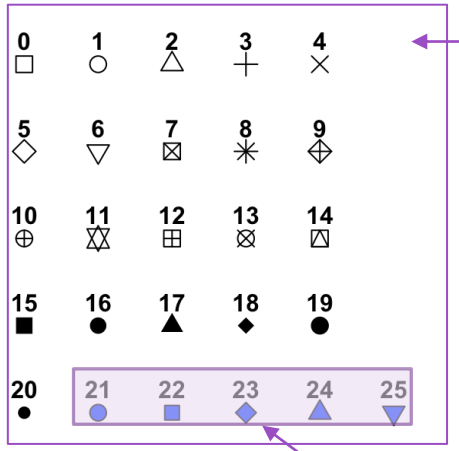
Lezione R Studio

Caterina Gozzi
Università di Firenze
Dipartimento di Scienze della Terra
email: caterina.gozzi@unifi.it

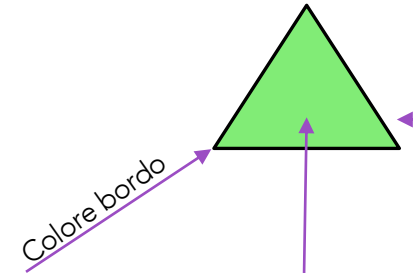
Diagrammi binari

- Caricare il database stream.xls in R




```
plot(stream$Ca, stream$SO4,pch=19,xlab="Ca(mg/L)", ylab="SO4(mg/L)",
col="darkgreen", cex=0.8)
```

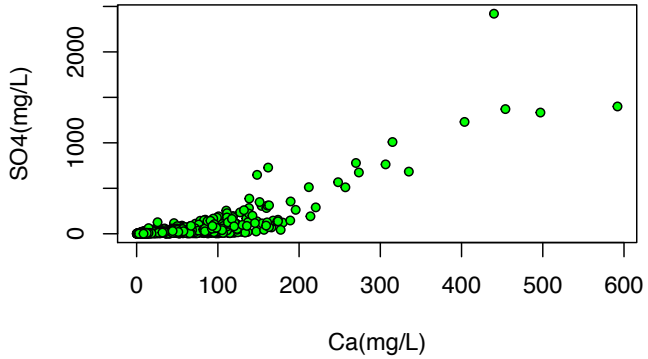
Dimensione dei simboli




```
plot(stream$Ca, stream$SO4,pch=21,xlab="Ca(mg/L)", ylab="SO4(mg/L)", col="black", bg="green", cex=0.8, lwd=1)

text(x=stream$Ca, y=stream$SO4, labels=stream$Country, cex=0.5, pos=4, col="red")
```

Etichette punti

labels= testo o testi da inserire
adj= allineamento del testo
pos= Posizione del testo rispetto al punto indicato dalle coordinate: 1 = sotto, 2 = a sinistra, 3 = sopra , 4 = a destra. Se specificato prevale su adj
offset= se pos è indicato, distanza delle etichette dal punto delle coordinate in frazioni delle dimensioni del carattere (default=0.5)
cex= dimensione del carattere in termini proporzionali
col= colore del carattere font



Diagrammi binari
(trasformazione in meq/L)

```
Caeq<-stream$Ca/40*2
HCO3eq<-stream$HCO3/61
SO4eq<-stream$SO4/96*2
Mgeq<-stream$Mg/24.3*2
NO3eq<-stream$NO3/62
Cleq<-stream$Cl/35.45
Naeq<-stream$Na/22.99
Keq<-stream$K/39.1
```

```
plot(Caeq, SO4eq,pch=21,xlab="Ca(meq/L)", ylab="SO4(meq/L)", col="Black",
bg="Green", cex=0.8, lwd=1)
```

Bubble plot (rappresentazione dati spaziali)

- Caricare il database stream.xls in R

- Caricare i seguenti pacchetti in R:

  - `library(ggplot2)`
  - `library(dplyr)`

- Creare un nuovo script:

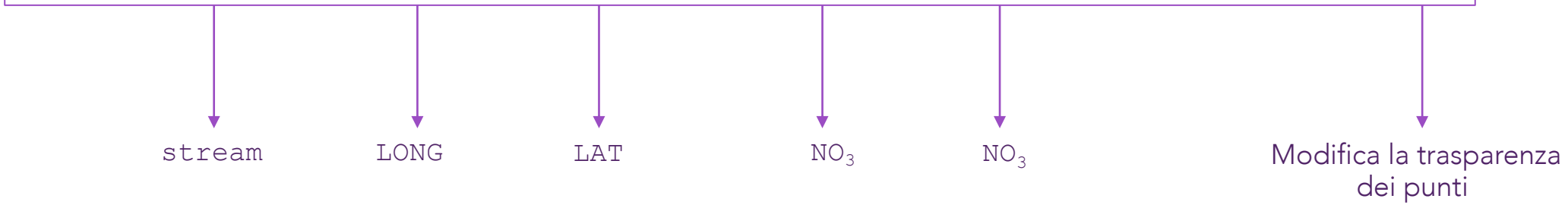
```
ggplot(data, aes(x=coordx, y=coordy, size =var, color=var)) +geom_point(alpha=0.7)
```

- Un bubble plot è uno scatterplot a cui viene aggiunta una terza dimensione: il valore della terza variabile è rappresentato dalla dimensione dei punti.

