

04-gRafici

Ottavia M. Epifania, Ph.D

Lezione di Dottorato @Università Cattolica del Sacro Cuore (MI)

8-9 Giugno 2023

Table of contents

① Grafici tradizionali

② ggplot2

③ Esportare i grafici

- Grafici base
- Grid graphics & ggplot2

Entrambi:

- High level functions → le funzioni che producono effettivamente il grafico
- Low level functions → Le funzioni che lo rendono più “bello”

Table of Contents

① Grafici tradizionali

② `ggplot2`

③ Esportare i grafici

demo(graphics) vi fornisce un tour guidato dei grafici

```
points() # Aggiunge punti al grafico
lines()  # Aggiunge linee al grafico
rect()
polygon()
abline() # aggiunge una riga con intercetta a e pendenza b
arrows() # aggiunge barre d'errore
text()   # aggiunge testo nel plot
mtext()  # aggiunge testo nei margini
axis()   # personalizza gli assi
box()    # box attorno al grafico
legend() # cambia parametri della legenda
```

Plot layout

Ogni plot è composto da due regioni:

- Plotting region (dove effettivamente sta il plot)
- La regione dei margini (contiene i margini e le varie etichette degli assi)

Plot layout

Ogni plot è composto da due regioni:

- Plotting region (dove effettivamente sta il plot)
- La regione dei margini (contiene i margini e le varie etichette degli assi)

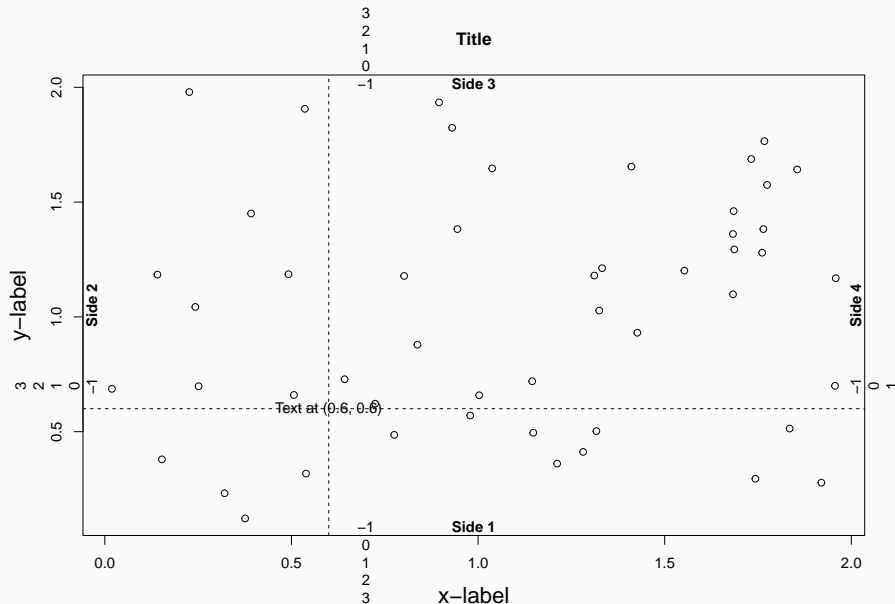
Uno scatter plot:

```
x <- runif(50, 0, 2) # 50 numeri random
y <- runif(50, 0, 2) # da una distr. uniforme
plot(x, y, main="Titolo",
      sub="Sottotitolo", xlab="x-label",
      ylab="y-label") # ecco il plot
```

Aggiunge del testo al plot

```
text(0.6, 0.6, "Testo @ (0.6, 0.6)")
abline(h=.6, v=.6, lty=2) # horizont. and vertic.
                           # lines
```


Margins region



Modificare il layout dei plot

Vanno creati dei pannelli

```
par(mfrow=c(nrighe, ncolonne)) # i pannelli vengono riempiti in righe  
par(mfcol=c(nrighe, ncolonne)) # i pannelli vengono riempiti in colonne
```

plot()

