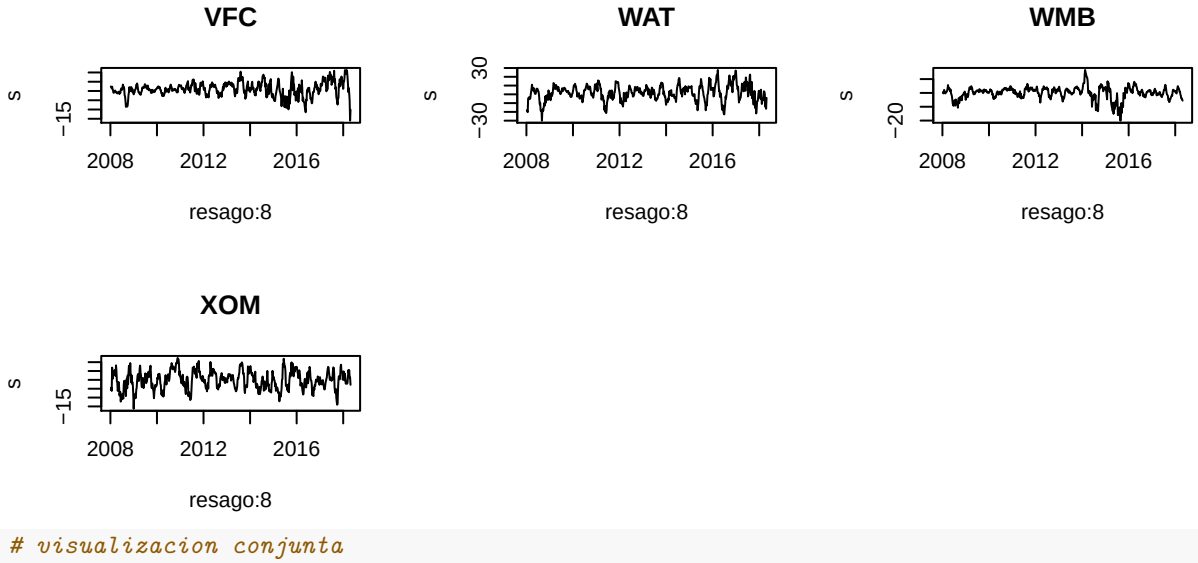
```
}
```


ADI
AFL
ALK
AMT
APH
BA
BK
CA
CDNS
CLX
COF
CSX
DAL
DLTR
DXC
EMR
ETN
FCX
s
2008
2012
2016
resago:8

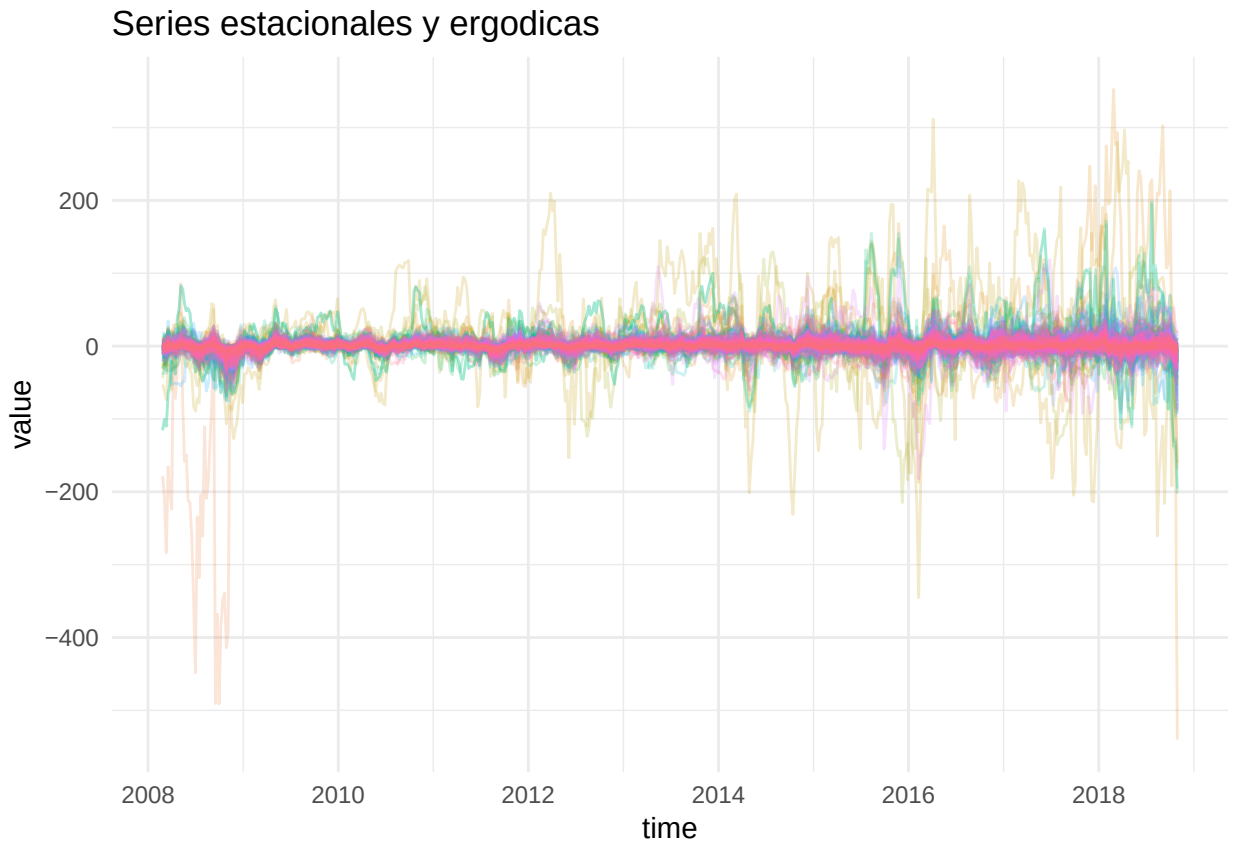

FLR
GILD
GS
HES
HRS
ILMN
ISRG
JPM
KR
LLL
MA
MCO
MMM
MTB
NFX
NUE
PAYX
PG
s
2008
2012
2016
resago:8




```
par(mfrow=c(1, 1))
```




```
# visualizacion conjunta
library(ggplot2)
a <- melt(serie.chida, id='time' )
#names(a)
ggplot(a, aes(x=time, y=value)) + geom_line(aes(colour=variable), alpha=0.2) +
  guides(colour=FALSE)+ theme_minimal()+ggtitle('Series estacionales y ergodicas')
```




```
#sp.500 <- ts(sp.500$price.close, start=2008, fre=54)
#plot.ts(sp.500, main='SP500', col='purple')
```

e. *Determine el número de componentes significativos adecuando un test derivado de la distribución de Marcenko-Pastur.*

