UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Fundació Politècnica de Catalunya

Métodos cuantitativos II: introducción a **R**

Jose M Sallan `jose.maria.sallan@upc.edu`

Outline

Introducción a **R** y RStudio

Estructuras de datos en **R**

Leyendo archivos en **R**

Bibliotecas (paquetes) de **R**

Introducción a análisis de datos con **R**

El lenguaje **R**

R es un conjunto integrado de programas para manipulación de datos, cálculo y gráficos
Entre otras características dispone de:

- una amplia, coherente e integrada colección de herramientas para análisis de datos
- posibilidades gráficas para análisis de datos
- un lenguaje de programación bien desarrollado, simple y efectivo, que incluye condicionales, ciclos, funciones recursivas y posibilidad de entradas y salidas

R es sofware de código abierto, que puede extenderse mediante paquetes según las necesidades del usuario.

RStudio

RStudio es un entorno de desarrollo integrado (IDE) para **R**, que incluye:

- una consola
- un editor de sintaxis que apoya la ejecución de código
- herramientas para el trazado, la depuración y la gestión del espacio de trabajo
- previsualización de gráficos y de ayuda del sistema

Utilizaremos RStudio desktop (gratuito)

Obteniendo **R** y RStudio

R y RStudio están disponibles para la mayoría de sistemas operativos

- Obtener e instalar R
- Obtener e instalar RStudio

Variables numéricas

```
> x <- 2
> X <- 3
> x
[1] 2
> X
[1] 3
> x <- x +1
> x
[1] 3
```

- Podemos almacenar valores en variables en una sesión de **R**
- Es preferible usar <- como operador de asignación
- Las variables se pueden actualizar, y distinguen mayúsculas y minúsculas

Variables de texto

R también pueden almacenarse en variables cadenas de texto. Se declaran con dobles comillas.

```
> a <- "Hello World"
> a

[1] "Hello World"

> b <- "203"
> b

[1] "203"

> c <- as.numeric(b)
> c

[1] 203
```

La variable **b** es una cadena y la variable **c** es numérica

Factores

Podemos representar variables categóricas mediante factores, en las que cada individuo toma el valor de un nivel especificado.

```
> estado <- c("tas", "qld", "sa", "sa", "sa", "vic", "nt",
+ "act", "qld", "nsw", "wa", "nsw", "nsw", "vic", "vic",
+ "vic", "nsw", "qld", "qld", "vic", "nt", "wa", "wa",
+ "qld", "sa", "tas", "nsw", "nsw", "wa", "act")
> estado <- factor(estado)
> levels(estado)

[1] "act" "nsw" "nt"  "qld" "sa"  "tas" "vic" "wa"
```

Vectores

Un conjunto de valores de variables pueden almacenarse en vectores

Vector numérico de longitud 5

```
> d <- numeric(5)
> d
```

```
[1] 0 0 0 0 0
```

Definiendo un vector usando c()

```
> e <- c(4,-1,2,3)
> e
```

```
[1]  4 -1  2  3
```

Vector de variables texto:

```
> f <- c("ab", "l", "fz", "a")
> f
```

```
[1] "ab" "l"  "fz" "a"
```

Vector de variables lógicas:

```
> g <- c(TRUE, FALSE, TRUE)
> g
```

```
[1]  TRUE FALSE  TRUE
```

Vectores

Subconjuntos de un vector

Podemos obtener un subconjunto de valores que cumplen determinada condición en un vector usando `which`

```
> s <- c(2, 3, 4, 6, 9, 1, 3)
> which(s>5)
```

```
[1] 4 5
```

```
> s[which(s>5)]
```

```
[1] 6 9
```

En **R** las posiciones de un vector comienzan por 1.

