



Parametri grafici in R

Corso di Bioinformatica

Nicola Vitacolonna

Corso di Laurea in Biotecnologie

Grafici esplorativi vs grafici finali

- Durante l'esplorazione dei dati non è necessario preoccuparsi troppo della qualità estetica dei grafici (a meno che ciò non sia funzionale all'analisi stessa)
 - Quando si prepara un grafico finale per una relazione, una tesi, un articolo o **per l'esame** è necessario curare il grafico nei dettagli
1. Etichettare gli assi (includendo le unità di misura)
 2. Aggiungere un titolo
 3. Curare altri aspetti estetici (colori, spessori delle linee, dimensione del testo, etc...)
 4. Aggiungere una legenda se necessario
 5. Preparare una didascalia per spiegare il grafico

Riepilogo: la funzione `plot()`

Funzione generica: il tipo di grafico dipende dalla classe degli argomenti

COMANDO	CLASSE DEGLI ARGOMENTI	RISULTATO
<code>plot(x,y)</code>	x e y vettori numerici	scatterplot
<code>plot(x,y, type = "l")</code>	x e y vettori numerici	grafico a linee
<code>plot(x,y)</code>	x factor, y vettore numerico	boxplot multiplo: equivalente a <code>boxplot(y ~ x)</code>
<code>plot(x)</code>	x vettore numerico	scatterplot
<code>plot(x, type = "l")</code>	x vettore numerico	grafico a linee
<code>plot(x)</code>	x factor	grafico a barre: equivalente a <code>barplot(table(x))</code>
<code>plot(x)</code>	x tabella di contingenza a 1 via	grafico a bastoncini
<code>plot(x)</code>	x tabella di contingenza a 2 o piú vie	grafico a mosaico
<code>plot(x)</code>	x serie temporale	grafico a linee
<code>plot(x)</code>	x data frame	grafici distribuzionali delle variabili nel data frame

Riepilogo: altre funzioni grafiche

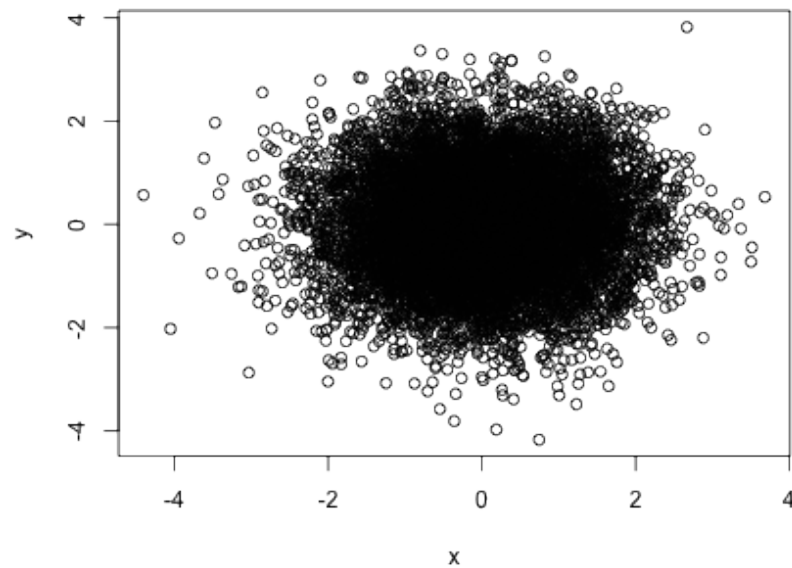
COMANDO	CLASSE DEGLI ARGOMENTI	RISULTATO
<code>hist(x)</code>	x vettore numerico	istogramma
<code>boxplot(x)</code>	x vettore numerico	boxplot
<code>boxplot(y ~ x)</code>	x factor, y vettore numerico	boxplot multiplo
<code>stripchart(y ~ x)</code>	x factor, y vettore numerico	strip chart
<code>barplot(x)</code>	x vettore numerico o tabella di contingenza a 1 via	grafico a barre semplice
<code>barplot(x)</code>	x matrice o tabella di contingenza a 2 vie	grafico a barre segmentato o multiplo
<code>dotchart(x)</code>	x vettore numerico o matrice	dot chart
<code>image(x,y,z)</code>	x e y vettori numerici, z matrice $ x \times y $	heat map
<code>mosaicplot(x)</code>	x tabella di contingenza a n vie	grafico a mosaico

