(Bio)Statistica con R

Parte III

UNIVERSITÀ DEGLI STUDI DI FOGGIA

Dr. Crescenzio Gallo – Università di Foggia – Dipartimenti di Area Medica

(Bio)Statistica con R — Parte III

FUNZIONI GRAFICHE PER LA STATISTICA

- Le funzioni grafiche di **R** sono utili per l'esplorazione visiva dei dati.
- Scarichiamo e salviamo nella directory di lavoro il file **Densplot.csv**, che contiene il sesso (M o F), l'età (in anni) e la concentrazione del colesterolo nel siero (in mg/dL) per 1000 soggetti; quindi eseguiamo il seguente codice:

```
# importo i dati
> mydata <- read.table("Densplot.csv", header=TRUE, sep=";")
# traccio un istogramma semplice dei dati
> hist(mydata$Colest, main = "Istogramma semplice",
     xlab = "Colesterolo totale in mg/dL",
     ylab = "Frequenza")
```

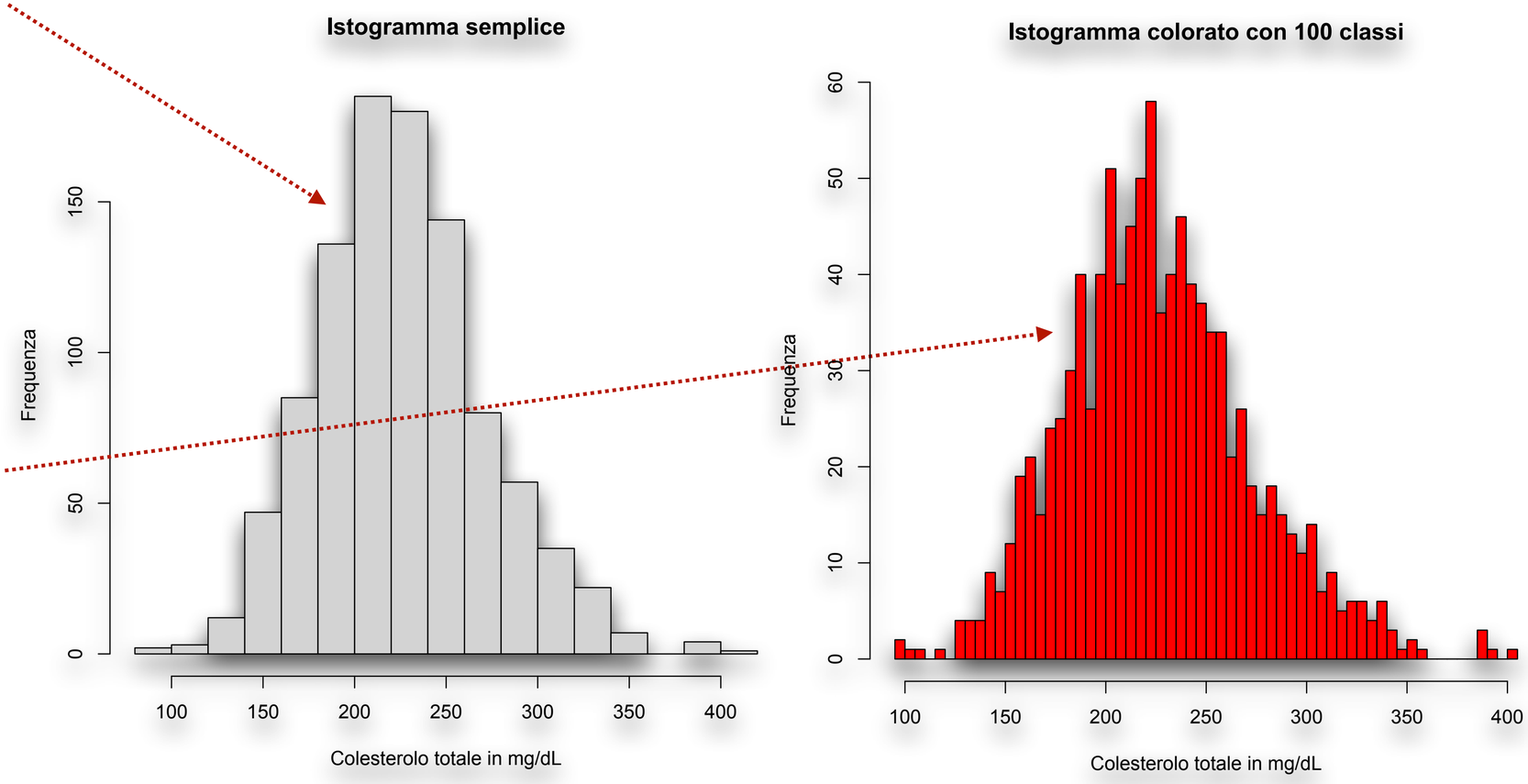
Sesso	Eta	Colest
M	13	172
F	14	132
M	14	176
F	15	156
F	16	190
F	17	175
F	18	118
M	18	160
M	18	161
F	18	174
F	18	178
M	18	190
F	18	192
F	18	201
F	18	226
M	19	142
M	19	148
M	19	208
M	19	248

(Bio)Statistica con R — Parte III

Istogrammi

- Istogramma semplice della distribuzione della concentrazione del colesterolo nel siero.
- Possiamo anche realizzare un istogramma colorato nel quale gestire anche il numero delle classi in cui suddividere i dati con questo codice:

```
# traccio un istogramma colorato, con 100 classi
> hist(mydata$Colest,
   breaks=100, col="red",
   main = "Istogramma
   colorato con 100 classi",
   xlab = "Colesterolo
   totale in mg/dL", ylab =
   "Frequenza")
```



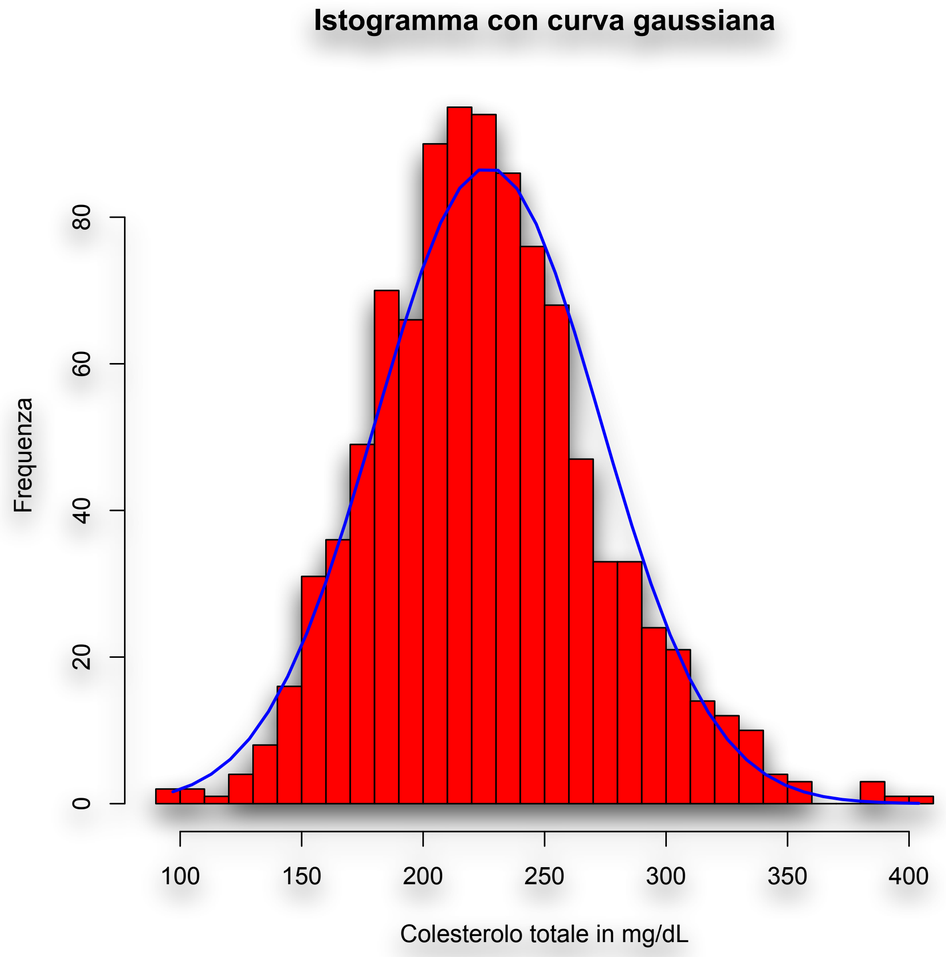
(Bio)Statistica con R — Parte III

Istogrammi

- Possiamo anche tracciare l'istogramma e sovraimporre ad esso la distribuzione gaussiana teorica:

```
> x <- mydata$Colest
> h <-hist(x, breaks=33, col="red",
     main = "Istogramma con curva gaussiana",
     xlab="Colesterolo totale in mg/dL",
     ylab = "Frequenza")
> xfit <- seq(min(x),max(x),length=40)
> yfit <- dnorm(xfit,mean=mean(x),sd=sd(x))
> yfit <- yfit*diff(h$mids[1:2])*length(x)
> lines(xfit, yfit, col="blue", lwd=2)
```

- In questo caso i dati sono stati suddivisi in 33 classi, seguendo la regola per cui **il numero delle classi dovrebbe essere uguale alla radice quadrata del numero dei dati** (qui abbiamo 1000 dati, la cui radice quadrata è circa 33).

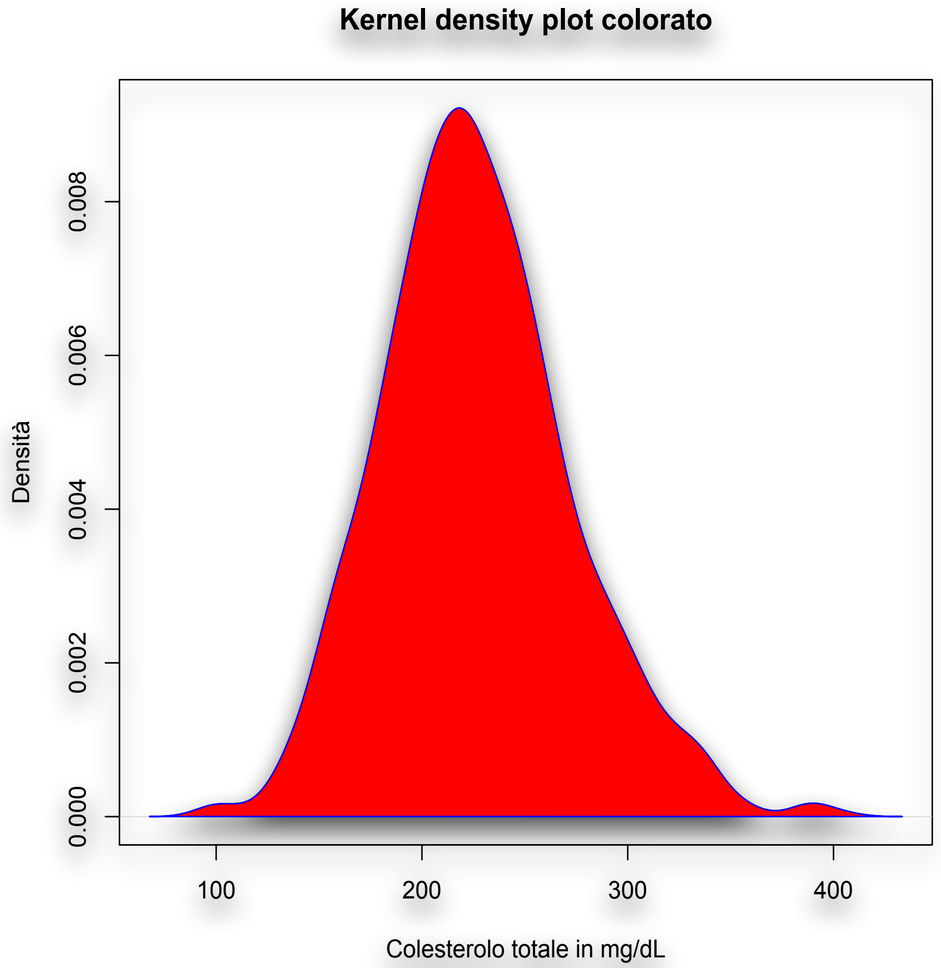


(Bio)Statistica con R — Parte III

Kernel Density Plot

- Il diagramma che rappresenta la distribuzione della densità delle osservazioni (**kernel density plot**) è una modalità di rappresentazione piuttosto interessante, che può essere utilizzata in alternativa al tradizionale istogramma.
- Riutilizziamo gli stessi dati del file precedente "Densplot.csv" che contiene sesso (M o F), età (in anni) e concentrazione del colesterolo nel siero (in mg/dL) di 1000 soggetti per eseguire i seguenti comandi:

```
> d <- density(mydata$Colest) # dati di densità
> plot(d, main = "Kernel density plot colorato",
  xlab="Colesterolo totale in mg/dL",
  ylab = "Densità") # traccio il kernel density plot
# coloro il grafico di rosso e il bordo della curva in blu
> polygon(d, col="red", border="blue")
```