Matrices

A partir de un vector, pueden definirse matrices de dos o más dimensiones. Se ha de especificar si se definen por filas o columnas.

```
> A <- matrix(1:16, 4, 4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

```
> B <- matrix(1:16, 4, 4,
+             byrow = TRUE)
> B
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

Matrices

Acceso a elementos de matrices

```
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

```
> A[2, 3]
[1] 10
> A[ ,3]
[1]  9 10 11 12
> A[2, ]
[1]  2  6 10 14
```

Listas

Una lista es una colección ordenada de objetos, que son sus componentes. Pueden ser de diferentes tipos:

```
> list <- list(albert = 54, bryan = A, carlos = c(1,2,3))
> list[[1]]

[1] 54

> list$carlos

[1] 1 2 3
```

Hojas de datos *(data frames)*

Una hoja de datos es una lista de vectores de la misma longitud. Tiene características específicas:

- Filas y columnas de una hoja de datos tienen nombres, accesibles con `rownames` and `colnames`, respectivamente.
- Podemos acceder a la fila i la columna j de `df` haciend `df[i, ]` and `df[, j]`.
- Del mismo modo que las listas, podemos acceder a las columnas por nombre usando `$`.
- Las columnas de texto de una hoja de datos son factores por defecto. Podemos cambiar esto haciendo `stringsAsFactors = FALSE` al cargar la hoja de datos.

Hojas de datos *(data frames)*

Extrayendo información de hojas de datos


```
> head(mtcars) #principio del df
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
> tail(mtcars)#final del df
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.7	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.5	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.5	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.6	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.6	1	1	4	2

Hojas de datos *(data frames)*

Accediendo a elementos de hojas de datos

```
> length(mtcars) #variables

[1] 11

> nrow(mtcars) #observaciones

[1] 32

> mtcars[3, ] #observación 3

            mpg cyl disp hp drat   wt  qsec vs am gear carb
Datsun 710 22.8   4  108 93 3.85 2.32 18.61  1  1    4    1

> mtcars$mpg[1:10] #primeros 10 valores de mpg

 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2
```

Formatos de archivo en **R**

En el contexto de análisis de datos, lo habitual es estructurarlos en forma de hojas de datos
Formatos usuales de lectura de archivo:

- Archivos de texto `.txt`: función `read.table`
- Archivos separados por comas `.csv`: función `read.csv`
- Otros formatos de archivo: paquete `foreign`

Para leer datos de hoja de cálculo lo más conveniente es guardarlos como `.csv`

Leyendo un archivo en **R**

Pasos a seguir:

- Fijar directorio de trabajo: función `setwd` o ventana Files de RStudio
- Leer archivo como *data frame* y asignarle un nombre
- Tener en cuenta si el archivo incluye nombres de las variables (parámetro `header`)

Bibliotecas (paquetes) en **R**

R se suministra con un conjunto de funciones `base`. En función de las necesidades, pueden añadirse nuevas funciones y datos con bibliotecas (paquetes) disponibles desde el repositorio CRAN

- Los paquetes se instalan en el sistema haciendo `install.packages("psych")`
- Cuando se quiere acceder al paquete: `library(psych)`

Bibliotecas (paquetes) en **R**

Algunos ejemplos de paquetes de **R** disponibles en CRAN

`psych`
`corrplot`
`car`
`AER`
`lavaan`
`igraph`
`dplyr`
`ggplot2`

Análisis de datos con **R**

R permite analizar datos de naturaleza diversa, y con objetivos diversos. En nuestro caso queremos usarlo para *análisis estadístico para investigación*.
Algunos pasos del análisis inicial de datos:

- Examinar inicio y final de los datos
- Examinar estructura de los datos con `str`
- Estadísticos de las variables con `summary`
- Detectar y tratar datos faltantes con `is.na`

Análisis de datos con **R**

Visualización del inicio y el final de los datos


```
> head(airquality)

  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6

> tail(airquality)

    Ozone Solar.R Wind Temp Month Day
148    14      20 16.6   63     9  25
149    30     193  6.9   70     9  26
150    NA     145 13.2   77     9  27
151    14     191 14.3   75     9  28
152    18     131  8.0   76     9  29
153    20     223 11.5   68     9  30
```

Análisis de datos con **R**

Estructura de los datos

```
> str(airquality)

'data.frame':        153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...

> summary(airquality)

     Ozone           Solar.R           Wind             Temp      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
 NA's   :37       NA's   :7                                       
     Month            Day      
 Min.   :5.000   Min.   : 1.0  
 1st Qu.:6.000   1st Qu.: 8.0  
 Median :7.000   Median :16.0  
 Mean   :6.993   Mean   :15.8  
 3rd Qu.:8.000   3rd Qu.:23.0  
 Max.   :9.000   Max.   :31.0  
```

