Práctica 2

1 Distribuciones bidimensionales. Regresión y correlación

1.1 Lectura de datos

Utilizaremos el fichero **ADLS.csv**, que ya fue empleado en la práctica anterior. Recordamos la importancia de establecer previamente el directorio de trabajo. Al tratarse de un fichero de texto con campos separados por punto y coma, en el que se utiliza la coma como separador decimal, haremos uso del comando `read.csv2` para leerlo. Recordemos que otros comandos relacionados con la lectura de ficheros de texto son `read.table`, `read.csv`, `read.delim` y `read.delim2`.

El fichero utilizado contiene información de los municipios de las provincias de Andalucía. Son datos reales obtenidos a través de la página web del Instituto de Estadística y Cartografía de Andalucía (IECA) relativos al año 2010. Cada observación corresponde a un municipio, en el que se han contemplado las siguientes variables:

- provincia: provincia a la que pertenece el municipio.
- codigo: código postal.
- municipio: nombre del municipio.
- lintelef: número de líneas de la compañía Telefónica en servicio.
- rdsi: número de líneas RDSI en servicio.
- adsl: número de líneas ADSL en servicio.
- poblacion: número de habitantes según el padrón.
- claspob: agrupación del número de habitantes.
  - P1: <200000
  - P2: [200000,500000)
  - P3: mayor o igual 500000
- ingresos: ingresos por habitante
- clasing: agrupación de los ingresos por habitante
  - I1: <1500
  - I2: [1500,3000)
  - I3: mayor o igual 3000
- cooperativas: número de cooperativas del sector Información y Comunicaciones.
- CentrosSalud: número de centros de salud.

El comando `read.csv2` devuelve los datos con estructura **data.frame**, como puede verse con el comando `str`. El comando `head` muestra los seis primeros casos.

```r
datos = read.csv2("ADSL.csv")
str(datos)
```

```
## 'data.frame':    758 obs. of  12 variables:
##  $ provincia   : Factor w/ 8 levels "Almería","Cádiz",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ codigo      : int  4001 4002 4003 4004 4005 4006 4007 4008 4009 4010 ...
##  $ municipio   : Factor w/ 758 levels "Abla","Abrucena",..: 1 2 4 13 15 18 29 31 32 45 ...
##  $ lintelef    : int  345 274 4124 248 188 2393 154 144 26 134 ...
##  $ rdsi        : int  12 9 184 10 9 115 8 0 0 5 ...
##  $ adsl        : int  149 103 2664 93 38 1419 61 1 0 52 ...
##  $ poblacion   : int  1463 1367 24512 814 667 11042 902 598 144 718 ...
##  $ claspob     : Factor w/ 3 levels "P1","P2","P3": 1 1 2 1 1 1 1 1 1 1 ...
##  $ ingresos    : num  1639 987 856 1051 1395 ...
##  $ clasing     : Factor w/ 3 levels "I1","I2","I3": 2 1 1 1 1 1 2 2 2 2 ...
##  $ cooperativas: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CentrosSalud: int  1 0 1 0 0 1 0 0 0 0 ...
```

```r
head(datos)
```

```
##   provincia codigo municipio lintelef rdsi adsl poblacion claspob ingresos
## 1   Almería   4001      Abla      345   12  149      1463      P1  1638.76
## 2   Almería   4002  Abrucena      274    9  103      1367      P1   987.08
## 3   Almería   4003      Adra     4124  184 2664     24512      P2   856.01
## 4   Almería   4004 Albánchez      248   10   93       814      P1  1051.14
## 5   Almería   4005 Alboloduy      188    9   38       667      P1  1395.17
## 6   Almería   4006     Albox     2393  115 1419     11042      P1   883.66
##   clasing cooperativas CentrosSalud
## 1      I2            0            1
## 2      I1            0            0
## 3      I1            0            1
## 4      I1            0            0
## 5      I1            0            0
## 6      I1            0            1
```

1.2 Distribución conjunta

Construimos la tabla de frecuencias conjuntas (absolutas) correspondiente a las variables **provincia** y **claspob**.

```r
conjunta = table(datos$provincia,datos$claspob)
conjunta
```

```
##
##            P1  P2  P3
##   Almería  94   3   3
##   Cádiz    28   7   8
##   Córdoba  65   7   1
##   Granada 157   6   2
##   Huelva   72   5   1
##   Jaén     89   4   2
##   Málaga   83   8   8
##   Sevilla  89  12   4
```