Parametri grafici

- I comandi grafici hanno numerosi parametri dai nomi spesso poco intuitivi
- Alcuni parametri, comuni a tutti i comandi grafici, sono descritti in `?plot`
- La descrizione di ulteriori parametri si ottiene con `?par`
- Alcuni parametri importanti:
 - `main`: titolo del grafico
 - `xlab`, `ylab`: nomi degli assi
 - `xlim`, `ylim`: limiti degli assi
 - `col`: specificazione dei colori
 - `cex`: dimensione del testo

Esempio

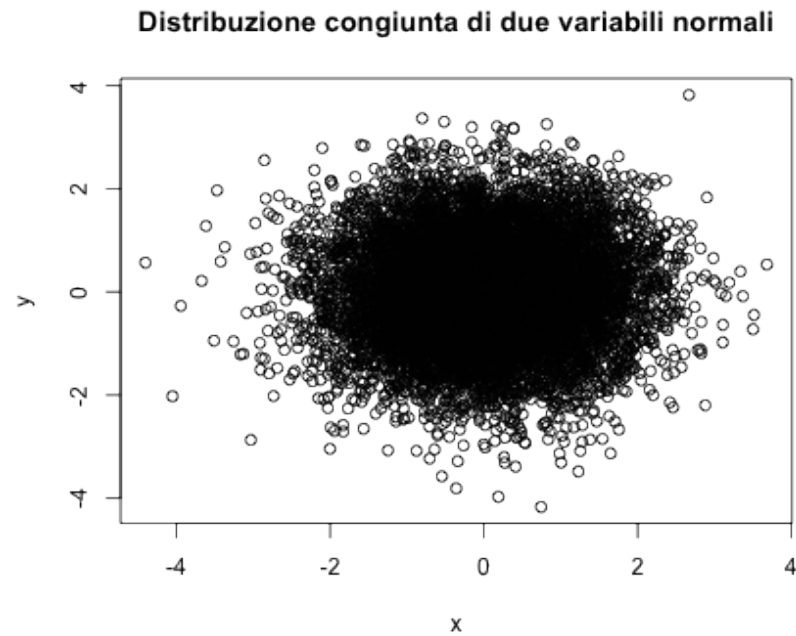
```
set.seed(1482)  
x <- rnorm(10000)  
y <- rnorm(10000)  
plot(x, y)
```



Titolo di un grafico

Parametro main

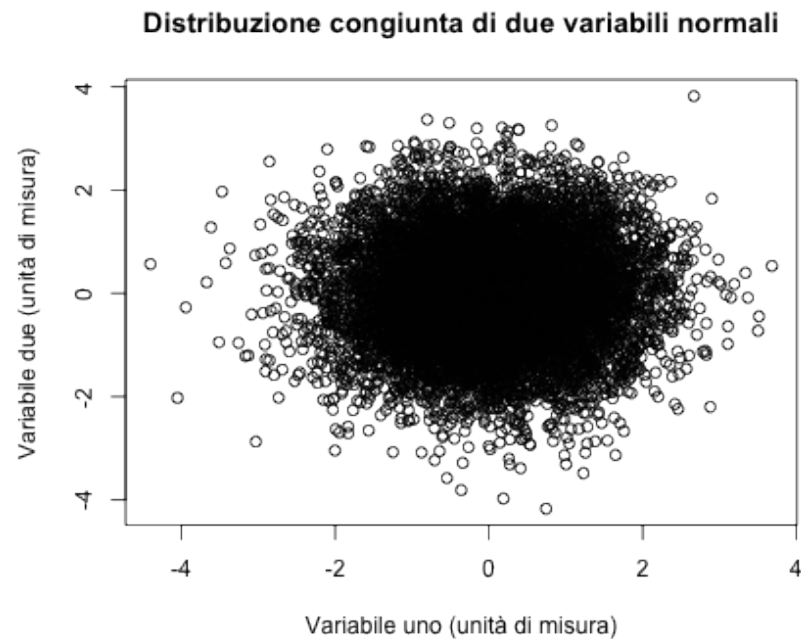
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali")
```



Nomi degli assi

Parametri `xlab` (asse orizzontale) e `ylab` (asse verticale)

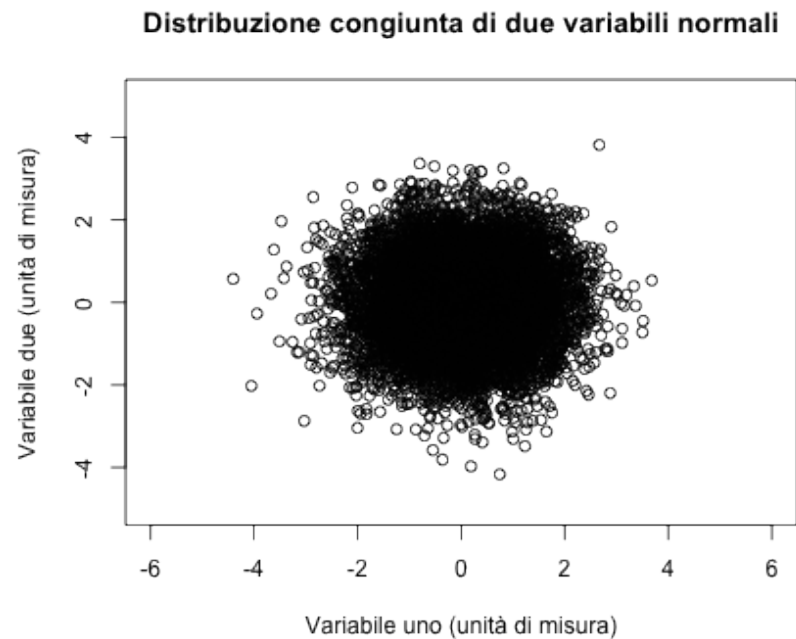
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)")
```