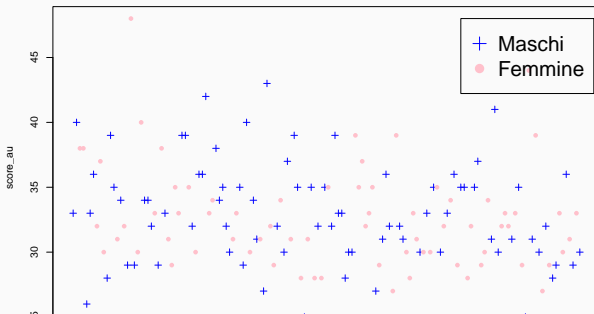
```
plot(x) # solo una variabile
```

```
plot(x, y) # due variabile (scatter plot)
```

```
plot(y ~ x) # due variabile, y in funzione di x
```

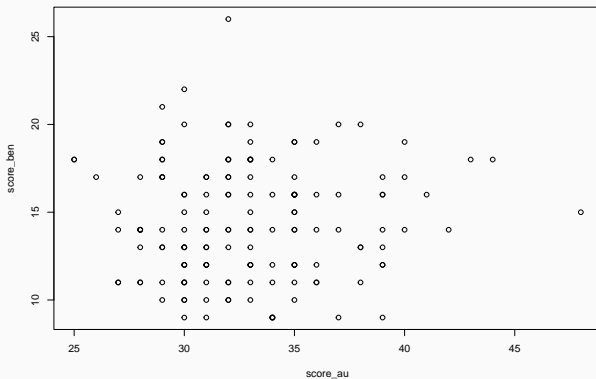
Esempi: plot(x)

```
with(benessere,  
     plot(score_au,  
          col = ifelse(genere == 1, "blue", "pink"),  
          pch = ifelse(genere == 1, 3, 16)))  
legend(x = 115, y = 48,  
       c("Maschi", "Femmine"), pch = c(3, 16), col = c("blue",
```



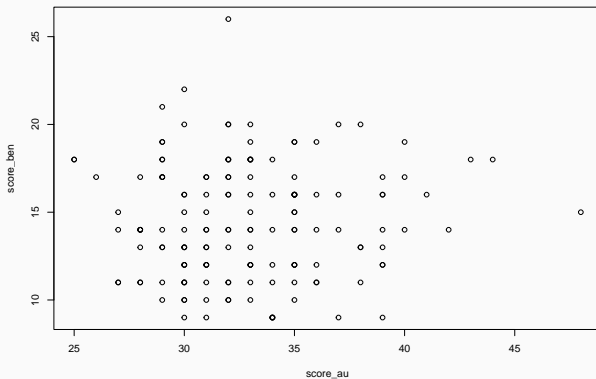
Esempi: plot(x, y)

```
with(benessere,  
      plot(score_au, score_ben))
```



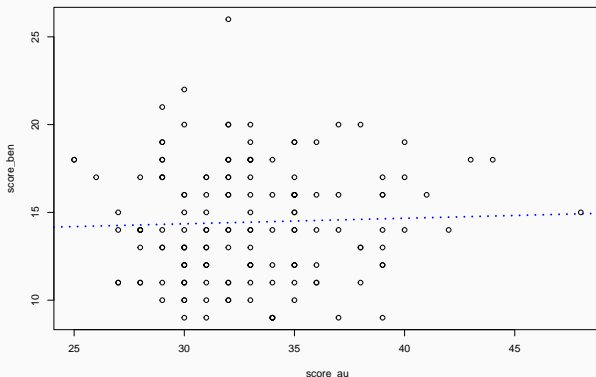
Esempi: `plot(y ~ x)`

```
with(benessere,  
      plot(score_ben ~ score_au))
```



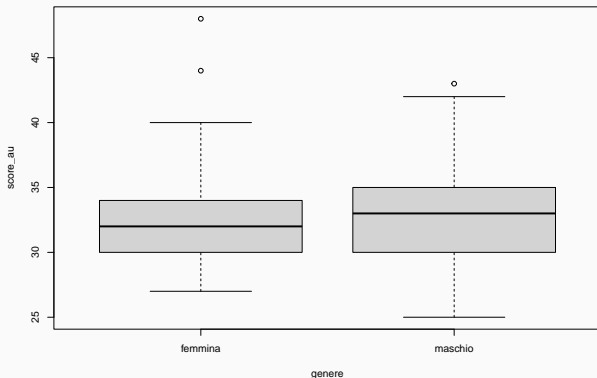
Esempi: `plot(y ~ x)` con retta di regressione

```
with(benessere,  
      plot(score_ben ~ score_au))  
abline(lm(score_ben ~ score_au, data = benessere),  
        col = "blue", lty = 3, lwd = 3)
```