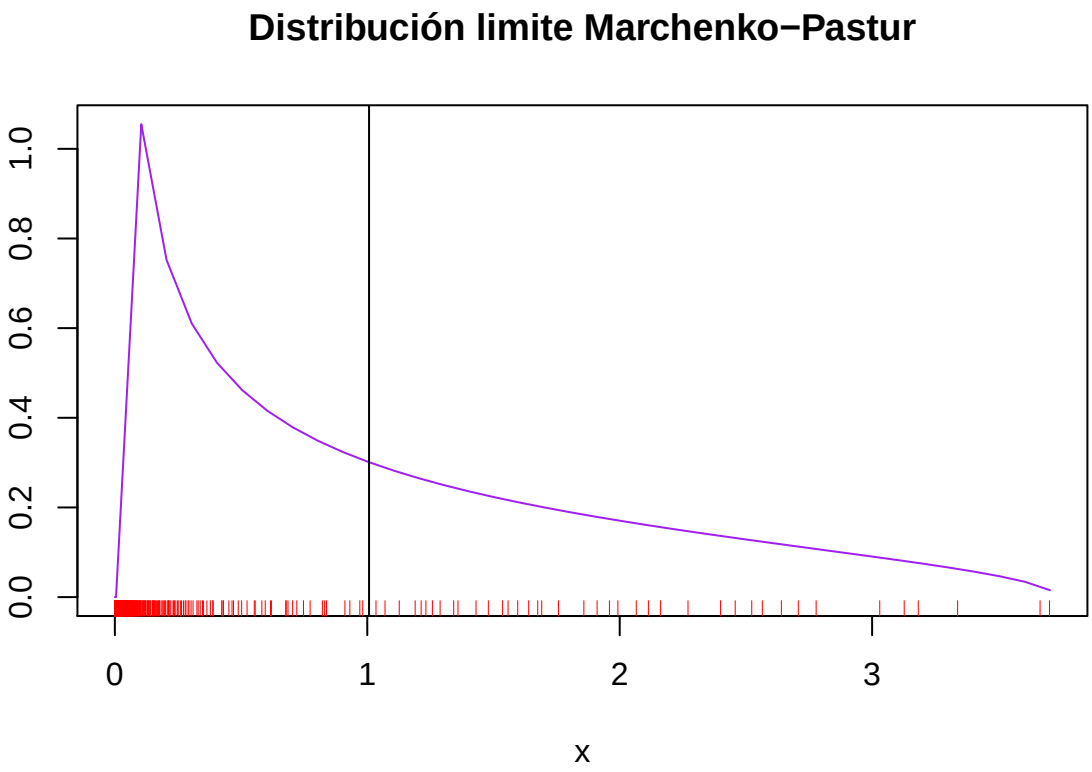
En vista de que conocemos la distribución asintótica de Marchenko Pastur, el criterio que determinamos se construye contemplando únicamente los valores propios que son mayores a la esperanza de la distribución, para ello la estimamos con una muestra 10 veces más grande que el número de registros de nuestro conjunto de datos.

```
##################################
# interseccion de series
##################################
names(SP500) <- c('time', 'price.close.SP')
m <- merge(serie.chida, SP500, by.x='time', by.y='time')
colnames(m) <- c(colnames(m)[1:(dim(m)[2]-1)], 'y')
tiempo <- m$time
#tiempo
index <- which(tiempo>ymd('2018-01-01'))
m$time <- NULL
train <- m[-index, ]
test <- m[index,]
vals <- eigen(cor(scale(train[, -dim(train)[2]])))$values
# minisimulacion
set.seed(0)
#names(train)
r <- (dim(train)[2]-1)/dim(train)[1]
```

```
x <- c(0,seq((1-r**.5)**2, (1+r**.5)**2, by=0.1)) #soporte de la distribucion M-P
plot(x,dmp(x, ndf=dim(train)[1], pdim=dim(train)[2]-1 ), col='purple', type='l',
     ylab='', main='Distribución limite Marchenko-Pastur')
rug(vals, col='red')
muestras <- dim(m)[1]*10
set.seed(0)
(limite <- mean(rmp(muestras, ndf=dim(train)[1], pdim=dim(train)[2]-1 )))
```

```
## [1] 1.007094
```

```
abline(v=limite)
```




```
vals <- vals[vals>limite]
(RMT.cota <- length(vals)) #60 cota criterio de M-P
```

```
## [1] 62
```

f. *Aplique regresión por componentes principales utilizando el número de componentes sugeridos por el resultado de matrices aleatorias y compare el resultado utilizando el criterio del 80% de la varianza. Se busca predecir el valor de apertura del índice S&P500 el lunes por la mañana a traves de los 500 mercados que lo componen.*

