Correlazione e regressione

(Le pagine che seguono sono tratte dal capitolo 4 del libro di G. Espa e R. Micciolo *Analisi esplorativa dei dati con* R. Apogeo, Milano, 2012.)

Correlazione

Quando oggetto di studio sono due caratteri quantitativi, come è ad esempio il caso delle variabili `df$altezza` e `df$peso`, la rappresentazione grafica prende il nome di *scatterplot* (*diagramma a dispersione*). È quanto abbiamo riprodotto nella figura 4.7, ottenuta con la funzione `plot`.

È evidente che i due caratteri sono associati fra loro; gli studenti più alti 'tendono' anche ad avere un peso più elevato, ma ci sono numerose 'eccezioni', come accade sempre nel mondo reale. Vogliamo trovare un indicatore in grado di quantificare il grado di associazione fra altezza e peso.

Diciamo subito che non esiste un unico indicatore, nel senso che, come sempre, dovremo fare riferimento più o meno esplicito ad uno o più assunti sottostanti; nel nostro caso, ad esempio, che la relazione fra altezza e peso possa essere di tipo 'lineare'.

Nella figura 4.7 sono stati inseriti due 'assi' in corrispondenza dei valori medi del peso e dell'altezza. Dobbiamo tuttavia fare attenzione. Quando chiediamo lo *scatterplot*, la mancanza di un dato anche solo per una delle due variabili comporta l'esclusione del soggetto dalla rappresentazione grafica; questo tuttavia non succede quando chiediamo la media di una singola variabile.

Calcoliamo il numero delle osservazioni mancanti per ciascuna delle due variabili in esame:

```
> sum(is.na(df$altezza))
[1] 0
> sum(is.na(df$peso))
[1] 4
```

Vediamo che mentre per l'altezza disponiamo di dati completi, nel caso del peso ci sono 4 osservazioni mancanti. È opportuno quindi lavorare su un *data-frame* 'completo', almeno per quanto riguarda queste due variabili. Ripetendo una procedura già

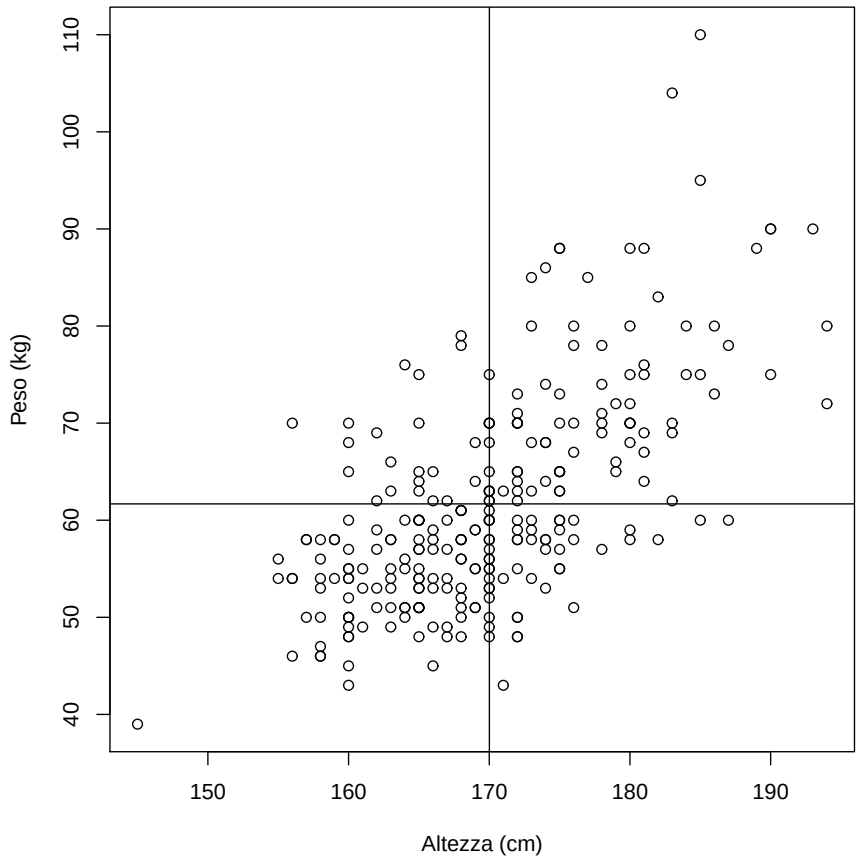


Figura 1. Altezza e peso degli studenti che hanno risposto al questionario. L'asse verticale è tracciato in corrispondenza dell'altezza media, quello orizzontale in corrispondenza del peso medio.

vista, impiegando la funzione `complete.cases` selezioniamo i soggetti che presentano entrambe le informazioni inserendoli in un *data-frame* temporaneo `tmp` che impiegheremo per le elaborazioni successive:

```
> ok <- complete.cases(df$altezza,df$peso)
> tmp <- df[ok,c(15,16)]
> dim(df)
[1] 268  49
> dim(tmp)
[1] 264   2
> str(tmp)
'data.frame':   264 obs. of  2 variables:
```

```
 $ altezza: int  170 172 173 165 165 170 165 174 ...
 $ peso   : int  65 60 58 51 54 58 51 57 49 58 ...
```

La funzione `dim` (così come `str`) ci dice che il *data-frame* `tmp` ha 264 righe e 2 colonne, contro le 268 righe e 49 colonne del *data-frame* `df`. Le due colonne del *data-frame* `tmp` sono state selezionate con l'istruzione `c(15,16)` inserita fra le parentesi quadre dopo la 'virgola' (vuol dire 'estrai le colonne 15 e 16', vale a dire l'altezza e il peso), mentre il vettore `ok` prima della 'virgola' permette di estrarre da `df` le righe (i soggetti) che non hanno dati mancanti nè sull'altezza nè sul peso.

La figura 4.7 è stata ottenuta con le seguenti istruzioni

```
> plot(tmp$altezza,tmp$peso)
> abline(v=mean(tmp$altezza),h=mean(tmp$peso))
> mean(tmp)
  altezza      peso
170.00758  61.68182
```

(abbiamo anche calcolato i valori medi di altezza e peso, che ci serviranno nel prosieguo).

Questo grafico mette in evidenza come i 'punti' (le coppie altezza, peso) tendano ad accumularsi nel primo e nel terzo quadrante, mentre sono meno rappresentate nel secondo e nel quarto. In altre parole, è (molto) più 'facile' vedere soggetti con altezza e peso *entrambi* sotto o sopra la media piuttosto che incontrare soggetti con valori "discordanti" per queste due variabili. Per contare quanti sono i 'punti' (le osservazioni) in ciascun quadrante basta dicotomizzare le due variabili prendendo come soglia la rispettiva media e producendo poi una tabella di contingenza:

```
> x <- tmp$altezza > mean(tmp$altezza)
> y <- tmp$peso > mean(tmp$peso)
> table(x,y)
       y
x       FALSE TRUE
  FALSE   122   32
  TRUE     34   76
```

Il risultato ottenuto conferma l'impressione visiva; le coppie 'concordanti' sono in totale 198, mentre quelle 'discordanti' sono 66, esattamente tre volte di meno. Un indicatore potrebbe essere rappresentato quindi dal rapporto fra numero di coppie concordanti e discordanti, oppure dal numero di coppie concordanti sul totale, oppure ancora, dal momento che i dati sono sotto forma di tabella 2×2, dall'*odds ratio* (circa 8.5 nel nostro caso). Il limite principale di questi indicatori consiste nella perdita di informazione a seguito della dicotomizzazione. Sarebbe opportuno conservare l'informazione numerica rispettando l''idea' di basare l'indicatore sulla posizione dei punti nei quattro quadranti.

Se non fosse per i valori riportati sui due assi, si potrebbe pensare che il punto dove gli stessi si incrociano sia l''origine', vale a dire la coppia (0,0) e che a destra di tale 'origine' (così come in alto rispetto all''origine') ci siano valori positivi, mentre a sinitra e in basso ci siano valori negativi. Possiamo ottenere questo risultato calcolando gli *scarti* (le differenze) rispetto alla media:

```
> tmp$x <- tmp$altezza - mean(tmp$altezza)
> tmp$y <- tmp$peso - mean(tmp$peso)
> head(tmp)
  altezza peso           x          y
1     170   65 -0.007575758   3.318182
2     172   60  1.992424242  -1.681818
3     173   58  2.992424242  -3.681818
4     165   51 -5.007575758 -10.681818
5     165   54 -5.007575758  -7.681818
6     170   58 -0.007575758  -3.681818
> tail(tmp)
    altezza peso          x          y
263     145   39 -25.007576 -22.681818
264     157   50 -13.007576 -11.681818
265     172   73   1.992424  11.318182
266     172   59   1.992424  -2.681818
267     159   58 -11.007576  -3.681818
268     180   70   9.992424   8.318182
```

Questa volta abbiamo preferito memorizzare gli scarti come elementi del *data-frame*; in questo modo, sfruttando le funzioni `head` e `tail` possiamo vedere le prime 6 righe e le ultime 6 righe del *data-frame* stesso, con i valori originali e i valori trasformati (scarti dalla media).