Obtenemos el número total de observaciones.

```r
n = sum(conjunta)
n
```

```
## [1] 758
```

Construimos la tabla de frecuencias relativas conjuntas (de dos formas distintas). El comando `prop.table` permite expresar las entradas de una tabla como fracción del total (es decir, en tanto por uno).

```r
relativa.conjunta = conjunta/n
relativa.conjunta
```

```
##
##                    P1          P2          P3
##   Almería 0.124010554 0.003957784 0.003957784
##   Cádiz   0.036939314 0.009234828 0.010554090
##   Córdoba 0.085751979 0.009234828 0.001319261
##   Granada 0.207124011 0.007915567 0.002638522
##   Huelva  0.094986807 0.006596306 0.001319261
##   Jaén    0.117414248 0.005277045 0.002638522
##   Málaga  0.109498681 0.010554090 0.010554090
##   Sevilla 0.117414248 0.015831135 0.005277045
```

```r
relativa.conjunta = prop.table(conjunta)    # también se puede hacer asi
relativa.conjunta
```

```
##
##                    P1          P2          P3
##   Almería 0.124010554 0.003957784 0.003957784
##   Cádiz   0.036939314 0.009234828 0.010554090
##   Córdoba 0.085751979 0.009234828 0.001319261
##   Granada 0.207124011 0.007915567 0.002638522
##   Huelva  0.094986807 0.006596306 0.001319261
##   Jaén    0.117414248 0.005277045 0.002638522
##   Málaga  0.109498681 0.010554090 0.010554090
##   Sevilla 0.117414248 0.015831135 0.005277045
```

1.3 Distribuciones marginales

Obtenemos la marginal de la variable **provincia**. El cualificador **margin** del comando `margin.table` indica la variable para la que se calculará la marginal: 1 para la primera variable de la tabla, etc.

```r
marginal.X = margin.table(conjunta,margin=1)    # marginal de provincia
marginal.X
```

```
##
## Almería   Cádiz Córdoba Granada  Huelva    Jaén  Málaga Sevilla
##     100      43      73     165      78      95      99     105
```

Obtenemos la distribución marginal de **provincia** en términos de frecuencias relativas (de dos formas distintas).

```r
relativa.marginal.X = margin.table(relativa.conjunta,margin=1)
relativa.marginal.X
```

```
##
##    Almería      Cádiz    Córdoba    Granada     Huelva       Jaén
## 0.13192612 0.05672823 0.09630607 0.21767810 0.10290237 0.12532982
```

```
##     Málaga    Sevilla
## 0.13060686 0.13852243
```

```
relativa.marginal.X = marginal.X/sum(marginal.X)    # también se puede hacer asi
relativa.marginal.X
```

```
##
##    Almería      Cádiz    Córdoba    Granada     Huelva       Jaén
## 0.13192612 0.05672823 0.09630607 0.21767810 0.10290237 0.12532982
##     Málaga    Sevilla
## 0.13060686 0.13852243
```

De forma análoga para la variable **claspob**...

```
marginal.Y = margin.table(conjunta,margin=2)    # marginal de claspob
marginal.Y
```

```
##
##  P1  P2  P3
## 677  52  29
```

1.4 Distribuciones condicionadas

Obtenemos las distribuciones condicionadas de la forma $Y/X = x_k$.

```
CondicionadasaX = prop.table(conjunta,1)    # distribuciones condicionadas Y/X=xk
CondicionadasaX
```

```
##
##                   P1         P2         P3
##   Almería 0.94000000 0.03000000 0.03000000
##   Cádiz   0.65116279 0.16279070 0.18604651
##   Córdoba 0.89041096 0.09589041 0.01369863
##   Granada 0.95151515 0.03636364 0.01212121
##   Huelva  0.92307692 0.06410256 0.01282051
##   Jaén    0.93684211 0.04210526 0.02105263
##   Málaga  0.83838384 0.08080808 0.08080808
##   Sevilla 0.84761905 0.11428571 0.03809524
```

Análogamente, obtenemos las distribuciones de la forma $X/Y = y_k$.

```
CondicionadasaY = prop.table(conjunta,2)        # distribuciones condicionadas X/Y=xk
CondicionadasaY
```

```
##
##                   P1         P2         P3
##   Almería 0.13884786 0.05769231 0.10344828
##   Cádiz   0.04135894 0.13461538 0.27586207
##   Córdoba 0.09601182 0.13461538 0.03448276
##   Granada 0.23190547 0.11538462 0.06896552
##   Huelva  0.10635155 0.09615385 0.03448276
##   Jaén    0.13146233 0.07692308 0.06896552
##   Málaga  0.12259970 0.15384615 0.27586207
##   Sevilla 0.13146233 0.23076923 0.13793103
```