Limiti numerici degli assi

Parametri `xlim` (asse orizzontale) e `ylim` (asse verticale)

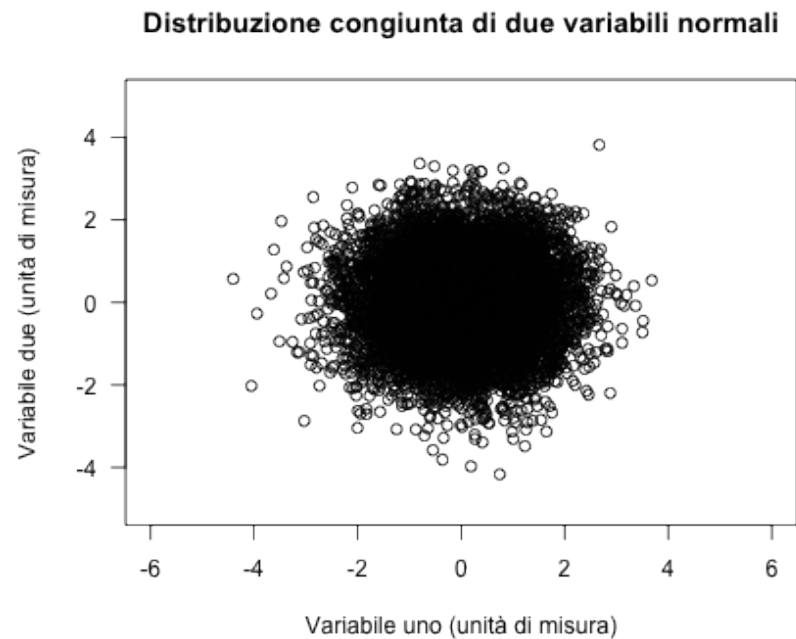
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
5))
```



Orientamento delle etichette degli assi

Parametro las: 0 = parallele all'asse, 1 = orizzontali, 2 = perpendicolari all'asse, 3 = verticali

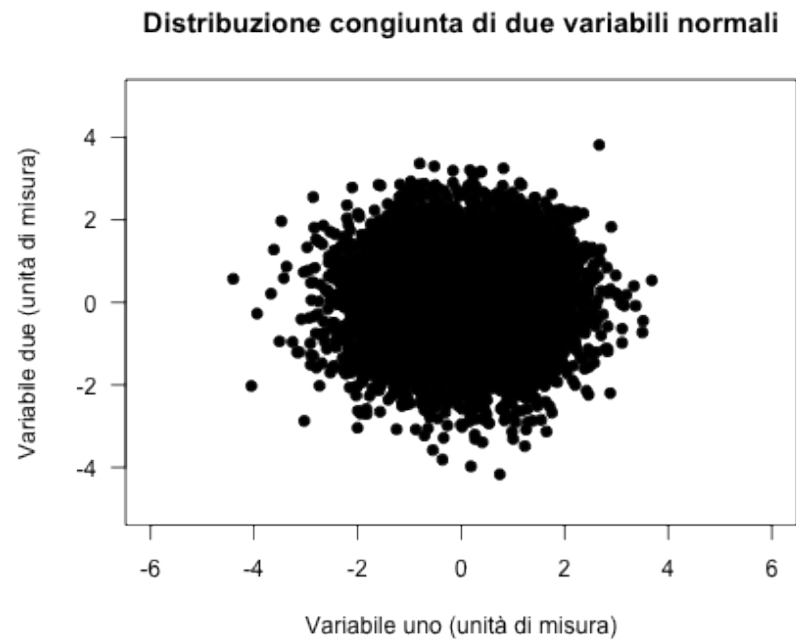
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
5), las = 1)
```