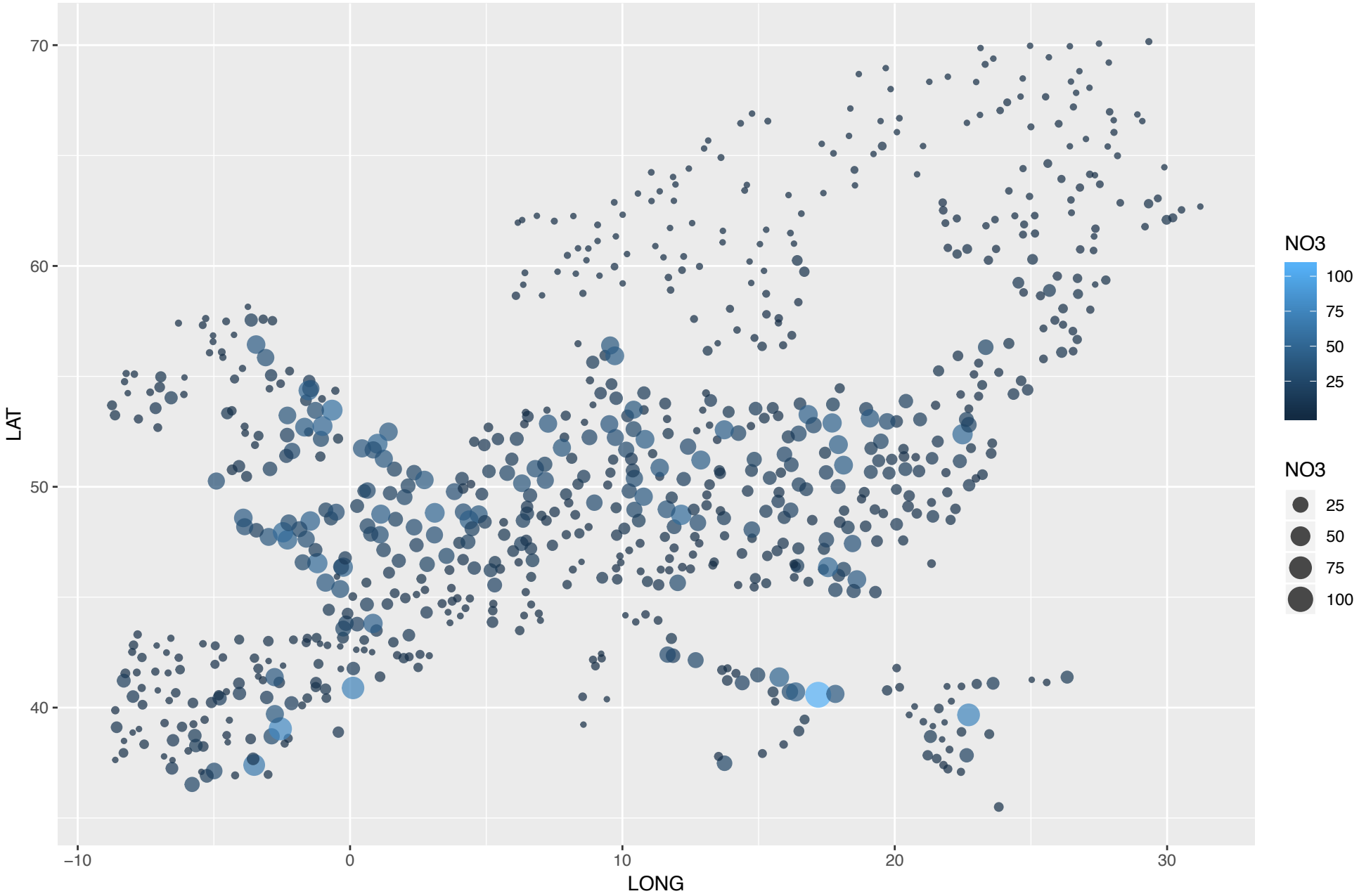
- Mediante il pacchetto ggplot2 il bubble plot viene costruito tramite la funzione geom_point(). Almeno tre variabili devono essere specificate in aes(): *x*, *y* e *size*. La legenda viene costruita in automatico da ggplot2.