Para poder medir el error en porcentaje utilizamos el SMAPE cuyo rango es [0,100] donde 0 es un error de 0% (predicción perfecta) y 100 una predicción totalmente erronea. También utilizamos el error promedio sobre las predicciones del año 2008. Los resultados son de 86% aprox. en ambos casos.

```
################## resultado con RMT
modelo.pcr.rmt <- pcr(y~., data=train, ncomp=RMT.cota)
summary(modelo.pcr.rmt)
```

```
## Data:    X dimension: 514 446
##  Y dimension: 514 1
```

```
## Fit method: svdpc
## Number of components considered: 62
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X   28.921    41.70    50.15    56.96    62.71    67.18    70.52    73.26
## y    6.131    11.36    16.51    17.92    21.57    21.89    22.03    25.51
##    9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X    75.35     77.19     78.98     80.26     81.32     82.25     83.15
## y    28.25     35.60     35.62     37.65     37.95     38.78     43.71
##    16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
## X     83.96     84.76     85.43     86.05     86.67     87.25     87.81
## y     46.76     46.82     47.12     47.51     47.63     47.88     49.47
##    23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29 comps
## X     88.34     88.83     89.30     89.73     90.14     90.52     90.87
## y     49.60     54.62     54.69     55.92     56.48     56.49     56.49
##    30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X     91.20     91.51     91.81     92.10     92.37     92.63     92.87
## y     57.07     58.39     58.42     58.58     58.61     58.62     58.67
##    37 comps  38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X     93.10     93.33     93.55     