Análisis de datos con **R**

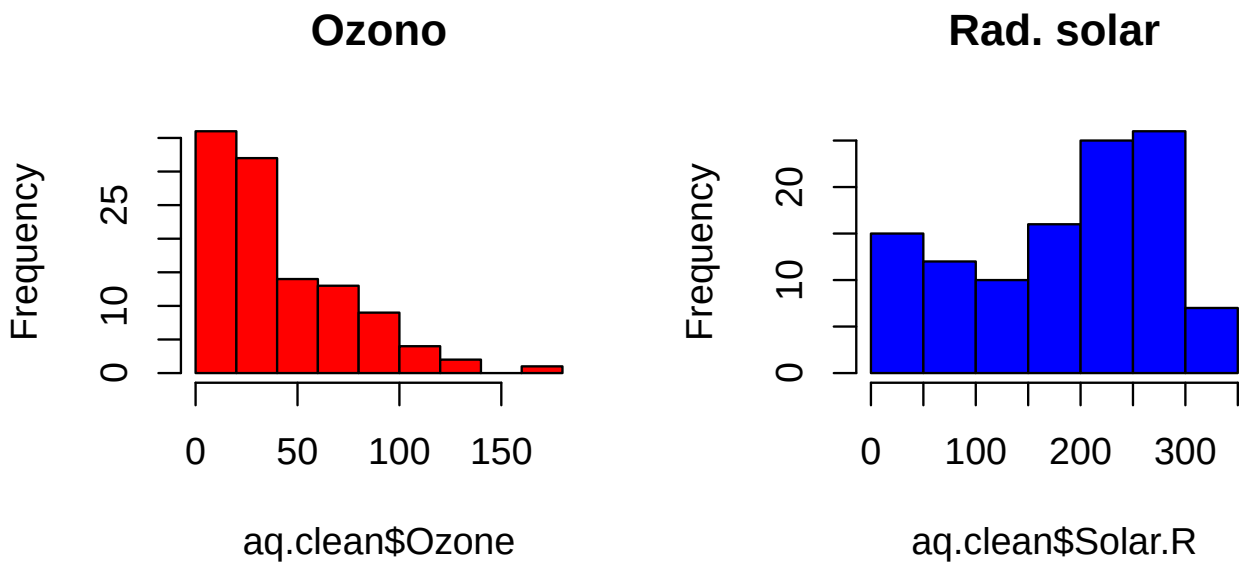
Análisis de datos faltantes

```
> aq.clean <- airquality[which(!is.na(airquality$Ozone)
+                              & !is.na(airquality$Solar.R)), ]
> nrow(airquality)

[1] 153

> nrow(aq.clean)

[1] 111

> summary(aq.clean)

     Ozone           Solar.R          Wind             Temp
 Min.   :  1.0   Min.   :  7.0   Min.   : 2.30   Min.   :57.00
 1st Qu.: 18.0   1st Qu.:113.5   1st Qu.: 7.40   1st Qu.:71.00
 Median : 31.0   Median :207.0   Median : 9.70   Median :79.00
 Mean   : 42.1   Mean   :184.8   Mean   : 9.94   Mean   :77.79
 3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:11.50   3rd Qu.:84.50
 Max.   :168.0   Max.   :334.0   Max.   :20.70   Max.   :97.00
     Month            Day
 Min.   :5.000   Min.   : 1.00
 1st Qu.:6.000   1st Qu.: 9.00
 Median :7.000   Median :16.00
 Mean   :7.216   Mean   :15.95
 3rd Qu.:9.000   3rd Qu.:22.50
 Max.   :9.000   Max.   :31.00
```

Análisis de datos con **R**

Histogramas

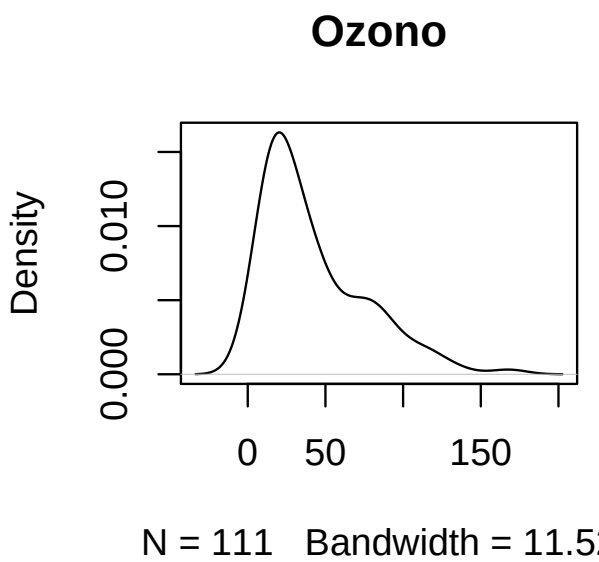
```
> par(mfrow=c(1,2))
> hist(aq.clean$Ozone, col="red", main="Ozono")
> hist(aq.clean$Solar.R, col="blue", main="Rad. solar")
```