Si queremos limitar el número de decimales a 3, por ejemplo:

```
round(CondicionadasaY,3)
```

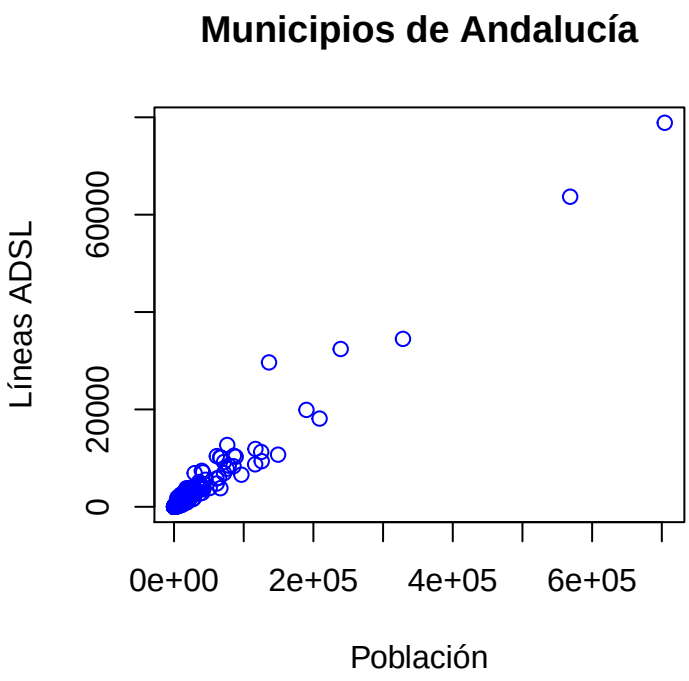
```
##
##              P1    P2    P3
##   Almería 0.139 0.058 0.103
##   Cádiz   0.041 0.135 0.276
##   Córdoba 0.096 0.135 0.034
##   Granada 0.232 0.115 0.069
##   Huelva  0.106 0.096 0.034
##   Jaén    0.131 0.077 0.069
##   Málaga  0.123 0.154 0.276
##   Sevilla 0.131 0.231 0.138
```

1.5 Diagrama de dispersión o nube de puntos

Representamos en un diagrama de dispersión el número de habitantes de cada municipio frente al número de líneas ADSL en servicio que posee.

```
x = datos$poblacion
y = datos$adsl

plot(x,y,main="Municipios de Andalucía",xlab="Población",
     ylab="Líneas ADSL",col="blue")
```

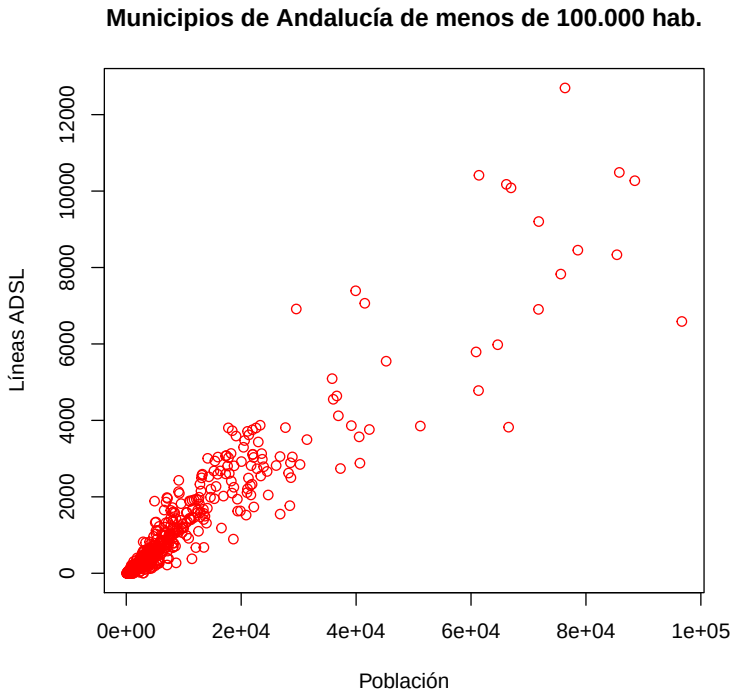


El gráfico queda distorsionado por la presencia de algunas observaciones con valores muy elevados en las variables consideradas. Nos centraremos en los municipios con un comportamiento no atípico seleccionando aquellos que tienen menos de 100.000 habitantes.

```
datos2 = datos[x<100000,]
```