Bubble plot

- `ggplot(data, aes(x=coordx, y=coordy, size =var, color=var)) +geom_point(alpha=0.7)`



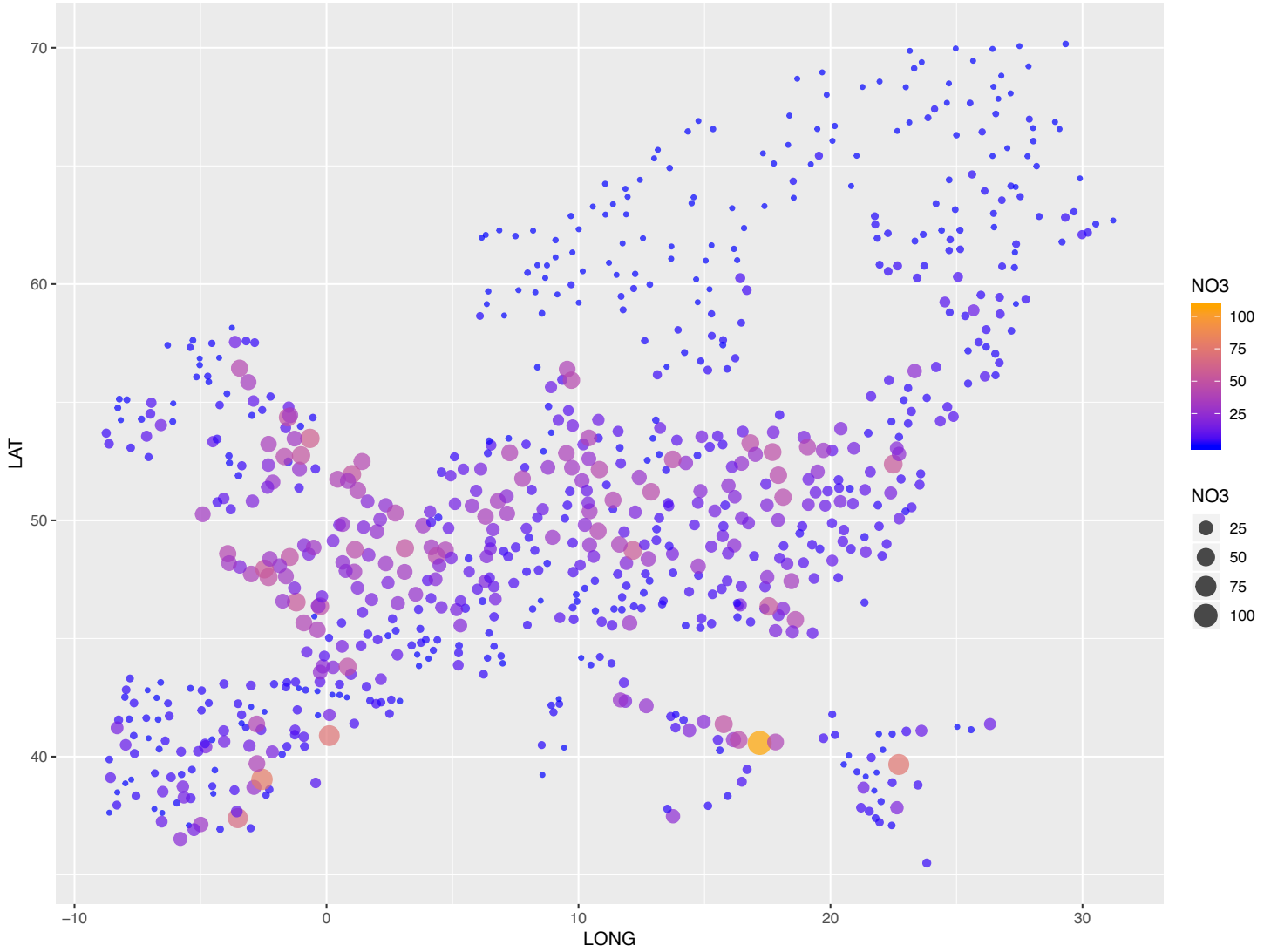
Concentrazione Nitrati in Europa



```
+   scale_colour_gradient(low="blue", high="orange")
```
#cambiare la scala di colore del grafico

Palette Colori

http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf



Bubble plot interattivo

- `library(plotly)`

Variabile che viene mostrata nel riquadro del plot interattivo spostando il cursore sul punto di interesse.