Tipo di punto

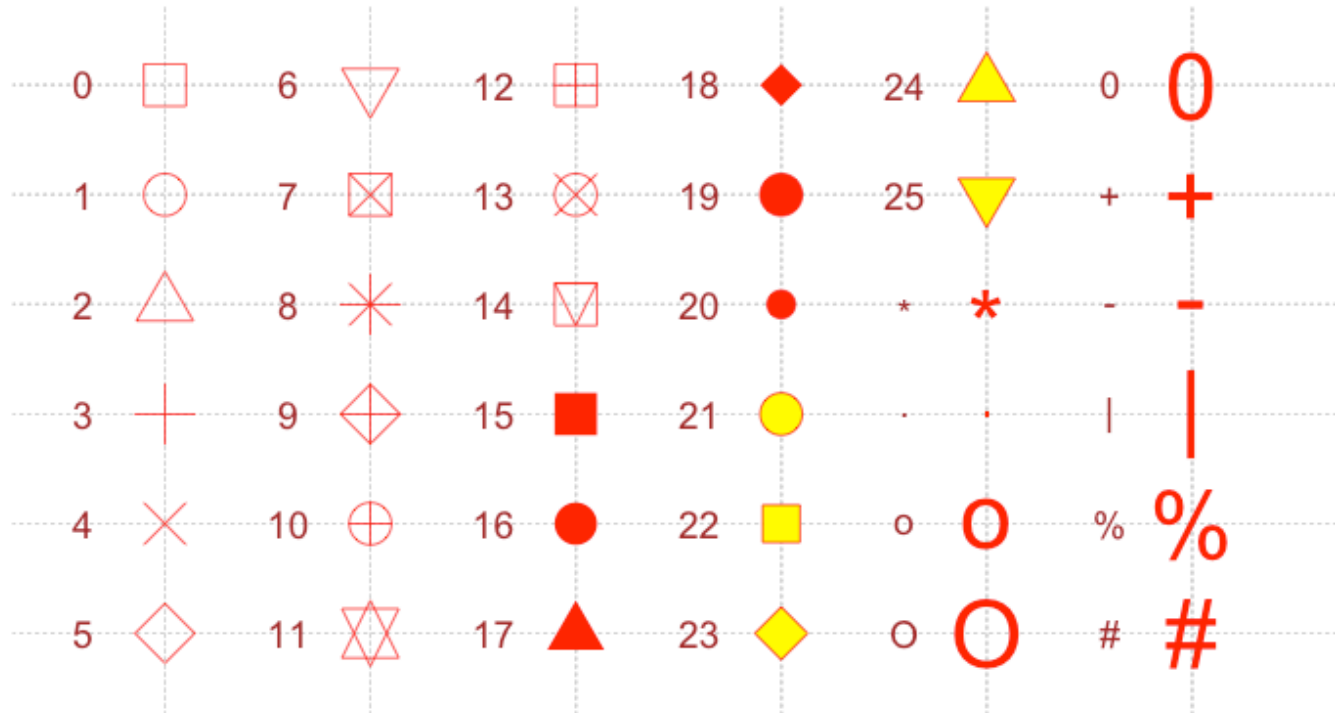
Parametro pch

```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
5), las = 1, pch = 19)
```