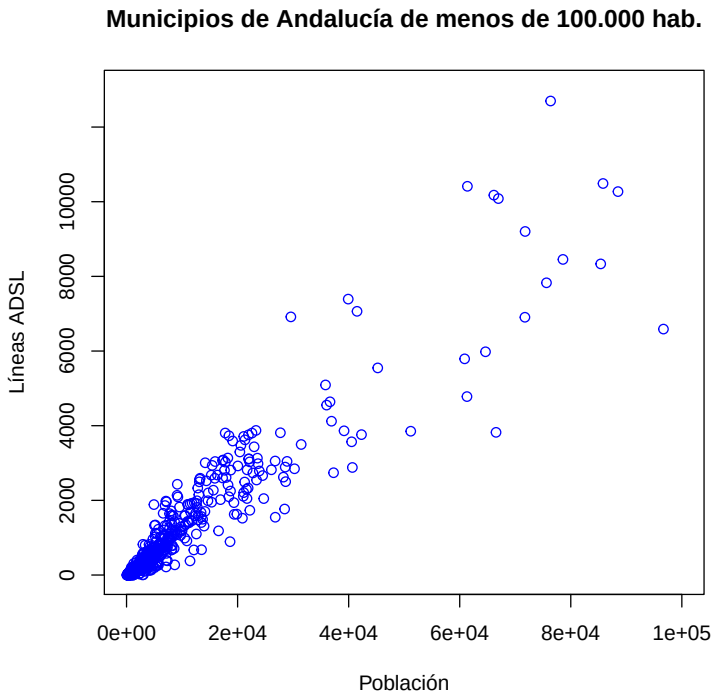
Dibujamos nuevamente el diagrama de dispersión.

```
plot(datos2$poblacion,datos2$adsl,main="Municipios de Andalucía de menos de 100.000 hab.",
     xlab="Población",ylab="Líneas ADSL",col="red")
```



Otra forma de hacerlo consiste en limitar las escalas horizontal y vertical para que el gráfico sólo presente los municipios de interés.

```
plot(x,y,main="Municipios de Andalucía de menos de 100.000 hab.",
     xlab="Población",ylab="Líneas ADSL",col="blue",xlim=c(0,100000),ylim=c(0,13000))
```



1.6 Cálculo de la covarianza

Recordemos que la función `cov` proporciona $\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})$. Por lo que para calcular la covarianza, será necesario realizar la siguiente transformación:

```
covarianza=((n-1)/n)*cov(x,y)
```

1.7 Recta de regresión Líneas ADSL / población

La recta de regresión puede calcularse utilizando el comando `lm` (ajuste de modelos lineales). Se recuerda que, en teclado español, la virgulilla (~) se puede obtener con AltGr 4 tanto en Linux como en Windows.

```
rYX = lm(y~x)
rYX
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     63.8817       0.1111
```

```
coef = as.vector(rYX$coefficients)
(a = coef[1])
```

```
## [1] 63.88172
```

```
(b = coef[2])
```

```
## [1] 0.1111127
```

Realizamos una predicción del número de líneas ADSL para poblaciones de 240000 y 300000 habitantes.

```
new = data.frame(x = c(240000,300000))
predict(rYX,new)
```

```
##        1        2
## 26730.93 33397.69
```

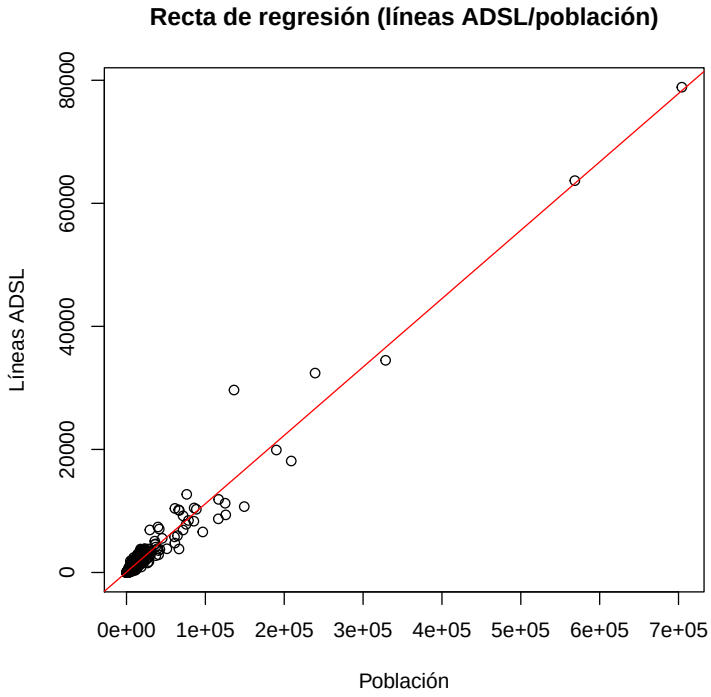
O bien

```
predict(rYX,newdata=data.frame(x = c(240000,300000)))
```

```
##        1        2
## 26730.93 33397.69
```