Esempi: `plot(y ~ x)` con `x` categoriale

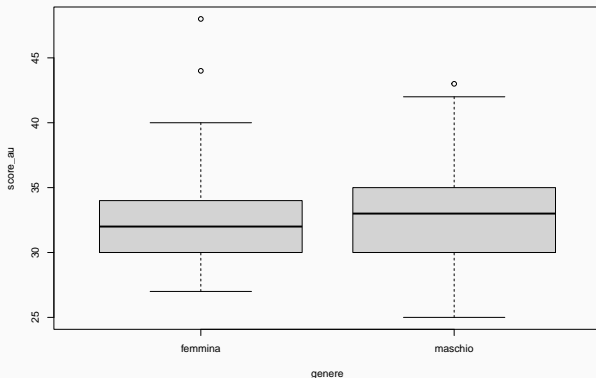
```
benessere$genere <- factor(ifelse(benessere$genere == 1,  
                                  "maschio", "femmina"))  
plot(score_au ~ genere, data = benessere)
```



Attenzione!

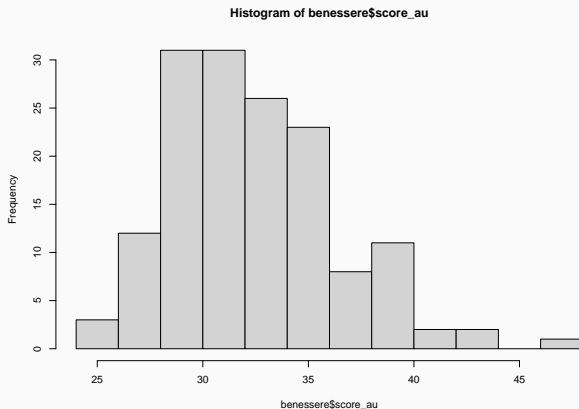
`plot(y ~ x)` con `x` categoriale è uguale a `boxplot(y ~ x)`

```
boxplot(score_au ~ genere, data = benessere)
```



hist(): Frequenze

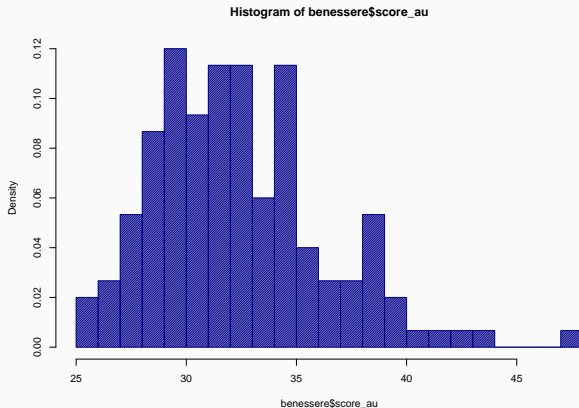
```
hist(benessere$score_au)
```



hist(): Densità

Densità

```
hist(benessere$score_au,  
      density=50, breaks=20,  
      prob=TRUE, col = "darkblue")
```

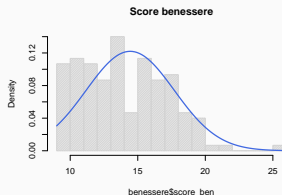
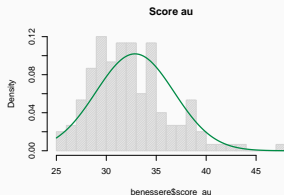


Multi plot (in riga)

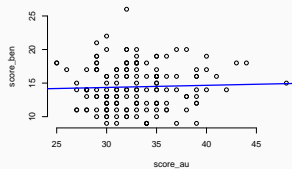
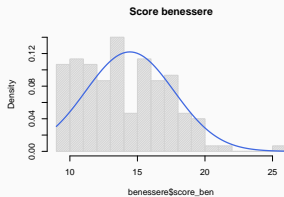
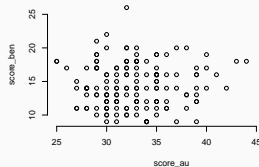
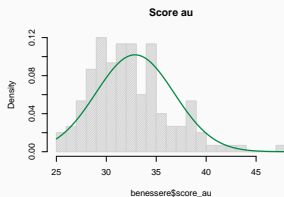
```
par(mfrow=c(1, 2))

hist(benessere$score_au,density=50, breaks=20, prob=TRUE,
     main = "Score au")
curve(dnorm(x, mean=mean(benessere$score_au),
             sd=sd(benessere$score_au)),
      col="springgreen4", lwd=2, add=TRUE, yaxt="n")

[...]
```