93.76     93.96     94.15     94.34
## y     59.49     60.65     60.85     61.71     63.83     63.87     64.25
##    44 comps  45 comps  46 comps  47 comps  48 comps  49 comps  50 comps
## X     94.52     94.70     94.87     95.04     95.20     95.34     95.49
## y     65.52     65.57     65.85     66.26     66.28     66.77     66.85
##    51 comps  52 comps  53 comps  54 comps  55 comps  56 comps  57 comps
## X     95.63     95.76     95.89     96.01     96.13     96.25     96.37
## y     66.87     67.65     70.17     72.47     72.64     72.70     73.63
##    58 comps  59 comps  60 comps  61 comps  62 comps
## X     96.47     96.57     96.67     96.77     96.86
## y     74.21     76.54     76.57     82.79     82.80
```

```
y.hat.test <- predict(modelo.pcr.rmt, ncomp=RMT.cota , newdata = test)
(100-SMAPE(test$y, as.numeric(y.hat.test))) #Presicion SMAPE
```

```
## [1] 86.58055
```

```
(100-ERROR(test$y, as.numeric(y.hat.test))) #Presicion promedio
```

```
## [1] 84.8645
```

```
res1 <- data.frame(y=test$y, y.hat=as.numeric(y.hat.test),
                   time=tiempo[-(1:(dim(train)[1])) ])
p1 <- ggplot(res1, aes(x=time, y=y))+geom_line(color=I('purple')) + theme_minimal()+
  geom_line(data=res1, aes(x=time, y=y.hat),color=I('orange'))+
  ggtitle('Pronostico para 2018,SP800, criterio MP y PCR')
################## resultado con 80 vars
modelo.pcr.rmt <- pcr(y~., data=train, ncomp=57)
y.hat.test <- predict(modelo.pcr.rmt, ncomp=57 , newdata = test)
(100-SMAPE(test$y, as.numeric(y.hat.test))) #Presicion SMAPE
```

```
## [1] 84.28605
```

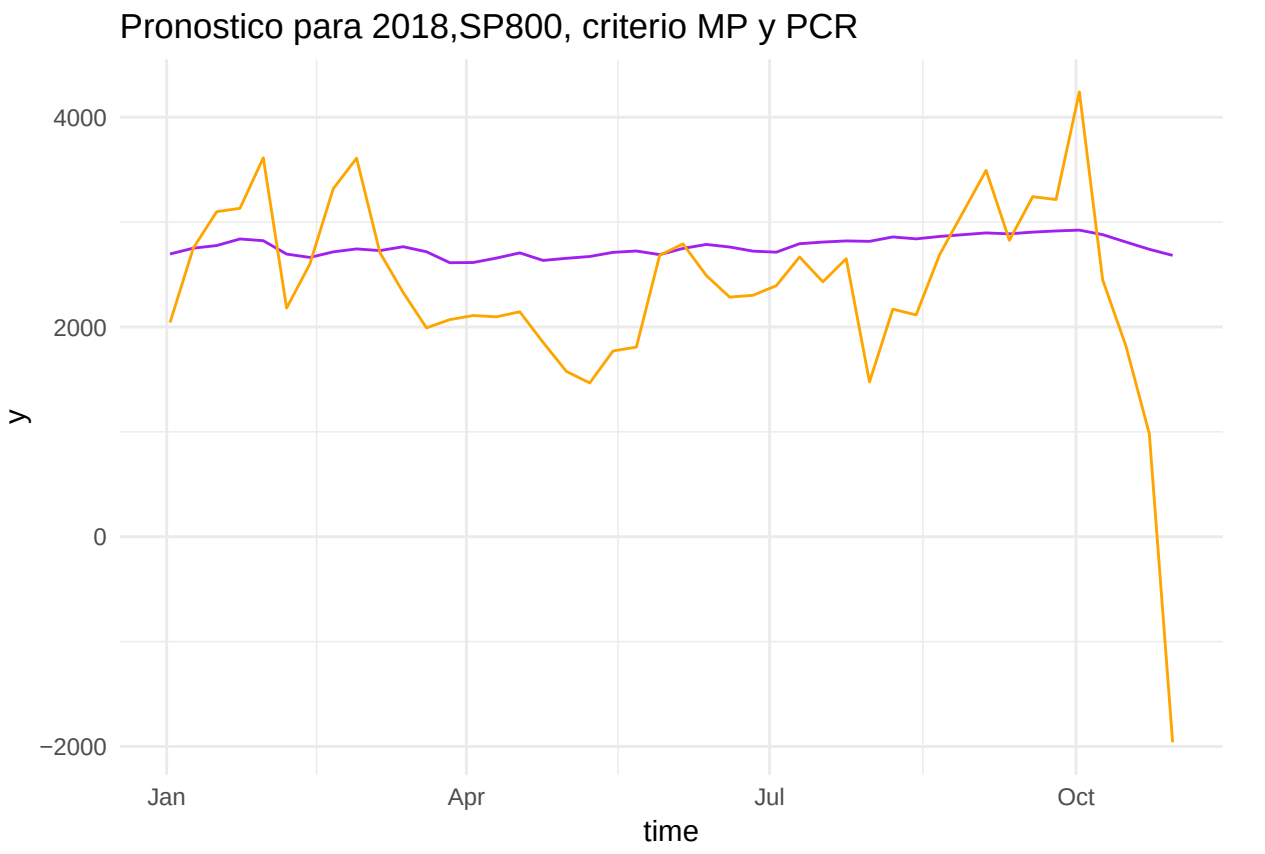
```
(100-ERROR(test$y, as.numeric(y.hat.test))) #Presicion promedio
```

```
## [1] 83.30924
```