Representamos el diagrama de dispersión y la recta de regresión.

```
plot(x,y,main="Recta de regresión (líneas ADSL/población)",xlab="Población",ylab="Líneas ADSL")
abline(rYX,col='red')
```



Obtenemos el coeficiente de correlación lineal y el coeficiente de determinación.

```
cor(x,y)
```

```
## [1] 0.9817457
```

```
(coef.det = cor(x,y)^2)
```

```
## [1] 0.9638246
```

De otra forma:

```
summary(rYX)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5946.1  -119.9   -75.3    31.7 14443.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.388e+01  3.267e+01   1.955   0.0509 .
## x           1.111e-01  7.829e-04 141.923   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 867.5 on 756 degrees of freedom
## Multiple R-squared:  0.9638, Adjusted R-squared:  0.9638
## F-statistic: 2.014e+04 on 1 and 756 DF,  p-value: < 2.2e-16
```

```
summary(rYX)$r.squared
```

```
## [1] 0.9638246
```

1.8 Recta de regresión Población / Líneas ADSL

Obtenemos la recta de regresión.

```
rXY = lm(x~y)
rXY
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Coefficients:
## (Intercept)            y
##    -155.250        8.674
```

```
coef.prim = as.vector(rXY$coefficients)
(a.prim = coef.prim[1])
```

```
## [1] -155.25
```

```
(b.prim = coef.prim[2])
```

```
## [1] 8.674298
```

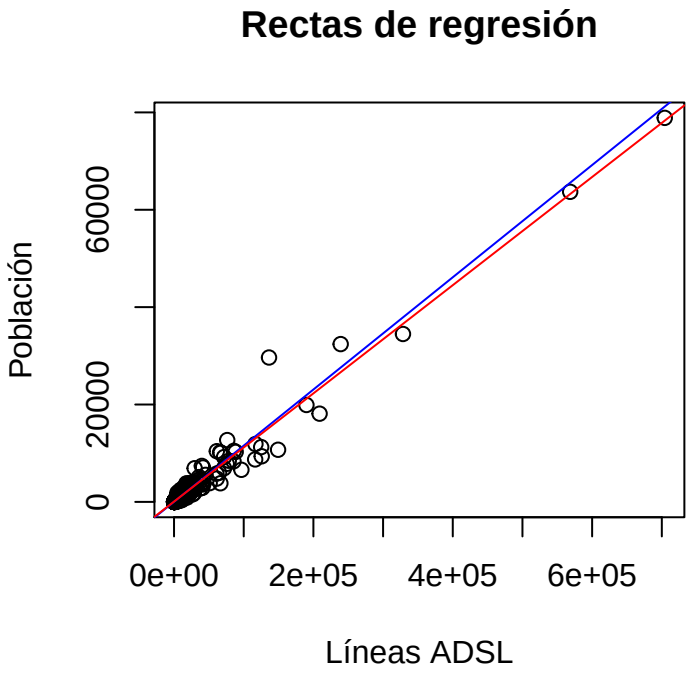
Realizamos una predicción de la población en municipios que poseen 6000, 7000 y 9000 líneas ADSL.

```
new = data.frame(y = c(6000,7000,9000))
predict(rXY,new)
```

```
##        1        2        3
## 51890.54 60564.84 77913.43
```

Representamos el diagrama de dispersión y las dos rectas de regresión.

```
plot(x,y,main="Rectas de regresión",xlab="Líneas ADSL",ylab="Población")
abline(a=-a.prim/b.prim,b=1/b.prim,col='blue')
abline(rYX,col="red")
```