Simboli per i punti

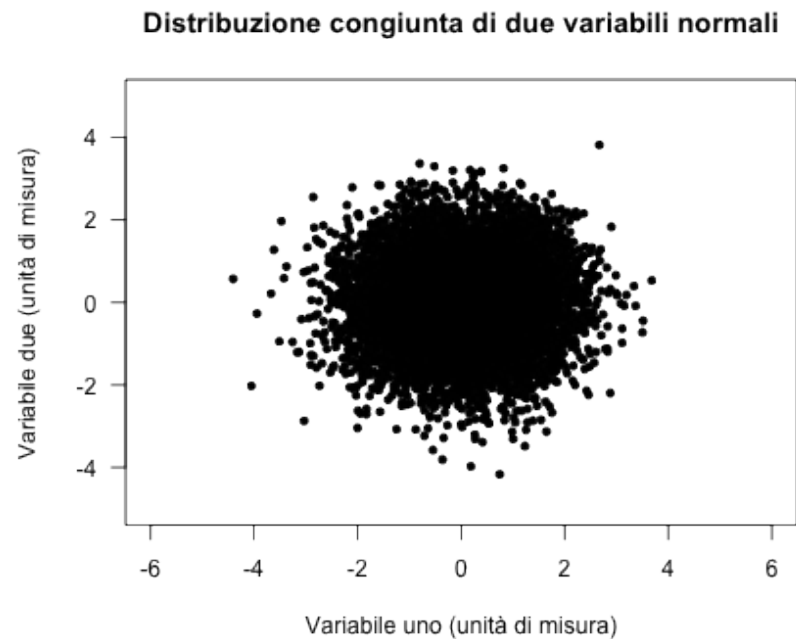
plot symbols: points(..., pch = *, cex = 3)



Dimensione dei punti

Parametro `cex`: specifica il fattore di scala dei punti (valore predefinito: 1)

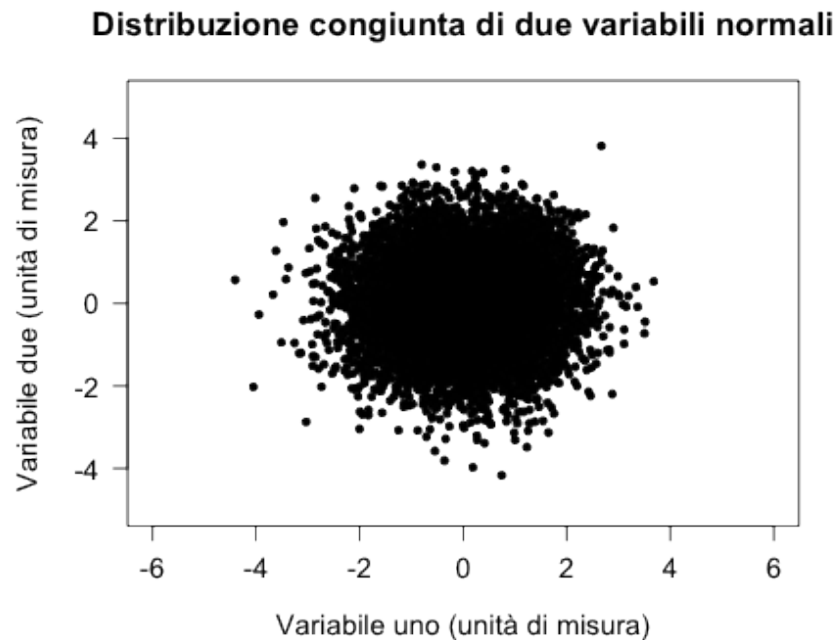
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
5), las = 1, pch = 19, cex = 0.7)
```



Dimensione del testo

Parametri `cex.axis` (etichette degli assi), `cex.lab` (nomi degli assi), `cex.main` (titolo)

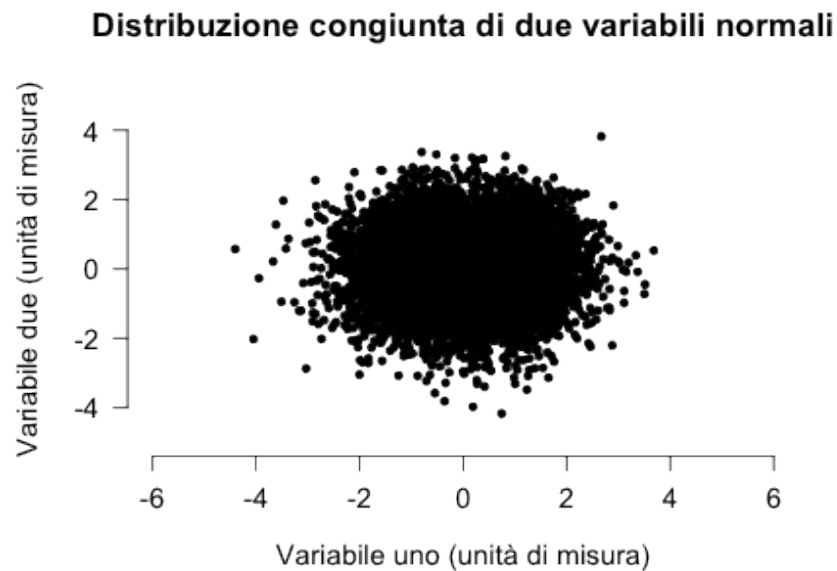
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u  
ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
5), las = 1, pch = 19, cex = 0.7, cex.main = 1.4, cex.axis = 1.2, cex.lab = 1.2)
```