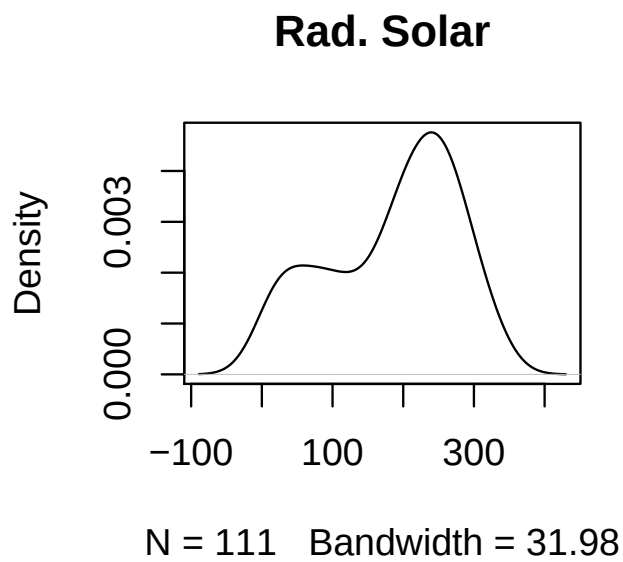


Análisis de datos con **R**

Densidad

```
> d.ozone <- density(aq.clean$Ozone)
> d.solar <- density(aq.clean$Solar.R)
> par(mfrow=c(1,2))
> plot(d.ozone, main="Ozono")
> plot(d.solar, main="Rad. Solar")
```