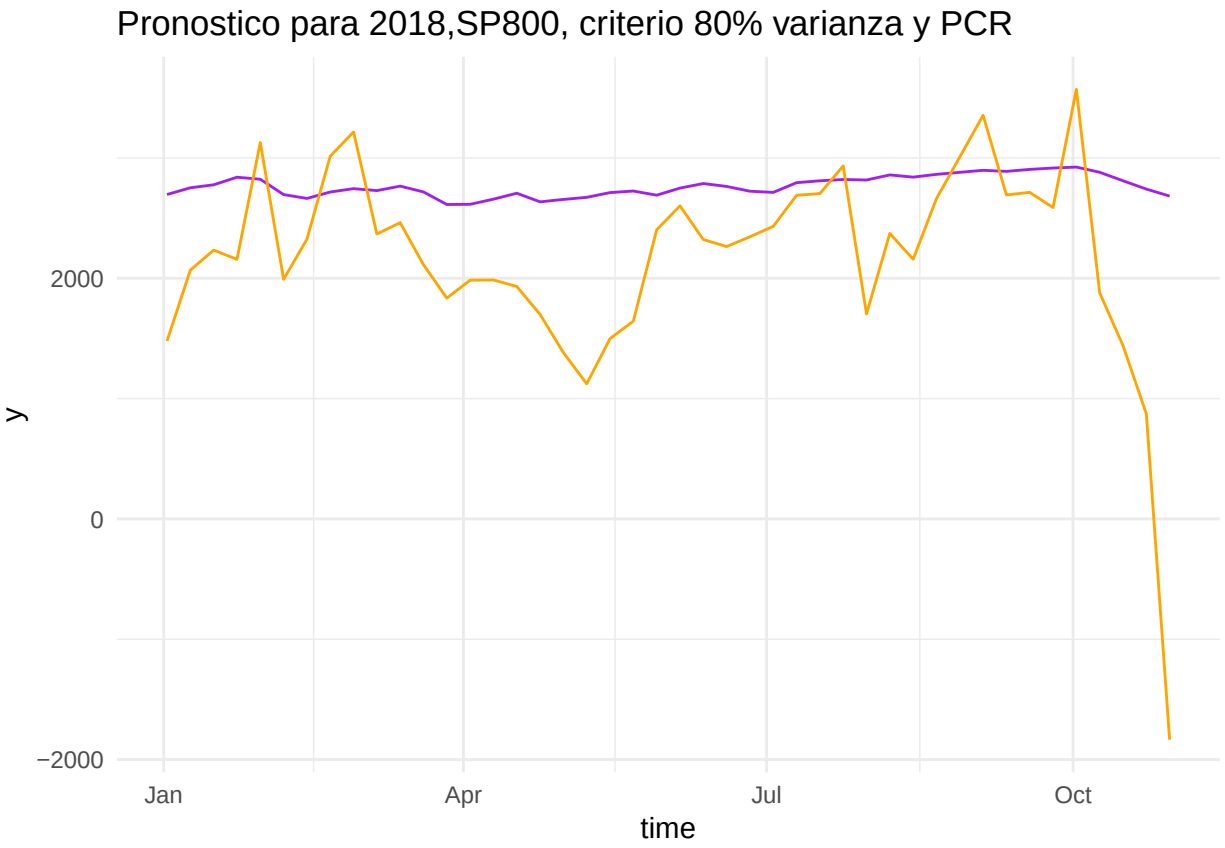
```
res1 <- data.frame(y=test$y, y.hat=as.numeric(y.hat.test),
                   time=tiempo[-(1:dim(train)[1])])
p2 <- ggplot(res1, aes(x=time, y=y))+geom_line(color=I('purple')) + theme_minimal()+
  geom_line(data=res1, aes(x=time, y=y.hat),color=I('orange')) +
  ggtitle('Pronostico para 2018,SP800, criterio 80% varianza y PCR')
```