Le coppie concordanti hanno scarti entrambi positivi o entrambi negativi, mentre quelle discordanti hanno scarti di segno opposto. Se facciamo il prodotto degli scarti, le coppie concordanti avranno un prodotto positivo, mentre quelle discordanti avranno un prodotto negativo:

```
> tmp$xy <- tmp$x * tmp$y
> sum(tmp$xy > 0)
[1] 198
> sum(tmp$xy < 0)
[1] 66
```

Vediamo confermato il dato precedente sul numero di coppie concordanti e discordanti. Per sfruttare appieno l'informazione numerica, calcoliamo sia la somma sia la media della nuova variabile `tmp$xy` contenente il prodotto degli scarti:

```
> sum(tmp$xy)
[1] 15575.64
> mean(tmp$xy)
[1] 58.99862
```

La prima quantità (la somma) prende il nome di *codevianza* (fra altezza e peso), mentre la seconda (la media) prende il nome di *covarianza*. Questa è già una misura del grado di associazione *lineare* fra i due caratteri in esame. Si potrebbe infatti far vedere che se i due caratteri fossero fra loro *indipendenti*, la covarianza sarebbe esattamente uguale a zero (vedremo più avanti che non è necessariamente vera l'affermazione contraria). La covarianza si può calcolare molto più rapidamente usando la funzione `cov` di R:

```
> cov(tmp$altezza,tmp$peso)
[1] 59.22295
```

anche se il risultato numerico non è proprio lo stesso. Chi è che sbaglia? Noi oppure R? In effetti, nessuno dei due (meno male!). Il fatto è che R, come già precisato nel capitolo 3, è orientato prevalentemente all'analisi dei risultati di ricerche empiriche basate, in generale, su campioni; in questo caso le formule per il calcolo della varianza e della covarianza tengono conto del campionamento, mettendo al denominatore non il numero delle osservazioni, ma il numero delle osservazioni *meno uno* (*gradi di libertà*). Infatti, se dividiamo la codevianza per i gradi di libertà otteniamo lo stesso risultato di R:

```
> sum(tmp$xy)/(nrow(tmp)-1)
[1] 59.22295
```

Lo stesso vale per il calcolo della varianza, come abbiamo già verificato nel capitolo 3; ricordiamo, infatti, come la varianza calcolata da R con la funzione `var` sia *più grande* di quella calcolata come media dei quadrati degli scarti:

```
> var(tmp$altezza)
[1] 67.44481
> mean(tmp$x^2)
[1] 67.18934
> sum(tmp$x^2)/(nrow(tmp)-1)
[1] 67.44481
> var(tmp$peso)
[1] 124.4079
> mean(tmp$y^2)
[1] 123.9366
> sum(tmp$y^2)/(nrow(tmp)-1)
[1] 124.4079
```

A differenza della varianza, che è sempre positiva, la covarianza può assumere anche valori negativi, qualora 'prevalgano' i valori nel secondo e nel quarto quadrante.

Un limite della covarianza è rappresentato dal fatto che essa non è limitata superiormente (in valore assoluto), quindi non è facile confrontare due covarianze; inoltre, essendo un prodotto, la covarianza ha l'unità di misura definita dal prodotto delle unità di misura delle variabili costituenti.

Si può ovviare a questi due'inconvenienti' sfruttando la *disuguaglianza di Schwartz* la quale afferma che la covarianza non può essere maggiore (in valore assoluto) del prodotto delle deviazioni standard. Pertanto il rapporto fra la covarianza (numeratore) e il prodotto delle deviazioni standard (denominatore) sarà compreso fra -1 e $+1$. A questo rapporto diamo il nome di *coefficiente di correlazione lineare*.

Nel nostro caso le deviazioni standard e il loro prodotto valgono

```
> sd(tmp$altezza)
[1] 8.212479
> sd(tmp$peso)
[1] 11.15383
> sd(tmp$altezza)*sd(tmp$peso)
[1] 91.60058
```

quindi il coefficiente di correlazione lineare vale

```
> cov(tmp$altezza,tmp$peso)/
+     (sd(tmp$altezza)*sd(tmp$peso))
[1] 0.6465347
```

Lo stesso risultato si ottiene impiegando la funzione `cor`

```
> cor(tmp$altezza,tmp$peso)
[1] 0.6465347
```

Fate attenzione che se avessimo fatto la stessa richiesta sui dati contenuti nel *data-frame* `df` avremmo ottenuto un `NA` come risultato, dovuto alla presenza di osservazioni mancanti:

```
> cor(df$altezza,df$peso)
[1] NA
```

Per ottenere un risultato numerico dobbiamo impiegare l'argomento `use`:

```
> cor(df$altezza,df$peso,use="complete.obs")
[1] 0.6465347
```

Il coefficiente di correlazione lineare, indicato con la lettera r, misura il grado di associazione lineare fra due variabili, ovvero quanto i punti tendono ad 'addensarsi' lungo una retta. Il segno del coefficiente (dovuto al segno della covarianza, dal momento che le deviazioni standard sono sempre positive) indica il 'verso' della relazione. Un valore di r esattamente uguale a -1 o a $+1$ si ottiene soltanto quando i punti sono perfettamente allineati lungo una retta con pendenza diversa da zero. Un valore assoluto vicino ad 1 indica (in generale) che i punti sono disposti a formare un 'ovale' abbastanza stretto, mentre un valore uguale a 0 si ottiene quando i punti sono dispersi nel piano.

Dal momento che una figura è meglio di molte righe di spiegazione lasciamo la parola ad una simulazione (i cui risultati sono riportati nella figura 4.8) che potrete riprodurre con lo `script` seguente:

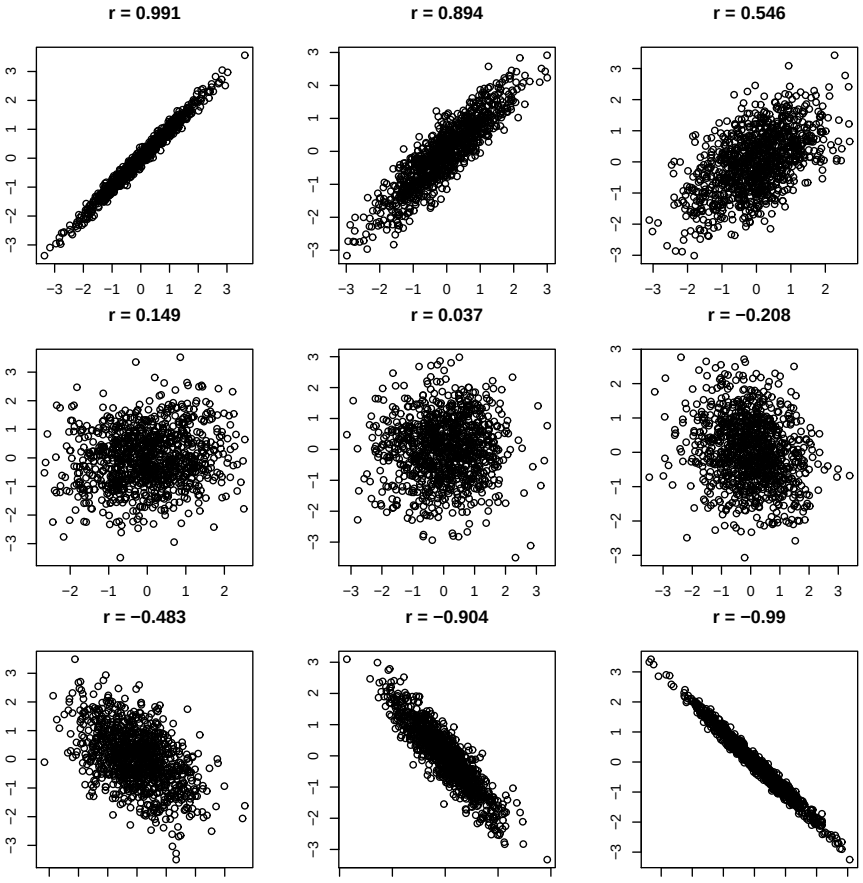


Figura 2. Risultati della simulazione di 9 campioni (ciascuno formato da 1000 coppie di osservazioni) con diverso grado di associazione.

```
> library(MASS)
> n <- 1000
> rv <- c(0.99,0.9,0.5,0.2)
> rv <- c(rv,0,rev(-rv))
> set.seed(123456)
> par(mfrow=c(3,3))
> par(mar=c(1,1,4,1))
> for (r in rv)
+ {
+     sigma <- matrix(c(1,r,r,1),nrow=2)
+     rand <- mvrnorm(n=n,mu=c(0,0),Sigma=sigma)
+     ccl <- cor(rand[,1],rand[,2])
+     plot(rand,main=round(ccl,3))
+ }
> par(mfrow=c(1,1))
```

Innanzitutto abbiamo richiamato la libreria `MASS` (che è una delle librerie che si installano 'automaticamente' quando si installa R); definiti dimensione del campione (`n`), valori di r che vogliamo simulare (`rv`) e 'innesco' del generatore di numeri pseudo-casuali (in modo da poter riprodurre esattamente la simulazione), attraverso un *ciclo* `for` estraiamo, impiegando la funzione `mvrnorm` (contenuta all'interno del *package* `MASS`) un campione di `n` osservazioni da una distribuzione normale bivariata con correlazione `r` e disegniamo il grafico delle coppie di punti così ottenuti (figura 4.8); il titolo di ciascun grafico riporta il valore della correlazione osservata sul campione. L'istruzione `par(mfrow=c(3,3))` chiede di organizzare i 9 grafici in un pannello contenente 3 righe e 3 colonne.