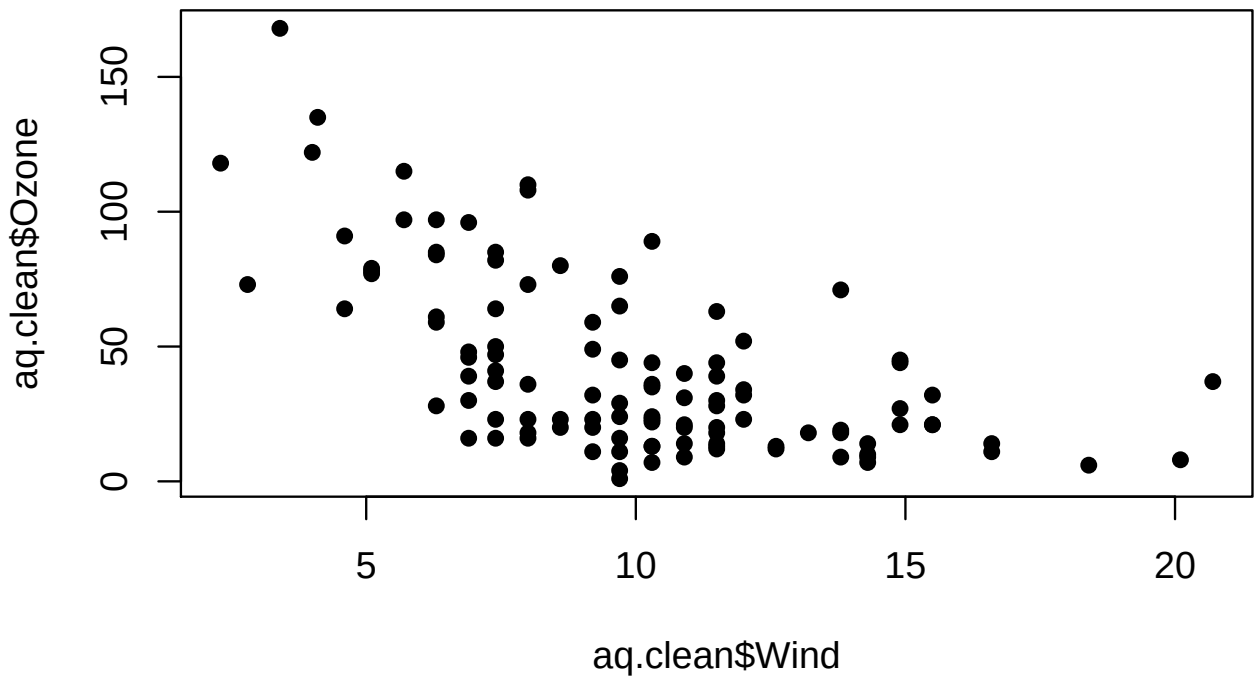




Análisis de datos con **R**

Gráfico de dispersión

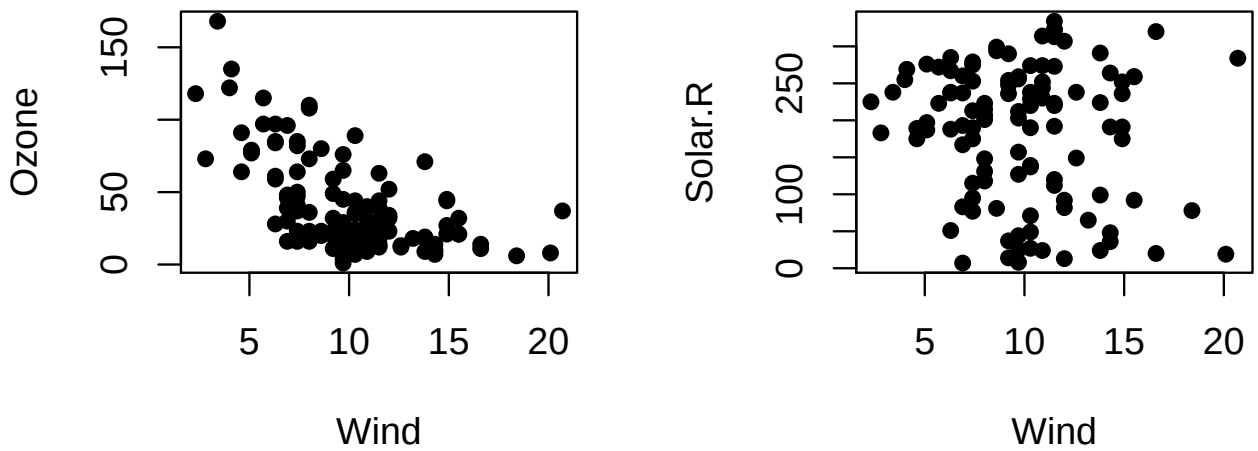
```
> plot(aq.clean$Wind, aq.clean$Ozone, pch=16)
```