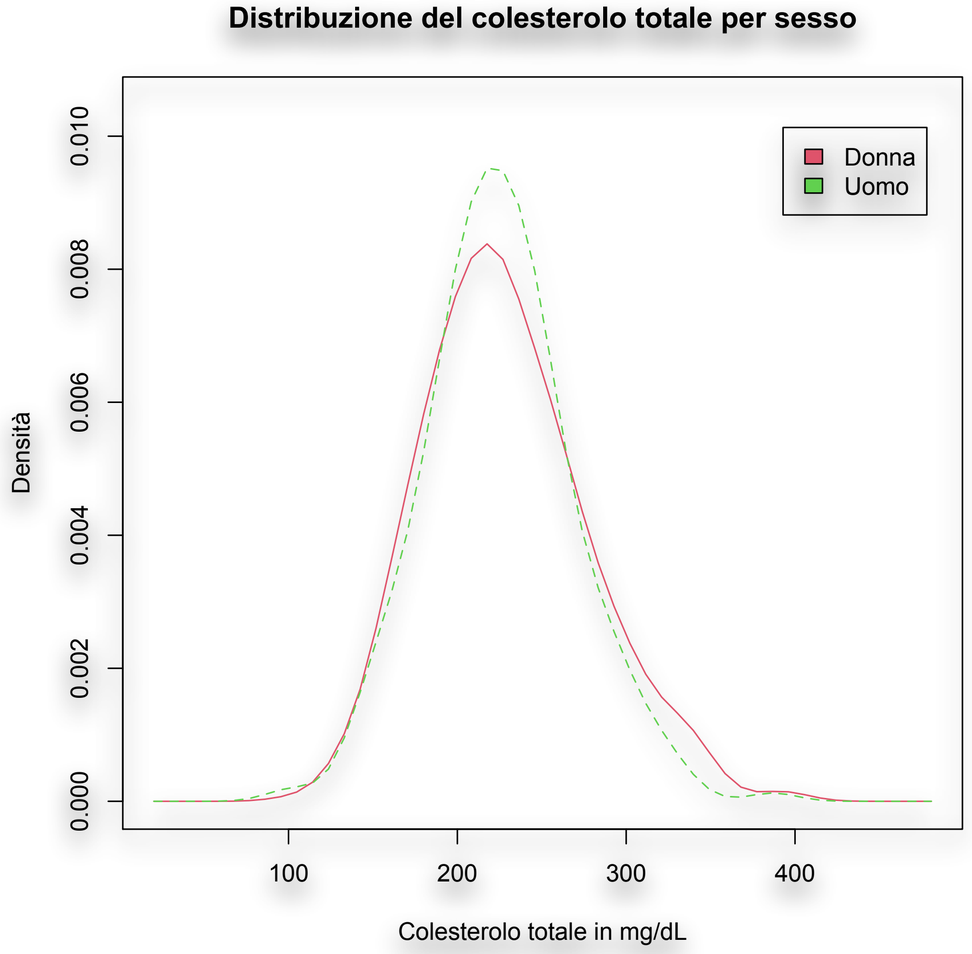


(Bio)Statistica con R — Parte III

Kernel Density Plot

- Eseguiamo ora il seguente codice (sempre sugli stessi dati) per tracciare kernel density plot sovrapposti (al termine fare click con il tasto sinistro del mouse nel punto in cui si vuole far comparire la legenda):

```
> install.packages("sm"); library(sm) # nonpar smoothing methods
> attach(mydata)
> Sesso.f <- factor(Sesso, levels= c("F","M"),
  labels = c("Donna", "Uomo")) # fattorizzo la variabile Sesso
> sm.density.compare(Colest, Sesso.f,
  xlab="Colesterolo totale in mg/dL",
  ylab="Densità") # traccio il grafico
> title(main = "Distribuzione del colesterolo totale
  per sesso") # aggiungo il titolo
> colfill <- c(2:(2+length(levels(Sesso.f))-1)) # colori 2:3,
cioè 2=rosso e 3=verde
> legend(locator(1), levels(Sesso.f), fill=colfill) #
posiziono la legenda: ricordarsi di fare click con il mouse sul punto desiderato nel grafico
```



(Bio)Statistica con R — Parte III

Box&Whiskers Plot

- I **Box & Whiskers Plot** consentono di confrontare in modo immediato la distribuzione di più variabili.
- La scatola rappresenta la mediana (al centro), il primo quartile (margine inferiore della scatola) e il terzo quartile (margine superiore della scatola). La scatola include pertanto il 50% delle osservazioni.
- I whiskers (baffi) includono tutti i dati osservati oppure lasciano all'esterno i dati che presentano uno scostamento eccessivo (**outliers**).
- I Box & Whiskers Plot forniscono una rappresentazione non-parametrica della distribuzione dei dati.
- Scarichiamo e salviamo il file **Boxplot.csv** (che abbiamo già utilizzato all'inizio della Parte II).
- I dati contenuti nel file sono i valori di concentrazione delle IgA (in g/L) in un gruppo di soggetti sani (Controlli) e di soggetti con cirrosi alcolica (AC), epatite cronica attiva (CAH), epatite cronica persistente (CPH), epatite alcolica non cirrotica (NCAH).

Diagnosi	IgA
Controlli	1.22
Controlli	2.81
Controlli	4.02
Controlli	2.23
Controlli	2.35
Controlli	1.64
Controlli	2.08
Controlli	1.96
Controlli	1.54
Controlli	1.63
Controlli	3.25
Controlli	2.9
Controlli	3.44
Controlli	2.55
Controlli	1.18
Controlli	1.78
Controlli	2.56
Controlli	1.36
Controlli	1.83

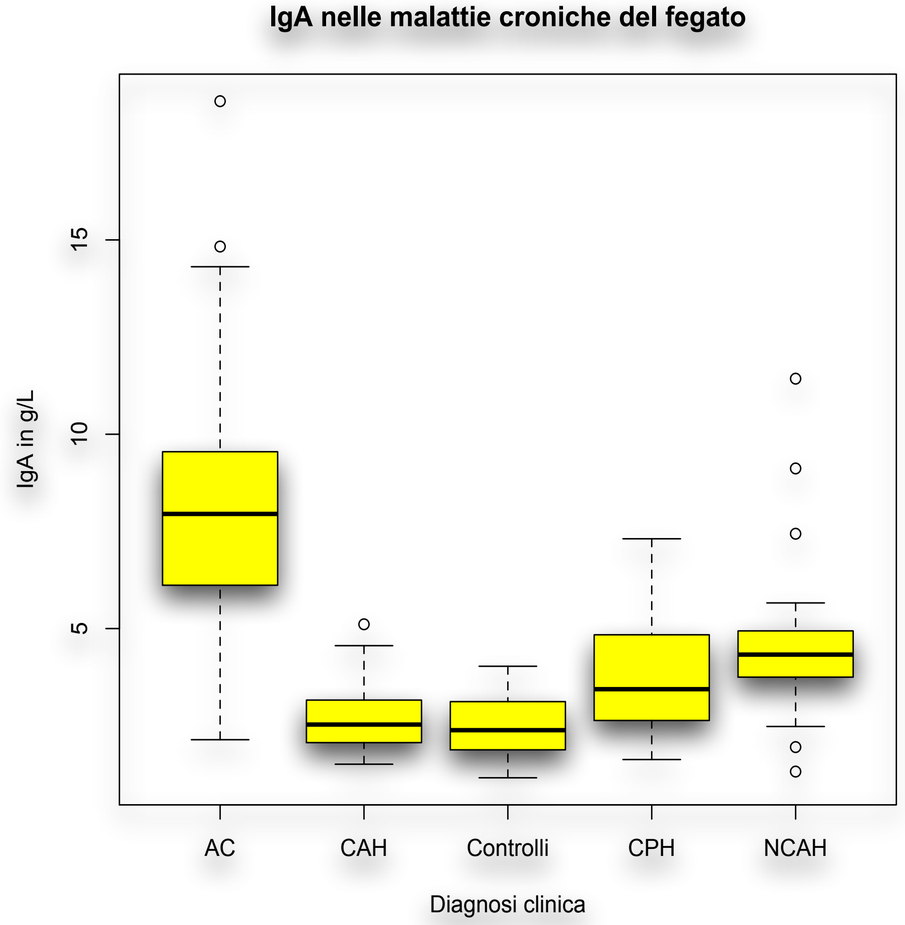
(Bio)Statistica con R – Parte III

Box&Whiskers Plot

- Eseguiamo il seguente codice:

```
# importo i dati
> mydata <- read.table("Boxplot.csv", header=TRUE,
   sep=";")
# traccio i boxplot delle IgA per ciascuna diagnosi
>boxplot(IgA~Diagnosi, data=mydata,
   main = "IgA nelle malattie croniche del fegato",
   xlab = "Diagnosi clinica", ylab = "IgA in g/L",
   notch = FALSE, outline = TRUE, col = "yellow")
```

- Non è necessario specificare il numero di box & whiskers plot da tracciare, che viene desunto direttamente dai dati aggregando i valori di IgA per Diagnosi (**IgA~Diagnosi**): quindi in questo caso è uguale al numero delle diverse diagnosi.
- Il parametro **outline=TRUE** indica di lasciare all'esterno dei baffi come punti separati i dati che presentano uno scostamento eccessivo (outliers) (vedi figura a sinistra):



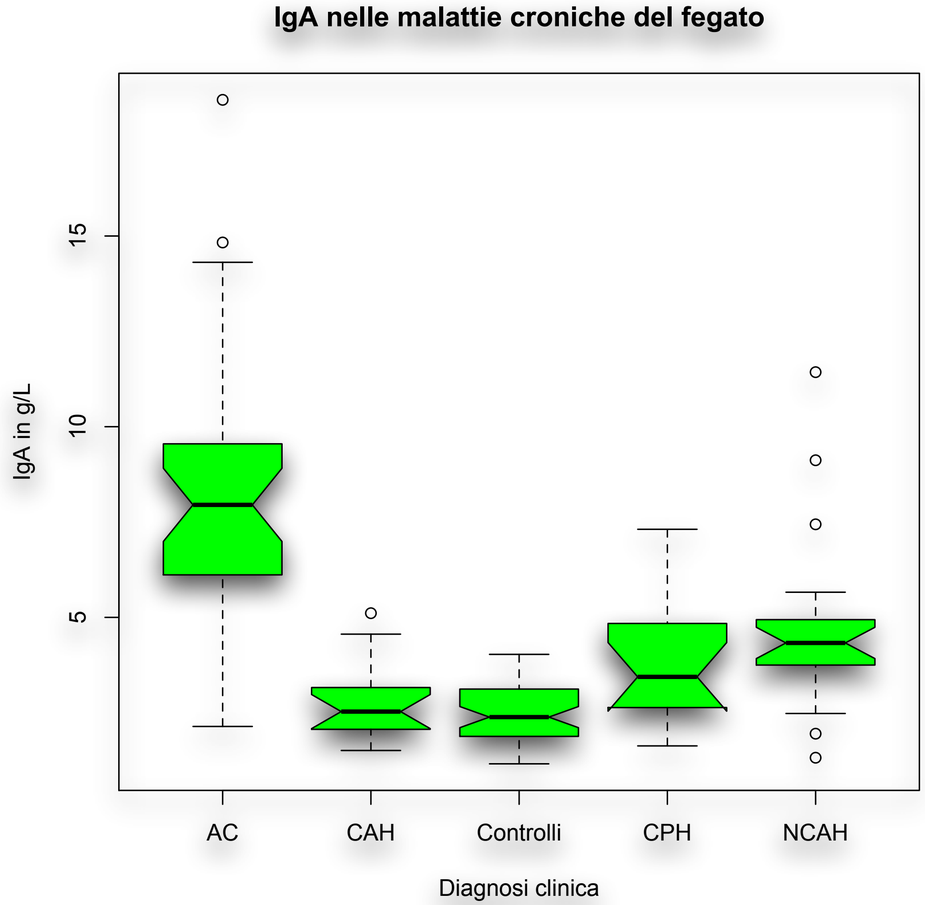
(Bio)Statistica con R — Parte III

Box&Whiskers Plot

- Ecco cosa accade invece con il parametro **notch=TRUE**:

```
# traccio i boxplot delle IgA per ciascuna diagnosi con i notch (incisure)
> boxplot(IgA~Diagnosi, data=mydata,
    main="IgA nelle malattie croniche del fegato",
    xlab="Diagnosi clinica", ylab="IgA in g/L",
    notch=TRUE, col="green")
```

- In questo caso sono tracciati i boxplot delle IgA per ciascuna diagnosi con una incisura che rappresenta i limiti di confidenza al 95% della mediana (figura a destra). Questo corrisponde ad un test per la significatività della differenza tra le mediane.
- Se le incisure di due boxplot non si sovrappongono la mediana delle due distribuzioni è significativamente diversa.