Il confronto fra ciascun grafico e il corrispondente valore di r presente nell'intestazione dovrebbe chiarire il significato di questo indicatore.

Abbiamo detto che se due caratteri sono indipendenti, allora $r = 0$. Tuttavia se $r = 0$ non è detto che i due caratteri siano indipendenti. Guardate il pannello di sinistra della figura 4.9, che riporta una serie di punti perfettamente disposti lungo la parabola $y = 2x^2 - 8x + 10$; i punti (e il grafico) sono stati prodotti dal seguente *script*:

```
> x <- c(-4:8); y <- 2*x**2-8*x+10
> cbind(x,y)
       x  y
 [1,] -4 74
 [2,] -3 52
 [3,] -2 34
 [4,] -1 20
 [5,]  0 10
 [6,]  1  4
 [7,]  2  2
```

```
 [8,]  3  4
 [9,]  4 10
[10,]  5 20
[11,]  6 34
[12,]  7 52
[13,]  8 74
> cor(x,y)
[1] 0
> plot(x,y,main=paste("r =",round(cor(x,y),3),"\n"))
```

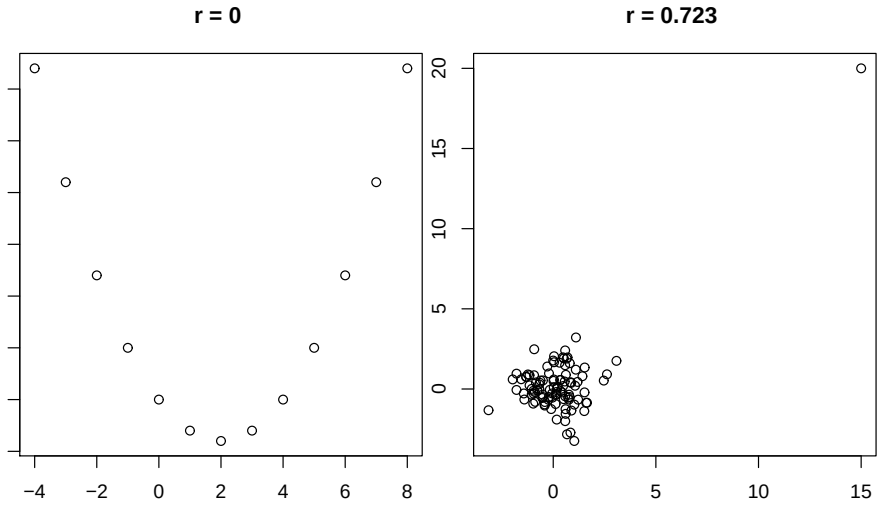


Figura 3. Due esempi in cui il coefficiente di correlazione lineare produce risultati "controversi".

Il coefficiente di correlazione lineare fra le variabili `x` e `y` vale esattamente 0 anche se esiste una relazione perfetta (ma non lineare) fra le due variabili. Osserviamo poi cosa succede se togliamo (con l'istruzione `-c(1,2)`) i primi due punti (quelli in alto a sinistra nel grafico della parabola):

```
> x <- x[-c(1,2)]; y <- y[-c(1,2)]
> cor(x,y)
[1] 0.5822225
```

la correlazione passa da 0 a 0.582, anche se la relazione fra `x` e `y` è rimasta la stessa.

Vediamo un secondo esempio. Se estraiamo un campione `x` di n osservazioni da una normale (ad esempio una normale standard) e poi ne estraiamo un secondo

y, della stessa numerosità, la correlazione fra x e y sarà molto vicina a zero (dal momento che i due campioni sono indipendenti).

```
> set.seed(123456)
> x <- rnorm(100000)
> y <- rnorm(100000)
> cor(x,y)
[1] -0.001194736
```

Se però aggiungiamo un punto 'estremo', cioè un punto molto 'distante' dagli altri, la correlazione diventa immediatamente alquanto diversa da zero (vedi il pannello di destra della figura 4.9):

```
> x <- rnorm(100); y <- rnorm(100)
> cor(x,y)
[1] 0.03230051
> x <- c(x,15); y <- c(y,20)
> cor(x,y)
[1] 0.7228463
> plot(x,y,main=paste("r =",round(cor(x,y),3),"\n"))
```

e, quindi, non basta un valore elevato per dire che il legame è forte; è sempre bene 'guardare' i dati.

Infine un altro *caveat*. Calcoliamo il coefficiente di correlazione lineare fra numero di scarpe e lunghezza dei capelli per gli studenti del nostro *data-frame*:

```
> cor(df$scarpe,df$capelli,use="complete.obs")
[1] -0.3309527
```

Per quanto non elevatissima, certamente esiste una correlazione negativa fra queste due variabili. Questo non vuol certo dire che riducendo il numero delle scarpe si allungano i capelli! Il coefficiente di correlazione lineare misura la forza di associazione (lineare) fra le due variabili e non può assolutamente essere scambiato per qualcosa che "certifica" l'esistenza di una relazione di tipo causa−effetto. In ogni caso il valore di r sta ad indicare che gli studenti con numero di scarpe più piccolo tendono ad avere i capelli più lunghi. Come possiamo spiegare questo risultato? Si tratta di una *correlazione spuria*. Infatti, il *data-frame* è formato da studenti di ambo i generi; è del tutto verosimile che le studentesse abbiano, in generale, i capelli più lunghi di quelli degli studenti così come un numero di scarpe più basso; ad esempio, le medie (quelle dei capelli sono in mm) sono

```
> tapply(df$capelli,df$sex,mean,na.rm=TRUE)
       F        M
221.8895  68.5600
```

```
> tapply(df$scarpe,df$sex,mean,na.rm=TRUE)
       F        M
38.40000 42.97436
```

Quindi la correlazione calcolata fra numero delle scarpe e lunghezza dei capelli riflette il dato appena illustrato e in questo senso è *spuria*; esiste una terza variabile (il genere) che non entra esplicitamente nell'analisi, ma ci entra 'implicitamente', attraverso la sua relazione con numero di scarpe e lunghezza dei capelli, 'inquinando', per così dire, il risultato finale (il termine più adatto è "distorcendo", dalla parola inglese *bias*). Vediamo cosa succede se ricalcoliamo la correlazione separatamente per maschi e femmine:

```
> mas <- which(df$sex == "M")
> tmp <- df[mas,]
> cor(tmp$scarpe,tmp$capelli,use="complete.obs")
[1] 0.07389281
> fem <- which(df$sex == "F")
> tmp <- df[fem,]
> cor(tmp$scarpe,tmp$capelli,use="complete.obs")
[1] 0.02056519
```

La correlazione è di segno positivo, ma comunque molto vicina a zero, per cui possiamo ritenere che i due caratteri non siano fra loro correlati quando sono valutati separatemente per genere.

Correlazione non parametrica

Consideriamo le seguenti coppie (x_i, y_i) di osservazioni: $(0, 1)$, $(1, 2)$, $(2, 4)$, $(3, 8)$, $(4, 16)$, $(5, 32)$. Il coefficiente di correlazione lineare è

```
> x <- c(0:5); y <- c(1,2,4,8,16,32)
> cor(x,y)
[1] 0.9057639
```

un valore molto alto, ma comunque diverso da 1. In effetti le cinque coppie considerate soddisfano la relazione $y = 2^x$ che non è di tipo lineare (è una relazione esponenziale).

Quando il ricercatore è interessato a valutare l'esistenza di una relazione *monotona* crescente (quando X cresce anche Y cresce), o decrescente (quando X cresce Y diminuisce), ma non di tipo lineare (come nell'esempio considerato) può fare ricorso al calcolo di un coefficiente di correlazione *non parametrico*, basato sulla trasformazione in *ranghi*.