Tipo di bordo

Parametro `bty`: valori ammissibili sono "l", "7", "c", "u", "]", "n" (nessun bordo)

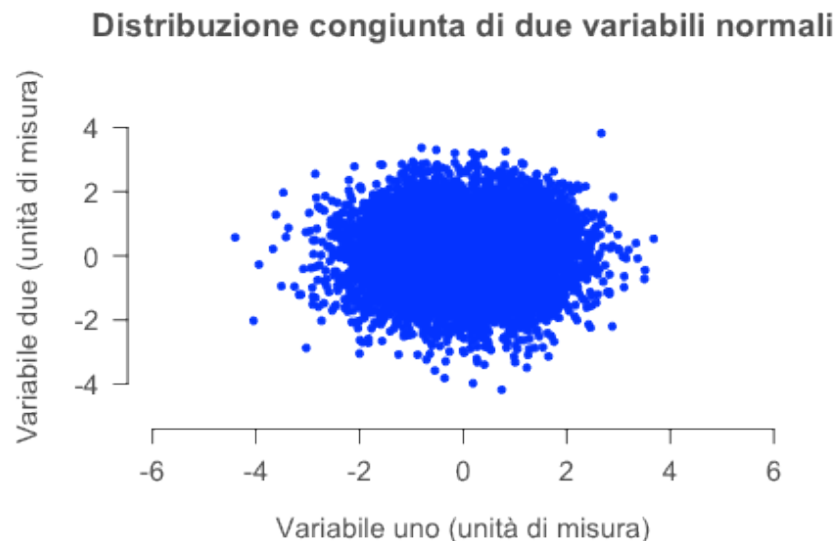
```
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u",  
      ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,  
        5), las = 1, pch = 19, cex = 0.7, cex.main = 1.4, cex.axis = 1.2, cex.lab = 1.2,  
      bty = "n")
```



Colore

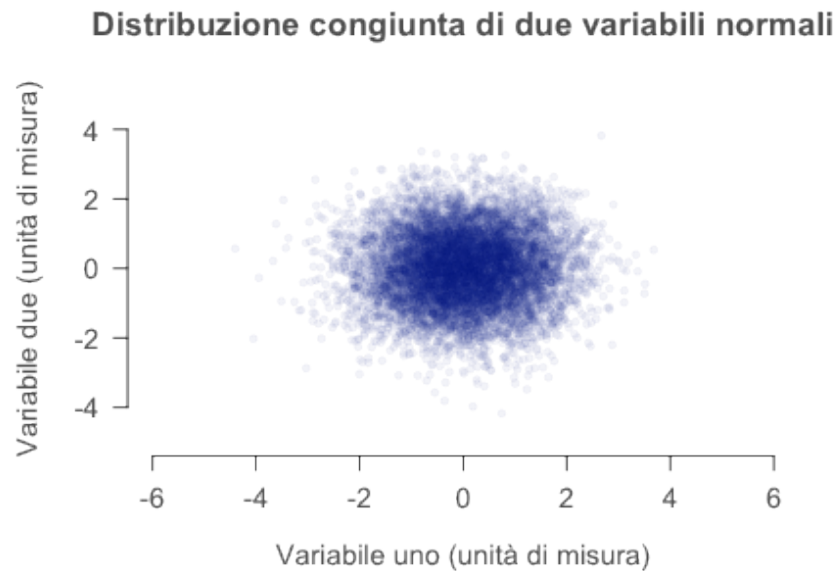
Parametri `col` (punti), `col.axis` (etichette degli assi), `col.lab` (nomi degli assi), `col.main` (titolo)

```
grigio <- rgb(red = 0.3, green = 0.3, blue = 0.3)
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u
      ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,
            5), las = 1, pch = 19, cex = 0.7, cex.main = 1.4, cex.axis = 1.2, cex.lab = 1.2,
      bty = "n", col = "blue", col.lab = grigio, col.main = grigio, col.axis = grigio)
```