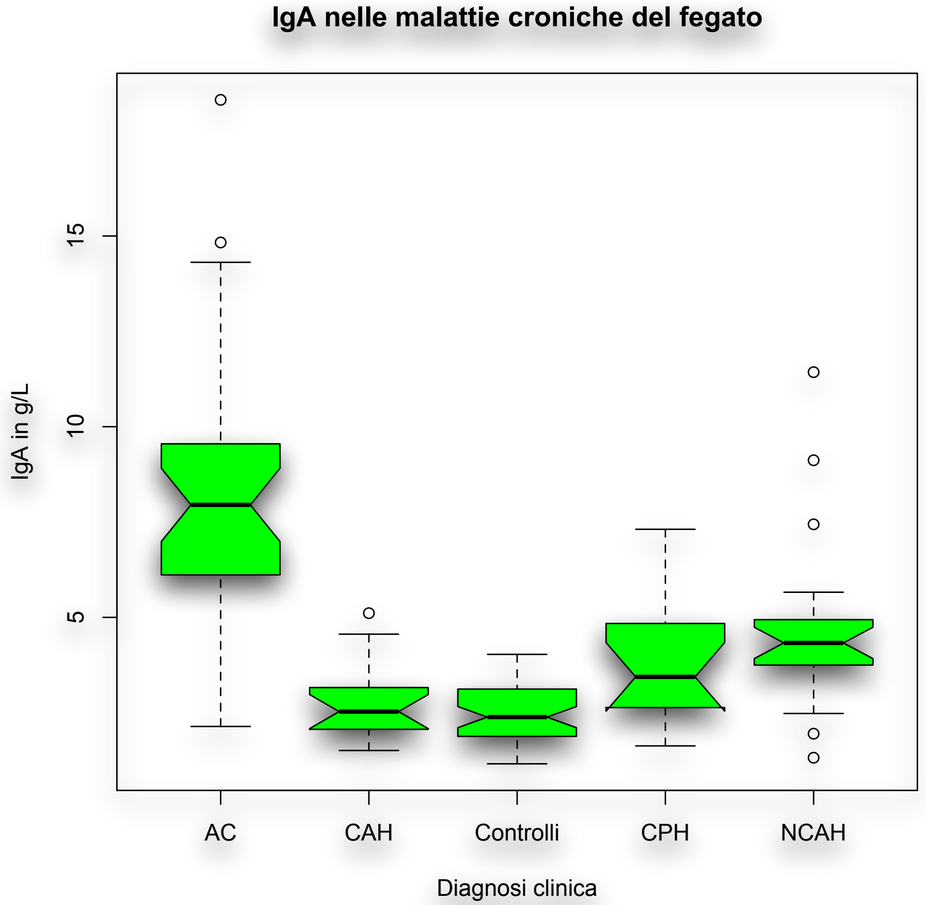
(Bio)Statistica con R — Parte III

Box&Whiskers Plot

- Il fatto interessante è che vediamo qui utilizzata una rappresentazione grafica per effettuare un test statistico (un confronto tra mediane).
- Si noti anche il messaggio che compare nella Console di R:

```
Warning message:
In bxp(list(stats = c(2.14, 6.115, 7.95, 9.55, 14.31, 1.51,
2.065, :
some notches went outside hinges ('box'): maybe set
notch=FALSE
```

- Il messaggio avverte che in alcuni casi le incisure sono uscite dai bordi della scatola (osservate i boxplot della CAH e della CPH). In questi casi il problema è determinato dal fatto che il numero delle osservazioni troppo ridotto determina un livello di incertezza che si estende al di la delle osservazioni.
- Vi sono solamente due modi per superare questo problema: **rinunciare** a trarre delle conclusioni da questi casi, o **aumentare** adeguatamente il numero delle osservazioni.



(Bio)Statistica con R — Parte III

Scatter Plot

- Si tratta di un diagramma cartesiano che mostra la "dispersione" dei dati.
- Scarichiamo e salviamo il file **Scatterplot.csv**, che una struttura molto tradizionale: una variabile per ogni colonna, nella prima riga i nomi delle variabili, nelle righe successive i loro valori:

GR	RGO	HB	HCT	HBA2	MCV	HBF	MCH	RDW	FERRO
4.90	97	13.3	40.6	1.8	82.8	0.6	27.1	17.3	106
4.66	81	10.8	34.3	2.6	73.6	1.6	23.2	21.5	148
5.43	57	11.5	36.1	4.8	66.5	2.5	21.1	21.0	104
5.41	63	10.8	39.7	2.5	73.4	1.8	20.0	19.9	74
4.94	60	10.4	32.3	1.4	65.0	0.7	21.1	23.7	17
4.30	97	12.1	35.8	1.9	83.3	0.7	28.2	18.3	43
5.03	96	13.7	40.7	2.4	81.0	0.9	27.3	19.0	101
5.12	93	13.6	42.2	2.9	82.6	0.8	26.7	17.8	50
4.04	82	11.9	36.1	1.9	89.4	0.6	29.4	16.0	79
4.48	95	13.1	40.5	1.5	90.4	0.5	29.2	16.8	77

(Bio)Statistica con R — Parte III

Scatter Plot

- Le variabili contenute nel file sono la concentrazione degli eritrociti (GR) espressa in 10^12/L, la resistenza globulare osmotica (RGO) in %, la concentrazione dell'emoglobina (HB), in g/dL, l'ematocrito (HCT) in %, l'emoglobina A2 (HBA2) espressa in % dell'emoglobina totale, il volume globulare medio (MCV) in fL, l'emoglobina F (HBF) espressa in % dell'emoglobina totale, l'emoglobina corpuscolare media (MCH) in pg, l'ampiezza della distribuzione dei globuli rossi (RDW) espressa in % (come coefficiente di variazione), e infine la concentrazione del ferro nel siero in µg/dL, misurati in 643 soggetti che includevano controlli sani, soggetti portatori di beta-talassemia, portatori di alfa-talassemia, e soggetti con anemia sideropenica.
- Da notare che sono utilizzate la libreria **car** e la libreria **gclus** che, se necessario, vanno installate e attivate con i comandi:

```
> install.packages("car"); install.packages("gclus")
> library(car); library(gclus)
```

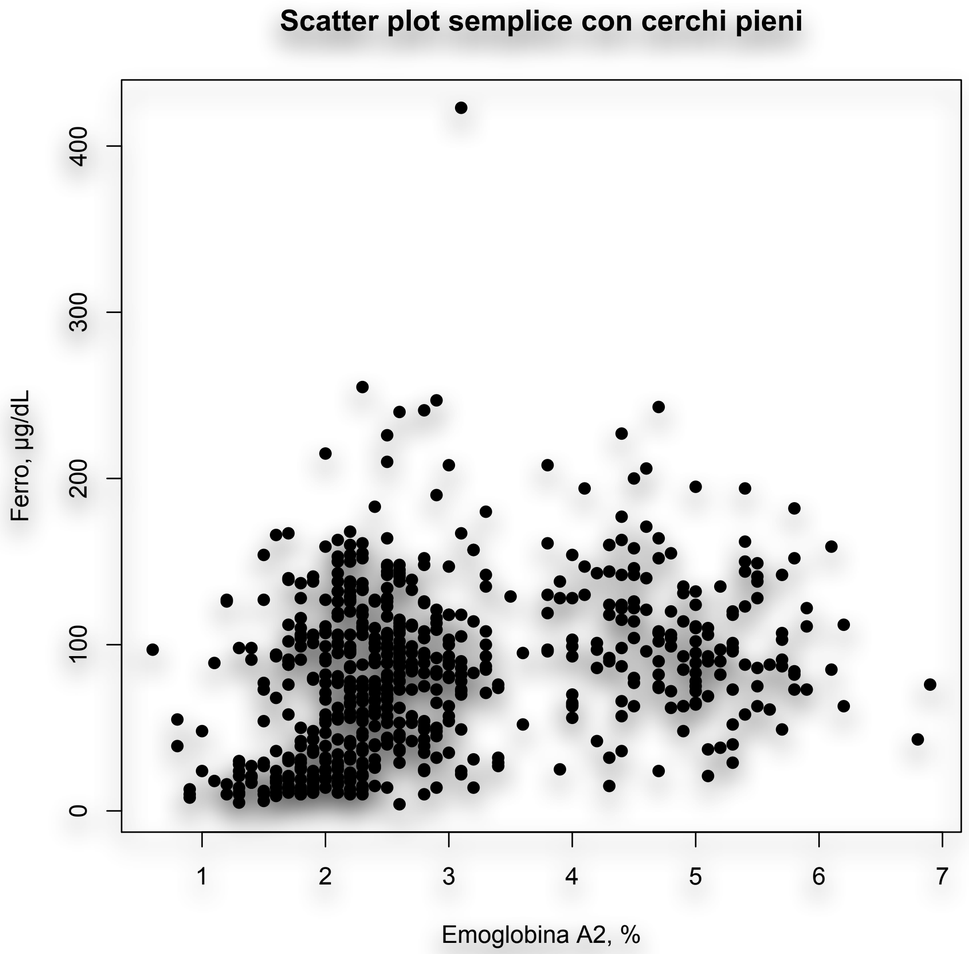
(Bio)Statistica con R — Parte III

Scatter Plot

- Eseguiamo il seguente codice:

```
# importo i dati
> mydata <- read.table("Scatterplot.csv",
    header=TRUE, sep=";")
# traccio uno scatter plot semplice
> attach(mydata)
> plot(HBA2, FERRO,
  main="Scatter plot semplice con cerchi pieni",
  xlab="Emoglobina A2, %", ylab="Ferro, µg/dL",
  pch=19)
```

- Viene prodotto un semplice diagramma cartesiano che rappresenta la concentrazione del ferro in funzione della concentrazione di emoglobina A2 (figura a sinistra).
- I singoli punti sono rappresentati mediante cerchi pieni.



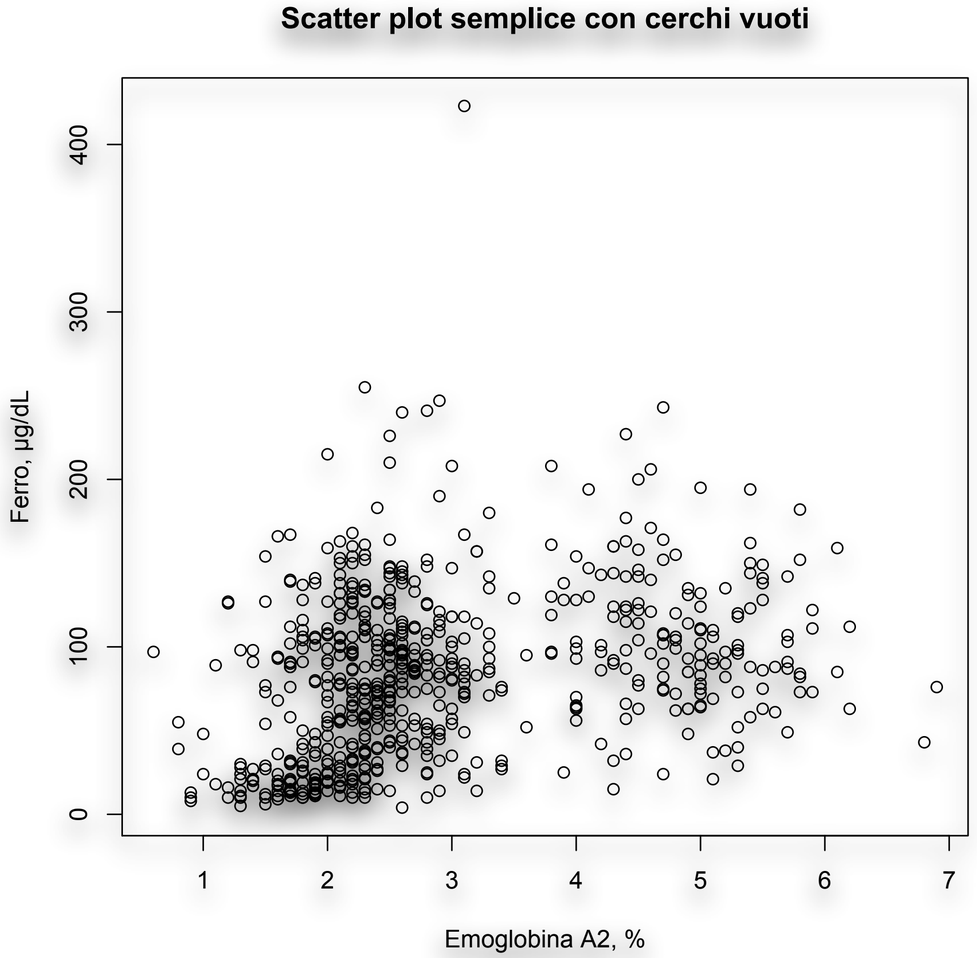
(Bio)Statistica con R — Parte III

Scatter Plot

- Cambiamo lo stile dei punti:

```
> plot(HBA2, FERRO,
  main="Scatter plot semplice con cerchi
  vuoti", xlab="Emoglobina A2, %",
  ylab="Ferro, µg/dL", pch=1)
```