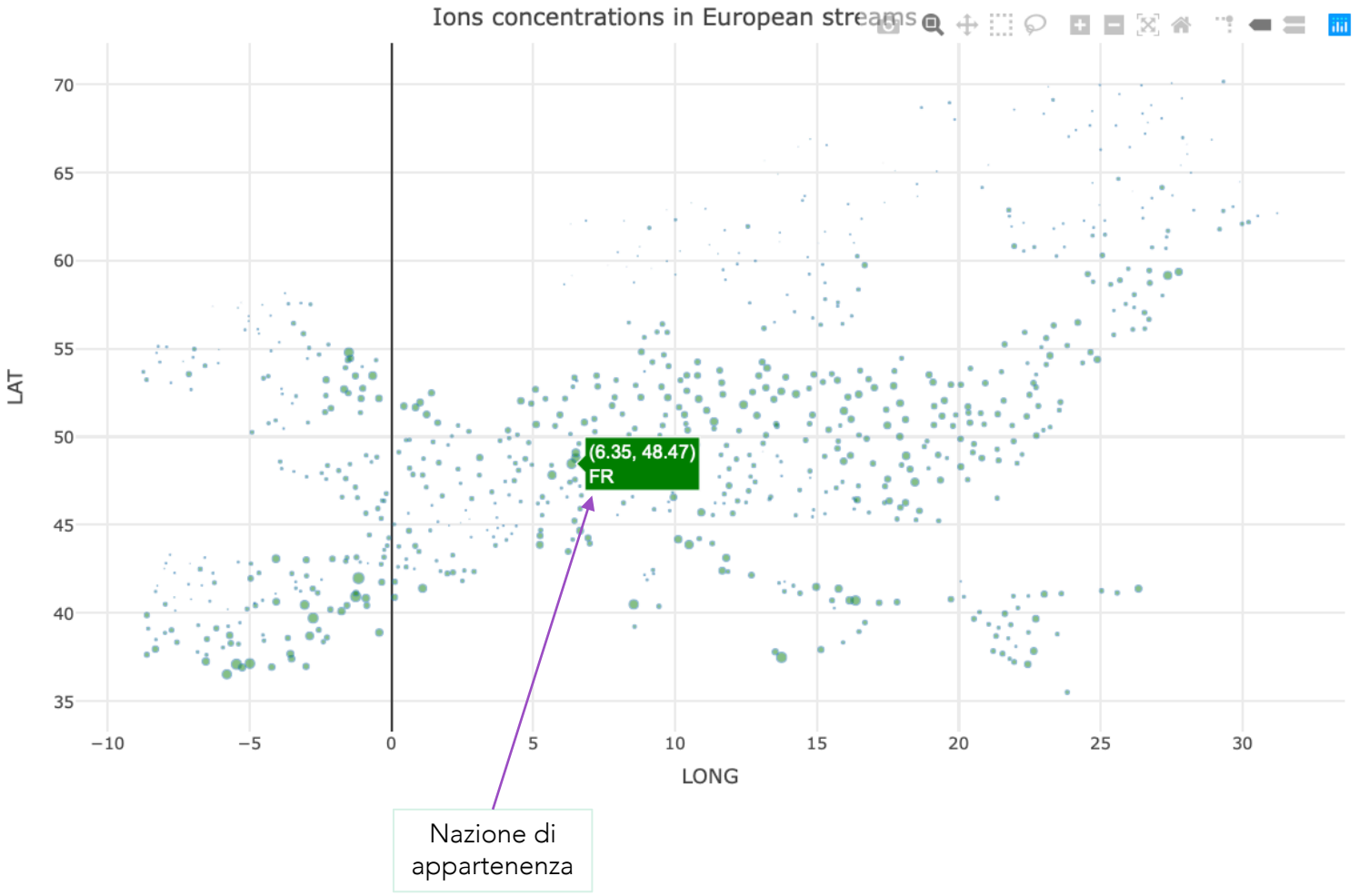
stream → `data`
LONG → `x = ~coordx`
LAT → `y = ~coordy`
Country → `text = ~var`

```
plot_ly(data, x = ~coordx, y = ~coordy, text = ~var, type = 'scatter', mode = 'markers',
        marker = list(size = ~var, opacity = 0.5, color='green')) %>%
  layout(title = 'Ions concentrations in European streams',
         xaxis = list(showgrid = TRUE),
         yaxis = list(showgrid = TRUE))
```

`size = ~var` → SO_4

`%>%` — Operatore per connettere le funzioni (utilizzato in alcuni pacchetti di R)

Bubble plot interattivo



Bubble plot interattivo

(mappare una variabile continua o categorica)