Trasparenza

```
blu_trasp <- rgb(red = 0, green = 0, blue = 0.5, alpha = 0.05)
grigio <- rgb(red = 0.3, green = 0.3, blue = 0.3)
plot(x, y, main = "Distribuzione congiunta di due variabili normali", xlab = "Variabile uno (u
      ylab = "Variabile due (unità di misura)", xlim = c(-6, 6), ylim = c(-5,
          5), las = 1, pch = 19, cex = 0.7, cex.main = 1.4, cex.axis = 1.2, cex.lab = 1.2,
      bty = "n", col = blu_trasp, col.lab = grigio, col.main = grigio, col.axis = grigio)
```



Il grafico precedente contiene molti punti sovrapposti. È perciò conveniente disegnare solo un campione casuale dell'intero data set. Estraete un campione di 1000 punti e disegnate lo scatterplot corrispondente. Qual è il comando corretto?

- `plot(sample(x, 1000), sample(y, 1000), ...)`
- `indici = sample(1:10000, 1000); plot(x[indici], y[indici], ...)`

Submit Show Hint Show Answer Clear

Grafico finale

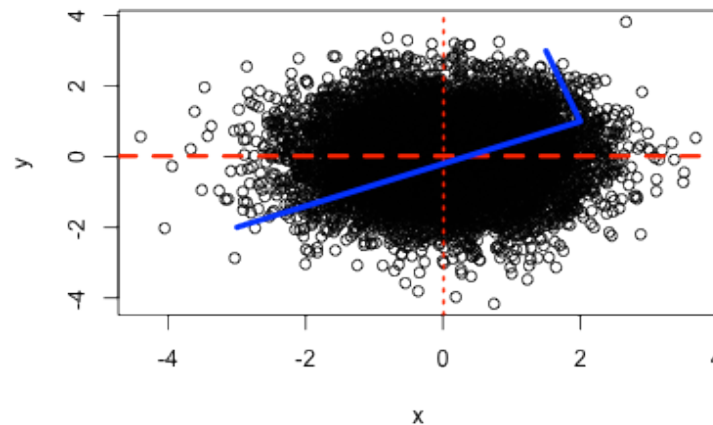
Distribuzione congiunta di due variabili normali



Sovrapporre linee a un grafico

- Funzioni `abline()` (linee rette), `lines()` (segmenti connessi)
- `lty`: tipo di linea (1 = continua, 2 = tratteggiata, 3 = punteggiata, ...)
- `lwd`: spessore della linea (valore predefinito: 1, dev'essere ≥ 1)

```
plot(x, y)
abline(mean(x), 0, col = "red", lty = 2, lwd = 3) # abline(intercetta, coeff. angolare)
abline(v = mean(y), col = "red", lty = 3, lwd = 2) # linea verticale
lines(c(-3, 2, 1.5), c(-2, 1, 3), col = "blue", lty = 1, lwd = 4) # Segmenti
```