Análisis de datos con **R**

Gráficos de dispersión

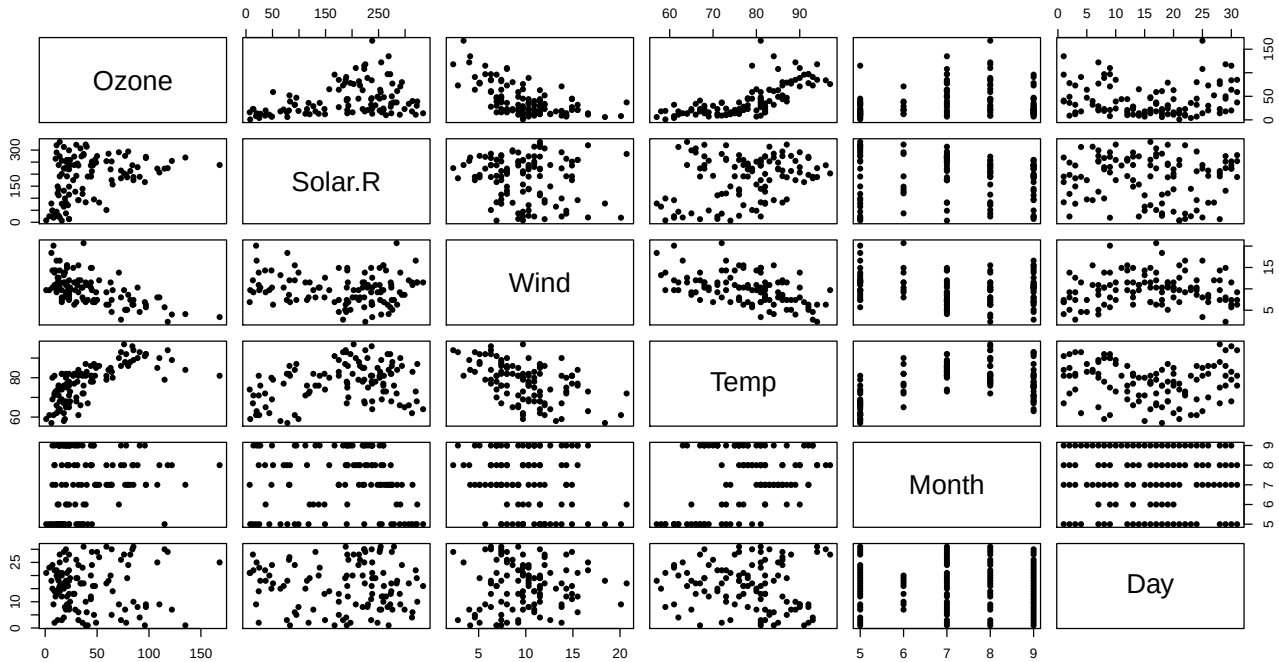
```
> par(mfrow=c(1,2))
> plot(aq.clean$Wind, aq.clean$Ozone, pch=16, xlab="Wind", ylab="Ozone")
> plot(aq.clean$Wind, aq.clean$Solar.R, pch=16, xlab="Wind", ylab="Solar.R")
```



Análisis de datos con **R**

Gráfico de dispersión

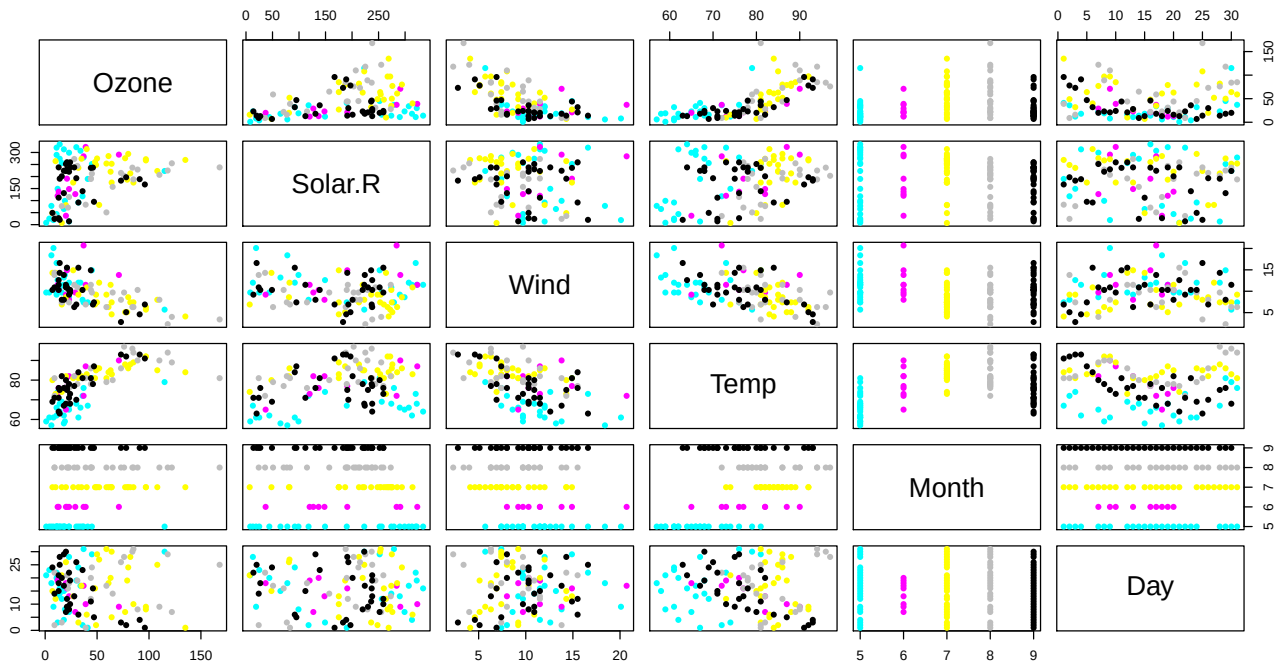
```
> plot(aq.clean, pch=16)
```



Análisis de datos con **R**

Gráfico de dispersión

```
> plot(aq.clean, pch=16,
+      col=aq.clean$Month)
```