Dato un insieme di N osservazioni $x_1, x_2, \ldots, x_N$, il *rango* dell'osservazione x_i è sostanzialmente la *posizione* occupata da quella osservazione nella graduatoria crescente. A scopo esemplificativo, consideriamo i seguenti 10 valori (estratti da una distribuzione uniforme):

```
> set.seed(123456)
> x <- runif(10)
> round(x,3)
 [1] 0.798 0.754 0.391 0.342 0.361
 [6] 0.198 0.535 0.097 0.988 0.168
```

I ranghi corrispondenti sono:

```
> rank(x)
 [1]  9  8  6  4  5  3  7  1 10  2
```

Il rango 1 è a carico della ottava osservazione estratta (0.097) che mostra appunto il valore più piccolo, mentre il rango 10 è a carico della nona osservazione estratta (0.988) che mostra appunto il valore più grande. La prima osservazione estratta (0.798) ha rango 9 perchè si trova al nono posto della graduatoria ordinata in modo crescente (sarebbe più appropriato dire non decrescente); la seconda osservazione estratta (0.754) ha rango 8 perchè si trova all'ottavo posto della graduatoria ordinata in modo crescente e così via. In effetti per assegnare i ranghi basta ordinare i dati:

```
> round(sort(x),3)
 [1] 0.097 0.168 0.198 0.342 0.361
 [6] 0.391 0.535 0.754 0.798 0.988
```

Una qualche 'complicazione' si ha quando ci sono più osservazioni con lo stesso valore; in questo caso si assegna ad esse il rango medio; vediamolo con i pesi dei primi 10 studenti:

```
> df$peso[1:10]
 [1] 65 60 58 51 54 58 51 57 49 58
> rank(df$peso[1:10])
 [1] 10.0  9.0  7.0  2.5  4.0  7.0  2.5  5.0  1.0  7.0
```

Ci sono due studenti il cui peso è di 51 kg e tre il cui peso è 58 kg. Nella graduatoria ordinata

```
> sort(df$peso[1:10])
 [1] 49 51 51 54 57 58 58 58 60 65
```

i due studenti che pesano 51 kg occupano la seconda e la terza posizione, quindi assegniamo a *ciascuno* rango $(2 + 3)/2 = 2.5$; i tre studenti che pesano 58 kg occupano le posizioni 6, 7, 8 e, quindi, assegniamo a *ciascuno* rango $(6+7+8)/3 = 7$ (notate che nessuno studente ha rango 2, 3, 6 o 8).

Se, date due variabili X e Y, trasformiamo i loro valori in ranghi r_X, r_Y, allora il coefficiente di correlazione lineare fra le coppie (r_X, r_Y) prende il nome di

coefficiente di correlazione non parametrico ρ (è la lettera greca *rho*) di Spearman. Questo coefficiente può essere calcolato direttamente da R assegnando all'argomento `method` della funzione `cor` il valore `"spearman"` (tra virgolette). Verifichiamolo con l'esempio introduttivo, che riproponiamo dall'inizio:

```
> x <- c(0:5); y <- c(1,2,4,8,16,32)
> cor(rank(x),rank(y))
[1] 1
> cor(x,y,method="spearman")
[1] 1
```

Come si vede questo coefficiente 'si accorge' che le 6 coppie sono ordinate monotonicamente e restituisce il valore massimo che è 1; il 'minimo' è -1, che indica una perfetta relazione monotona decrescente:

```
> x <- c(0:5); y <- 1/c(1,2,4,8,16,32)
> cor(x,y,method="spearman")
[1] -1
```

Calcoliamo il coefficiente ρ fra altezza e peso sia come correlazione fra ranghi, sia chiedendo ad R di calcolarlo 'autonomamente' (anche in questo caso preferiamo lavorare dopo aver eliminato i dati mancanti):

```
> ok <- complete.cases(df$altezza,df$peso)
> tmp <- df[ok,c(15,16)]
> cor(rank(tmp$altezza),rank(tmp$peso))
[1] 0.6158899
> cor(tmp$altezza,tmp$peso,method="spearman")
[1] 0.6158899
```

Approfondimento: La correlazione in simboli

Date due variabili X e Y rilevate su n unità statistiche, indichiamo con x_i e y_i la $i-$esima osservazione.
La covarianza è data da

$$\sigma_{XY} = \frac{\sum_{i=1}^{N}(x_i - \mu_X)(y_i - \mu_Y)}{N}$$

La varianza della variabile X è data da

$$\sigma_X^2 = \frac{\sum_{i=1}^{N}(x_i - \mu_X)^2}{N}$$

La varianza della variabile Y è data da

$$\sigma_Y^2 = \frac{\sum_{i=1}^{N}(y_i - \mu_Y)^2}{N}$$

Il coefficiente di correlazione lineare è dato da

$$r = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Nota. Nel caso di dati provenienti da un campione casuale semplice, tutti i valori N al denominatore vanno sostituiti con $N - 1$ (i gradi di libertà). Questo non influisce comunque sul calcolo del coefficiente di correlazione lineare.

Regressione

Guardiamo lo *scatterplot* riportato nella figura 4.7; è abbastanza evidente che gli studenti più alti *tendono* anche ad avere un peso più elevato e viceversa. Possiamo cercare di approssimare la relazione fra altezza e peso mediante una *retta* di equazione $\hat{y} = b_0 + b_1 x$, dove $\hat{y}$ è il peso (in kg) *previsto* per un soggetto la cui l'altezza (in cm) è x e b_0 e b_1 sono due *parametri* (che vanno *stimati* dai dati) necessari per poter disegnare la retta. Y è chiamata anche *variabile dipendente*, mentre X è chiamata *variabile indipendente* o *covariata*.

Naturalmente, dal momento che i punti riportati in figura 4.7 non giacciono su una retta, è necessario stabilire un *criterio* in base al quale poter trovare un'unica retta che lo soddisfi. Il criterio che prenderemo in considerazione è quello dei *minimi quadrati*. Senza entrare in dettagli tecnici, diremo soltanto che il parametro b_1 viene stimato dividendo la covarianza fra le variabili X e Y per la varianza di X. Pertanto la quantità al numeratore di r e di b_1 è la stessa e, quindi, il *segno* di r e quello di b_1 coincidono. Inoltre, il parametro b_0 viene stimato a partire dalle medie μ_Y e μ_X come segue: $b_0 = \mu_Y - b_1 \mu_X$. Nell'esempio che abbiamo considerato abbiamo quindi:

```
> ok <- complete.cases(df$altezza,df$peso)
```

```
> tmp <- df[ok,c(15,16)]
> (b1 <- cov(tmp$altezza,tmp$peso)/var(tmp$altezza))
[1] 0.878095
> (b0 <- mean(tmp$peso) - b1*mean(tmp$altezza))
[1] -87.60099
```

ovvero $\hat{y} = -87.6 + 0.878x$ è la retta stimata con il metodo dei minimi quadrati.

La figura 4.10 ripropone lo *scatterplot* fra altezza e peso e la retta stimata con il metodo dei minimi quadrati, che è stata aggiunta al grafico impiegando la funzione `abline` i cui primi argomenti sono l'*intercetta* e la *pendenza* della retta di regressione.

```
> plot(tmp$altezza,tmp$peso,
+      xlab="Altezza (cm)",ylab="Peso (kg)")
> abline(b0,b1,lwd=2)
```

Il parametro b_1 prende il nome di *coefficiente di regressione lineare* (o *coefficiente angolare*) e rappresenta la *pendenza* della retta, ovvero ci dice di quanto varia *in media* il peso (la variabile dipendente) quando l'altezza (la variabile indipendente) varia di 1 unità (1 cm nel nostro caso). Quindi nel nostro esempio ci aspettiamo che in media il peso di un soggetto alto, ad esempio, 170 cm sia superiore di 878 grammi (0.878 kg) rispetto al peso di un soggetto alto 169 cm. I due valori 170 cm e 169 cm non hanno nulla di particolare; l'affermazione rimane la stessa per qualsiasi coppia di valori la cui differenza è di 1 cm (ad esempio 165 e 164 oppure 183 e 182).

Il parametro b_0 prende il nome di *intercetta* (o, anche, *ordinata all'origine*) ed è il peso medio previsto per un soggetto alto 0 cm! In questi termini, ovviamente, b_0 non ha molto significato, anche se è necessario per poter 'disegnare' la retta ovvero calcolare il *valore previsto* $\hat{y}_i$ in corrispondenza del valore x_i relativo alla i-esima osservazione ($\hat{y}_i = b_0 + b_1 x_i$).

