



R: Un Ambiente y Lenguaje para el Cálculo y la Graficación Estadística

Gabriel Nuñez Antonio
Ernesto Barrios Zamudio

Seminario Aleatorio del Departamento de Estadística, ITAM, 2007

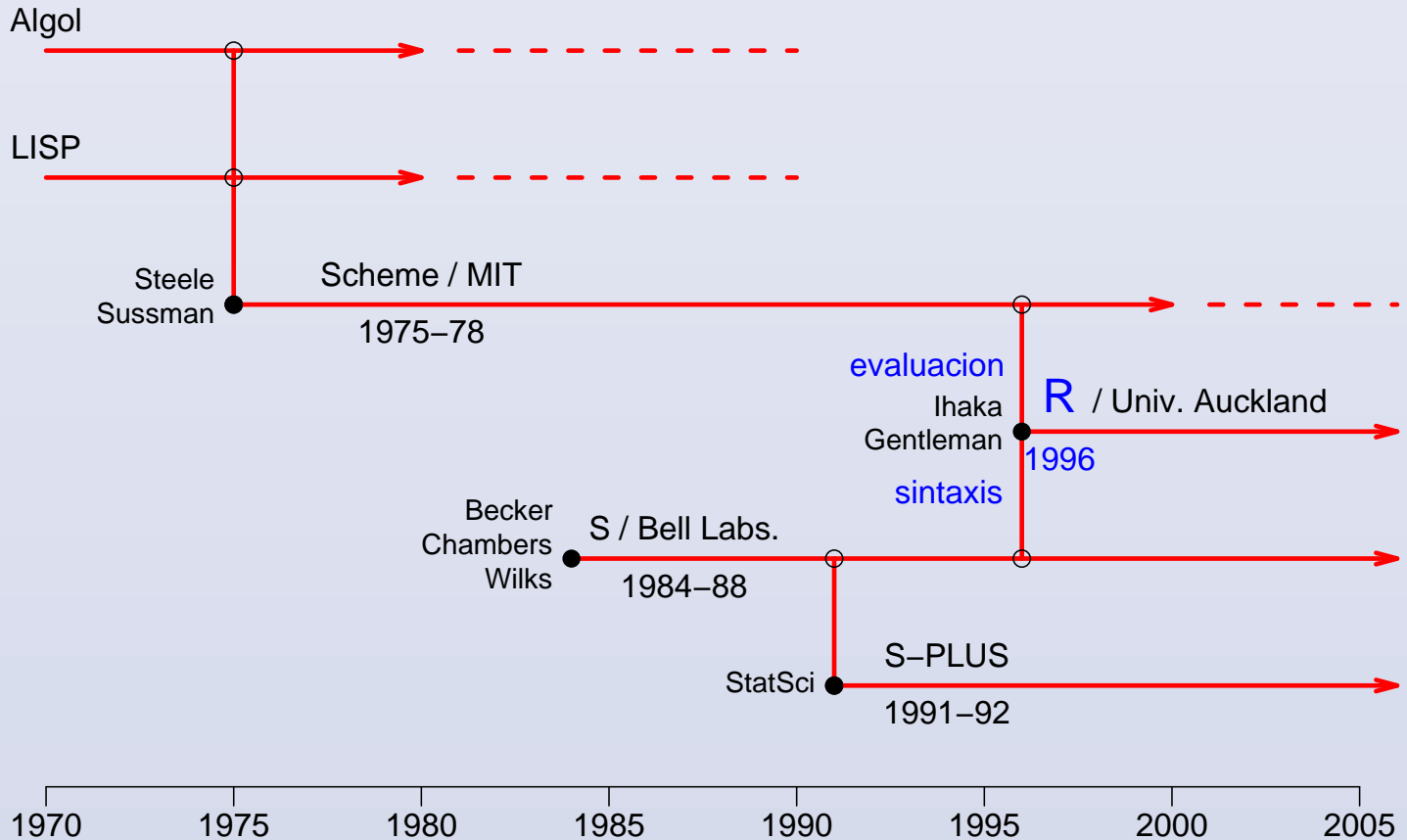


Contenido

- Historia.
- Introducción.
- Manipulación de Datos.
- Gráficos.
- Análisis Estadístico.



Genealogia de R





R: A Language for Data Analysis and Graphics

ROSS IHAKA and Robert GENTLEMAN

In this article we discuss our experience designing and implementing a statistical computing language. In developing this new language, we sought to combine what we felt were useful features from two existing computer languages. We feel that the new language provides advantages in the areas of portability, computational efficiency, memory management, and scoping.

Key Words: Computer language; Statistical computing.

Ross Ihaka is Senior Lecturer, and Robert Gentleman is Senior Lecturer, Department of Statistics, University of Auckland, Private Bag 92019, Auckland, New Zealand. e-mail: ihaka@stat.auckland.ac.nz.

©1996 American Statistical Association, Institute of Mathematical Statistics,
and Interface Foundation of North America

Journal of Computational and Graphical Statistics, Volume 5, Number 3, Pages 299–314

Estatutos de “*The R Foundation for Statistical Computing*”



1 Nombre, Lugar y Campo de Actividad

- a) La organización es llamada “*The R Foundation for Statistical Computing*”, abreviado “Fundación R”, que se usará en este documento.

2 Objetivos

1. Fundamentales

- (b) La “Fundación R” es una organización no lucrativa que trabaja por el interés público.

2. Los objetivos de la “Fundación R” son:

- (a) El avance del proyecto R del cálculo estadístico para proveer de un ambiente para el análisis de datos y graficación gratis y de código abierto.

3 Medios para cumplir los objetivos

1. Para cumplir estos objetivos la organización especialmente:

- (a) Apoyará en el desarrollo de R y proyectos de código abierto relacionados.