g. *Grafique la efectividad del pronóstico durante este año.*

```
p1
```




```
p2
```



h. *¿Cómo mejoraría el pronóstico? si obtiene un promedio en la efectividad mayor al 50% gana puntos extras en proporción a como este valor se acerque al 100% (Puede explorar otros métodos de pronóstico en busca de mayor efectividad, pero siempre contrastando con el criterio de matrices aleatorias).*

En vista de que estamos trabajando con configuraciones de datos de dimensionalidad mediana, consideramos un método de regresión, PLS,que a su vez no requiere de supuestos distribucionales, reduce dimensionalidad y podemos utilizar el criterio de Marchenko Pastur como cota superior para evaluar el número de componentes (a.k.a variables latentes de PLS) y disminuir el tiempo de cómputo. Los resultados mejoran a los anteriores alcanzando una precisión de 85%

```
################## resultado con PLS
modelo.pcr.rmt <- plsr(y~., data=train, ncomp=RMT.cota)
summary(modelo.pcr.rmt)
```

```
## Data:    X dimension: 514 446
##  Y dimension: 514 1
## Fit method: kernelpls
## Number of components considered: 62
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X    24.72    37.87    45.23    51.10    56.20    59.28    63.49    67.09
## y    20.58    35.61    49.21    59.29    65.81    73.59    77.58    81.72
##    9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X    69.66     72.63     74.07     76.36     78.02     79.28     80.14
## y    85.76     87.90     90.17     91.14     91.95     92.80     93.64
##    16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
## X     80.70     81.30     82.26     83.39     84.07     84.61     84.98
## y     94.39     95.09     95.52     95.83     96.16     96.47     96.86
```

```
##    23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29 comps
## X     85.47     86.04     86.45     86.86     87.14     87.64     88.00
## y     97.08     97.26     97.46     97.61     97.81     97.92     98.06
##    30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X     88.50     88.81     89.07     89.42     89.73     90.03     90.33
## y     98.12     98.21     98.31     98.39     98.46     98.53     98.59
##    37 comps  38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X     90.69     90.97     91.33     91.57     91.82     92.13     92.30
## y     98.64     98.70     98.75     98.80     98.85     98.88     98.92
##    44 comps  45 comps  46 comps  47 comps  48 comps  49 comps  50 comps
## X     92.54     92.75     92.92     93.08     93.24     93.44     93.63
## y     98.95     98.98     99.02     99.06     99.09     99.12     99.14
##    51 comps  52 comps  53 comps  54 comps  55 comps  56 comps  57 comps
## X     93.80     93.94     94.12     94.27     94.42     94.55     94.70
## y     99.17     99.19     99.21     99.23     99.25     99.26     99.28
##    58 comps  59 comps  60 comps  61 comps  62 comps
## X     94.84     94.96     95.08     95.20     95.33
## y     99.30     99.31     99.33     99.34     99.35
```

```
error <- rep(0, RMT.cota)
for (i in 1:RMT.cota){
y.hat.test <- predict(modelo.pcr.rmt, ncomp=i , newdata = test)
error[i] <- SMAPE(test$y, as.numeric(y.hat.test)) #100-19.15192
}
which.min(error)
```

```
## [1] 8
```