Multiplot (in colonna)



Multiplot (in colonna), codice

```
par(mfcol= c(2,2))
hist(benessere$score_au,density=50, breaks=20, prob=TRUE,
     main = "Score au")
curve(dnorm(x, mean=mean(benessere$score_au),
                sd=sd(benessere$score_au)),
      col="springgreen4", lwd=2, add=TRUE, yaxt="n")

hist(benessere$score_ben,density=50, breaks=20, prob=TRUE,
     main = "Score benessere")
curve(dnorm(x, mean=mean(benessere$score_ben),
                sd=sd(benessere$score_ben)),
      col="royalblue", lwd=2, add=TRUE, yaxt="n")

with(benessere,
     plot(score_au, score_ben, frame = FALSE))
```

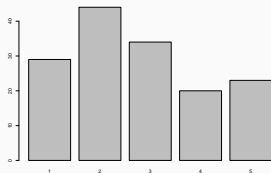
barplot()

Per creare i grafici a barre quando si hanno variabili discrete o categoriali

Richiede uno step in più → la creazione della tabella delle frequenze

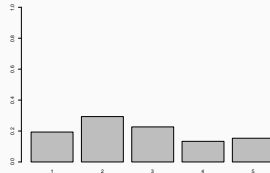
Frequenze assolute

```
freq_item1 = table(benessere$item1)
barplot(freq_item1)
```



Frequenze relative

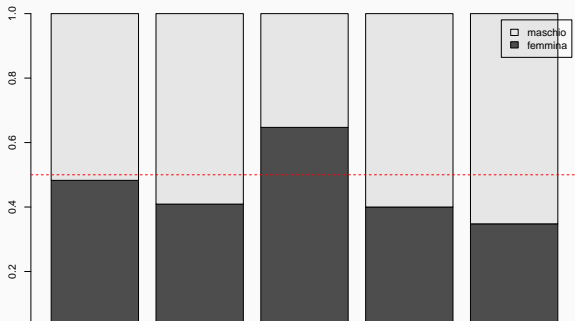
```
perc_item1 = freq_item1/sum(freq_item1)
barplot(perc_item1, ylim = c(0, 1))
```



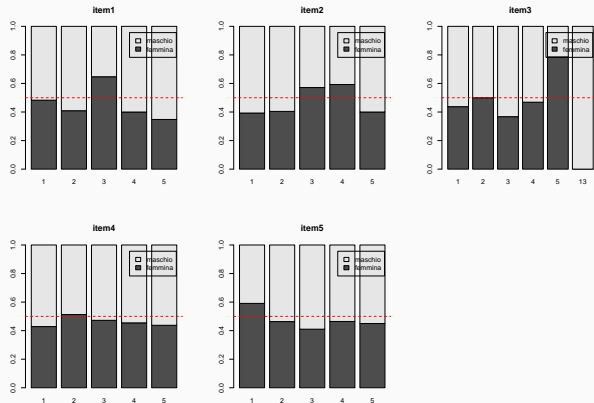
barplot() con più variabili

```
perc_item1_gender = table(benessere$genere, benessere$item1)
perc_item1_gender[1,] = perc_item1_gender[1,]/table(benessere
perc_item1_gender[2,] = perc_item1_gender[2,]/table(benessere

barplot(perc_item1_gender, ylim=c(0,1), legend = rownames(per
abline(h = .5, lty = 2, col = "red")
```



Un esempio di multiplot



Un esempio di multiplot (codice)

```
item_ben = benessere[, grep("item", colnames(benessere))]  
  
par(mfrow = c(2, round(ncol(item_ben)/2 + 0.2)))  
temp = NULL  
for (i in 1:ncol(item_ben)) {  
  temp = table(benessere$genere, item_ben[,i])  
  for (j in 1:nrow(temp)) {  
    temp[j,] = temp[j,]/table(item_ben[,i])  
  }  
  barplot(temp, ylim=c(0,1), legend = rownames(temp), main = colnames(item_b  
  abline(h = .5, lty = 2, col = "red")  
}
```

interaction.plot()