Miembros y Benefactores de R



The R Foundation - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

The R Project for Statistical Computing The R Foundation The R Foundation

Pascal Heus (USA)
Paul Hewson (UK)
Anja von Heydebreck (Germany)
Giles Heywood (UK)
Eugène Horber (Switzerland)
Torsten Hothorn (Germany)
Wolfgang Huber (UK)
Brad Hunting (USA)
Johannes Husing (Germany)
Joe Ibrahim (USA)
Rafael Irizarry (USA)
Andrew Irwin (Canada)
Simon Jackman (USA)
David A. James (USA)
Paulo Justiniano (Brasil)
Landon Jensen (USA)

Klemens Vierlinger (Austria)
Scott R. Waichler (USA)
Rainer Walke (Germany)
Mario Walter (Germany)
Ko-Kang Kevin Wang (New Zealand)
Harald Weedon-Fekjær (Norway)
Bernd Weiß (Germany)
Stefan Werner (Finland)
Brandon Whitcher (UK)
Matthew Wilkes (UK)
Kevin Wright (USA)
Zhijin (Jean) Wu (USA)
Jean Yee Hwa Yang (Australia)
Jianhua Zhang (USA)
Victor D. Zurkowski (Canada)

Ordinary Members

Douglas Bates (USA)
Roger Bivand (Norway)
John Chambers (USA)
Peter Dalgaard (Denmark)
John Fox (Canada)
Robert Gentleman (USA)
Kurt Hornik (Austria)
Stefano Iacus (Italy)
Ross Ihaka (New Zealand)
Friedrich Leisch (Austria)

Thomas Lumley (USA)
Martin Machler (Switzerland)
Duncan Murdoch (Canada)
Paul Murrell (New Zealand)
Martyn Plummer (France)
Brian Ripley (UK)
Duncan Temple Lang (USA)
Luke Tierney (USA)
Simon Urbanek (Germany)
Bill Venables (Australia)

Miembros y Benefactores de R



The R Foundation - Mozilla Firefox

The R Project for Statistical Computing | The R Foundation | The R Foundation

R Foundation Members & Supporters

- [Benefactors, Institutions & Donors](#)
- [Supporting Members](#)
- [Ordinary Members](#)

Benefactors	Donors
<ul style="list-style-type: none">• Burns Statistics Ltd., London, U.K.• Department of Statistics, Brigham Young University, Utah, USA• Institute of Mathematical Statistics (IMS), Ohio, USA• Loyalty Matrix Inc., California, USA• Mango Solutions, Chippenham, UK• Marc Schwartz, USA• Merck and Co., Inc., USA• Numbers Internation Pty Ltd, Australia• Statisticon AB, Uppsala, Sweden	<ul style="list-style-type: none">• Adelchi Azzalini (Italy)• BC Cancer Agency, Vancouver (Canada)• Biostatistics and Research Decision Sciences, Merck Research Laboratories (USA)• Brian Caffo (USA)• David W. Crawford (USA)• Department of Economics, University of Milano (Italy)• Dipartimento di Scienze Statistiche e Matematiche di Palermo (Italy)• Emanuele De Rinaldis (Italy)• Faculty of Economics, University of Groningen (Netherlands)• Peter L. Flom (USA)• Manfred Georg (USA)• Google Inc., Mountain View, California (USA)• Hort Research Institute (New Zealand)• Peeter Luikmel (Estland)• Vittorio de Martino (Italy)• Shigeru Mase (Japan)• Merck and Co., Inc. (USA)• Telecom New Zealand (New Zealand)
<h3>Supporting Institutions</h3> <ul style="list-style-type: none">• Astra Zeneca R&D Mölndal, Mölndal, Sweden• Baxter AG, Vienna, Austria• Baxter Healthcare Corp., California, USA• Boehringer Ingelheim Austria GmbH, Vienna, Austria• Breast Center at Baylor College of Medicine, Houston, Texas, USA• Center für digitale Systeme, Freie Universität Berlin, Germany• Dana-Farber Cancer Institute, Boston, USA• Department of Biostatistics, Johns Hopkins University, Maryland, USA• Department of Biostatistics, Vanderbilt University School of Medicine, USA• Department of Economics, Stockholm University, Sweden	



Introducción

- R ofrece una gran cantidad de funciones para realizar análisis gráfico y estadístico.
- ¿Cómo trabaja R?
- R es un lenguaje orientado a objetos. Es un intérprete no un compilador.
- Una vez que se abre R aparece el prompt de default “>”, lo que indica que R espera algún comando.



Introducción

- El nombre de un objeto debe empezar con una letra (A-Z y a-z) y puede incluir dígitos y puntos.
- R discrimina para el nombre de los objetos letras mayúsculas de minúsculas, por lo que **x** y **X** nombrarán a diferentes objetos.
- En R para ejecutar una función, ésta siempre se debe escribir con paréntesis aunque no haya nada dentro de ellos. Por ejemplo:

```
ls()
```

desplegará el contenido del directorio de trabajo actual.

- Los argumentos de una función pueden ser en si objetos (datos, fórmulas, matrices, tablas, etc.)



Creando Objetos

- La forma de asignar objetos en R es a través del simbolo $<-$. Por ejemplo:

```
> x<- 56
```

```
> X<- 23
```

```
>x;X
```

```
[1] 56 [1] 23
```

```
>
```

```
> n<- sqrt(X)
```

```
> n
```

```
[1] 4.795832
```

```
> m<- 3+n
```

```
> m
```

```
[1] 7.795832
```

```
> m.aux<-10*n
```

```
> m.aux
```



Ayudas

- R ofrece ayuda en línea a través de la función `help()`.

```
>help(rm)
```

```
>?rm
```

- R hace también búsquedas inteligentes

```
>help.search("rose diagram")
```

```
Help files with alias or concept or title matching 'rose diagram'  
using fuzzy matching:
```

```
rose.diag(CircStats)      Rose Diagram
```

```
rose.diag(circular)      Rose Diagram
```

```
Type 'help(FOO, package = PKG)' to inspect entry 'FOO(PKG) TITLE'.
```

- Adicionalmente, se pueden hacer búsquedas en el sitio web de R.

```
>RSiteSearch("rose diagram")
```



Lectura de Datos

- Para leer datos desde un archivo se pueden utilizar las funciones `read.table()` o `scan()`. Esta última es más flexible y permite especificar el tipo de cada variable.

```
> mydata<-scan(file="./NMV.dat2", what=list("",0,0))
```

```
Read 15 records
```

```
> mydata
```

```
[[1]]  
[1] "18.61664" "19.43575" "20.20695" "21.84337" "21.34864"  
[[2]]  
[1] 20.48832 17.99986 21.38629 17.97225 22.99391  
[[3]]  
[1]18.70510 20.18638 21.75702 22.66418 20.04545
```

- Se puede notar que la primera variables es de tipo caracter y las otras dos de tipo numérico



Generación de Datos

Secuencias Regulares

- Algunas susesiones se pueden generar de la siguiente manera:

```
> x<-1:15
```

```
> x
```

```
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

```
> y<-seq(1,5,0.5)
```

```
> y
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
> w<-seq(length=9,from=1,to=5)
```