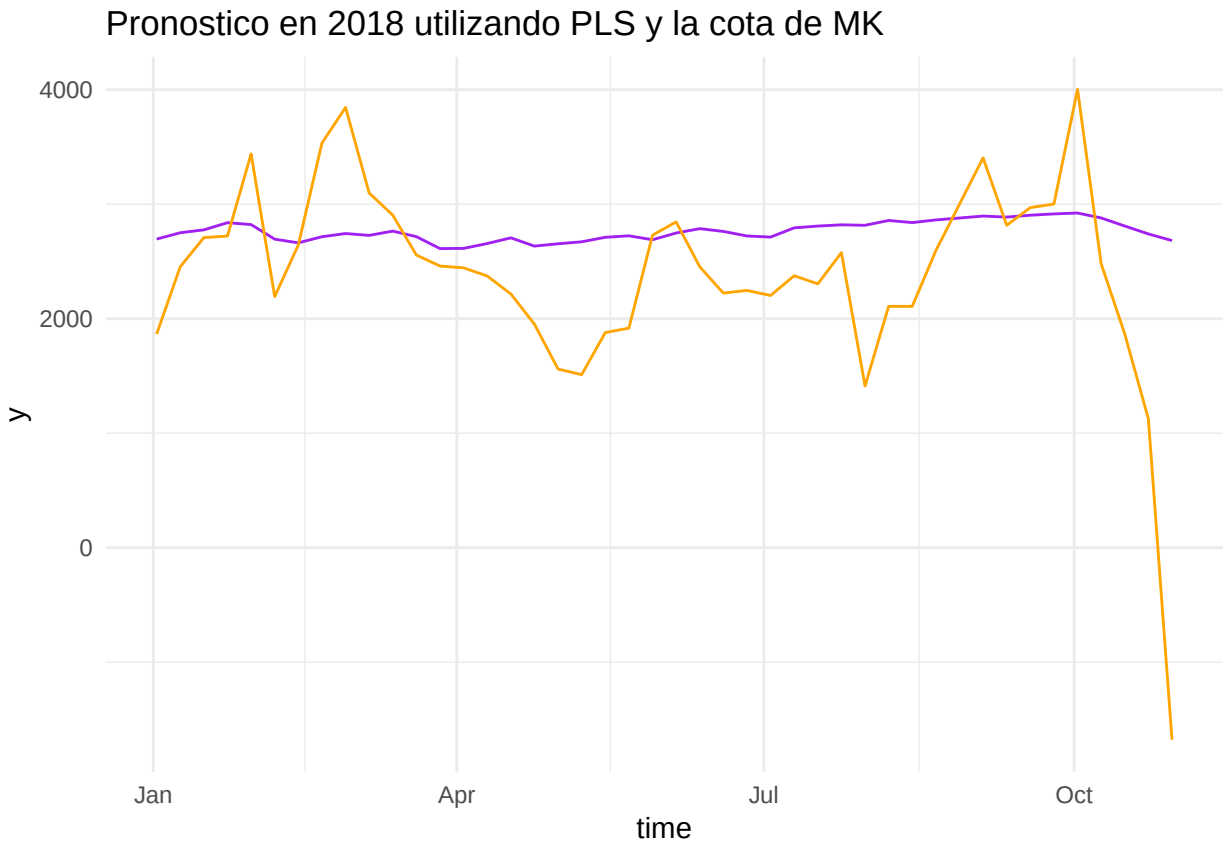
```
modelo.pcr.rmt <- plsr(y~., data=train, ncomp=which.min(error))
y.hat.test <- predict(modelo.pcr.rmt, ncomp=which.min(error) , newdata = test)
(100-SMAPE(test$y, as.numeric(y.hat.test))) #Presicion SMAPE
```

```
## [1] 87.38741
```

```
(100-ERROR(test$y, as.numeric(y.hat.test))) #Presicion promedio
```

```
## [1] 85.8537
```

```
res1 <- data.frame(y=test$y, y.hat=as.numeric(y.hat.test),
                   time=tiempo[-(1:dim(train)[1])])
ggplot(res1, aes(x=time, y=y))+geom_line(color=I('purple')) + theme_minimal()+
  geom_line(data=res1, aes(x=time, y=y.hat),color=I('orange'))+ggtitle('Pronostico en 2018 utilizando Pl
```


Pronostico en 2018 utilizando PLS y la cota de MK
4000
2000
0
y
Jan
Apr
Jul
Oct
time