A titolo di esempio, consideriamo una serie di altezze e calcoliamo con R i valori previsti dal nostro *modello lineare* (che saranno tutti sulla retta disegnata nella figura 4.10):

```
> (x <- seq(150,180,by=5))
[1] 150 155 160 165 170 175 180
> y <- b0 + b1*x
> round(y,2)
[1] 44.11 48.50 52.89 57.28 61.68 66.07 70.46
```

Ovviamente possiamo allo stesso modo calcolare il peso previsto in base all'altezza di ciascuno studente mediante l'istruzione `tmp$yhat <- b0 + b1*tmp$altezza` e osservare che la correlazione fra valori previsti e valori osservati è esattamente la stessa calcolata in precedenza fra altezza e peso:

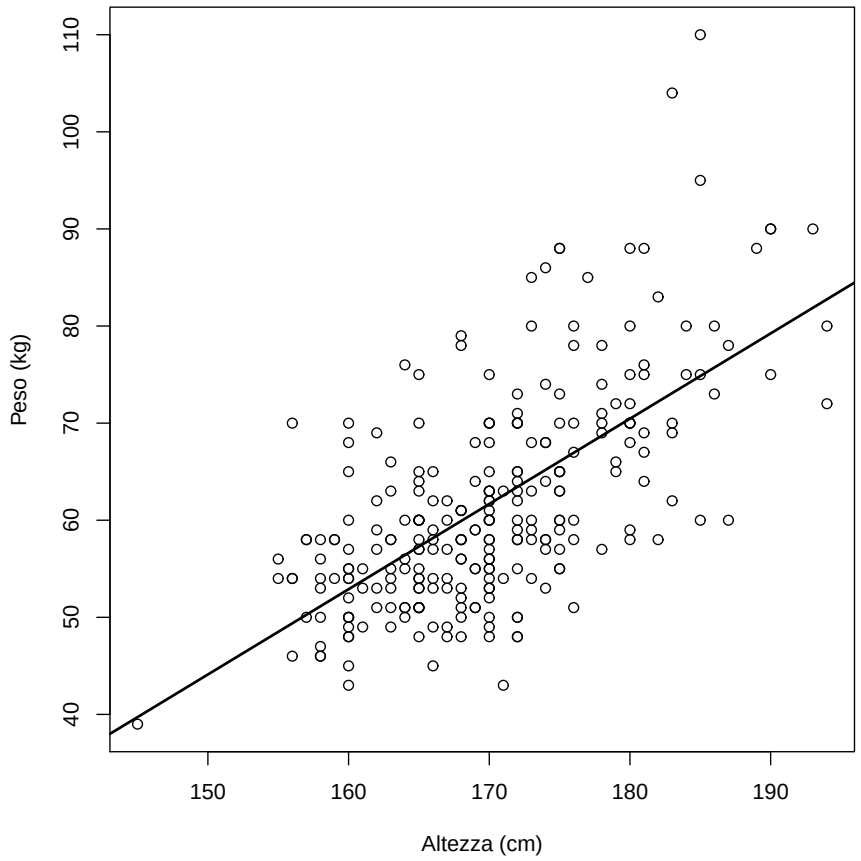


Figura 4. Peso e altezza degli studenti con la retta di regressione interpolante.

```
> cor(tmp$yhat,tmp$peso)
[1] 0.6465347
> cor(tmp$altezza,tmp$peso)
[1] 0.6465347
```

Naturalmente è possibile calcolare in modo esplicito la retta a minimi quadrati, impiegando la funzione `lm` e quella che R chiama una *formula*:

```
> lm(peso ~ altezza, data=tmp)

Call:
lm(formula = peso ~ altezza, data = tmp)

Coefficients:
```

```
(Intercept)      altezza
   -87.6010       0.8781
```

L'argomento principale di `lm` è una *formula* che impiega al suo interno un carattere speciale: la *tilde*; `peso ~ altezza` si legge "peso modellato dall'altezza"; attraverso il secondo argomento (`data`) specifichiamo il nome del *data-frame* e possiamo evitare di scrivere il nome delle variabili per esteso (vale a dire includendo il nome del *data-frame* e il simbolo $). La funzione `lm` permette anche, tra le tante 'cose' che produce, di disegnare la retta di regressione, ma per fare questo è necessario assegnare l'*output* di `lm` ad un oggetto (che abbiamo chiamato `fit`):

```
> fit <- lm(peso ~ altezza, data=tmp)
```

Nello *script* che ha disegnato la retta di regressione, avremmo quindi potuto sostituire `abline(b0,b1,lwd=2)` con `abline(fit,lwd=2)` ottenendo lo stesso risultato (provate).

Come si può vedere dalla figura 4.10, fissata una altezza, i pesi osservati sono in generale diversi da quelli previsti; calcoliamo ora i *residui*, ovvero la differenza fra valori osservati e valori previsti, sfruttando il vettore `tmp$yhat` ottenuto in precedenza:

```
> tmp$res <- tmp$peso - tmp$yhat
> head(tmp)
  altezza peso     yhat       res
1     170   65 61.67517  3.324834
2     172   60 63.43136 -3.431356
3     173   58 64.30945 -6.309451
4     165   51 57.28469 -6.284691
5     165   54 57.28469 -3.284691
6     170   58 61.67517 -3.675166
```

Abbiamo anche visualizzato i dati relativi ai primi 6 studenti a titolo di esempio. Il primo studente ha un'altezza di 170 cm e un peso di 65 kg; il valore previsto dal modello è pari a 61.7 kg e, quindi, il peso osservato supera di circa 3.3 kg quello previsto. Nel caso del secondo studente, alto 172 cm il peso osservato (60 kg) risulta inferiore di circa 3.4 kg rispetto a quello previsto. Anche gli altri 4 residui sono negativi, ma questo non deve far pensare che sia così per tutti. In generale si dovrebbe osservare un numero simile di residui positivi e negativi; possiamo verificarlo sfruttando la funzione `sign` già impiegata in precedenza:

```
> table(sign(tmp$res))

 -1   1
146 118
```

La somma dei quadrati dei residui ottenuta con la retta stimata con il metodo dei minimi quadrati sui nostri dati è pari a

```
> sum(tmp$res^2)
[1] 19042.38
```

Se provassimo a sostituire i valori stimati per b_0 e per b_1 con qualsiasi altra coppia di valori (disegnando quindi una retta diversa da quella rappresentata in figura 4.10), la somma dei quadrati dei residui risulterebbe invariabilmente più grande. Facciamo soltanto una prova, lasciando eventualmente al lettore il compito di insistere con qualche altro tentativo:

```
> yyy <- -90 + 0.9*tmp$altezza
> sum((tmp$peso - yyy)^2)
[1] 19514.38
```

La somma dei quadrati dei residui prende il nome di *devianza residua*, mentre la somma dei quadrati delle differenze fra i valori osservati della variabile dipendente e la loro media prende il nome di *devianza totale*:

```
> (devRes <- sum(tmp$res^2))
[1] 19042.38
> (devTot <- sum((tmp$peso - mean(tmp$peso))^2))
[1] 32719.27
```

Osserviamo che la devianza residua è minore della devianza totale; la differenza prende il nome di *devianza spiegata* dalla regressione e può anche essere calcolata come somma dei quadrati delle differenze fra i valori previsti e la loro media (che è poi anche la media dei valori osservati):

```
> devTot - devRes
[1] 13676.89
> (devSpi <- sum((tmp$yhat - mean(tmp$yhat))^2))
[1] 13676.89
> mean(tmp$peso)
[1] 61.68182
> mean(tmp$yhat)
[1] 61.68182
```

Da questa ultima uguaglianza si ricava che la somma dei residui (calcolati con il metodo dei minimi quadrati) è nulla (a meno di errori di arrotondamento)

```
> sum(tmp$res)
[1] -4.916956e-12
```

Si potrebbe dimostrare che la somma della devianza spiegata dalla regressione e della devianza residua è sempre uguale alla devianza totale. Pertanto il rapporto fra la devianza spiegata dalla regressione e la devianza totale sarà compreso fra 0 e 1 e può essere interpretato come la proporzione della variabilità della variabile dipendente *spiegata* dalla (o attribuibile alla) regressione. A questo rapporto si dà il nome di *coefficiente di determinazione lineare* e il suo valore numerico è il *quadrato* del coefficiente di correlazione lineare, tanto è vero che si indica con il simbolo r^2:

```
> devSpi/devTot
[1] 0.4180071
> cor(tmp$altezza,tmp$peso)^2
[1] 0.4180071
```

Il coefficiente di determinazione lineare r^2 è zero quando r è zero, cioè quando la covarianza è nulla (e quindi anche b_1 è nullo); in questo caso la retta a minimi quadrati è rappresentata da una retta orizzontale con intercetta b_0 uguale alla media della variabile dipendente. r^2 vale 1 quando la devianza residua è zero, quindi quando tutti i residui sono nulli (il che vuol dire che i valori osservati sono esattamente allineati lungo una retta). La funzione `reglin` di `rmf` calcola pendenza e intercetta con il metodo dei minimi quadrati e dà anche il valore di r^2 (moltiplicato per 100 e indicato come 'percentuale di variabilità spiegata'), eliminando 'automaticamente' eventuali dati mancanti:

```
> reglin(df$altezza,df$peso)
Variabile dipendente  : $ df peso
Variabile indipendente: $ df altezza
Intercetta: -87.60099
Pendenza  : 0.878095
% variabilità spiegata: 41.80071
Numero di coppie di osservazioni: 264
Test t per l'assenza di regressione lineare: 13.71777
p-value = 1.212655e-32
```