```
> w
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```



Generación de Variables Aleatorias

- Generando observaciones de variable aleatorias

```
> x<-rnorm(1000)
> y<-rnorm(1000,5,3)
> u<-runif(2000,-1,1)
> z<-rexp(1000,2)
```

- No sólo se pueden generar observaciones de variable aleatorias. También se puede obtener la densidad, la probabilidad acumulada y los cuantiles de la correspondiente variable.

```
dnorm(x, mean=0, sd=1)
pnorm(q, mean=0, sd=1)
qnorm(p, mean=0, sd=1)
rnorm(n, mean=0, sd=1)
```

Manipulación de vectores y matrices



- La forma más común de crear vectores es con la función `c()`.

```
> x<-c(1,2,3,4,5,6)
> x
[1] 1 2 3 4 5 6
> y<-c(6,7)
> z<-c(y,x,y)
> z
[1] 6 7 1 2 3 4 5 6 6 7
```

- Las operaciones aritméticas (+, -, *, /, %, etc.) entre vectores se realizan elemento a elemento.

```
> X<-c(10,11,12,100,-5,-6)
> x*X
[1] 10 22 36 400 -25 -36
> X+1
[1] 11 12 13 101 -4 -5
```

Manipulación de vectores y matrices



- Las matrices pueden ser creadas a partir de vectores o directamente usando la función `matrix()`.

```
> matrix(data=5,nrow=2,ncol=3)
```

```
      [,1] [,2] [,3]  
[1,]    5    5    5  
[2,]    5    5    5
```

```
> matrix(0,ncol=3,nrow=2)
```

```
      [,1] [,2] [,3]  
[1,]    0    0    0  
[2,]    0    0    0
```

```
> matrix(1:6,2,3)
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
> matrix(1:6,2,3,byrow=T)
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6
```


Manipulación de vectores y matrices



* Al igual que con los vectores las operaciones aritméticas se realizan elemento a elemento.

```
> m1+m2
```

```
      [,1] [,2]
[1,]    3    3
[2,]    3    3
```

```
> m1*m2
```

```
      [,1] [,2]
[1,]    2    2
[2,]    2    2
```

```
> m1/m2
```

```
      [,1] [,2]
[1,]  0.5  0.5
[2,]  0.5  0.5
```

```
> m1<-matrix(1,2,2)
> m1
```

```
      [,1] [,2]
[1,]    1    1
[2,]    1    1
```

```
> m1<-matrix(2,2,2)
> m2
```

```
      [,1] [,2]
[1,]    2    2
[2,]    2    2
```



Funciones

- R tiene funciones especiales para matrices, por ejemplo `solve()` para invertir, `qr()` para la descomposición QR, `eigen()` para obtener los eigenvalores y eigenvectores, `svd()` para obtener la descomposición de valor singular, etc.
- En R uno puede encontrar:
 - Funciones matemáticas básicas: `log`, `exp`, `log10`, `log2`, `sin`, `cos`, `tan`, `asin`, `acos`, `abs`, `sqrt`, etc.
 - Funciones especiales: `gamma`, `digamma`, `beta`, `besselI`, ...
 - Funciones estadísticas: `mean`, `median`, `lm`,...
 - Algunas otras funciones como: `sum(x)`=suma de los elementos de `x`, `max(x)`, `min(x)`, `wich`, `wich.max`, etc.



Creando tus propias funciones

- Una función en **R** puede tener cualquier número de argumentos y las operaciones que esta realice pueden ser producto de expresiones en **R**.
- La sintaxis general para la definición de una función es:

```
function(arguments){expression}
```

donde **arguments** son los argumentos de la función separados por comas y **expression** es cualquier estructura permitida en **R**. El valor de la última línea dentro de la estructura **expression** será el valor que retorne la función.



Creando tus propias funciones

Ejemplo 1.

- La siguiente función retorna la suma de los cuadrados de los elementos del vector **x**.

```
> myfunction<-function(x){ sum(x*x) }
```

- La función **myfunction** ahora puede ser usada de la siguiente manera:

```
> z<-1:50
```

```
> y<-myfunction(z)
```

```
> y
```

```
[1] 42925
```



Creando tus propias funciones

Ejemplo 2.

- La siguiente función realiza cierta operación en cada punto de una malla (“grid”) de valores.

```
> grid.calc<-function(x,y){
+     # Esta funcion calcula sqrt(x*x+y*y)
+     # en cada punto del grid definido por x y y.
grid<-matrix(0,length(x),length(y))
        # Define la matriz para almacenar los resultados.
+   for(i in 1:length(x)){
      for(j in 1:length(y))
        grid[i,j]<-sqrt(x[i]*x[i]+y[j]*y[j])
+   }
grid
}
```



Graficando con R

- R ofrece una gran variedad de gráficos, aunado a la posibilidad y flexibilidad de crearlos y personalizarlos.
- Para tener una idea de las gráficas que ofrece R se puede ejecutar el siguiente comando:

```
> demo(graphics)
```

- Sería difícil exponer en esta presentación todas las opciones y posibilidades que ofrece R en términos gráficos. De manera particular, cada función gráfica tiene un gran número de opciones (argumentos), lo que resulta en una amplia flexibilidad en la construcción de gráficos.