2 Series temporales

2.1 Lectura de datos

Utilizaremos el fichero **ParoRegistrado.csv**, que contiene las cifras mensuales de paro registrado en España desde octubre de 1997 hasta septiembre de 2002. Se trata de un fichero de texto con campos separados por punto y coma, que utiliza como separador decimal la coma. Para su lectura utilizamos el comando `read.csv2`.

```
datos = read.csv2("ParoRegistrado.csv")
str(datos)
```

```
## 'data.frame':    60 obs. of  3 variables:
##  $ Anno : int  1997 1997 1997 1998 1998 1998 1998 1998 1998 1998 ...
##  $ Mes  : int  10 11 12 1 2 3 4 5 6 7 ...
##  $ Valor: int  2072858 2093888 2075659 2091329 2067830 2039130 1967980 1902166 1860627 1786051 ...
```

```
head(datos,n=12)
```

```
##    Anno Mes   Valor
## 1  1997  10 2072858
## 2  1997  11 2093888
## 3  1997  12 2075659
## 4  1998   1 2091329
```

```
## 5  1998   2 2067830
## 6  1998   3 2039130
## 7  1998   4 1967980
## 8  1998   5 1902166
## 9  1998   6 1860627
## 10 1998   7 1786051
## 11 1998   8 1777134
## 12 1998   9 1788415
```

2.2 Creación de la serie temporal

Los valores de la serie se encuentran en el campo **Valor**. Los campos **Anno** y **Mes** contienen, respectivamente, el año y el mes a que corresponde cada valor. Construimos la serie temporal con el comando `ts`.

```
r=12
serie = ts(data=datos$Valor,frequency=r,start=c(datos$Anno[1],datos$Mes[1]))
serie
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 1997
## 1998 2091329 2067830 2039130 1967980 1902166 1860627 1786051 1777134
## 1999 1804234 1783940 1757159 1708000 1649120 1612494 1550958 1554459
## 2000 1670578 1659820 1628535 1578858 1531169 1500147 1488785 1487606
## 2001 1620699 1598920 1578456 1535090 1478133 1460586 1451469 1459007
## 2002 1651728 1666049 1649046 1636268 1588987 1567390 1548449 1552002
##          Sep     Oct     Nov     Dec
## 1997         2072858 2093888 2075659
## 1998 1788415 1803692 1804518 1785692
## 1999 1569978 1591689 1623676 1613750
## 2000 1501442 1530143 1556879 1556382
## 2001 1488551 1540003 1572847 1574844
## 2002 1590264
```

2.3 Descomposición de la serie en componentes

Se asume un modelo multiplicativo.

```
componentes = decompose(serie,type="multiplicative")
str(componentes)
```

```
## List of 6
##  $ x       : Time-Series [1:60] from 1998 to 2003: 2072858 2093888 2075659 2091329 2067830 2039130 19
##  $ seasonal: Time-Series [1:60] from 1998 to 2003: 0.995 1.014 1.013 1.051 1.048 ...
##  $ trend   : Time-Series [1:60] from 1998 to 2003: NA NA NA NA NA ...
##  $ random  : Time-Series [1:60] from 1998 to 2003: NA NA NA NA NA ...
##  $ figure  : num [1:12] 0.995 1.014 1.013 1.051 1.048 ...
##  $ type    : chr "multiplicative"
##  - attr(*, "class")= chr "decomposed.ts"
```

2.4 Obtención de la medias móviles centradas

```
mmcent = componentes$trend
mmcent
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 1997
## 1998      NA      NA      NA 1949040 1925768 1901629 1877585 1853794
## 1999 1745242 1726168 1707789 1689854 1673485 1658786 1646052 1635312
## 2000 1586878 1581501 1575860 1570440 1565093 1559919 1555450 1550835
## 2001 1531205 1528459 1526730 1526604 1527680 1529114 1531177 1535267
## 2002 1574557 1582472 1590585      NA      NA      NA      NA      NA
##          Sep     Oct     Nov     Dec
## 1997              NA      NA      NA
## 1998 1830216 1807635 1786259 1765376
## 1999 1624781 1614041 1603745 1594149
## 2000 1546210 1542300 1538267 1534408
## 2001 1541005 1548162 1556997 1566066
## 2002      NA
```

2.5 Obtención de los índices de variación estacional

```
varest1 = componentes$figure
varest1
```

```
##  [1] 0.9951100 1.0136828 1.0133820 1.0510420 1.0480092 1.0357461 1.0102859
##  [8] 0.9821433 0.9691924 0.9519417 0.9570041 0.9724604
```