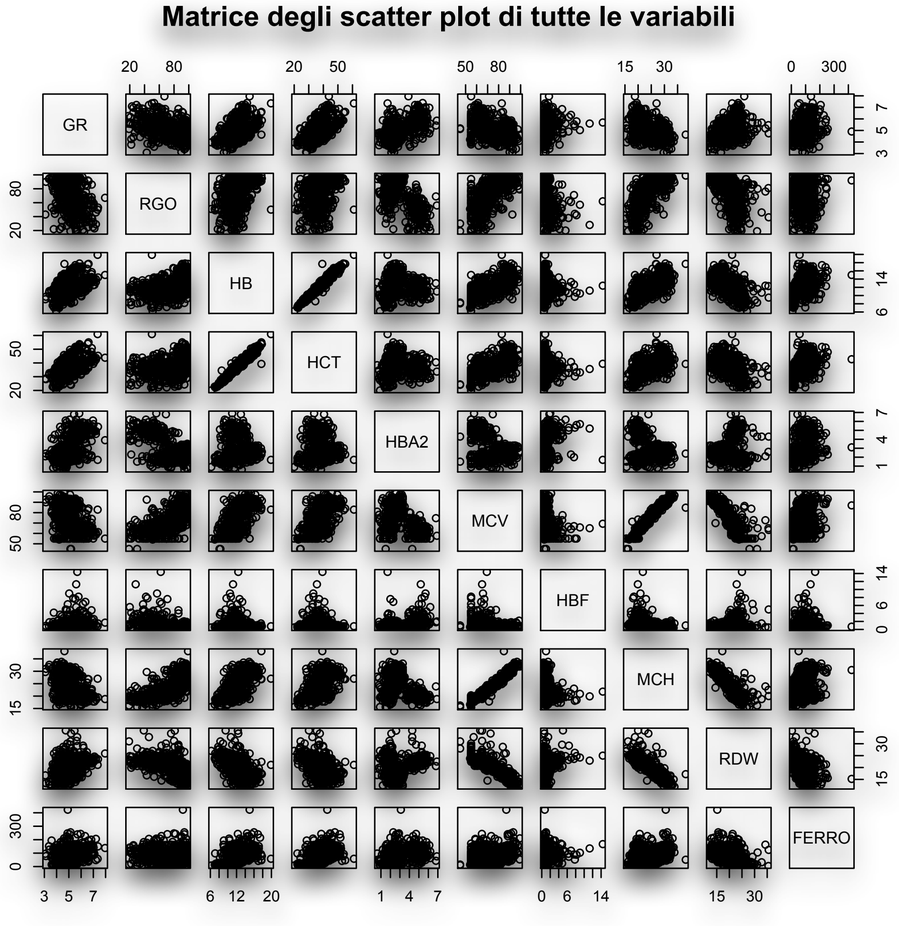
- Viene prodotto lo stesso diagramma cartesiano del caso precedente, ma questa volta il simbolo per rappresentare i dati è rappresentato da un cerchio vuoto (figura a destra).
- Le potenzialità di **R** nella rappresentazione di scatter plot vanno però ben oltre (vedi la prossima diapositiva).



(Bio)Statistica con R — Parte III

Scatter Plot

```
# traccio lo scatterplot con la matrice completa di tutte le variabili
> pairs( ~ GR + RGO + HB + HCT + HBA2 + MCV +
  HBF + MCH + RDW + FERRO, data=mydata,
  main="Matrice degli scatter plot di tutte le
  variabili")
```



- Viene generata la matrice degli scatter plot incrociando tra di loro tutte le variabili, e per ogni coppia di variabili viene effettuata una duplice rappresentazione, prima con l'una e poi con l'altra variabile in ascissa.
- La matrice degli scatter plot include controlli sani, soggetti portatori di beta-talassemia, portatori di alfa-talassemia, e soggetti con anemia sideropenica.

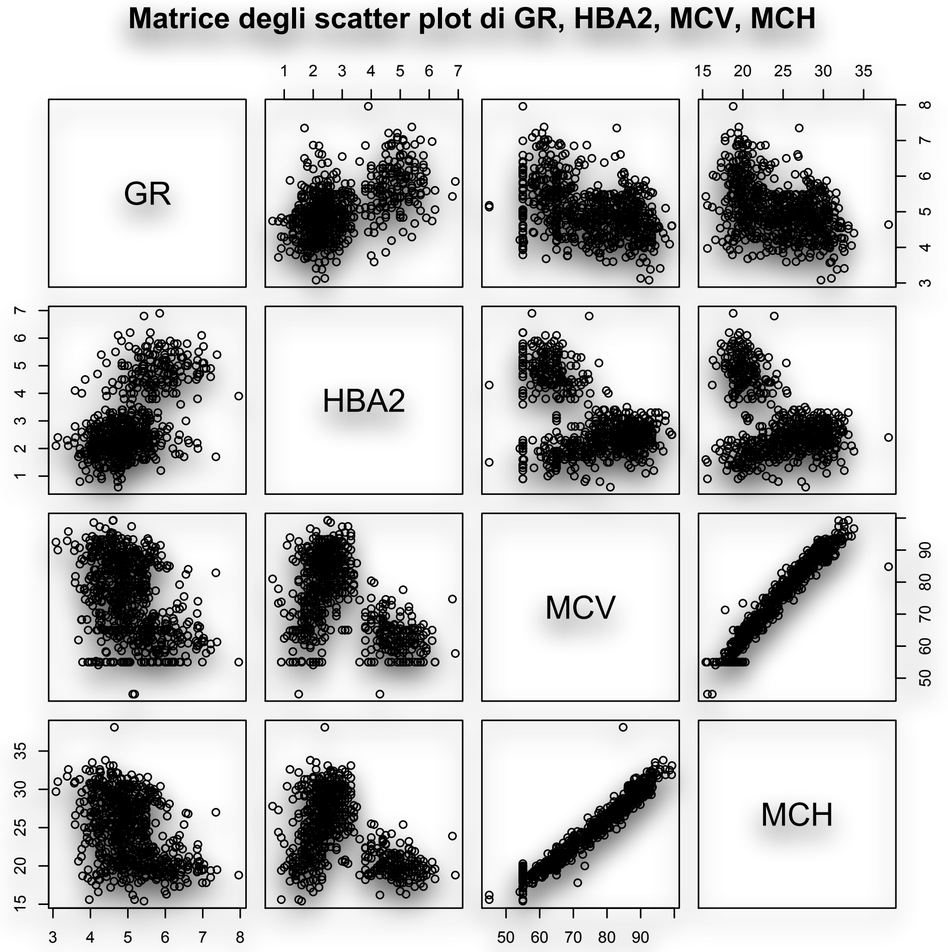
(Bio)Statistica con R — Parte III

Scatter Plot

- Come al punto precedente, ma con matrice parziale limitata a quattro variabili:

```
> pairs(~GR+HBA2+MCV+MCH, data = mydata,
  main = "Matrice degli scatter plot di GR,
  HBA2, MCV, MCH")
```

- La matrice degli scatter plot viene limitata a eritrociti (GR), emoglobina A2 (HBA2), volume globulare medio (MCV) ed emoglobina corpuscolare media (MCH).



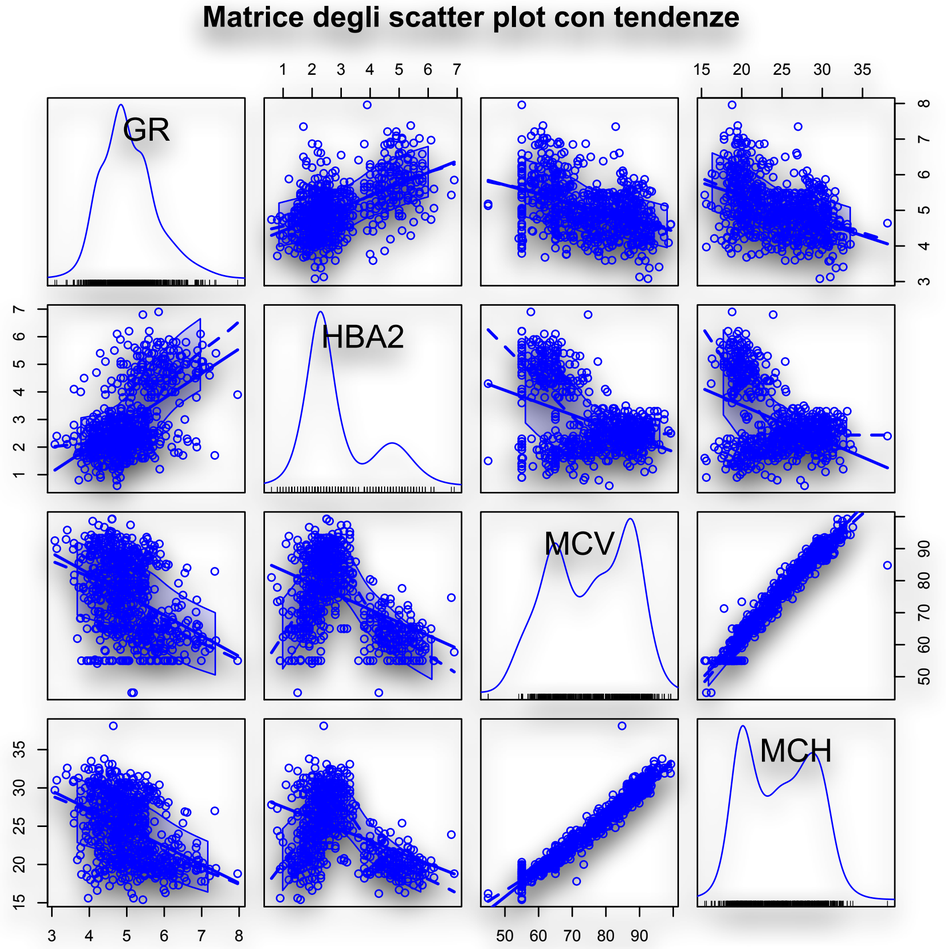
(Bio)Statistica con R — Parte III

Scatter Plot

- Altra rappresentazione; notare la nuova libreria **car** e il parametro **diagonal = TRUE**:

```
> library(car)
> scatterplotMatrix(~GR+HBA2+MCV+MCH,
    reg.line = lm, smooth = TRUE, span = 0.5,
    diagonal = "none", data = mydata,
    main = "Matrice degli scatter plot con
    tendenze")
```

- In questa rappresentazione ottenuta con l'impiego della libreria **car** sono riportate le curve che esprimono le tendenze medie dei dati a variare congiuntamente.
- Si noti che la relazione tra MCV e MCH è chiaramente lineare.