Permette di vedere l'interazione tra due variabili a seconda di una variabile categoriale:

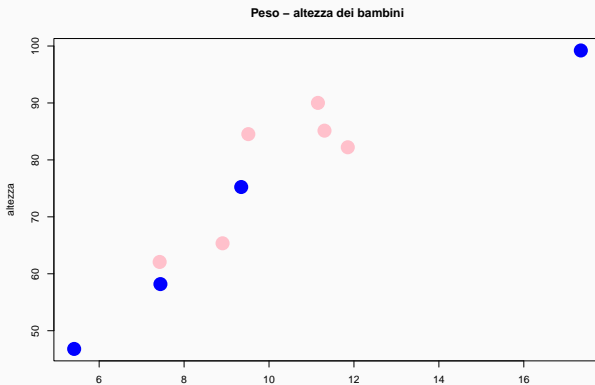
```
interaction.plot(x, v. categoriale, y)
```

interaction.plot()

Permette di vedere l'interazione tra due variabili a seconda di una variabile categoriale:

```
interaction.plot(x, v. categoriale, y)
```

La relazione tra peso e altezza cambia a seconda del genere?



```
babies$cat_weight = with(babies,
                          ifelse(peso <=
quantile(babies$peso)[2], "light",
ifelse(peso > quantile(babies$peso)[2] & peso > quantile(babi
"medium", "heavy")))
babies$cat_weight = factor(babies$cat_weight,
                           levels = c("light", "medium", "hea
```

```
babies
```

	id	genere	peso	altezza	cat_weight
1	baby1	f	7.424646	62.07722	light
2	baby2	m	7.442727	58.18877	light
3	baby3	f	9.512598	84.52737	heavy
4	baby4	f	11.306349	85.13573	medium
5	baby5	m	9.345165	75.23783	heavy
6	baby6	m	5.411290	46.80163	light
7	baby7	m	17.342840	99.21825	medium

```
with(babies,  
      interaction.plot(cat_weight, genere, altezza))
```

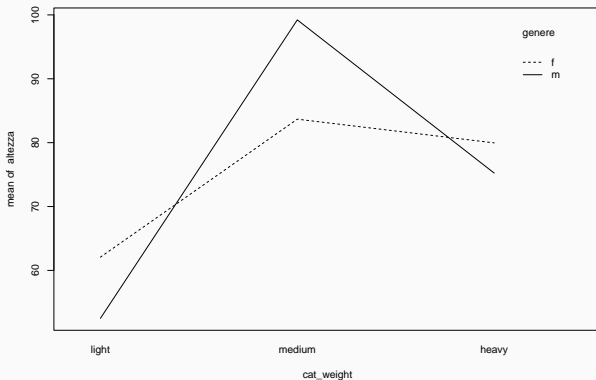


Table of Contents

① Grafici tradizionali

② `ggplot2`

③ Esportare i grafici

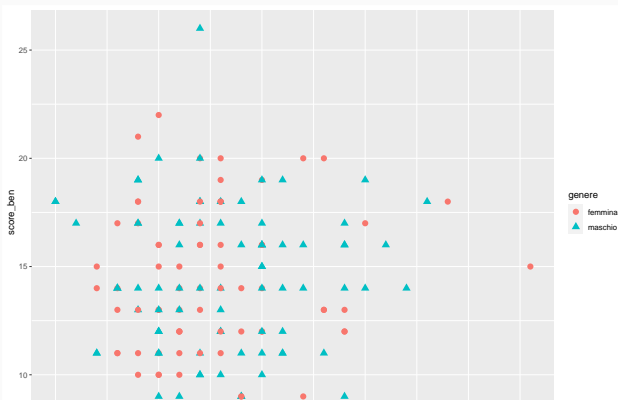
è più difficile ma è anche più facile:

```
ggplot(dati,  
  ars(x = variabile.x,  
    y = variabile.y,  
    col = variabile.colore.contorno,  
    fill = variabile.colore.filling,  
    shape = variabile.shape,  
    size = variabile.size,  
    ...)) + geom_tipo.grafico() + ...
```

