Università degli Studi di Trieste

Dipartimento di Matematica e Geoscienze

QUADERNI DIDATTICI

Massimo Borelli

SUMMER SCHOOL 2012 IN BIOSTATISTICA:

Dall'analisi univariata alle analisi multivariate negli studi cross-section e negli studi longitudinali

Quaderno n.60
Dicembre 2012

Edizione fuori commercio

SONO STATI ADEMPIUTI GLI OBBLIGHI DI LEGGE
D.P.R. 03/05/06 nr. 252 (G.U. nr. 191 del 18/06/06)

Dipartimento di Matematica e Geoscienze
Università degli Studi di Trieste

ASSOCIAZIONE NAZIONALE
BIOTECNOLOGI ITALIANI

Biotech Lifelong Learning Network

SUMMER SCHOOL 2012 IN BIOSTATISTICA

Dall'analisi univariata alle analisi multivariate
negli studi cross-section e negli studi longitudinali

Università degli Studi di Bologna, 14-15 giugno 2012.

Introduction

The following pages contain the lecture notes of a two-days course in biostatistics held at the University of Bologna, organized by the Associazione Nazionale Biotecnologi Italiani (ANBI) in collaboration with the Federación Española de Biotecnólogos (Febiotec) within the '*Biotech Lifelong Learning Network*' project.

The aims of the Summer School were to refresh or to introduce participants into some typical questions arising in data analysis both in cross-section and in longitudinal / repeated measures frameworks: to summarize and present data; to assess differences between groups; to perform 'multivariate' analysis; to evaluate time-to-event (i.e. survival) data.

Dealing with such questions require to manage a statistical package, and our experience acquired with the open source software R [23] allow to say that, actually, this can be the best software in biotech scenery, together its companion Bioconductor. Moreover, for the novices, the graphic interface *R Commander* helps to overcome some initial difficulties in manipulating data and perform basic analysis.

Our course has been structured in a way that at the end of every chapter participants can exercise with real dataset similar to those presented, and the results are discussed together. Clearly a two-day summer school can not achieve the level of depth of a two-year class, and therefore I surely refer the readers to some basic books as the following:

- [3] Bland M. (2000). *An introduction to medical statistics*, Oxford University Press.

- [4] Bretz F., Hothorn T., Westfall P. (2010). *Multiple Comparisons Using R*, CRC Press.
- [10] Crawley M. J. (2007). *The R Book*, Wiley.
- [13] Espa G., Micciolo R. (2012). *Analisi esplorativa dei dati con R*, Apogeo.
- [14] Faraway J. (2004). *Linear Models with R*, Chapman Hall/CRC.
- [15] Faraway J. (2005). *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*, Chapman Hall/CRC.

Indice

Introduction **i**

1 Richiami di statistica descrittiva **1**
1.1 La descrizione dei dati 1
1.1.1 La media e la deviazione standard 1
1.1.2 La mediana e i quartili 3
1.1.3 .. ed inoltre: 6
1.1.4 Laboratorio 10

2 Cenni di inferenza statistica **13**
2.1 I test statistici più consueti nei disegni Cross-Section 13
2.1.1 APPROFONDIMENTO: *When variances are different, do not compare the means!* 18
2.1.2 APPROFONDIMENTO: Perché si deve preferire l'approccio non parametrico ai dati non distribuiti normalmente? 20
2.2 I test statistici più consueti nei disegni longitudinali 21
2.2.1 Laboratorio 24

3 Stabilire associazioni tra due variabili **25**
3.1 I test statistici di associazione 25
3.1.1 Variabili fattore e variabili numeriche 25
3.1.2 Scegliere un cut-off appropriato 28
3.1.3 APPROFONDIMENTO: Il p-value dipende dalla dimensione del campione 33
3.1.4 APPROFONDIMENTO: Scegliere la dimensione corretta del campione 35
3.1.5 Laboratorio 36

4 Differenze tra gruppi **37**
4.1 L'analisi della varianza negli studi cross-section 37

4.1.1 Il caso introduttivo a due livelli 37
4.1.2 APPROFONDIMENTO: analisi delle medie o analisi della varianza? 41
4.1.3 Il caso di un predittore a più livelli 45
4.1.4 APPROFONDIMENTO: Post-hoc in presenza di un *gold standard* 49
4.2 Anova a due vie 50
4.2.1 APPROFONDIMENTO: l'Anova e la matrice del modello lineare 51
4.3 Le ipotesi matematiche dell'Anova 52
4.4 In conclusione: perché parliamo di modello Anova? 54
4.5 Laboratorio 55

5 Calibrare in R un insieme di dati con una retta di regressione 56
5.1 La retta di regressione negli esperimenti longitudinali 56
5.2 La retta di regressione negli esperimenti cross-section 57
5.2.1 APPROFONDIMENTO: Come *nascono* le stelline? . . 60
5.3 La diagnostica del modello lineare 61
5.4 Quando una covariata non è un predittore? 63
5.4.1 APPROFONDIMENTO: la parola *lineare* non vuol dire *rettilineo* 63
5.4.2 Laboratorio 64