2.6 Reordenación de los índices para empezar desde enero

En el vector de índices de variación estacional aparece en la primera posición el que corresponde al primer mes de la serie, que es octubre. Por tanto, el índice correspondiente a enero se encuentra en la posición 4 del vector. Los índices reordenados se utilizarán al realizar predicciones, para añadir la estacionalidad a la predicción de la tendencia.

```
varest1_ord = varest1[c(4,5,6,7,8,9,10,11,12,1,2,3)]
varest1_ord
```

```
##  [1] 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
##  [8] 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
```

O bien:

```
varest1_ord = varest1[c(4:12,1:3)]
varest1_ord
```

```
##  [1] 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
##  [8] 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
```

2.7 Obtención de los índices de variación estacional repetidos para cada año

```
varest2 = componentes$seasonal
varest2
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 1997
## 1998 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
## 1999 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
```

```
## 2000 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
## 2001 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
## 2002 1.0510420 1.0480092 1.0357461 1.0102859 0.9821433 0.9691924 0.9519417
##            Aug       Sep       Oct       Nov       Dec
## 1997                     0.9951100 1.0136828 1.0133820
## 1998 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
## 1999 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
## 2000 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
## 2001 0.9570041 0.9724604 0.9951100 1.0136828 1.0133820
## 2002 0.9570041 0.9724604
```

2.8 Desestacionalización la serie

```
serie_des = serie/varest2
serie_des
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 1997
## 1998 1989767 1973103 1968755 1947944 1936750 1919770 1876219 1856976
## 1999 1716615 1702218 1696515 1690611 1679103 1663750 1629257 1624297
## 2000 1589449 1583784 1572330 1562783 1559008 1547832 1563946 1554441
## 2001 1541993 1525674 1523980 1519461 1505007 1507013 1524746 1524557
## 2002 1571515 1589727 1592133 1619609 1617877 1617212 1626622 1621730
##          Sep     Oct     Nov     Dec
## 1997         2083044 2065625 2048249
## 1998 1839062 1812555 1780160 1762111
## 1999 1614439 1599511 1601759 1592440
## 2000 1543962 1537662 1535864 1535830
## 2001 1530706 1547571 1551617 1554048
## 2002 1635299
```

2.9 Obtención de los tiempos en que ha sido medida la serie

```
Time = time(serie_des)
Time
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 1997
## 1998 1998.000 1998.083 1998.167 1998.250 1998.333 1998.417 1998.500
## 1999 1999.000 1999.083 1999.167 1999.250 1999.333 1999.417 1999.500
## 2000 2000.000 2000.083 2000.167 2000.250 2000.333 2000.417 2000.500
## 2001 2001.000 2001.083 2001.167 2001.250 2001.333 2001.417 2001.500
## 2002 2002.000 2002.083 2002.167 2002.250 2002.333 2002.417 2002.500
##           Aug      Sep      Oct      Nov      Dec
## 1997                    1997.750 1997.833 1997.917
## 1998 1998.583 1998.667 1998.750 1998.833 1998.917
## 1999 1999.583 1999.667 1999.750 1999.833 1999.917
## 2000 2000.583 2000.667 2000.750 2000.833 2000.917
## 2001 2001.583 2001.667 2001.750 2001.833 2001.917
## 2002 2002.583 2002.667
```

Se observa que la codificación viene dada por $año + \frac{mes-1}{12} = año + \frac{mes-1}{r}$.

2.10 Obtención de la tendencia secular

```
rTendT = lm(serie_des~Time)                 # recta de regresión
rTendT
```

```
##
## Call:
## lm(formula = serie_des ~ Time)
##
## Coefficients:
## (Intercept)         Time
##   181333665       -89822
```

```
coef = as.vector(rTendT$coefficients)      # extraer los coeficientes
coef
```

```
## [1] 181333665.17    -89822.43
```

```
(a = coef[1])
```

```
## [1] 181333665
```

```
(b = coef[2])
```

```
## [1] -89822.43
```