Gráficos en R



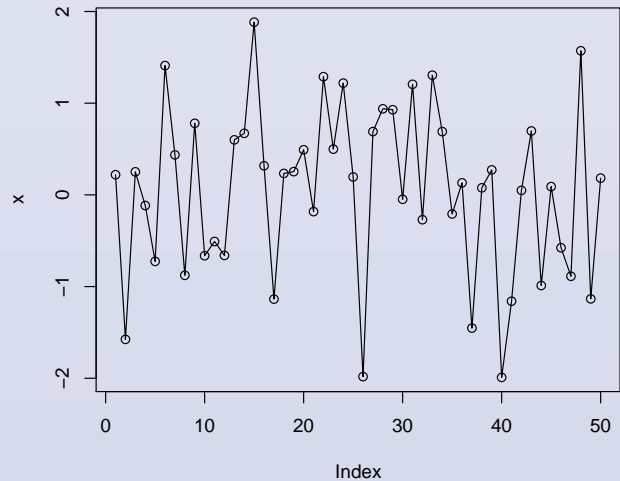
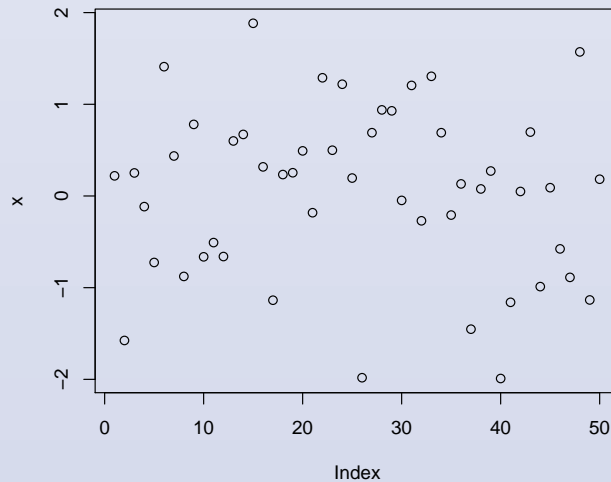
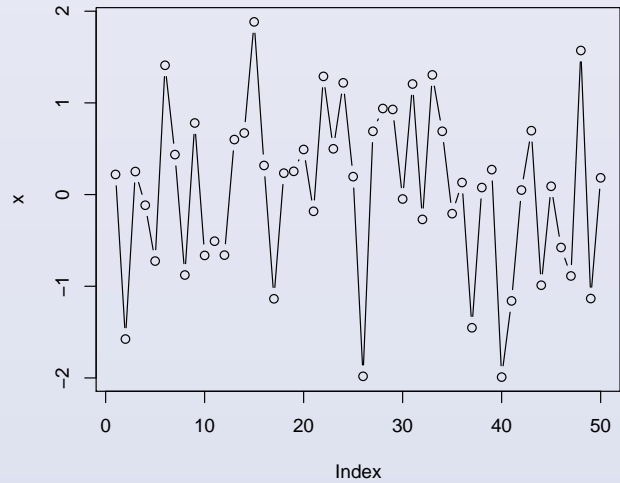
- Notemos la flexibilidad de las funciones gráficas.

```
> x<-rnorm(50)
```

```
> plot(x)
```

```
> plot(x,type="b")
```

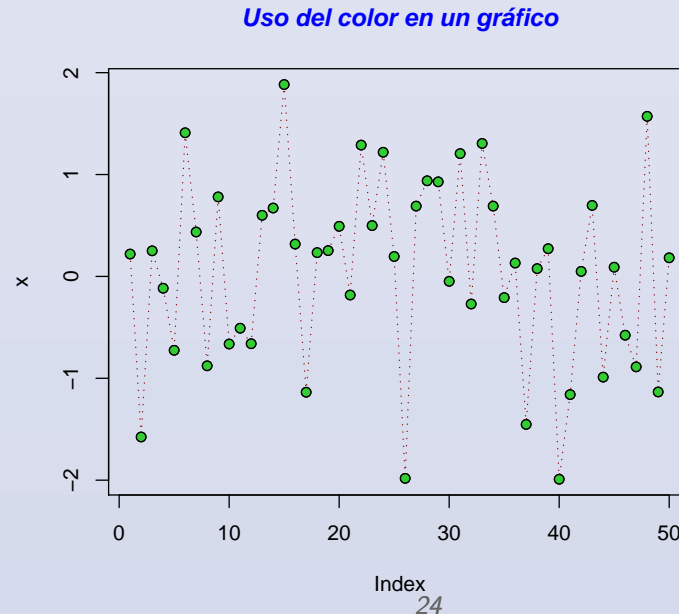
```
> plot(x,type="o")
```





- Uso del color en los gráficos.

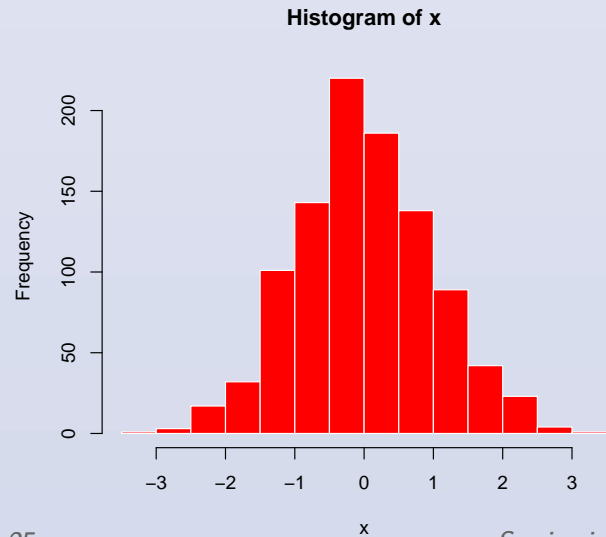
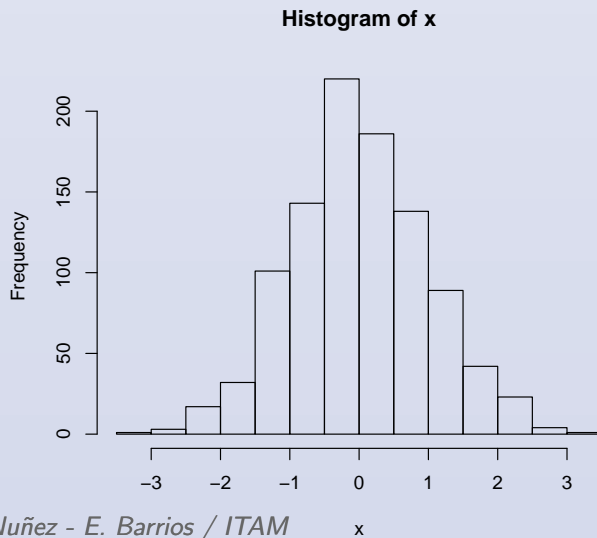
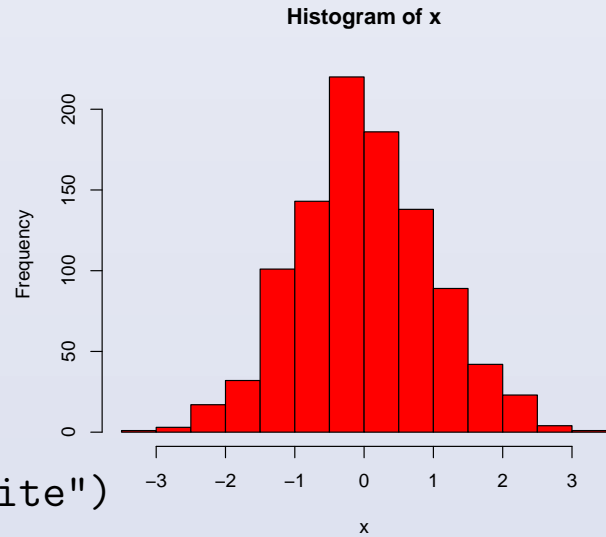
```
>plot(x)
>lines(x, col = "red4", lty = "dotted")
>points(x, bg="limegreen", pch = 21)
>title(main = "Uso del color en un grafico",cex.main =
+ 1.2,font.main = 4,col.main = "blue")
```