- `library(plotly)`

Mappare una variabile continua

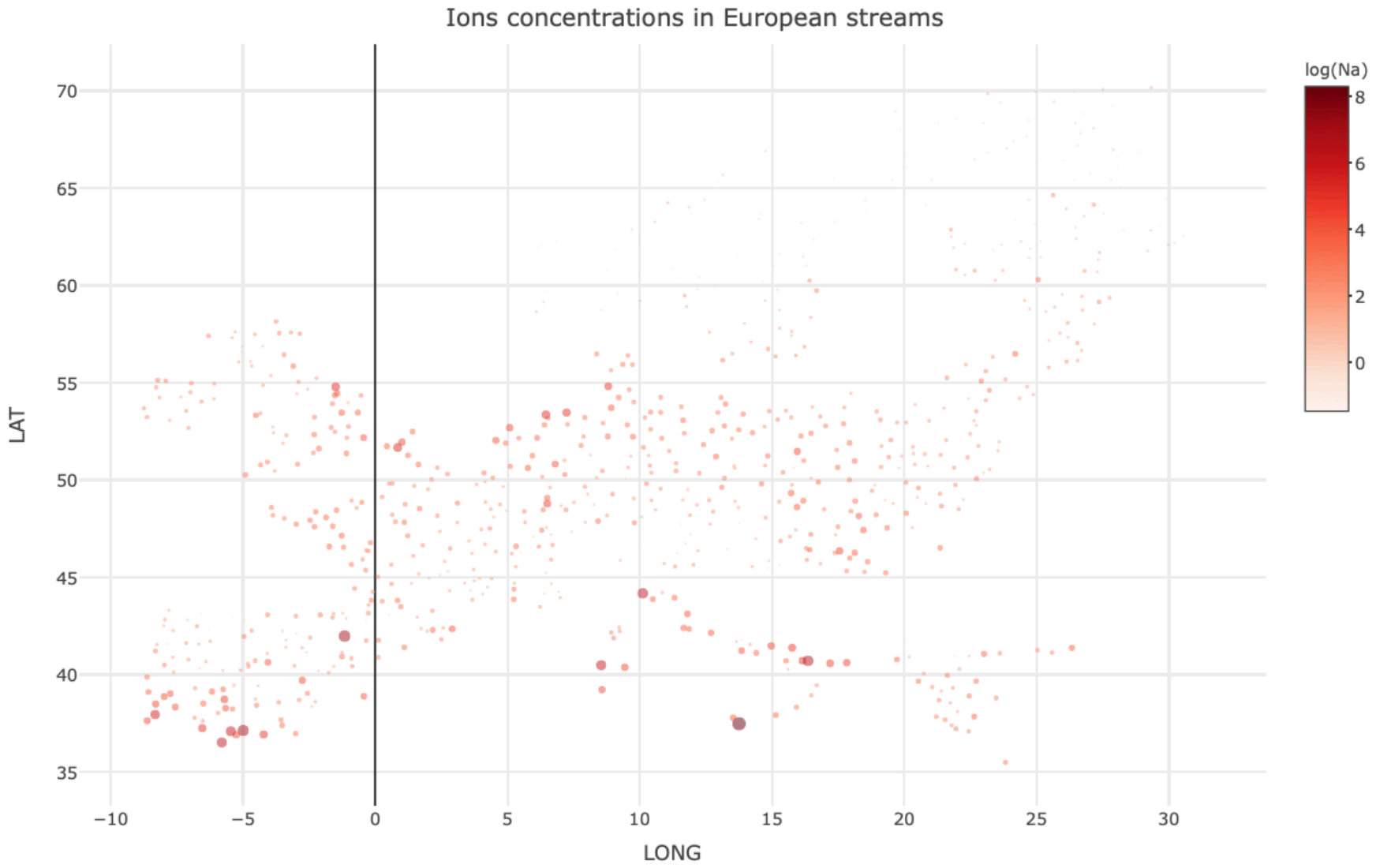
```
plot_ly(stream, x = ~LONG, y = ~LAT, text = ~Country, type = 'scatter', mode = 'markers', color = ~log(Na), colors = 'Reds',
        marker = list(size = ~log(Na), opacity = 0.5)) %>%
  layout(title = 'Ions concentrations in European streams',
         xaxis = list(showgrid = TRUE),
         yaxis = list(showgrid = TRUE))
```

Mappare una variabile categorica (esempio: colori diversi per nazione)

```
plot_ly(stream, x = ~LONG, y = ~LAT, text = ~Na, type = 'scatter', mode = 'markers', size = ~log(Na), color = ~Country, colors
= 'Paired',
        sizes = c(1, 10),
        marker = list(opacity = 0.5, sizemode = 'diameter')) %>%
  layout(title = 'Ions concentrations in European streams',
         xaxis = list(showgrid = TRUE),
         yaxis = list(showgrid = TRUE),
         showlegend = TRUE)
```