Solitamente vuole i dati in formato long

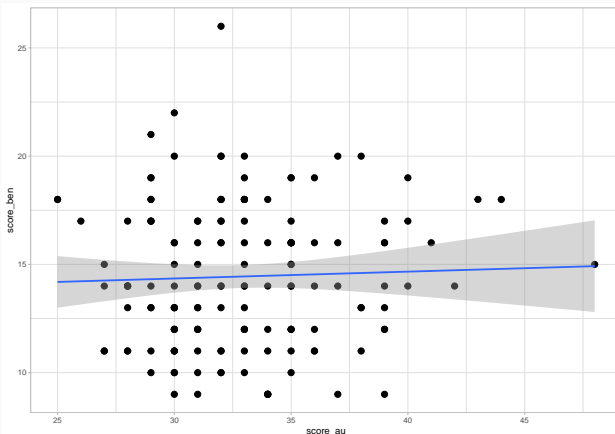
Scatter plot

```
ggplot(benessere,  
  aes(x = score_au, y = score_ben,  
    col = genere,  
    shape = genere)) +  
  geom_point(size = 3)
```



Scatter plot con retta di regressione

```
ggplot(benessere,  
       aes(x = score_au, y = score_ben)) +  
  geom_point(size = 3) +  
  geom_smooth(method = "lm", formula = y ~ x) + theme_light()
```



boxplot() e violin_plot()

Sono i grafici prediletti per far vedere le distribuzioni dei dati (specie il violin)

Richiedono che i dati siano in formato long:

	id	condition	mean_time
1	sbj1	A	4.136174
2	sbj1	B	2.639523
3	sbj2	A	2.547628
4	sbj2	B	4.319068
5	sbj3	A	4.265100
6	sbj3	B	4.113846

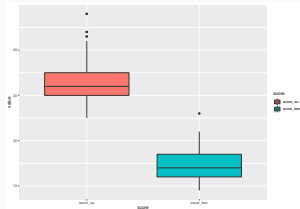
```
small = benessere[, c("ID", "score_au", "score_ben")]  
score_long = reshape(small,  
  idvar = "ID",  
  times = names(small)[-1],  
  timevar = "score", v.names = "value",  
  varying = list(names(small)[-1]),  
  direction = "long")  
head(score_long)
```

	ID	score	value
1.score_au	1	score_au	33
2.score_au	2	score_au	40
3.score_au	3	score_au	38
4.score_au	4	score_au	38
5.score_au	5	score_au	26
6.score_au	6	score_au	33

boxplot vs violinplot

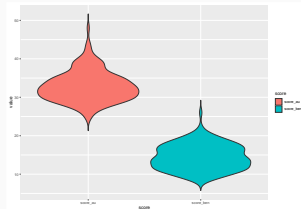
Boxplot

```
ggplot(score_long,
  aes(x = score, y = value,
    fill = score)) + geom_boxplot()
```



Violinplot

```
ggplot(score_long,
  aes(x = score, y = value,
    fill = score)) + geom_violin(t
```



```
score_long = merge(score_long, benessere[, c("ID", "genere")])
ggplot(score_long,
  aes(x = score, y = value,
      fill = genere)) + geom_violin(trim = FALSE)
```

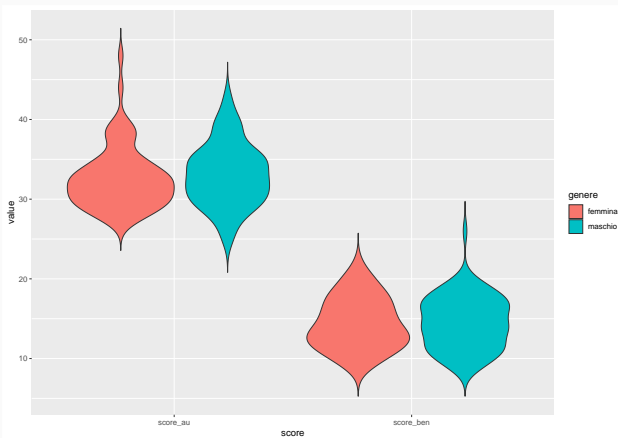


Table of Contents

① Grafici tradizionali

② `ggplot2`

③ Esportare i grafici

Esportare i grafici

```
postscript()  # vector graphics
pdf()

png()          # bitmap graphics
tiff()
jpeg()
bmp()
```

Remember to run off the graphic device once you've saved the graph:

```
dev.off()
```

(You can do it also manually)