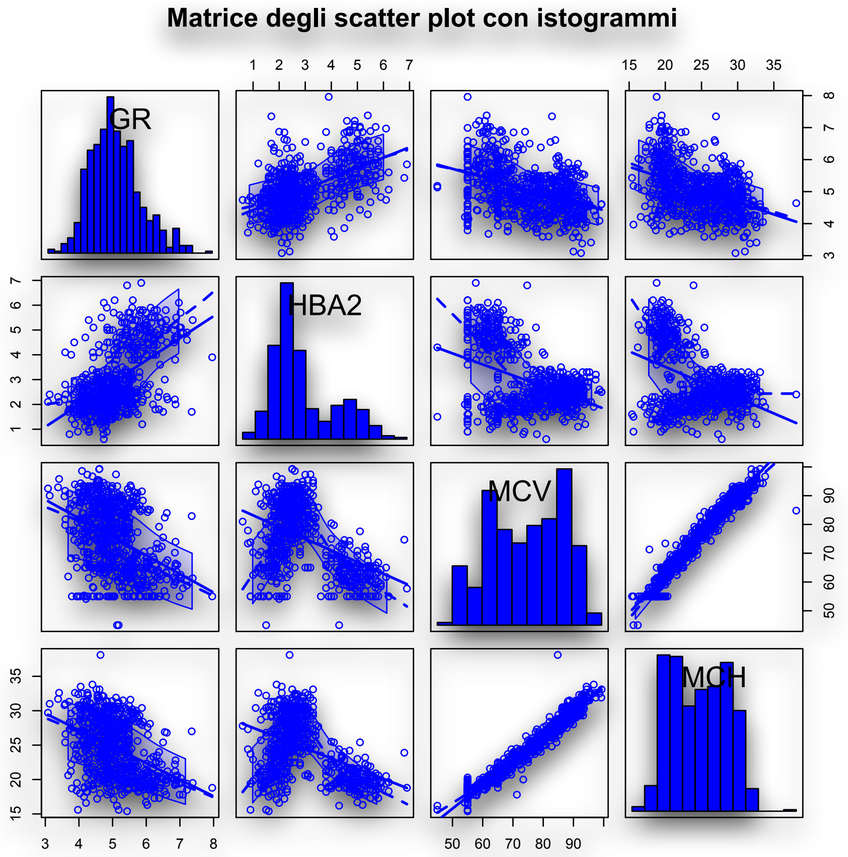
(Bio)Statistica con R — Parte III

Scatter Plot

- Come al punto precedente; da notare il parametro **diagonal** (nella diagonale sono ora rappresentati gli istogrammi delle distribuzioni):

```
> scatterplotMatrix(~GR+HBA2+MCV+MCH,
  reg.line=lm, smooth=TRUE, span=0.5,
  diagonal=list(method ="histogram",
  breaks="FD"), data=mydata, main="Matrice degli
  scatter plot con istogrammi")
```

- Il parametro **diagonal** ammette, oltre ai valori precedenti **"none"** e **"histogram"**, anche i seguenti valori: **"boxplot"**, **"density"**,**"oned"**, **"qqplot"**.



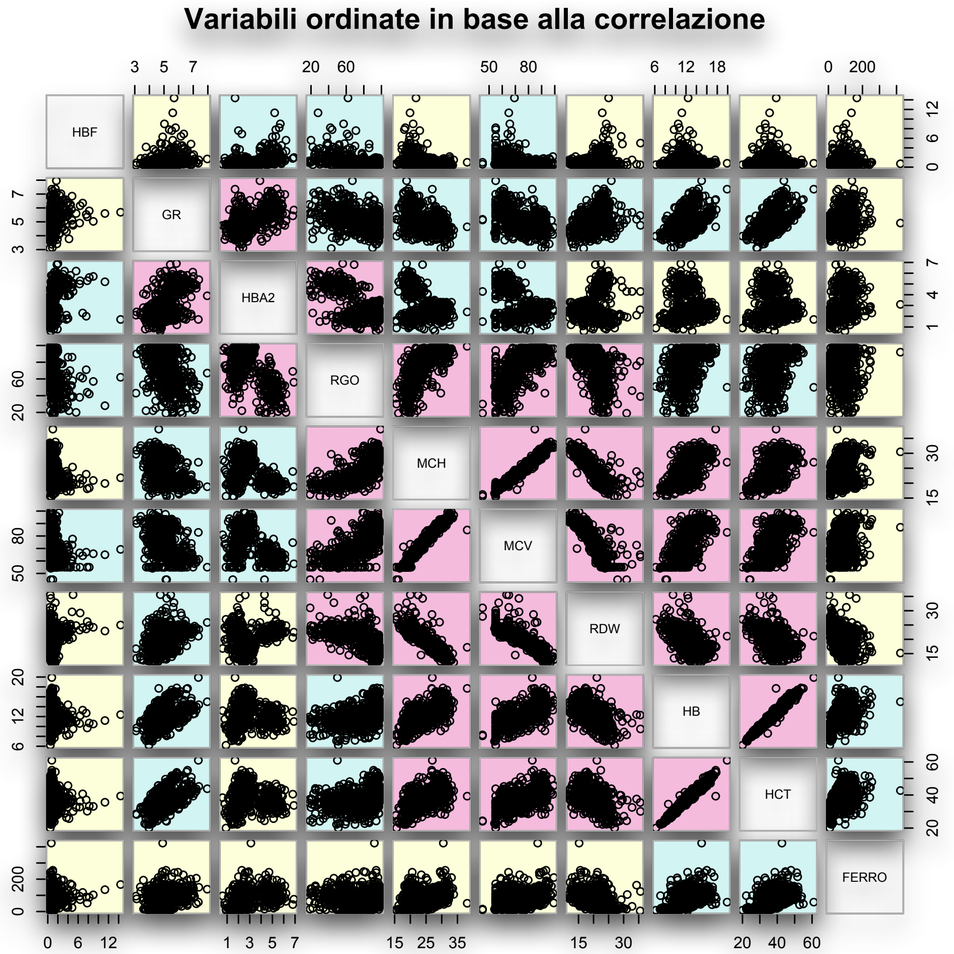
(Bio)Statistica con R – Parte III

Scatter Plot

- Una nuova opportunità nella rappresentazione degli scatter plot sotto forma di matrici è offerta dalla libreria **gclus** con il codice che segue:

```
library(gclus) # questo scatterplot necessita della libreria gclus
> df <- mydata[c(1,2,3,4,5,6,7,8,9,10)] # recupero i dati dalle colonne
> df.r <- abs(cor(df)) # calcolo la correlazione
> df.col <- dmat.color(df.r) # applico i colori
> df.o <- order.single(df.r) # riordino le variabili in modo che quelle meglio correlate siano vicine alla diagonale
> cpairs(df, df.o, panel.colors=df.col, gap=0.5,
  main="Variabili ordinate in base alla correlazione")
```

- Le variabili sono colorate e ordinate in base alla maggiore o minore correlazione esistente tra di loro: quelle meglio correlate (in rosa) sono collocate accanto alla diagonale, le altre sono collocate andando dalla diagonale verso la periferia via via che la correlazione diminuisce (vedi figura a lato).



(Bio)Statistica con R — Parte III

Scatter Plot 3D

- Per gli scatte plot tridimensionali (3D) utilizzeremo gli stessi dati precedenti (file Scatterplot.csv) le librerie **scatterplot3d**, **rgl** e **Rcmdr**, che vanno installate ed attivate.

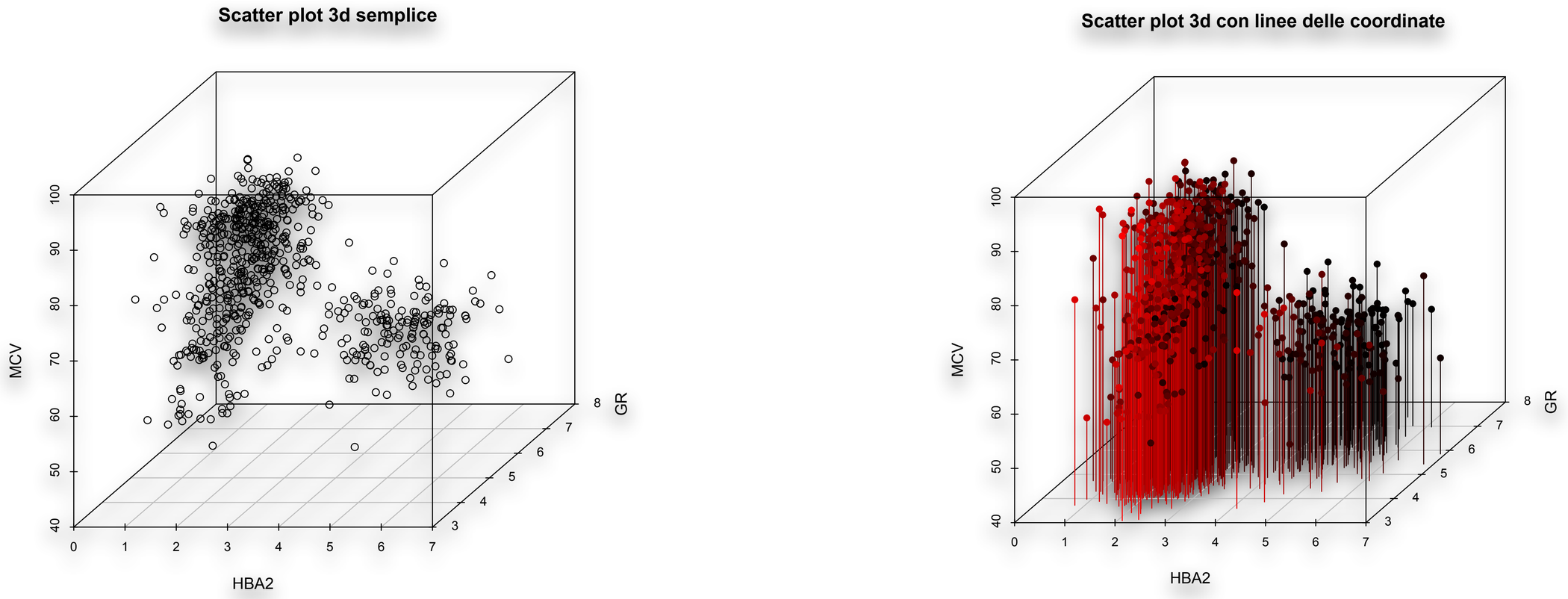
```
> mydata <- read.table("Scatterplot.csv", header=TRUE, sep=";") # importo i dati
# Traccio lo scatter plot tridimensionale (3D) semplice (serve la libreria apposita)
>library(scatterplot3d)
>attach(mydata)
>scatterplot3d(HBA2,GR,MCV, main="Scatter plot 3d semplice")
```

- Lo scatter plot 3D ottenuto è quello molto semplice di base (figura a sinistra della diapositiva seguente).
- Il seguente comando traccia invece la proiezione dei punti sulle coordinate orizzontali per meglio identificare la loro posizione (figura a destra della diapositiva seguente):

```
> scatterplot3d(HBA2, GR, MCV, pch=16, highlight.3d=TRUE, type="h",
  main="Scatter plot 3d con linee delle coordinate”)
```

(Bio)Statistica con R — Parte III

Scatter Plot 3D



Scatter plot 3D (x,y,z) della concentrazione dell'emoglobina A2 negli eritrociti (HBA2 in %, asse x), dei globuli rossi (GR in 10^{12}/litro, asse y) e volume globulare medio (MCV in fL, asse z), con i soli punti (a sinistra) e con la proiezioni dei punti sul piano delle coordinate orizzontali x,y (a destra).

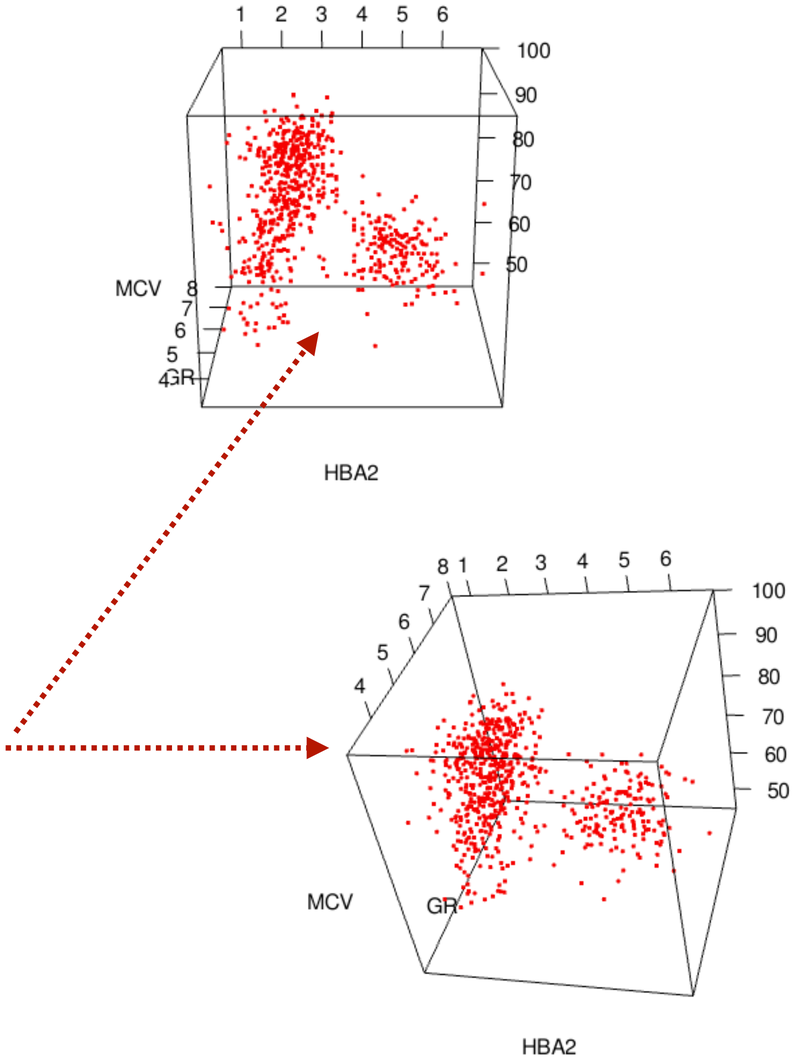
(Bio)Statistica con R — Parte III

Scatter Plot 3D

- Con la libreria **rgl** è possibile realizzare un grafico 3D che può essere ruotato al fine di orientare i dati secondo la prospettiva che li coglie al meglio:

```
# spinning 3D scatter plot (necessita della libreria rgl)
> install.packages("rgl"); library(rgl)
> mydata <- read.table("Scatterplot.csv", header=TRUE,
    sep=";"); attach(mydata)
> axes3d(); bg3d("white")
> plot3d(HBA2,GR,MCV, type="p", col="red", size=3)
```

- Se si "afferra" il grafico 3D facendo click con il tasto sinistro del mouse e lo si tiene premuto senza rilasciarlo, lo si può ruotare a piacimento (vedi figura a lato).