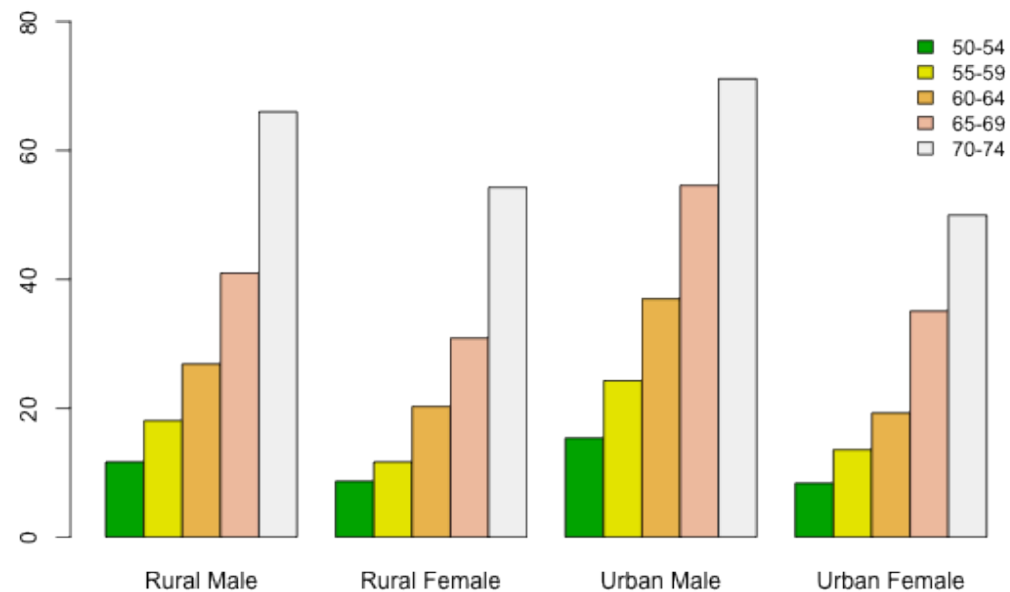
Aggiungere una legenda

- La funzione `legend()` aggiunge una legenda a un grafico
- La posizione della legenda può essere "topleft", "topright", "bottomleft", "bottomright" o una coppia di coordinate x,y
- Il parametro `legend` consente di specificare un vettore di descrizioni per le etichette della legenda
- Il parametro `fill` specifica, nell'ordine, i colori di ciascuna etichetta nella legenda
- I colori devono coincidere con quelli usati nel grafico (R non li sceglie automaticamente)!

```
data(VADeaths)
colori <- terrain.colors(5) # Genera 5 colori
barplot(VADeaths, beside = TRUE, col = colori)
legend("topright", legend = row.names(VADeaths), fill = colori)
```

Aggiungere una legenda

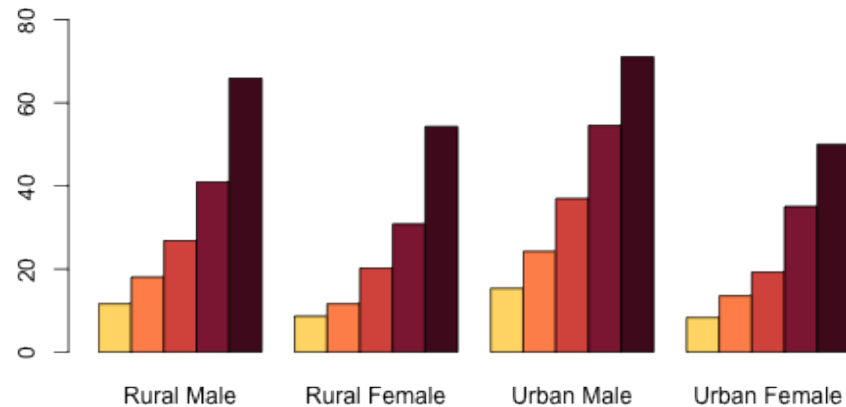
```
data(VADeaths)
colori <- terrain.colors(5) # Genera 5 colori
barplot(VADeaths, beside = TRUE, col = colori, ylim = c(0, 80))
legend("topright", legend = row.names(VADeaths), fill = colori, bty = "n", cex = 0.9)
```



Colori

- Le palette di colori di base di R non sono un granché...
- È facile tuttavia creare i propri vettori di colori
- [Adobe Kuler \(https://kuler.adobe.com/create/color-wheel/\)](https://kuler.adobe.com/create/color-wheel/)

```
colori <- c(rgb(255, 212, 98, max = 255), rgb(252, 125, 73, max = 255), rgb(207,  
66, 60, max = 255), rgb(122, 22, 49, max = 255), rgb(63, 11, 27, max = 255))
```



Altre funzioni utili per disegnare grafici

- `points()`: simile a `lines()`, ma per aggiungere punti
- `text()`: testo arbitrario in una specificata posizione
- `mtext()`: testo arbitrario sui margini del grafico
- `axis()`: manipolazione fine degli assi
- ...

Salvare un grafico

```
png(file = "nome_grafico.png")  
# Qui vanno scritti i comandi per disegnare il grafico  
dev.off() # Salva il grafico su disco
```

```
set.seed(1482)  
x <- rnorm(10000)  
y <- rnorm(10000)  
blu_trasp <- rgb(red = 0, green = 0, blue = 0.5, alpha = 0.1)  
indici = sample(1:10000, 6000)  
png(file = "bivariata_normale.png")  
plot(x[indici], y[indici], main = "Distribuzione congiunta di due variabili normali",  
      xlab = "Variabile uno (unità di misura)", ylab = "Variabile due (unità di misura)",  
      las = 1, xlim = c(-4, 4), ylim = c(-4, 4), pch = 19, cex = 0.7, cex.main = 1.6,  
      cex.axis = 1.4, cex.lab = 1.4, bty = "n", col = blu_trasp)  
dev.off()
```