Gráficos en R

- Continuemos con la flexibilidad de las funciones gráficas.

```
> x<-rnorm(1000)
>
>
> hist(x)
> hist(x,col="red")
> hist(x,col="red",border="white")
```

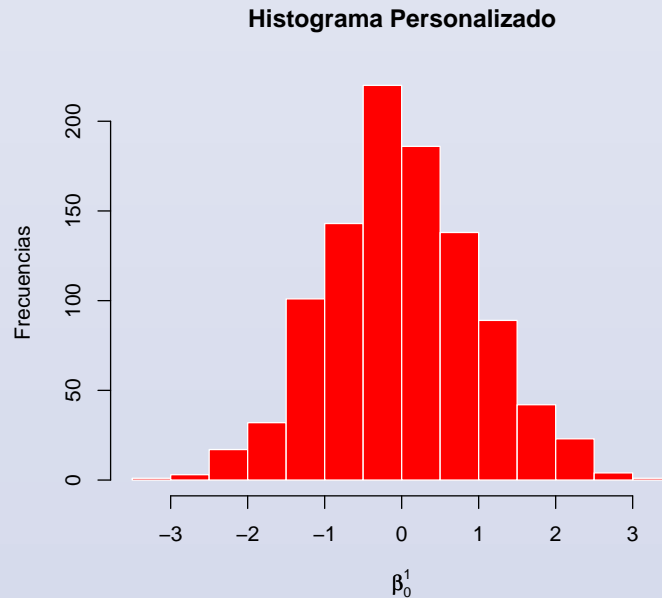


Gráficos en R



- Con los parámetros gráficos `xlab`, `ylab` y `main` uno puede agregar etiquetas a los ejes X , Y , y darle un título al gráfico, respectivamente.

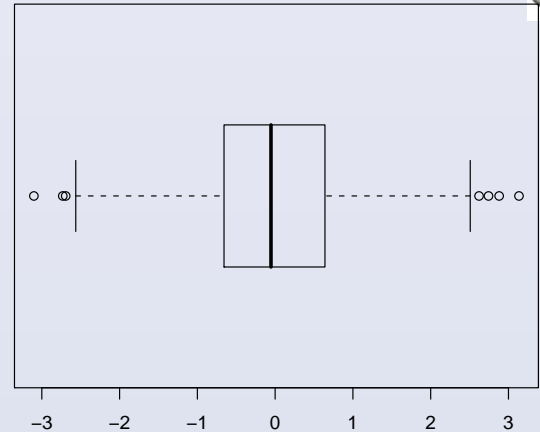
```
> hist(x,col="red",border="white",xlab=expression(beta[0]^1),  
+ ylab="Frecuencias",main="Histograma Personalizado")
```





Más Gráficos

- En R se pueden graficar diagramas de caja y brazo, de tallo y hoja, distribuciones discretas de probabilidad, etc.



```
> boxplot(x, horizontal=T)
> stem(x[1:50])
```

The decimal point is at the |

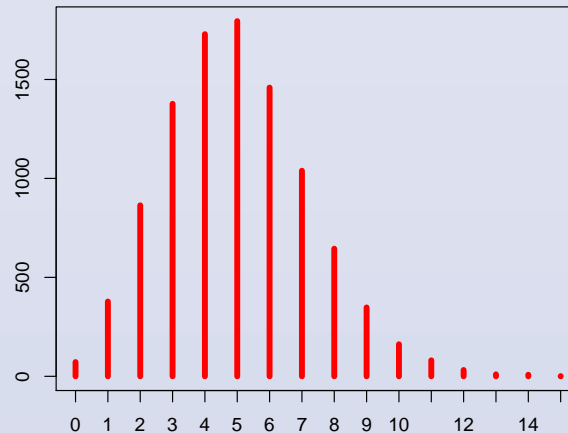
```
-3 | 1
-2 |
-1 | 72110
-0 | 99998544333332222110
 0 | 2222234456678889
 1 | 011356
 2 | 2
```



Más Gráficos

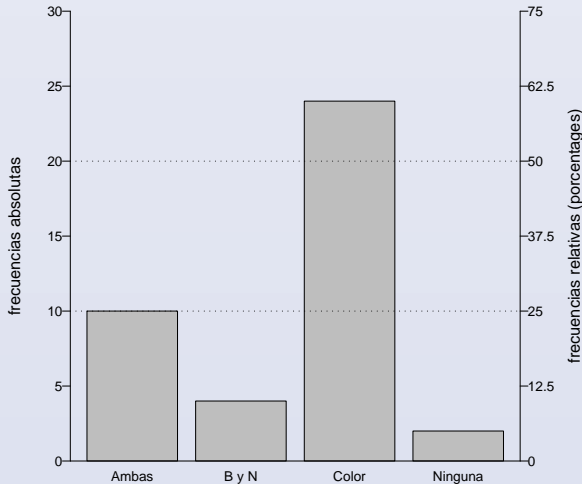
```
> plot(table(rpois(10000,5)), type = "h", col = "red", lwd=5,  
+ main="Distribucion de 10000 Poissones con lambda=5",ylab="")
```

Distribución de 10000 Poissones con lambda=5

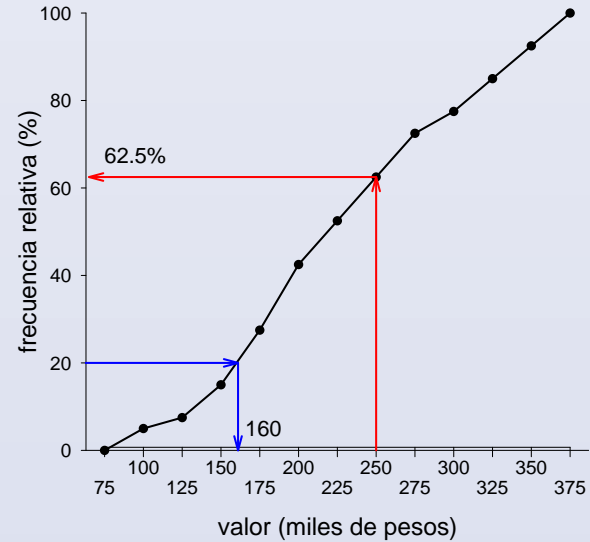


Gráficos Personalizados

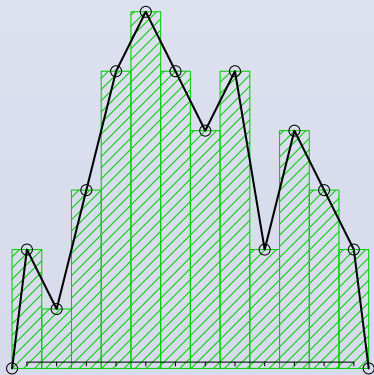
Distribucion de Tipo de Television por Colonia (porcentajes)