Análisis de datos con **R**

Comparación de medias

Evaluación de la temperatura media en los meses 5 y 9 de airquality

```
> aq.clean$Month <- factor(aq.clean$Month)
> airquality.59 <- aq.clean[which(aq.clean$Month==5 | aq.clean$Month==9), ]
> t.test(Temp ~ Month, data=airquality.59)

        Welch Two Sample t-test

data:  Temp by Month
t = -5.0182, df = 50.847, p-value = 6.752e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -14.614451  -6.261986
sample estimates:
mean in group 5 mean in group 9
       66.45833        76.89655
```

Análisis de datos con **R**

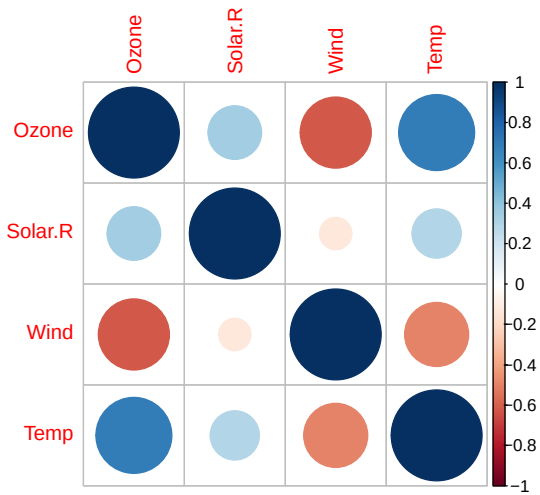
Correlaciones con psych

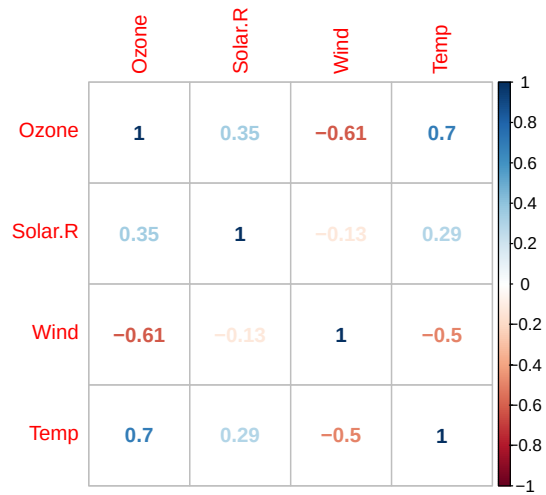
```
> library(psych)
> corr.test(aq.clean[, 1:4])

Call:corr.test(x = aq.clean[, 1:4])
Correlation matrix
        Ozone Solar.R  Wind  Temp
Ozone    1.00    0.35 -0.61  0.70
Solar.R  0.35    1.00 -0.13  0.29
Wind    -0.61   -0.13  1.00 -0.50
Temp     0.70    0.29 -0.50  1.00
Sample Size
[1] 111
Probability values (Entries above the diagonal are adjusted for multiple tests.)
        Ozone Solar.R Wind Temp
Ozone       0    0.00 0.00    0
Solar.R     0    0.00 0.18    0
Wind        0    0.18 0.00    0
Temp        0    0.00 0.00    0

 To see confidence intervals of the correlations, print with the short=FALSE option
```

Análisis de datos con **R**

Correlaciones con `corrplot`

```
> library(corrplot)
> par(mfrow=c(1,2))
> cor.aq <- cor(aq.clean[, 1:4])
> corrplot(cor.aq, method = "circle")
> corrplot(cor.aq, method = "number")
```





Conclusiones

1. **R** proporciona una plataforma potente para análisis de datos (no sólo estadística para investigación)
2. La curva de aprendizaje de **R** puede reducirse con RStudio
3. **R** puede extenderse mediante librerías (packages)
4. **R** permite un análisis previo de los datos, necesario antes de análisis más sofisticados