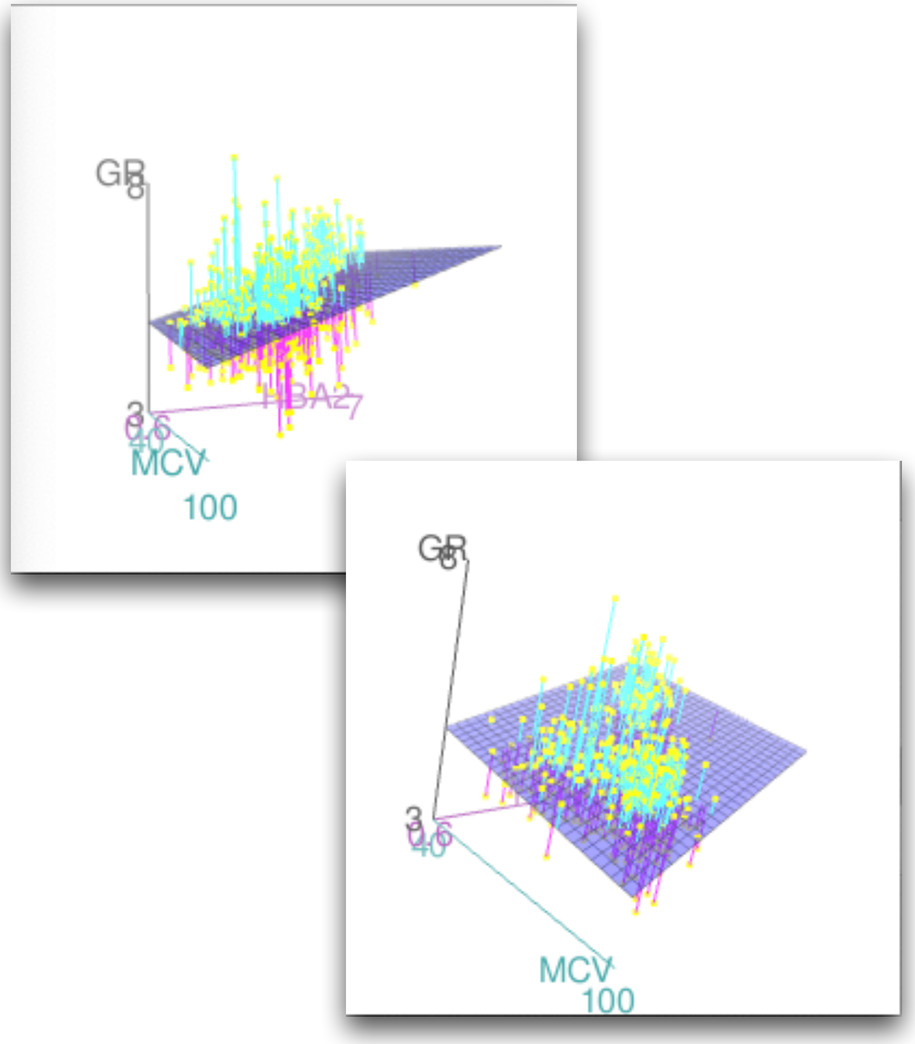


(Bio)Statistica con R — Parte III

Scatter Plot 3D

- Per eseguire lo script che segue occorre chiudere completamente **R** e riaprirlo per inizializzarlo.
- Lo script prevede l'utilizzo della libreria **Rcmdr** che consente anche in questo caso di realizzare in grafico 3D che può essere ruotato al fine di orientare i dati secondo la prospettiva che li coglie al meglio:

```
# spinning 3D scatter plot, necessita la libreria Rcmdr
> install.packages("Rcmdr"); library(Rcmdr)
> mydata <- read.table("Scatterplot.csv",
    header=TRUE, sep=";"); attach(mydata)
> scatter3d(HBA2,GR,MCV)
```



(Bio)Statistica con R — Parte III

Correlogrammi

- Per realizzare i correlogrammi viene utilizzata la libreria **corrgram** che va installata e attivata.
- Per gli esempi riutilizziamo il file **Scatterplot.csv** già visto nella parte precedente dedicata agli scatter plot (bidimensionali):

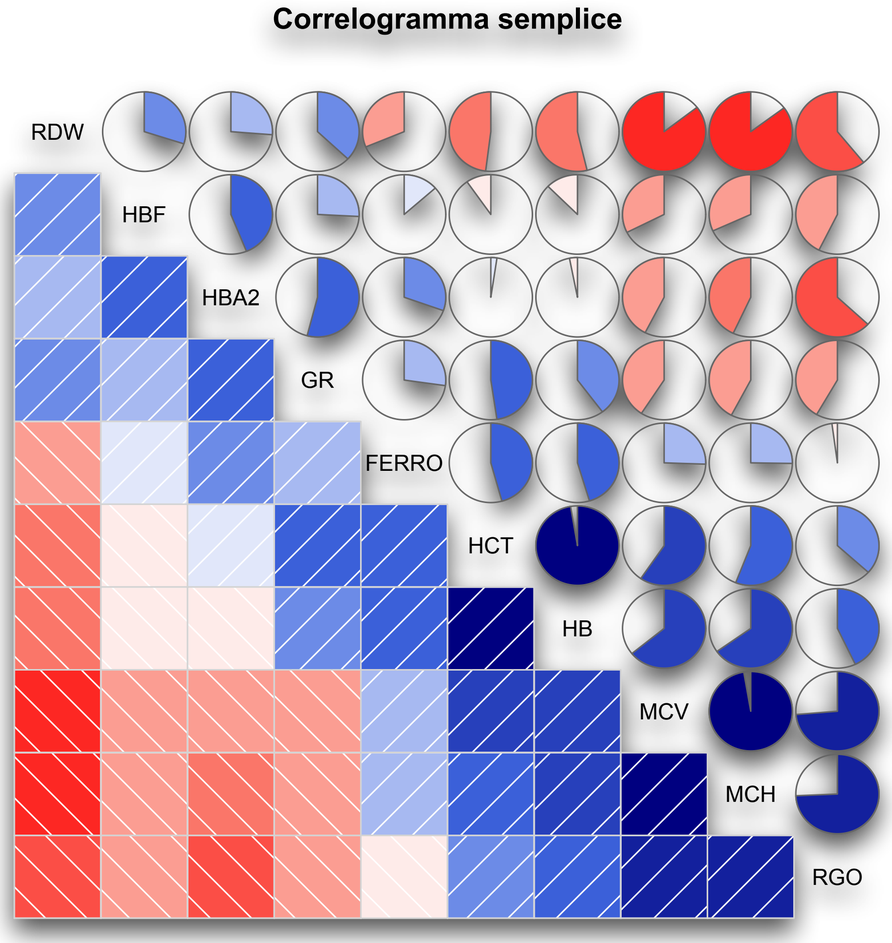
```
# importo i dati
> mydata <- read.table("Scatterplot.csv", header=TRUE, sep=";")
# correlogramma semplice
> install.packages(“corrgram")
> library(corrgram)
> corrgram(mydata, order=TRUE, lower.panel=panel.shade, upper.panel=panel.pie,
    text.panel=panel.txt, main="Correlogramma semplice")
```

- Nella figura della diapositiva seguente vediamo come appare il correlogramma.

(Bio)Statistica con R — Parte III

Correlogrammi

- L'ampiezza della colorazione della torta misura il coefficiente di correlazione (torta completamente bianca $r = 0$, torta completamente colorata $r = 1$).
- I valori dei coefficienti di correlazione vanno decrescendo dalla diagonale centrale verso la periferia.
- In blu sono riportati i valori positivi di r (le due grandezze aumentano e diminuiscono congiuntamente), in rosso i valori negativi di r (all'aumentare di una delle due grandezze l'altra diminuisce e viceversa).



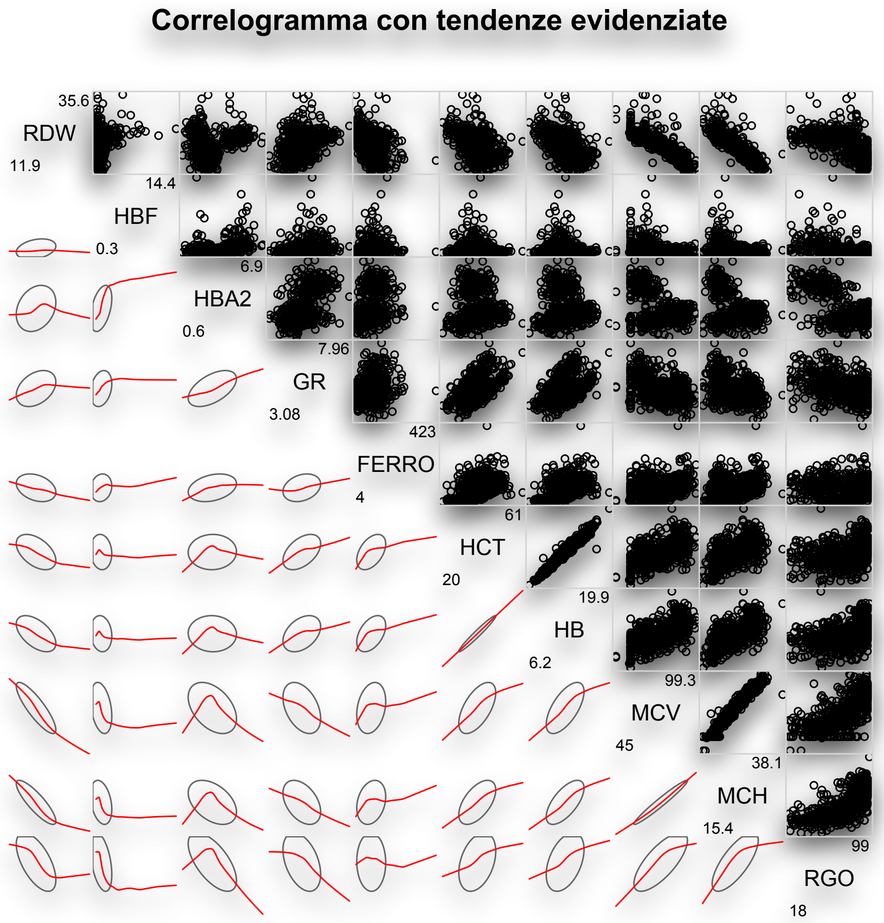
(Bio)Statistica con R — Parte III

Correlogrammi

- Correlogramma con tendenze evidenziate:

```
> corrgram(mydata, order=TRUE,
  lower.panel=panel.ellipse,
  upper.panel=panel.pts, text.panel=panel.txt,
  diag.panel=panel.minmax,
  main="Correlogramma con tendenze evidenziate")
```

- In questo caso nel quadrante superiore sono riportati i diagrammi di dispersione (scatter plot) e nel quadrante inferiore sono riportate le rette o le curve che esprimono le tendenze medie dei dati a variare congiuntamente.