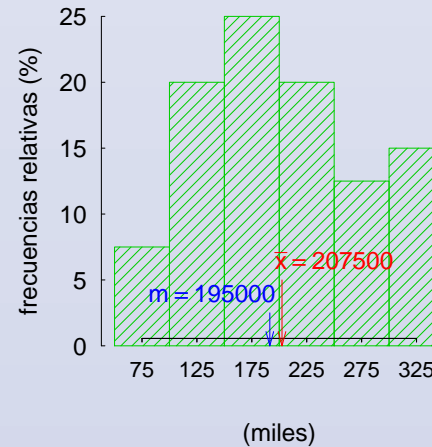
Ojiva de la variable <valor>



Histograma y poligono de frecuencias



Medidas de tendencia central



Análisis Estadísticos usando R



- R no sólo ofrece una gran flexibilidad en la construcción de gráficos, también ofrece una amplia gama de posibilidades para realizar análisis estadísticos (tanto descriptivos como inferenciales). A continuación se muestran sólo algunos ejemplos.

Ejemplo 1

```
>datos<-rnorm(100, 2, 4)
```

```
#Muestra de 100 observaciones normales con media 2 y desv. est.
```

- La función `summary()` calcula algunas estadísticas descriptivas.

```
>summary(datos)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-6.2090  0.2729  2.5220  2.6040  5.0040 11.5000
```



Ejemplo 2

- A continuación se muestra el ajuste de un modelo de **regresión lineal múltiple**.

```
> y=rnorm(10)
> x1=c(1:10)
> x2=c(rep(1,5),rep(0,5))
> x3=rnorm(10)
```

- La función **lm()** es usada para ajustar modelos lineales.

```
>modelo1<-lm(y~x1+x2+x3) # Se define y ajusta el modelo.
> modelo1
Call: lm(formula = y ~ x1 + x2 + x3)
Coefficients: (Intercept)          x1          x2          x3
           2.8520         -0.3599         -1.8483          0.3328
```

★ Observe de qué tipo es el objeto **modelo1**.



Ejemplo 2 (Continuación ...)

- Con la ayuda de la función genérica `summary()` se pueden obtener algunos resultados resumen del ajuste.

```
>summary(modelo1)
```

```
Call: lm(formula = y ~ x1 + x2 + x3)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-0.6709 -0.3335 -0.0218  0.2858  0.7165
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.8520	1.0690	2.668	0.0371 *
x1	-0.3599	0.1305	-2.759	0.0329 *
x2	-1.8483	0.7467	-2.475	0.0481 *
x3	0.3328	0.3616	0.920	0.3930

```
---
```

```
Residual standard error: 0.5758 on 6 degrees of freedom Multiple
R-Squared: 0.5797, Adjusted R-squared: 0.3695 F-statistic: 2.758
on 3 and 6 DF, p-value: 0.1342
```




Ejemplo 2 (Continuación ...)

- Se puede notar que la función `summary()` se puede aplicar tanto a un vector de datos como a un `modelo`. He aquí la flexibilidad de R.
- La tabla de ANOVA para un análisis de regresión múltiple se obtiene usando la función `anova()`

```
> anova(modelo1)
```

```
Analysis of Variance Table
```

```
Response: y
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	0.17703	0.17703	0.5339	0.49250
x2	1	2.28595	2.28595	6.8942	0.03929 *
x3	1	0.28074	0.28074	0.8467	0.39299
Residuals	6	1.98945	0.33158		

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Ejemplo 2 (Continuación ...)

- En realidad `modelo1` es un objeto tipo `lista`, todos sus componentes se pueden obtener con ayuda de la función `names()`.

```
> names(modelo1)
[1] "coefficients" "residuals"    "effects"      "rank"
[5] "fitted.values" "assign"       "qr"           "df.residual"
[9] "xlevels"      "call"        "terms"       "model"
```

- Así , sus componentes se pueden acceder, por ejemplo, con:

```
> modelo1$residuals
      1          2          3          4          5
-0.23464471  1.18946063 -0.05046907  0.17749660 -1.08184345
      6          7          8          9         10
-0.49368395 -1.58228384  2.06572986 -0.11623393  0.12647186
```

Ejemplo 2 (Continuación ...)



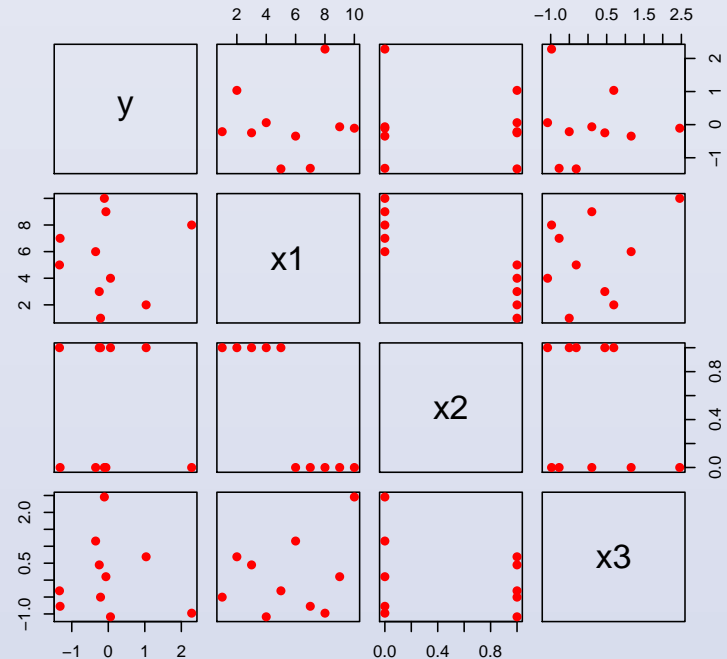
- Se puede notar que la función `plot()` es una función genérica, es decir:

```
>datos=data.frame(y,x1,x2,x3)
```

```
> datos
```

	y	x1	x2	x3
1	-0.21193520	1	1	-0.5032437
2	1.03567146	2	1	0.6893553
3	-0.24656601	3	1	0.4493651
4	0.06023359	4	1	-1.0839041
5	-1.33581988	5	1	-0.3160428
6	-0.34656175	6	0	1.1543616
7	-1.31937170	7	0	-0.7734411
8	2.28289129	8	0	-0.9766752
9	-0.06481415	9	0	0.1010843
10	-0.10739500	10	0	2.4550769

```
>plot(datos)
```