Se "centrassimo" i valori della variabile indipendente (vale a dire se li sostituissimo con gli scarti dalla media), si potrebbe vedere che la stima di b_1 rimarrebbe invariata, mentre quella di b_0 diventerebbe uguale alla media aritmetica dei valori osservati della variabile dipendente:

```
> x <- tmp$altezza - mean(tmp$altezza)
> lm(peso ~ x, data=tmp)

Call:
lm(formula = peso ~ x, data = tmp)
```

```
Coefficients:
(Intercept)            x
    61.6818       0.8781

> mean(tmp$peso)
[1] 61.68182
```

È un risultato che dovevamo aspettarci. Infatti abbiamo detto che b_0 è il valore previsto in media quando la variabile indipendente vale zero; ma una variabile 'centrata' vale zero quando la variabile di origine ha un valore uguale alla media. Inoltre, una variabile 'centrata' ha media uguale a zero.

```
> mean(x)
[1] 6.889901e-15
```

Quindi da $b_0 = \mu_Y - b_1\mu_X$ segue che, quando $\mu_X = 0$, $b_0 = \mu_Y$. Inoltre, da tutto questo segue anche che, in una retta a minimi quadrati, quando la variabile indipendente assume un valore uguale alla sua media, il valore previsto per la variabile dipendente è la media della variabile dipendente:

```
> b0 + b1*mean(tmp$altezza)
[1] 61.68182
> mean(tmp$peso)
[1] 61.68182
```

In altre parole, qualsiasi retta di regressione a minimi quadrati passa per il punto di coordinate (μ_X, μ_Y).

Nell'esempio trattato abbiamo interpolato un'unica retta di regressione per tutta la casistica considerata (vedi la figura 4.10). Noi però sappiamo che parte della casistica è di genere femminile e parte di genere maschile. Potrebbe essere pù opportuno interpolare due distinte rette di regressione selezionando femmine e maschi impiegando un vettore logico (o due).

Tuttavia, nel caso della regressione, R possiede un argomento specifico (`subset`) che agevola il compito del ricercatore. Vediamolo all'opera nello *script* seguente dove (dopo aver selezionato come al solito soltanto i dati completi) abbiamo stimato pendenza e intercetta prima sulle femmine e poi sui maschi e, infine, abbiamo disegnato su un unico grafico sia i 'punti osservati', sia le due rette interpolate (vedi la figura 4.11).

```
> ok <- complete.cases(df$sex,df$altezza,df$peso)
> tmp <- df[ok,c(3,15,16)]
> (fit.f <- lm(peso ~ altezza, data=tmp,
+     subset=(sex=="F")))
```

```
Call:
lm(formula = peso ~ altezza, data = tmp,
subset = (sex == "F"))

Coefficients:
(Intercept)      altezza
   -12.5778       0.4197

> (fit.m <- lm(peso ~ altezza, data=tmp,
+     subset=(sex=="M")))

Call:
lm(formula = peso ~ altezza, data = tmp,
subset = (sex == "M"))

Coefficients:
(Intercept)      altezza
   -85.1864       0.8814

> plot(tmp$altezza,tmp$peso,type="n",
+      xlab="Altezza (cm)",ylab="Peso (kg)")
> points(tmp$altezza[tmp$sex=="F"],
+        tmp$peso[tmp$sex=="F"],pch=16)
> points(tmp$altezza[tmp$sex=="M"],
+        tmp$peso[tmp$sex=="M"],pch=0)
> abline(fit.f,lty=2,lwd=2)
> abline(fit.m,lty=3,lwd=2)
> legend(145,110,legend=c("F","M"),pch=c(16,0),
+    lty=c(2,3))
```

Osserviamo che i parametri sembrano differire abbastanza tra maschi e femmine, in particolare il coefficiente di regressione che prevede un aumento di circa 420 g/cm nelle donne e un valore più che doppio negli uomini (881 g/cm). Possiamo anche calcolare i valori previsti con i due modelli per alcuni valori di altezza, tenendo presente che in alcuni casi stiamo facendo una *estrapolazione* (ad esempio non ci sono studentesse la cui altezza supera 182 cm e non ci sono studenti di altezza inferiore a 170 cm).

```
> x <- seq(150,190,by=10)
> b.f <- coef(fit.f)
> b.m <- coef(fit.m)
> cbind(x,b.f[1]+b.f[2]*x,b.m[1]+b.m[2]*x)
       x
[1,] 150 50.37528 47.02175
```

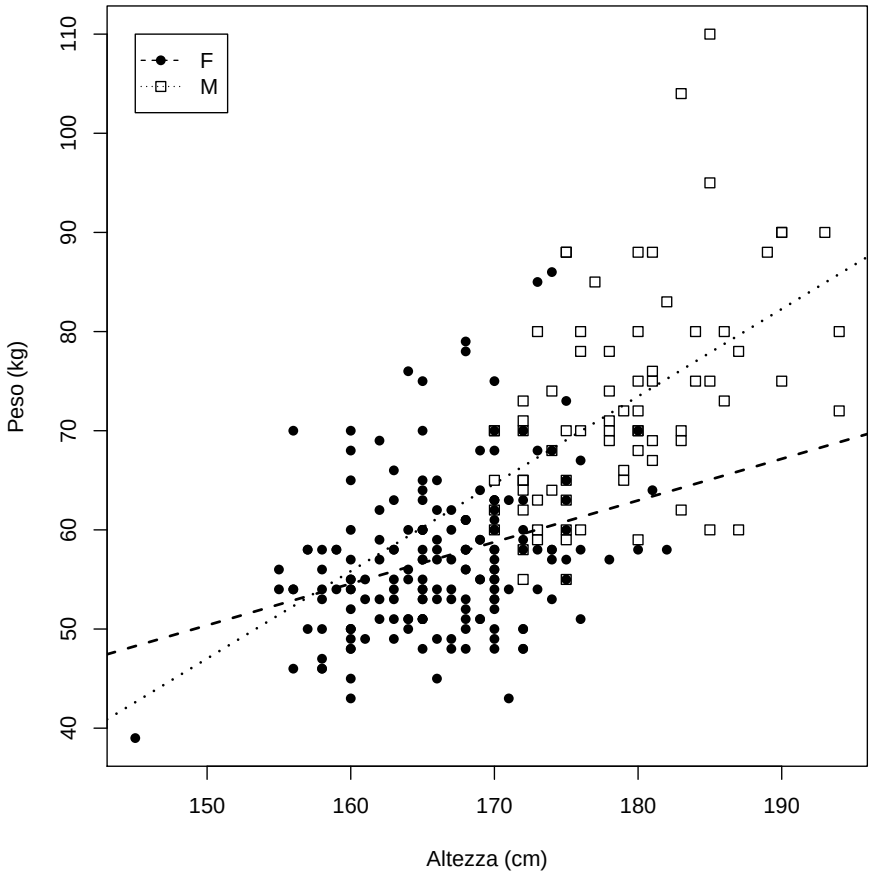


Figura 5. Peso e altezza di studentesse (cerchi) e di studenti (quadrati) con le due rette di regressione interpolanti.

```
[2,] 160 54.57215 55.83563
[3,] 170 58.76902 64.64951
[4,] 180 62.96589 73.46339
[5,] 190 67.16277 82.27726
```

Abbiamo impiegato la funzione `coef` per *estrarre* dagli oggetti prodotti da `lm` i parametri stimati, che sono stati copiati nei vettori `b.f` e `b.m`. Il primo elemento di ciascun vettore (al quale accediamo con `[1]`) contiene l'intercetta, mentre il secondo elemento (`[2]`) contiene la pendenza.

Qualche parola di commento sulla 'produzione' del grafico presentato in figura 4.11. Abbiamo innanzitutto impiegato l'argomento `type` con valore `"n"` per impostare la finestra grafica senza però disegnare i punti (se avrete la pazienza di eseguire lo *script* istruzione per istruzione ve ne accorgerete). Altri valori possibili per l'ar-

gomento `type` sono `"p"` (che è quello di *default* e disegna dei punti), `"l"` (che connette i punti, che non si vedono, con delle linee), `"b"` (che disegna sia i punti sia le linee che li connettono), `"o"` (con funzione simile alla precedente, salvo che le linee passano anche sopra i punti), `"h"` (che disegna dei segmenti verticali, come nella funzione `bastonicini` di `rmf`), `"s"` (che disegna dei segmenti a gradini). La funzione `points` disegna i 'punti', ma *aggiungendoli* su un grafico preesistente, mentre `abline` disegna le due rette un po' spesse (`lwd=2`), una 'tratteggiata' (`lty=2`) e una 'punteggiata' (`lty=3`), estraendo le informazioni dai due oggetti prodotti da `lm`. Infine, la funzione `legend` aggiunge una legenda al grafico.