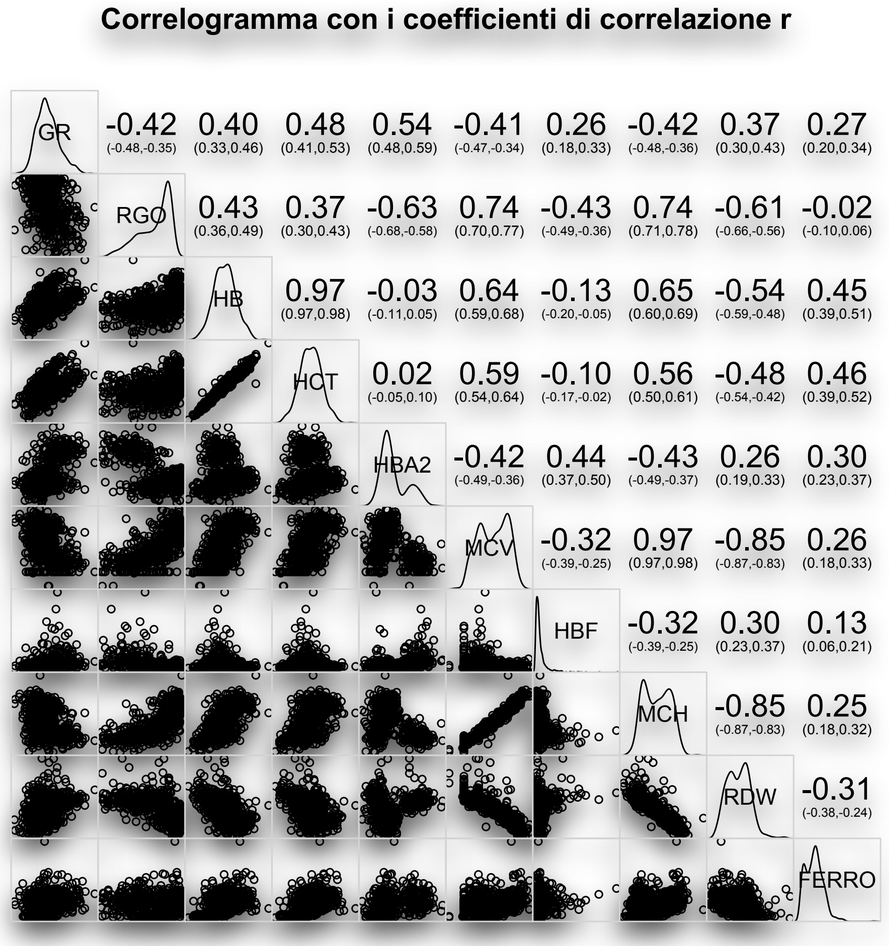
(Bio)Statistica con R — Parte III

Correlogrammi

- Correlogramma con i coefficienti di correlazione e i loro limiti di confidenza:

```
> corrgram(mydata, lower.panel=panel.pts,
    upper.panel=panel.conf, diag.panel=panel.density,
    main="Correlogramma con i coefficienti di
    correlazione r")
```

- In questa forma di correlogramma nella diagonale sono riportate le distribuzioni delle variabili sotto forma di kernel density plot, nel quadrante inferiore i diagrammi di distribuzione (scatter plot) e nel quadrante superiore il valore del coefficiente di correlazione r con i limiti di confidenza al 95%.

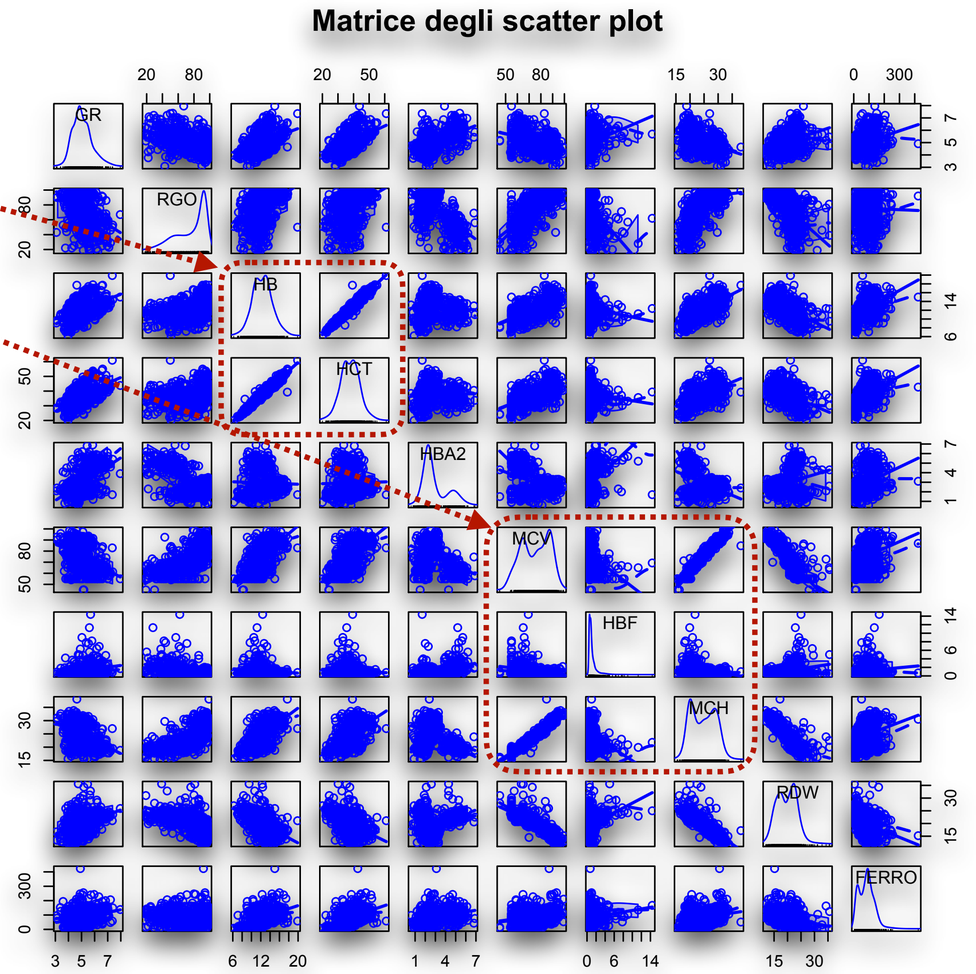


(Bio)Statistica con R — Parte III

Correlogrammi

- In realtà anche un più tradizionale scatter plot aiuta a cogliere le forti correlazioni che intercorrono tra emoglobina (HB) ed ematocrito (HCT) e tra emoglobina corpuscolare media (MCH) e volume globulare medio (MCV):

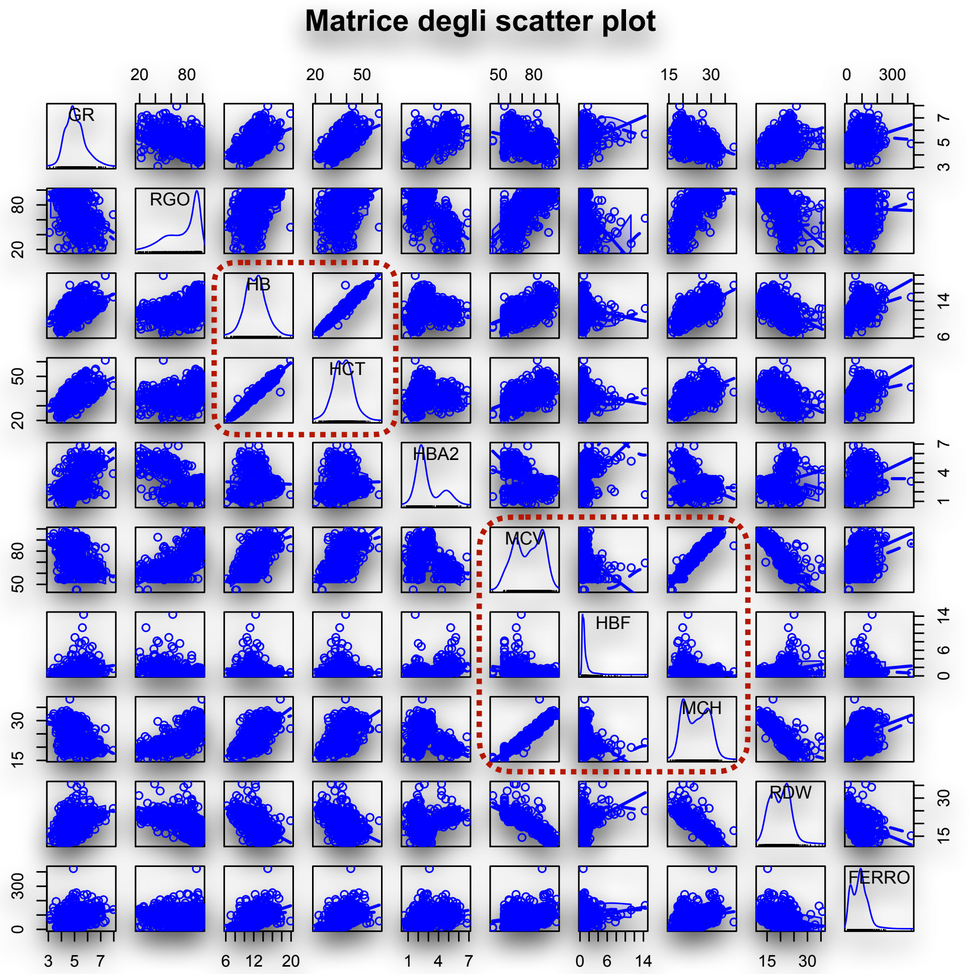
```
> library(car)
> scatterplotMatrix( ~ GR + RGO + HB + HCT +
  HBA2 + MCV + HBF + MCH + RDW + FERRO,
  reg.line=lm, smooth=TRUE,
  span=0.5, diagonal = "density",
  main="Matrice degli scatter plot",
  data=mydata)
```



(Bio)Statistica con R — Parte III

Correlogrammi

- Osserviamo che il fatto che due variabili siano correlate non dice nulla sui possibili rapporti causa-effetto.
- Anzi, è possibile che siano "evidentemente" correlati dal punto di vista statistico fatti che in realtà sono completamente slegati tra di loro.
- Nonostante ciò, quando utilizzata in modo appropriato la correlazione può essere utile.
- Ed è quello che accade quando, come nel casi dei correlogrammi, il coefficiente di correlazione viene integrato con una rappresentazione grafica dei dati che aiuta a fare emergere i legami fra le variabili in esame.



(Bio)Statistica con R — Parte III

Curve ROC

- Scarichiamo e salviamo i file **CurveROC.csv** e **CurveROCbis.csv**. Il contenuto di entrambi i file apparirà così (cambiano solamente i valori), con i nomi delle variabili nella prima riga e i dati dei singoli casi nelle righe successive:

predictions	labels
19	0
22	0
22	1
24	1
24	1
26	0
27	1
28	0
29	0

- La variabile "predictions" contiene i valori misurati (in questo caso il risultato numerico di una analisi di laboratorio) mentre la variabile "labels" contiene la classificazione dei casi, e riporta 0 per i controlli (soggetti sani) e 1 per i soggetti malati.
- Utilizzeremo per gli esempi le librerie **pROC** e **sm** che vanno installata e attivate (se non lo sono già).