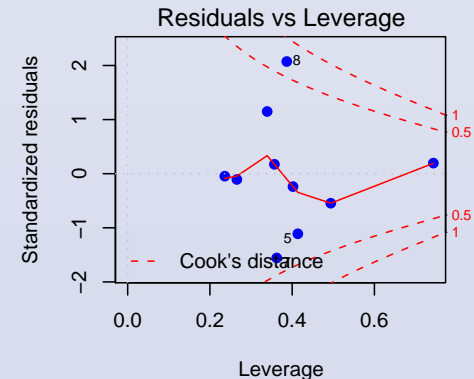
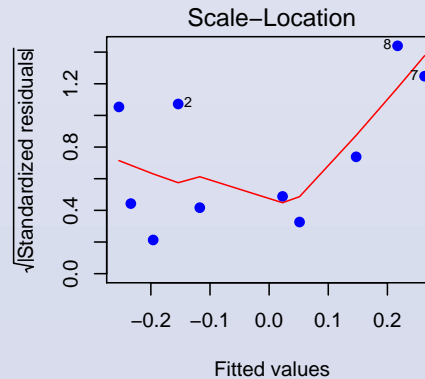
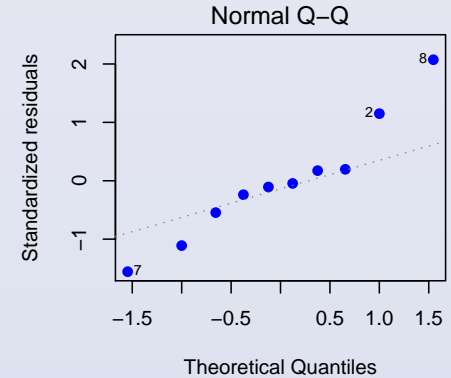
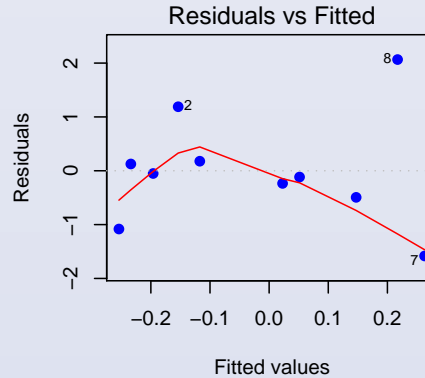




Ejemplo 2 (Continuación ...)

- Ahora, recordando que `modelo1` es un objeto de tipo lista, al que se le asigno un ajuste de regresión lineal, ...

```
>plot(modelo1)
```



Ejemplo 3. Series de Tiempo



Datos, ACF y gráficos

```
presidents      package:datasets      R Documentation
```

```
Quarterly Approval Ratings of US Presidents
```

```
Description:
```

The (approximately) quarterly approval rating for the President of the United States from the first quarter of 1945 to the last quarter of 1974.

```
> presidents
```

	Qtr1	Qtr2	Qtr3	Qtr4
1945	NA	87	82	75
1946	63	50	43	32
1947	35	60	54	55
....
1972	49	61	NA	NA
1973	68	44	40	27
1974	28	25	24	24

```
> plot(presidents,type="o",main="Serie Presidentes")
```

```
> acf(presidents,na.action=na.pass)
```

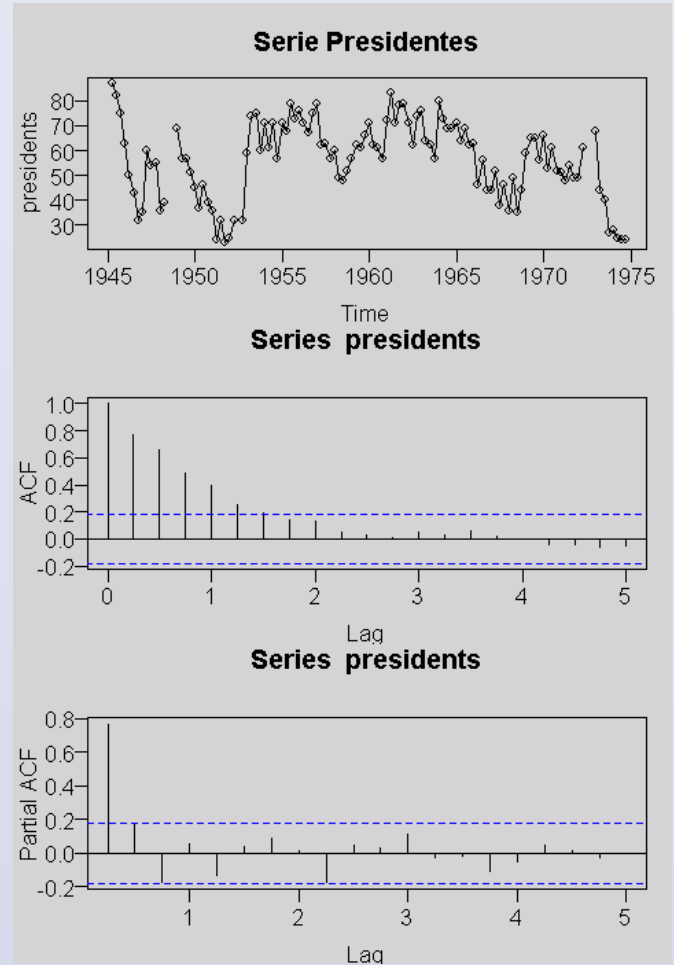
Autocorrelations of series 'presidents', by lag

	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
1.000	0.768	0.660	0.484	0.397	0.250	0.189	0.145	0.131	

```
> pacf(presidents,na.action=na.pass)
```

Partial autocorrelations of series 'presidents', by lag

	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00
0.768	0.171	-0.172	0.055	-0.133	0.039	0.087	0.013	





Series de Tiempo (cont.)