Galton e la regressione verso la media

Il termine "regressione" porta con sè una connotazione negativa e non fa certo pensare ad una procedura per interpolare un modello lineare. Il nome ha origini storiche e venne usato da Francis Galton in un suo lavoro del 1886 intitolato "*Regression towards mediocrity in hereditary stature*". In questo lavoro Galton mise in relazione le altezze dei figli in età adulta con la media delle altezze dei loro genitori (che chiamò *mid-parent height*). Questi dati sono riportati nel *data-frame* `galton` disponibile in `rmf` e sono tratti dalla tabella I del suo articolo, alla quale si rimanda per ulteriori dettagli.

```
> data(galton)
> str(galton)
'data.frame':   892 obs. of  2 variables:
 $ mph: num  72.5 72.5 72.5 72.5 72.5 72.5 72.5 ...
 $ cah: num  68.2 69.2 69.2 70.2 71.2 71.2 72.2 ...
```

Il *data-frame* è costituito da 892 osservazioni e due variabili; la prima (`galton$mph`) riporta l'altezza (in *inches*) del *mid-parent* (ovvero la media delle altezze dei genitori); la seconda (`galton$cah`) riporta l'altezza (in *inches*) del figlio in età adulta.

Possiamo disegnare uno *scatterplot* dei dati (vedi il pannello di sinistra della figura 4.12) che, però, risulta scarsamente soddisfacente in quanto quasi ogni 'punto' del grafico corrisponde a più di una osservazione.

```
> plot(galton$mph,galton$cah,
+     xlab="Mid-parent height (Inches)",
+     ylab="Child adult height (Inches)")
```

Si può migliorare un po' l'effetto grafico, sfruttando la funzione `jitter` che aggiunge a ciascuna osservazione una 'piccola' quantità *random*, che permette di separare un po' i punti coincidenti (vedi il pannello di destra della figura 4.12)

```
> plot(jitter(galton$mph),jitter(galton$cah),
+      xlab="Mid-parent height (Inches)",
+      ylab="Child adult height (Inches)")
```

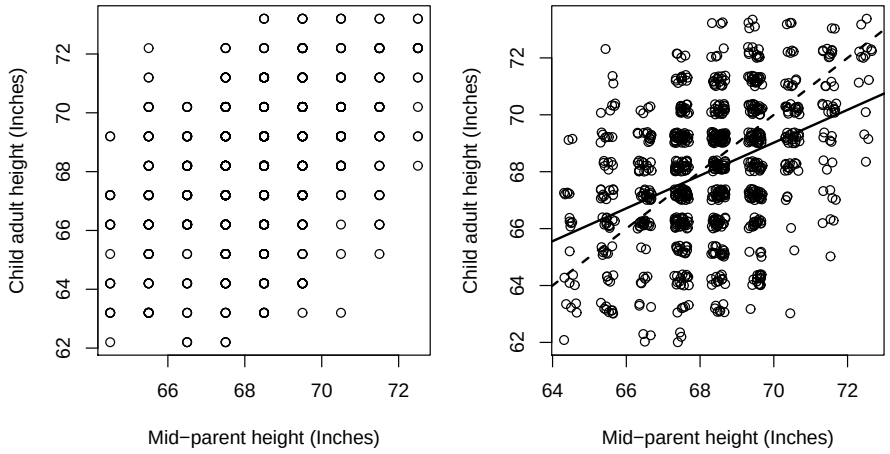


Figura 6. La regressione verso la media illustrata impiegando i dati originali di Galton.

Per avere il massimo dettaglio dei dati impieghiamo la funzione `table`. In colonna sono riportate le altezze (medie) dei genitori, mentre in riga quelle dei figli. Per giustificare i valori assegnati alle altezze dei figli (il valore decimale .2 invece di .5) Galton riferisce che le altezze dei figli tra 62 e 63 *inches*, 63 e 64, e così via sono distribuite in modo non uniforme, con una evidente distorsione a favore di valori interi e conclude: "*After careful consideration, I concluded that the headings, as adopted, best satisfied the conditions. This inequality was not apparent in the case of Mid-parents*".

```
> table(galton$cah,galton$mph)
```

	64.5	65.5	66.5	67.5	68.5	69.5	70.5	71.5	72.5
62.2	1	0	3	3	0	0	0	0	0
63.2	4	9	3	5	7	1	1	0	0
64.2	4	5	5	14	11	16	0	0	0
65.2	1	7	2	15	16	4	1	1	0
66.2	5	11	17	36	25	17	1	3	0
67.2	5	11	17	38	31	27	3	4	0
68.2	0	7	14	28	34	20	12	3	1
69.2	2	7	13	38	48	33	18	5	2
70.2	0	5	4	19	21	25	14	10	1
71.2	0	2	0	11	18	20	7	4	2
72.2	0	1	0	4	4	11	4	9	7
73.2	0	0	0	0	3	4	3	2	2

I parametri (stimati con il metodo dei minimi quadrati) della retta di regressione relativa all'altezza dei figli in funzione dell'altezza media dei genitori sono i seguenti

```
> (fit <- lm(cah ~ mph, data=galton))

Call:
lm(formula = cah ~ mph, data = galton)

Coefficients:
(Intercept)          mph
    28.6159       0.5772
```

e la retta di regressione interpolata $\hat{y} = 28.6 + 0.58x$ è presentata nella figura 4.12, insieme alla retta (tratteggiata) di intercetta uguale a 0 e pendenza uguale a 1 (la bisettrice).

```
> abline(fit)
> abline(0,1,lty=2)
```

La pendenza della retta interpolata prevede che se la differenza in altezza dei genitori è di 1 *inch*, il figlio dei genitori più alti sarà (in media) più alto del figlio di genitori più bassi, ma 'soltanto' di circa 0.58 *inches*.

Con il termine "regressione verso la media" (in realtà nel titolo del suo articolo Galton parla di *"Regression towards mediocrity"*) Galton voleva indicare proprio questo: i figli dei genitori più alti tendono ad essere più alti dei figli di genitori più bassi, ma non quanto ci si aspetterebbe in base all'altezza dei genitori. Possiamo verificarlo calcolando il valore medio dell'altezza dei figli all'interno dei singoli valori delle altezze dei genitori:

```
> round(tapply(galton$cah,galton$mph,mean),1)
64.5 65.5 66.5 67.5 68.5 69.5 70.5 71.5 72.5
65.6 66.8 67.1 67.6 68.0 68.6 69.5 69.9 71.4
```

Quando i genitori sono alti 64.5 *inches*, i figli sono alti, in media, 65.6 *inches* mentre quando i genitori sono alti 72.5 *inches*, i figli sono alti, in media, 71.4 *inches*, ovvero più dei figli dei genitori più bassi, ma meno dei propri genitori.

Altezza dei genitori e altezza dei figli sono fra loro correlate:

```
> cor(galton$cah,galton$mph)
[1] 0.4056477
> cor(galton$cah,galton$mph)^2
[1] 0.1645501
```

La correlazione è di poco superiore a 0.4, il che vuol dire (elevando 0.4 al quadrato) che poco più del 16% della variabilità dell'altezza dei figli è attribuibile all'altezza dei genitori (per lo meno considerando una relazione di tipo lineare).

La presenza di 'repliche' (ovvero di più osservazioni in corrispondenza dei valori della variabile indipendente), ci permette anche di calcolare il valore dell'indice 'eta' (il rapporto di correlazione).

```
> (devt <- sum((galton$cah - mean(galton$cah))^2))
[1] 5009.169
> tmp <- tapply(galton$cah,galton$mph,
+     function(z) sum((z-mean(z))^2))
> (deve <- sum(tmp))
[1] 4150.723
> (devf <- devt - deve)
[1] 858.4465
> devf/devt
[1] 0.171375
> sqrt(devf/devt)
[1] 0.4139747
```

Come si può vedere il valore di r^2 (0.165) e quello di η^2 (0.171) sono molto vicini fra loro. Dal momento che il rapporto di correlazione misura il grado di avvicinamento ad una relazione funzionale, possiamo pensare che non ci sono forti evidenze per ritenere che tale relazione sia diversa da quella lineare.