6 Calibrare in R una serie nonlineare di dati con una 'curva di regressione' 65
6.1 Stime dei parametri di modelli completamente nonlineari . . . 65
6.2 Stime dei parametri di modelli riconducibili a modelli lineari . 66
6.2.1 Il modello iperbolico. 67
6.2.2 Il modello esponenziale. 67
6.2.3 Il modello *maxima function*. 68
6.2.4 Il modello potenza. 69
6.2.5 Il modello logistico. 71
6.2.6 Il modello sigmoidale. 72

7 Ancova: alla ricerca del modello lineare minimale adeguato in R 73
7.1 La *Ancova* 73
7.1.1 APPROFONDIMENTO: nell'Ancova l'ordine è importante 76
7.1.2 APPROFONDIMENTO: il modello lineare e l'algebra 76

8 La regressione logistica e la regressione di Poisson con R negli studi cross-section **81**
8.1 La regressione logistica 81
8.1.1 Come interpretare i parametri della regressione logistica 83
8.1.2 Il problema della sovradispersione 84
8.2 La regressione di Poisson 85
8.3 Laboratorio 86

9 Curve di sopravvivenza con R **87**
9.1 La distribuzione dei *time-to-event* non è gaussiana 87
9.2 L'analisi di sopravvivenza 88
9.3 Laboratorio 90

10 Studi longitudinali e misure ripetute **91**
10.1 La questione delle misure ripetute nei design longitudinali . . 91
10.2 Laboratorio intermedio 94
10.3 La questione delle misure ripetute negli studi cross-section . . 95
10.3.1 APPROFONDIMENTO: Come '*si scrive*' in un paper tutto ciò? 96

Bibliografia **97**

Capitolo 1

Richiami di statistica descrittiva

1.1 La descrizione dei dati

In questa sezione ci occupiamo di ciò che non può mancare in un articolo scientifico nel quale vengano analizzati dei dati: la **statistica descrittiva univariata**. Si tratta di assegnare con raziocinio un opportuno **indice di centralità** ed un altrettanto opportuno **indice di dispersione** che riassumano i dati in esame in maniera di trasmettere al lettore sufficiente informazione sul loro 'comportamento'. Non ci si deve limitare esclusivamente nel riportare nei propri lavori un indice di centralità, pena l'equivoco fatale magistralmente cantato dal poeta Carlo Alberto Salustri (Trilussa) nella sua poesia intitolata *La Statistica*[1].

1.1.1 La media e la deviazione standard

Forse non è azzardato paragonare la **media** alla *Gioconda* di Leonardo da Vinci: la conoscono tutti e non ne vale nemmeno la pena di parlarne.. o forse sì, visto che si sorride amaramente sulla famosa frase '*mediamente, gli esseri umani hanno 1.99 gambe*'; o sorridiamo anche se diciamo che su cinque lanci di una moneta ci aspettiamo in media 2.5 teste e 2.5 croci. Il problema è proprio questo: la media ha a che fare sia con il 'baricentro' dei dati che analizziamo, ma anche con ciò che è il **valore atteso** di un certo fenomeno casuale; e non è detto che quel valore atteso, nonostante la sua alta plausibilità, sia effettivamente un **evento** possibile dell'**esperimento** aleatorio che stiamo studiando.

[1] `http://it.wikipedia.org/wiki/Trilussa#Il_pollo_di_Trilussa`

Gli indici di dispersione che *vanno d'accordo* con la media sono due: la **varianza** e la **deviazione standard**. In pratica i due indici di dispersione sono la stessa cosa, visto che il secondo si ottiene elevando al quadrato il primo. La deviazione standard però è preferibile perché ha la medesima unità di misura dei dati (e della loro media). Tanto per fare una battuta, se mi dicono che ci sono tre donne che mi stanno cercando, di età media 26 *anni* e deviazione standard 37 *anni* (cioè una varianza di 1339 *anni quadrati*), intuisco immediatamente che non si tratta di tre studentesse del corso, ma di mia mamma e delle mie due bimbe!

C'è da dire che la varianza e la deviazione standard sono dei buoni indici di dispersione perché hanno in sé la nozione di 'distanza dal centro', come si capisce dalla analogia con la formula della distanza di due punti sul piano cartesiano[2]. Infatti, nella formula algebrica compare un'espressione del tipo:

$$(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2$$

che ci ricorda immediatamente il teorema di Pitagora.