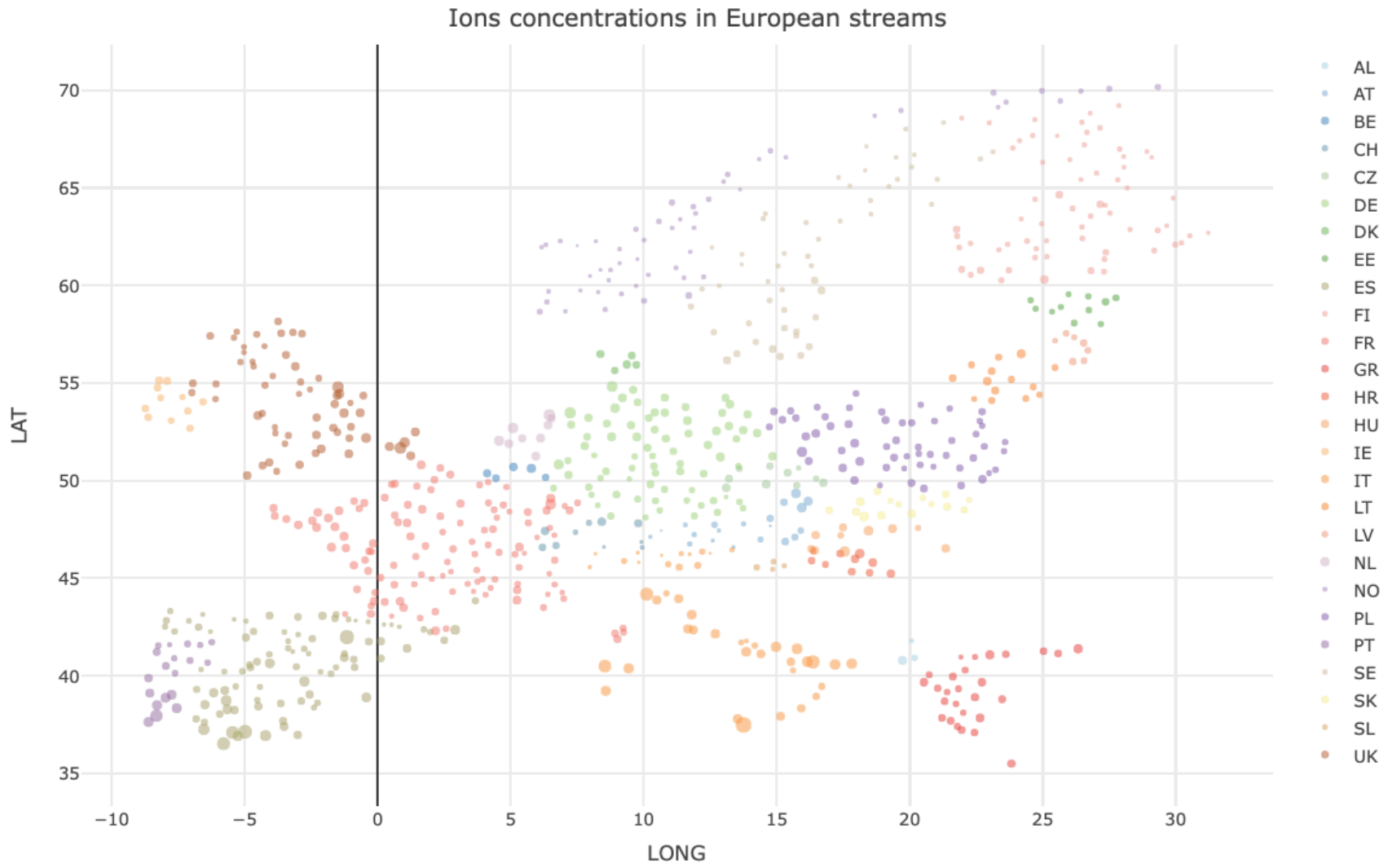
Bubble plot interattivo

mappare una variabile continua

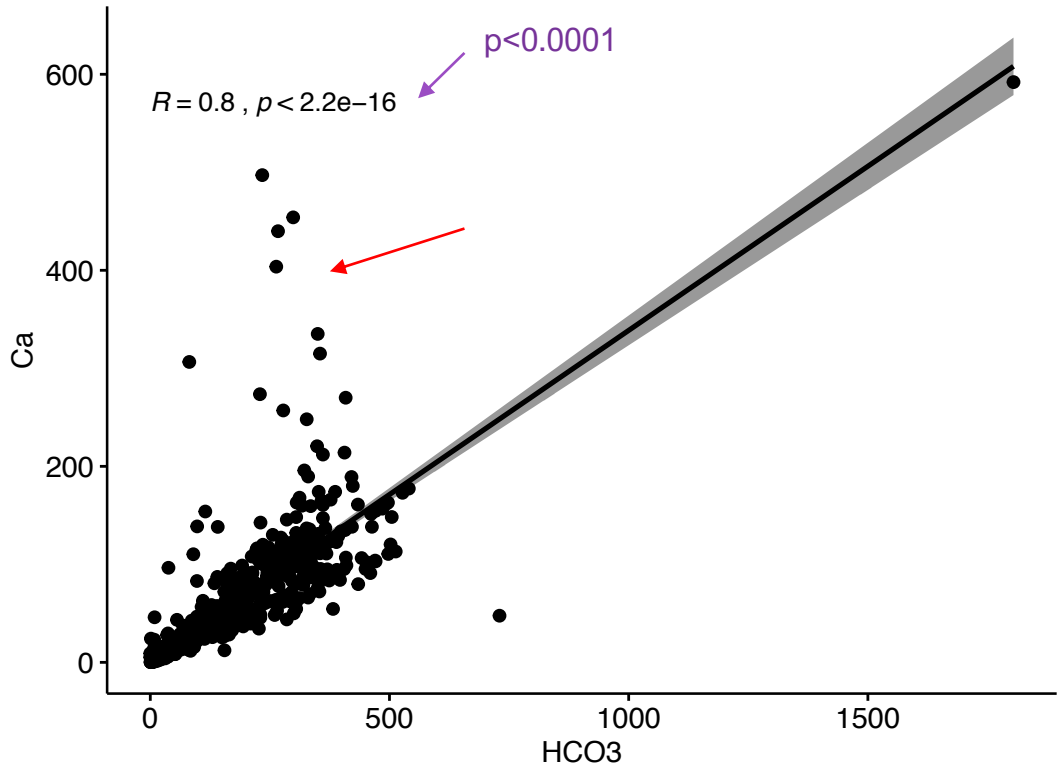


Bubble plot interattivo

mappare una variabile categorica



Is the covariation linear?

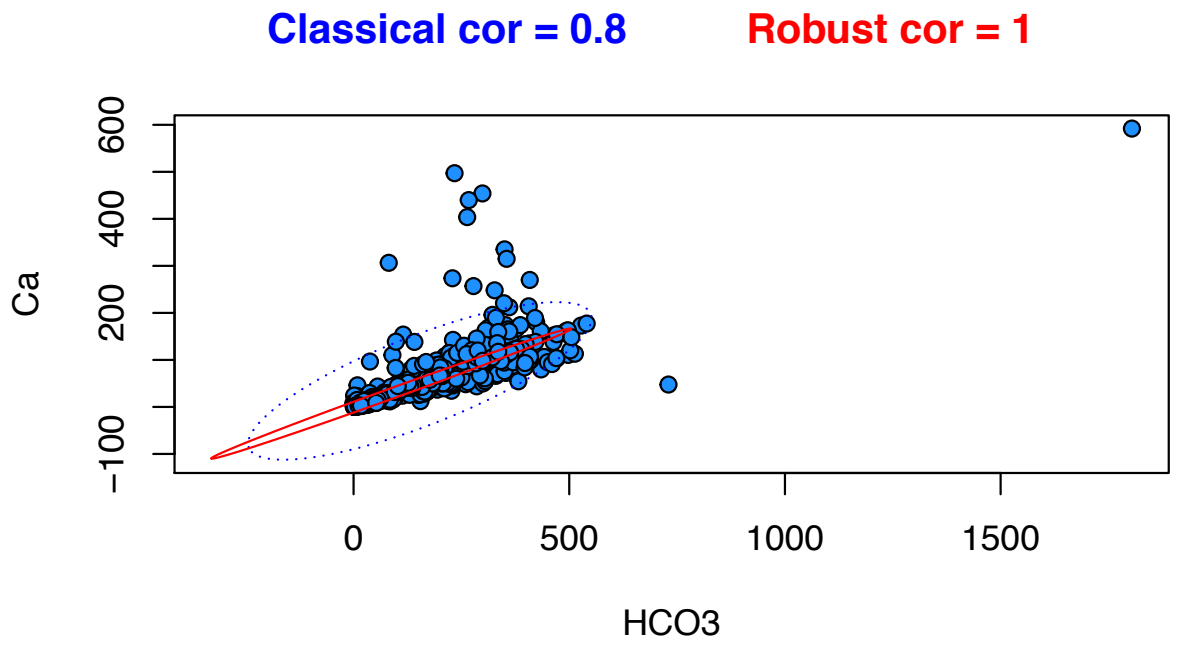


Correlazioni fra variabili

```
library("ggpubr")
ggscatter(stream, x = "HCO3", y = "Ca",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "HCO3", ylab = "Ca")
```

Correlazioni classiche e robuste

```
library(mvoutlier)

par(mfrow=c(1,2))

corr.plot(HCO3,Ca, xlab="HCO3", ylab="Ca", pch=21, bg='dodgerblue')
```