Ajuste y diagnóstico

```
> fit1 <- arima(presidents, c(1, 0, 0))  
> fit1
```

```
Call:  
arima(x = presidents, order = c(1, 0, 0))
```

```
Coefficients:  
      ar1  intercept  
    0.8242  56.1505  
s.e.  0.0555   4.6434
```

```
sigma^2 estimated as 85.47:  
log likelihood = -416.89, aic = 839.78
```

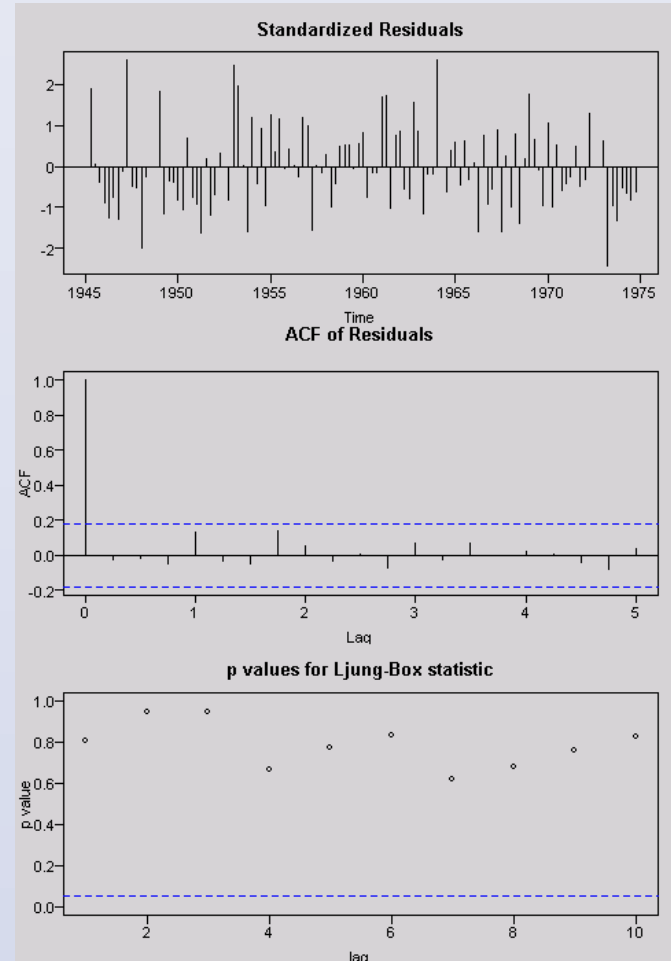
```
> fit3 <- arima(presidents, c(3, 0, 0))  
> fit3
```

```
Call:  
arima(x = presidents, order = c(3, 0, 0))
```

```
Coefficients:  
      ar1      ar2      ar3  intercept  
    0.7496  0.2523 -0.1890   56.2223  
s.e.  0.0936  0.1140  0.0946    4.2845
```

```
sigma^2 estimated as 81.12:  
log likelihood = -414.08, aic = 838.16
```

```
> tsdiag(fit3)
```





- Pruebas de hipótesis, paramétricas y no paramétricas:

```
{ t.test(), var.test(), cor()  
  wilcox.test(), cor.test(datos,method="spearman")  
  binom.test(), kruskal.test()  
  :  
}
```

- Análisis y ajuste de modelos:

```
{ lm()           Modelos de regresión  
  glm()         Modelos lineales generalizados  
  survfit(), coxph() Análisis de supervivencia  
  prcomp()      Análisis multivariado  
  hclust(), plclust  
  density()     Estimadores de kernel  
  ts.plot(), acf() Series de tiempo  
  :  
}
```

Paquetes Adicionales (Contribuciones)



Grafica de Datos Circulares

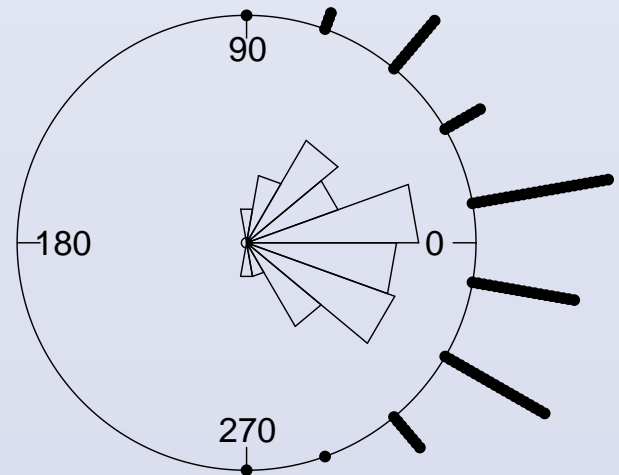
Title: Circular Statistics Package:

CircStats Author:

S-plus original by Ulric Lund <ulund@calpoly.edu>,

R port by Claudio Agostinelli <claudio@unive.it>

```
> #install.packages(CircStats)
> library(CircStats)
> data.vm <- rvm(100, 0, 3)
> rose.diag(data.vm, bins = 18,
+ pts = TRUE, shrink=1.5,prop=1.5)
> title("Grafica de Datos Circulares")
```





Comunicación con Fortran y C

```
obj.list <- .Fortran("bm",X,Y,N,COLS,BLKS,MXFAC,MXINT,  
                    PI,INDGAM,INDG2,GAM2,NGAM,  
                    GAMMA,NTOP,mdcnt,ptop,sigtop,  
                    nftop,jtop,del,sprob,pgam,  
                    prob,ind,PACKAGE="BsMD")
```

```
names(obj.list) <- c("X","Y","N","COLS","BLKS","MXFAC","MXINT",  
                    "PI", "INDGAM","INDG2","GAM2","NGAM",  
                    "GAMMA","NTOP", "mdcnt","ptop","sigtop",  
                    "nftop","jtop","del","sprob","pgam",  
                    "prob","ind")
```



Consideraciones Finales

- Existen más de 865 paquetes, en el sitio web, que ajustan modelos específicos, grafican datos muy particulares (datos direccionales), realizan simulaciones de variables específicas, etc, etc, ...
- R resulta no sólo en una opción sino una **buena opción** para la graficación y el análisis estadístico.
- Aunque en esta plática no se muestra, en términos prácticos el manejo de memoria en R es más eficiente que en S-plus y esto es redituable al momento de programar.



Referencias

1. Ihaka, R. y Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, **5**, 3, 299-314.
2. <http://cran.r-project.org>
3. Dalgaard, P. (2002). *Introductory Statistics with R*. Springer-Verlag. New York.
4. Everitt, S. B. (1996). *A handbook of Statistical Analyses using S-PLUS*. Chapman & Hall. New York.