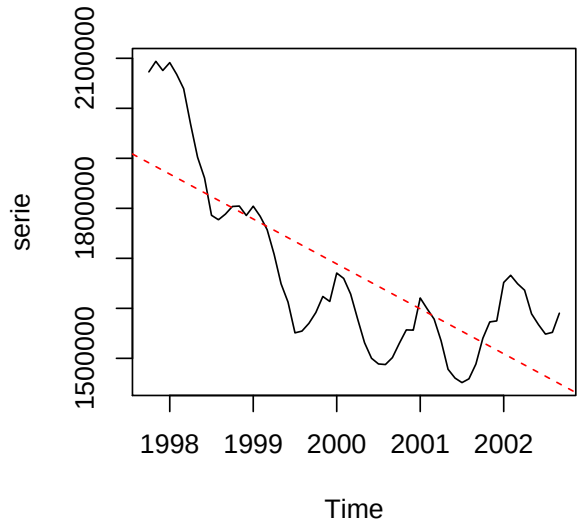
```
summary(rTendT)$r.squared
```

```
## [1] 0.6424844
```

2.11 Representación gráfica de la serie

Representamos en el mismo gráfico la serie temporal y la recta de tendencia.

```
plot(serie,type="l",ylim=c(min(serie),max(serie)))
abline(rTendT,lty=2,col=2)
```



2.12 Predicciones

Realizamos una predicción del valor de la serie temporal para el mes de marzo de 2003. Por tanto, la estación sería 3 y el instante de tiempo 2003+(3-1)/12.

```
frec = 3
(t = 2003+(frec-1)/r)                          # Predicción para marzo de 2003
```

```
## [1] 2003.167
```

```
new = data.frame(Time = t)                     # Calculamos la tendencia secular
(tsecular =predict(rTendT,new))
```

```
##       1
## 1404369
```

```
(prediccion = tsecular*varest1_ord[frec])      # Predicción
```

```
##       1
## 1454569
```

2.13 Resolución a través de la codificación temporal utilizada en teoría

En el tema 5, lo habitual era codificar el tiempo con números enteros. Así, el cálculo del modelo de regresión para la obtención de la tendencia secular, el gráfico y las predicciones se obtendría como sigue:

```
Time = 1:length(datos$Valor)
Time
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
## [47] 47 48 49 50 51 52 53 54 55 56 57 58 59 60
```

```
rTendT = lm(serie_des~Time)                    # recta de regresión
rTendT
```

```
##
## Call:
## lm(formula = serie_des ~ Time)
##
## Coefficients:
## (Intercept)         Time
##     1898392        -7485
```

```
coef = as.vector(rTendT$coefficients)          # extraer los coeficientes
coef
```

```
## [1] 1898391.877   -7485.202
```

```
(a = coef[1])
```

```
## [1] 1898392
```

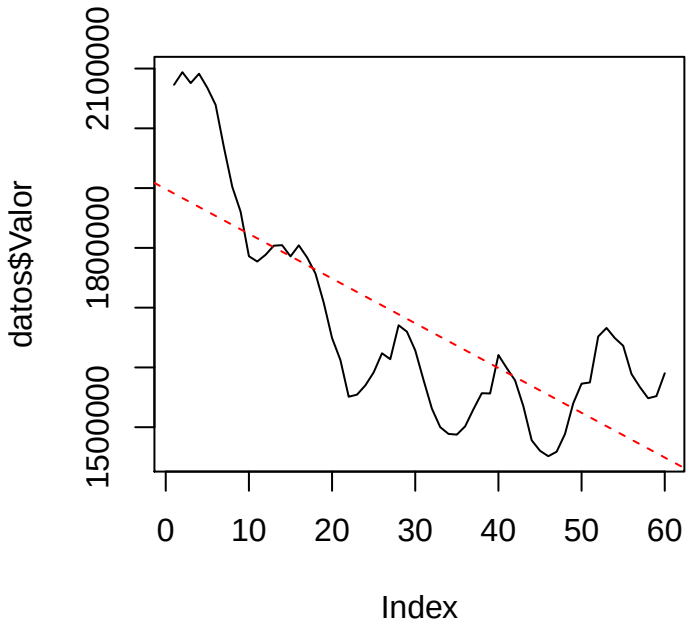
```
(b = coef[2])
```

```
## [1] -7485.202
```

```
summary(rTendT)$r.squared
```

```
## [1] 0.6424844
```

```
plot(datos$Valor,type="l",ylim=c(min(serie),max(serie)))
abline(rTendT,lty=2,col=2)
```




```
frec = 3
t = 66                                  # Predicción para marzo de 2003
new = data.frame(Time = t)              # Calculamos la tendencia secular
(tsecular =predict(rTendT,new))
```

```
##       1
## 1404369
```

```
(prediccion = tsecular*varest1_ord[frec])   # Predicción
```

```
##       1
## 1454569
```

Se puede comprobar que, aunque cambian los coeficientes de la recta de regresión, los resultados de la predicción son los mismos.