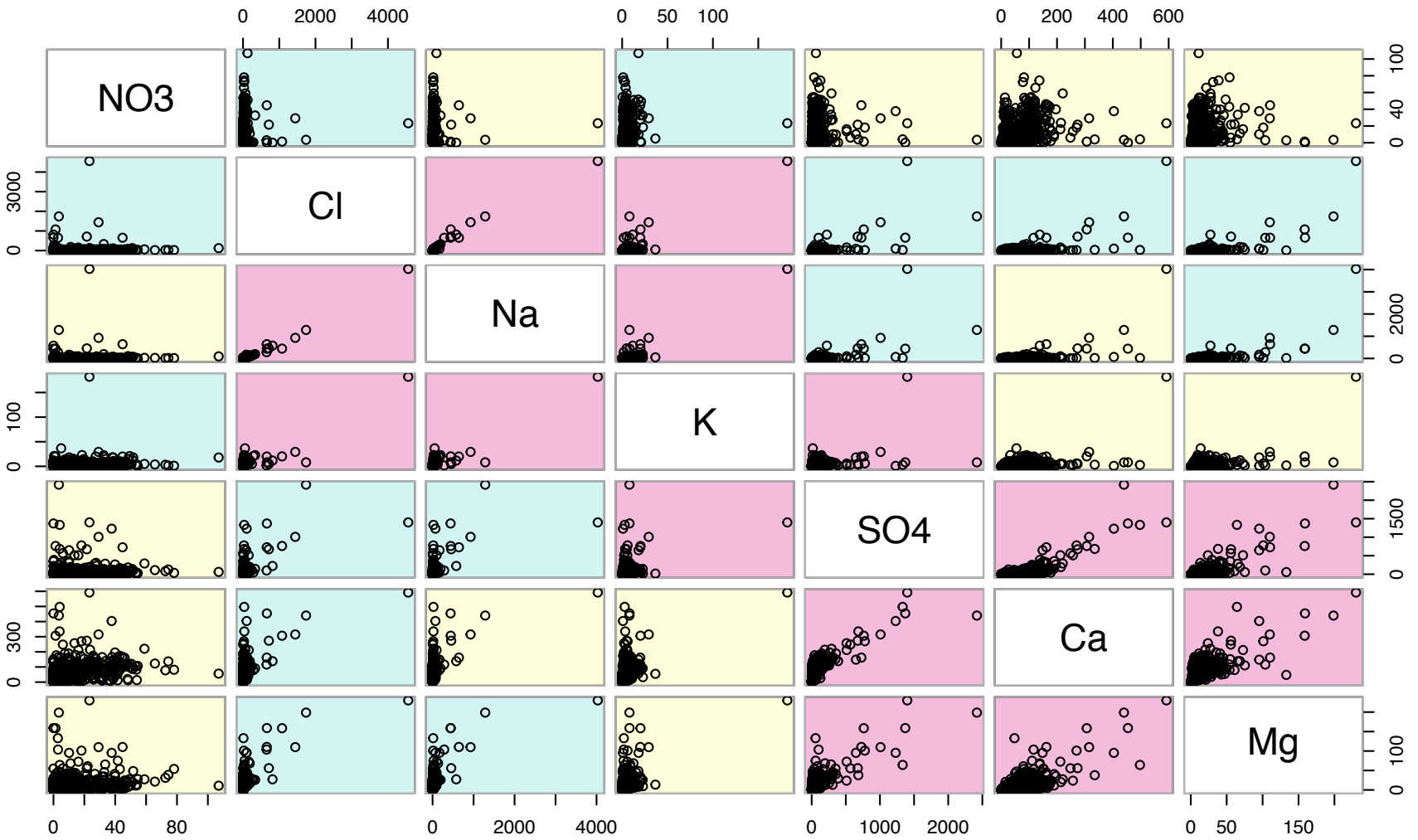


Matrici di correlazione

```
library(gclus)

stream1<-stream[c(7,8,9,10,11,12,13)] # selezionare le variabili di interesse da stream

dta <- stream1 # ottenere i dati

dta.r <- abs(cor(dta, method = "spearman")) # calcolare le correlazioni

dta.col <- dmat.color(dta.r) # definire i colori

dta.o <- order.single(dta.r) # ordinare le variabili in modo che quelle con la maggiore correlazione si trovino più vicino alla diagonale

cpairs(dta, dta.o, panel.colors=dta.col, gap=0.5)
```

gap=dimensione spazio fra i riquadri

Matrici di correlazione



Le matrici di correlazione sono un ottimo modo per determinare se sono presenti correlazioni lineari tra più variabili. Per un set di variabili di dati X1, X2,…, Xn, il diagramma mostra tutti i grafici a coppie delle variabili in formato di matrice.

Matrici di correlazione
(ellissi)

```
library(ellipse)
library(RColorBrewer)

stream1<-stream[c(6,7,8,9,10,11,12,13,14)]
data=cor(stream1, method="spearman")

# Costruzione della scala di colori
my_colors <- brewer.pal(5, "Spectral")
my_colors=colorRampPalette(my_colors)(100)

# Ordinare la matrice di correlazione
ord <- order(data[1, ])
data_ord = data[ord, ord]
plotcorr(data_ord , col=my_colors[data_ord*50+50],
mar=c(1,1,1,1))

#mar=grandezza del grafico
```

L'uso delle ellissi permette di visualizzare più velocemente le correlazioni.