In realtà, più che una spiegazione genetica, la regressione verso la media ha una spiegazione statistica, come fanno vedere Bland e Altman. Proviamo infatti a 'scambiare' il ruolo delle due variabili

```
> (fit <- lm(mph ~ cah, data=galton))

Call:
lm(formula = mph ~ cah, data = galton)

Coefficients:
(Intercept)          cah
    48.9198       0.2851

> round(tapply(galton$mph,galton$cah,mean),1)
62.2 63.2 64.2 65.2 66.2 67.2 68.2 69.2 70.2 71.2
72.2 73.2 66.6 66.8 67.8 67.8 67.7 67.9 68.3 68.5
69.0 69.1 70.2 70.2
```

e vediamo che i genitori di figli più alti sono più bassi dei loro figli!

La spiegazione 'statistica' si comprende meglio se si introduce una formula 'alternativa' per il calcolo del coefficiente di regressione b_1. Infatti, questo può essere calcolato sia come rapporto fra la covarianza delle due variabili divisa per la varianza della variabile indipendente, sia nel modo seguente

$$b_1 = r\frac{\sigma_Y}{\sigma_X}$$

dove σ_Y è la deviazione standard della variabile dipendente e σ_X è la deviazione standard della variabile indipendente.

Dal momento che $|r|$ è sempre minore di 1 (salvo il caso di una perfetta relazione lineare, che non esiste (quasi) mai nella realtà), questo vuol dire che quando la variabile indipendente X cambia di una deviazione standard, la variabile dipendente Y cambia di r deviazioni standard (con $|r| < 1$). Pertanto, per ciascun dato valore di X, il valore previsto di Y è distante dalla propria media meno deviazioni standard di quanto sia distante X dalla propria media.

Inoltre, dalla relazione precedente che lega b_1 ad r, segue che, qualora le deviazioni standard σ_X e σ_Y delle due variabili siano uguali, il coefficiente di regressione b_1 sarà esattamente uguale a r e, quindi, minore di 1 in valore assoluto (abbiamo già detto che il caso $|r| = 1$ non è quasi mai un caso reale). A maggior ragione, nel caso in cui $\sigma_Y < \sigma_X$, si avrà $|b_1| < |r| < 1$. Soltanto se $\sigma_Y > \sigma_X$ sarà possibile che $|b_1| > 1$, ma questo dipenderà sia dall'entità di $|r|$ sia dal valore del rapporto σ_Y/σ_X.

Nel caso descritto da Galton le varianze (e le deviazioni standard) delle altezze di genitori e figli sono alquanto diverse fra loro.

```
> apply(galton,2,sd)
     mph      cah
1.666310 2.371068
> apply(galton,2,var)
     mph      cah
2.776589 5.621963
```

In particolare la varianza delle altezze dei genitori (2.78) è circa la metà della varianza delle altezze dei figli (5.62). Dobbiamo però ricordare che le altezze dei genitori sono medie di due osservazioni e che le medie hanno una variabilità più bassa di quella dei valori che la costituiscono; in particolare la varianza della media di due osservazioni è la metà della varianza dei dati originali. Questo spiega il risultato osservato.

Grazie a questo il rapporto σ_Y/σ_X fra le deviazioni standard delle due variabili risulta ben maggiore di 1:

```
> sd(galton$cah)/sd(galton$mph)
[1] 1.422945
```

(è quasi uguale a $\sqrt{2}$) ma, data la bassa correlazione fra le due variabili (circa 0.4), il loro prodotto non riesce a superare il valore 1. Per ottenere questo risultato sarebbe stata necessaria una correlazione almeno pari a 0.7.

Come fanno notare Bland e Altman, il fenomeno della regressione verso la media è presente in numerose circostanze. Fra i vari esempi proposti, i due Autori citano il caso di uno studio in cui era stato chiesto ad un campione di persone di riferire (senza pesarsi) il proprio peso. I soggetti erano stati successivamente pesati ed il peso misurato era stato confrontato con quello riferito impiegando un modello di regressione lineare in cui la variabile dipendente era il peso riferito e la variabile indipendente era il peso effettivamente misurato. La pendenza della retta di regressione

era risultata minore di uno. Quindi, il peso riferito da un soggetto "pesante" era (mediamente) inferiore al peso reale, mentre il peso riferito da un soggetto "leggero" era (mediamente) superiore al peso reale. È immediata (ma superficiale) l'interpretazione "psicologica": le persone sovrappeso tendono a sottostimare il loro peso, mentre quelle sottopeso tendono a sovrastimarlo. Infatti, se scambiando la variabile dipendente e quella indipendente, trovassimo ancora una volta una pendenza minore di uno, l'interpretazione sarebbe opposta: le persone sovrappeso sovrastimano il loro peso, mentre quelle sottopeso lo sottostimano. Il risultato osservato può essere invece attribuito (almeno in parte) alla regressione verso la media.

Lo psicologo Daniel Kahneman, premio Nobel per l'Economia, afferma che la regressione verso la media potrebbe spiegare la teoria secondo cui i rimproveri migliorerebbero la *performance*, mentre le lodi la peggiorerebbero.

Anche l'interpretazione dei risultati di test che valutano il livello di conoscenza deve tener conto della regressione verso la media. Uno studente che supera un test conseguendo un punteggio molto elevato tenderà (in media) a conseguire un voto più basso quando il test viene ripetuto e viceversa. Per valutare se un intervento produce un reale miglioramento è fondamentale la presenza di un gruppo di controllo.

Anche nello sport è presente la regressione verso la media. Un atleta che in un certo anno ottiene delle ottime prestazioni, tenderà (in media) a non ripeterle l'anno successivo (e viceversa un giocatore con basse prestazioni tenderà, in media, a migliorarle). Basta guardare le statistiche sui giocatori americani di *baseball*.

È possibile verificare quanto abbiamo appena detto mediante una simulazione. Supponiamo che un test sia costituito da 100 domande di tipo Vero/Falso e che ogni studente risponda a caso a ciascuna domanda. Quello che ci aspettiamo è che, *in media*, uno studente risponda correttamente a (circa) 50 domande. Simuliamo la distribuzione del punteggio ottenuto da una classe di 200 studenti impiegando la funzione `rbinom`:

```
> set.seed(123456)
> x <- rbinom(200,100,0.5)
> mean(x)
[1] 50.265
```

Non entriamo nei dettagli di questa funzione, per i quali rimandiamo al nostro testo "*Problemi ed Esperimenti di Statistica con R*". Prendiamo ora il 10% degli studenti che hanno riportato il punteggio peggiore:

```
> y <- sort(x)
> (y <- y[1:20])
 [1] 37 37 37 38 39 41 41 41 41 41 42 42
[13] 42 42 43 43 43 43 43 43
> mean(y)
[1] 40.95
```

e osserviamo che la loro media è, del tutto comprensibilmente, alquanto inferiore a 50. Supponiamo ora di programmare un intervento educativo su questi 20 studenti per cercare di migliorare il loro risultato, ma che l'intervento non abbia alcun esito; vale a dire che gli studenti sceglieranno ancora a caso la risposta. Ripetiamo a questo punto il test su questi studenti alla fine dell'intervento educativo:

```
> x <- rbinom(20,100,0.5)
> mean(x)
[1] 50.45
```

e osserviamo che il loro punteggio medio è migliorato, anche se gli studenti hanno continuato a rispondere a caso. La spiegazione è data dalla regressione verso la media. Per verificare concretamente che l'intervento educativo non ha avuto alcun esito sarebbe stato necessario dividere a caso i 20 studenti in due gruppi, sottoponendo all'intervento solo uno dei due gruppi, lasciando l'altro come controllo. Si sarebbe allora visto che il punteggio medio dei due gruppi sarebbe stato simile (con un valore intorno a 50) e migliore del punteggio iniziale. Come illustra questo esempio, è molto importante tener conto della regressione verso la media quando si programma un esperimento. In particolare è fondamentale avere un gruppo di controllo.

Approfondimento: La regressione lineare in simboli

Date due variabili X (variabile indipendente) e Y (variabile dipendente) rilevate su N unità statistiche, indichiamo con x_i e y_i la $i-$esima osservazione, con σ_X^2 la varianza della variabile indipendente, con σ_Y^2 la varianza della variabile dipendente, con σ_{XY} la covarianza fra X e Y.
Il coefficiente di regressione lineare è dato da

$$b_1 = \frac{\sigma_{XY}}{\sigma_X^2} = r\frac{\sigma_Y}{\sigma_X}$$

L'intercetta è data da

$$b_0 = \mu_Y - b_1\mu_X$$

I valori previsti sono dati da

$$\hat{y}_i = b_0 + b_1 x_i$$

I residui sono dati da

$$e_i = y_i - \hat{y}_i$$

La devianza totale è data da

$$DT = \sum_{j=1}^{N}(y_i - \mu_Y)^2$$

La devianza spiegata (dalla regressione) è data da

$$DS = \sum_{j=1}^{N}(\hat{y}_i - \mu_Y)^2$$

La devianza residua è data da

$$DR = \sum_{j=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{j=1}^{N} e_i^2$$

Il coefficiente di determinazione lineare è dato da

$$r^2 = \frac{DS}{DT}$$