(Bio)Statistica con R — Parte III

Curve ROC

- Eseguiamo il seguente codice:

```
# importo i dati
> mydata <- read.table("CurveROC.csv", header=TRUE, sep=";")
# visualizzo i nomi delle variabili
> names(mydata)
# elenco i primi 10 casi
> head(mydata, n=10)
# elenco gli ultimi 5 casi
> tail(mydata, n=5)
# utilizzo la libreria pROC e imposto la variabile mydata
> library(pROC)
> attach(mydata)
# traccio la curva ROC e calcolo l'area sotto la curva (AUC)
> roc(mydata$labels, mydata$predictions, smooth = FALSE, auc = TRUE, ci = FALSE, plot = TRUE,
  identity = TRUE, main = "Curva ROC", xlab="1-specificità", ylab = "Sensibilità")
```

(Bio)Statistica con R — Parte III

Curve ROC

- Esaminiamo il risultato dell'esecuzione del comando **roc**:

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Call:

roc.default(response = mydata$labels, predictor = mydata$predictions,

smooth = FALSE, auc = TRUE, ci = FALSE, plot = TRUE, identity = TRUE,

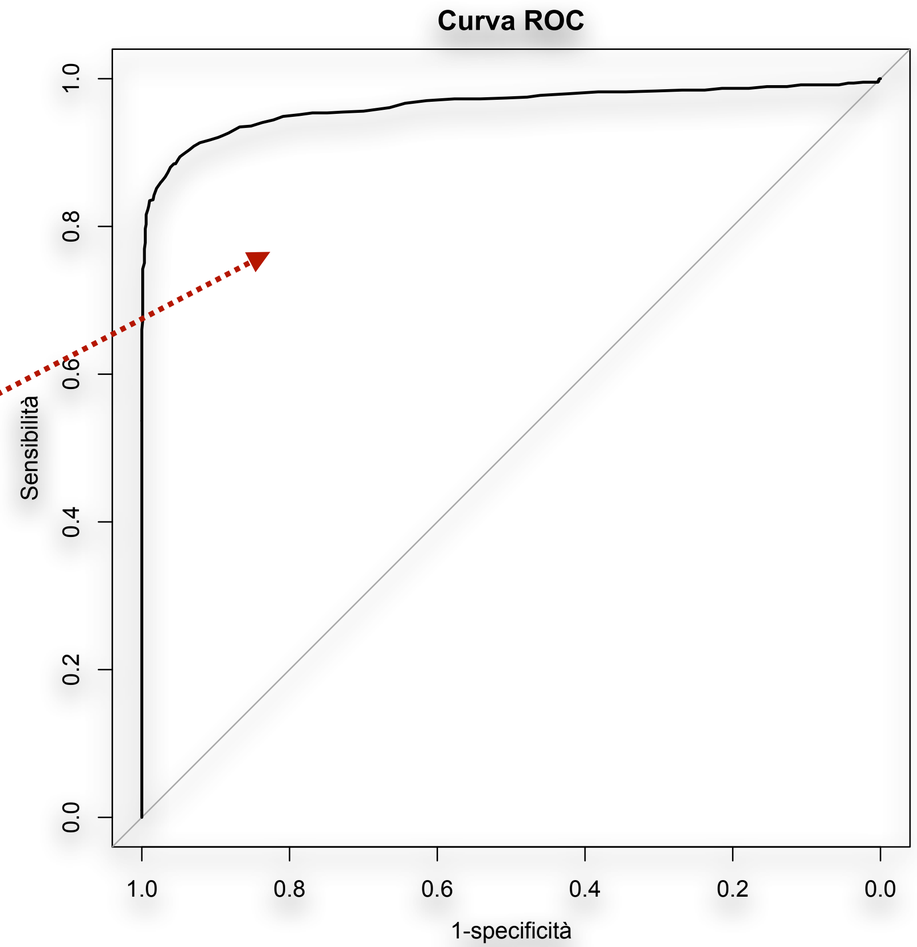
main = "Curva ROC", xlab = "1-specificità", ylab = "Sensibilità")

Data:

mydata$predictions in 853 controls (mydata$labels 0)

< 842 cases (mydata$labels 1).

Area under the curve: 0.9633



(Bio)Statistica con R — Parte III

Curve ROC

- Con il codice che segue sono infine calcolate le statistiche della curva ROC.

```
# intervallo di confidenza al 95% dell'area sotto la curva, metodo di DeLong:
> ci.auc(mydata$labels, mydata$predictions, conf.level = 0.95)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
95% CI: 0.9537-0.9729 (DeLong)
# intervallo di confidenza al 95% della sensibilità per valori di specificità da 0 a 1 con passo 0.1:
> ci.se(mydata$labels, mydata$predictions, specificities=seq(0,1,.1), conf.level = 0.95, boot.n = 100)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
  |=========================================================================================| 100%
95% CI (100 stratified bootstrap replicates):
  sp se.low se.median se.high
 0.0 1.0000    1.0000  1.0000
 0.1 0.9857    0.9917  0.9988
 …
 0.9 0.9025    0.9192  0.9425
 1.0 0.6358    0.6805  0.7874
```

(Bio)Statistica con R — Parte III

Curve ROC

intervallo di confidenza al 95% della specificità per valori di sensibilità da 0 a 1 con passo 0.1:

```
> ci.sp(mydata$labels, mydata$predictions, sensitivities=seq(0,1,.1), conf.level = 0.95, boot.n = 100)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
  |=========================================================================================| 100%
95% CI (100 stratified bootstrap replicates):
  se sp.low sp.median sp.high
 0.0 1.0000  1.000000 1.00000
 0.1 1.0000  1.000000 1.00000
 0.2 1.0000  1.000000 1.00000
 0.3 1.0000  1.000000 1.00000
 0.4 1.0000  1.000000 1.00000
 0.5 1.0000  1.000000 1.00000
 0.6 1.0000  1.000000 1.00000
 0.7 0.9965  0.998800 1.00000
 0.8 0.9875  0.994100 0.99880
 0.9 0.8903  0.941200 0.96140
 1.0 0.0000  0.001172 0.01263
```

(Bio)Statistica con R — Parte III

Curve ROC

calcolo il miglior valore soglia tra sani e malati e l'intervallo di confidenza al 95% della sensibilità e della specificità corrispondenti:

```
> ci.thresholds(mydata$labels, mydata$predictions, thresholds="best", conf.level = 0.95, boot.n = 100)

Setting levels: control = 0, case = 1
Setting direction: controls < cases
  |=========================================================================================| 100%
95% CI (100 stratified bootstrap replicates):
 thresholds sp.low sp.median sp.high se.low se.median se.high
       74.5 0.9379     0.949  0.9637 0.8747    0.8955  0.9133
```

calcolo per le principali grandezze i valori corrispondenti al valore soglia tra sani e malati:

```
> myroc <-roc(mydata$labels, mydata$predictions, plot = FALSE)

Setting levels: control = 0, case = 1
Setting direction: controls < cases

> coords(myroc, "best", best.method = "youden", ret=c("threshold", "specificity", "sensitivity",
  "accuracy", "tn", "tp", "fn", "fp", "npv", "ppv"))
          threshold specificity sensitivity  accuracy  tn  tp fn fp       npv      ppv
threshold      74.5   0.9484174   0.8942993 0.9215339 809 753 89 44 0.9008909 0.944793
```

N.B.: **tn**=True Negative | **tp**=True Positive | **fn**=False Negative | **fp**=False Positive | **npv**=Negative Predicted Value | **ppv**=Positive Predicted Value (Precision)

(Bio)Statistica con R — Parte III

Curve ROC

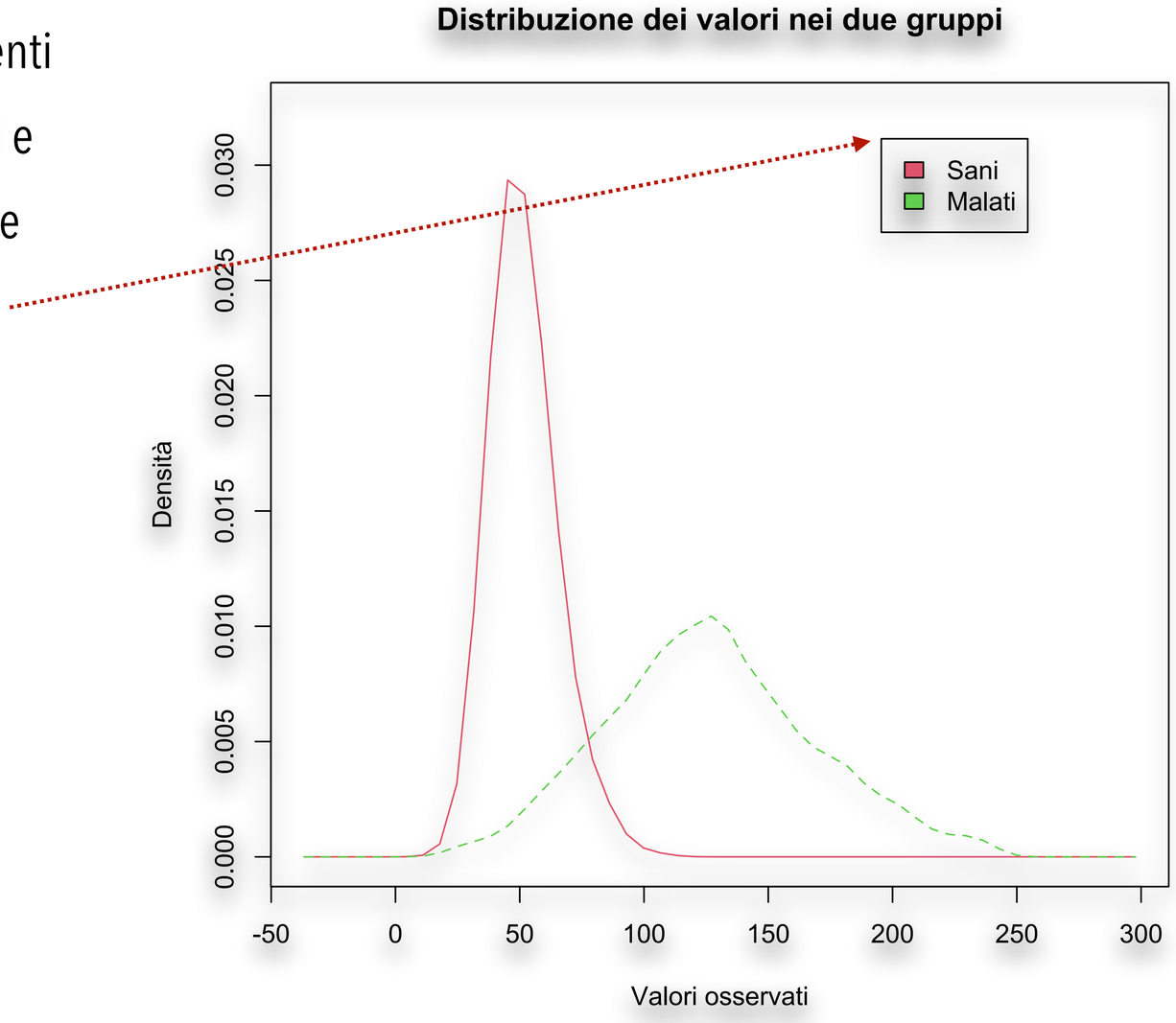
- Vediamo un grafico che mostra, sovrapposte, le distribuzioni dei valori nei sani e nei malati.

```
# traccio i kernel density plot sovrapposti dei valori osservati per controlli sani (0) e malati (1)
> library(sm)
> attach(mydata)
# attenzione: il primo "labels" è la variabile che contiene i valori osservati, il secondo "labels" sono le etichette da applicare come legenda
> myplot <- factor(labels, levels= c("0","1"), labels = c("Sani", "Malati"))
# traccio i due grafici sovrapposti
> sm.density.compare(predictions, labels, xlab="Valori osservati", ylab="Densità")
> title(main="Distribuzione dei valori nei due gruppi")
# aggiungiamo la legenda (fare click con tasto sinistro del mouse dove la si desidera far comparire)
> colfill<-c(2:(2+length(levels(myplot))-1)) # colori da rosso in poi…
> legend(locator(1), levels(myplot), fill=colfill)
```

(Bio)Statistica con R — Parte III

Curve ROC

- Il codice traccia due kernel density plot indipendenti e sovrapposti dei valori osservati nei controlli sani e nei malati e rimane in attesa: posizionare il mouse dove si vuole far comparire la legenda e fare click con il tasto sinistro.



(Bio)Statistica con R — Parte III

Curve ROC

- Eseguiamo ora questo codice, con il quale sono importate due serie di dati, le cui curve ROC sono poi sovrapposte sullo stesso sistema di assi cartesiani:

```
# importo i dati per le due curve ROC
> mydata <- read.table("CurveROC.csv", header=TRUE, sep=";")
> mydatabis <- read.table("CurveROCbis.csv", header=TRUE, sep=";")
> library(pROC)
# traccio la prima curva ROC
> roc(mydata$labels, mydata$predictions, smooth = FALSE, auc = TRUE, ci = FALSE,
  plot = TRUE, identity = FALSE, main = "Curve ROC sovrapposte", xlab="1-specificità",
  ylab = "Sensibilità")
# traccio la seconda curva ROC
> roc(mydatabis$labels, mydatabis$predictions, smooth = FALSE, auc = TRUE, ci = FALSE,
  plot = TRUE, add = TRUE, col = "red", lty = 4)
```

(Bio)Statistica con R — Parte III

Curve ROC

- L'argomento **add = TRUE** consente, quando viene tracciata la seconda curva ROC, di sovrapporla alla prima.
- Inoltre specificando il colore **col = "red"** e la linea tratteggiata **lty = 4** le due curve ROC possono essere meglio distinte.
- Infatti, le due curve ROC sovrapposte consentono di evidenziare come il test con la curva in nero continuo fornisca una informazione maggiore (853 controlli, 842 casi; AUC=0.9633) di quello con la curva ROC in colore rosso trattegiato (696 controlli, 999 casi; AUC=0.8).