In R la media si calcola con la funzione `mean`, mentre la deviazione standard e la varianza si calcolano con le funzioni `sd` e `var`. Vediamo dunque di importare il dataset `metabolismo` per scoprire che il peso medio dei partecipanti al tempo iniziale del *trial* era di circa 67 chilogrammi, con una deviazione standard di circa 6 Kg:

```
rm(list = ls())
www <- "http://www.dmi.units.it/~borelli/dataset/metabolismo.txt"
metabolismo <- read.table(www, header = TRUE)
attach(metabolismo)
names(metabolismo)
mean(pesoT0)
sd(pesoT0)
```

Se raffiguriamo i dati con un istogramma, ci accorgiamo che essi hanno un andamento confortante (che fa l'immediata felicità di molti studiosi): le barre dell'istogramma sembrano essere in accordo con una possibile forma a campana, tipica della **distribuzione normale** (o **gaussiana**) dei dati. E se esaminiamo gli stessi dati con un particolare grafico - del quale rimandiamo la spiegazione fra qualche pagina - che si chiama **QQplot** (ovvero, **grafico quantile - quantile**), vediamo che i dati si adagiano lungo la diagonale: segno patente di normalità.

[2] `http://en.wikipedia.org/wiki/Distance`

```
par(mfrow = c(1,2))
hist(pesoT0)
qqnorm(pesoT0)
qqline(pesoT0)
detach(metabolismo)
rm(metabolismo)
```

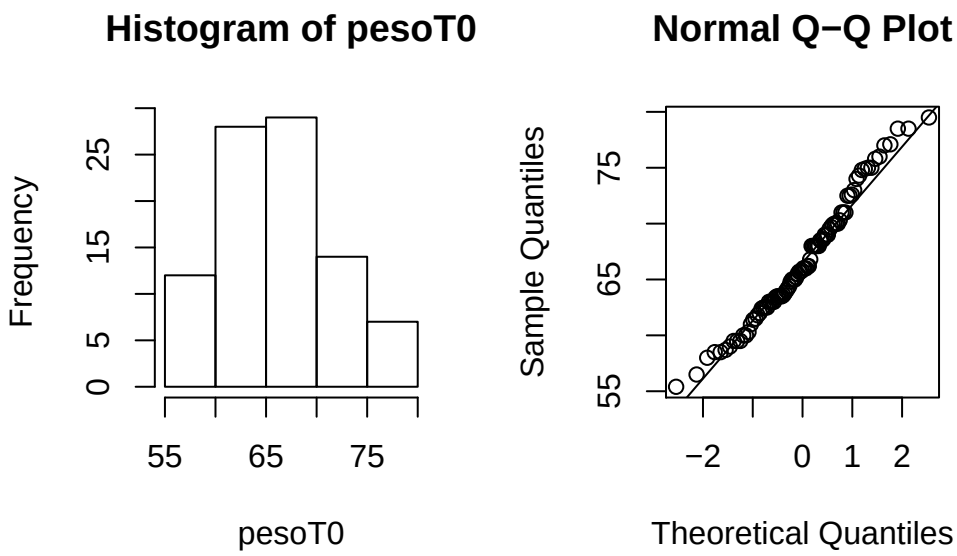

Figura 1.1: Istogramma e QQplot del `pesoT0` del dataset `metabolismo`.

E' bello avere a che fare con dati normali, perché in quel caso l'uso della media e della deviazione standard sono proprio azzeccati; avrebbe anche un certo senso scrivere nelle tabelle riassuntive, come spesso vediamo, la dicitura $67(\pm 6)$, perché in questo modo suggeriamo implicitamente al lettore che circa i due terzi[3] dei dati che stiamo esaminando sono compresi nel *range* che va da $67 - 6 = 61$ chilogrammi e $67 + 6 = 73$ chilogrammi. Ritorneremo fra poco su questo argomento, segnalndo un frequente misuso.

1.1.2 La mediana e i quartili

Dicevamo che è bello avere a che fare con dati normali. Ma, non so voi, a me capita di rado! E se i dati non sono normali, la media e la deviazione standard

[3] `http://en.wikipedia.org`, Gaussian distribution, Standard deviation and confidence intervals

continuano a mantenere il loro irresistibile *appeal*? Beh, non proprio. L'età media nel dataset `candida` è circa 76 anni, con una deviazione standard di 15 anni.

```
rm(list = ls())
www <- "http://www.dmi.units.it/~borelli/dataset/candida.txt"
candida <- read.table(www, header = TRUE)
attach(candida)
mean(eta)
sd(eta)
```

Ma in questo caso non è affatto vero che - come accade per definizione con i dati normali, vedi Figura 1.3 - circa i 2/3 dei nostri pazienti, ossia 97 persone, abbiano eta compresa tra $76 - 15 = 61$ anni e $76 + 15 = 91$ anni; al contrario, su 145 pazienti totali, solo 12 superano i 91 anni d'età e 23 invece hanno età inferiore alla soglia di 61 anni. La distribuzione dei dati dunque non è simmetrica; inoltre, il 24 per cento circa del campione si discosta (in più o in meno, si intende) di **una deviata** dalla media.

```
length(eta)
sum(eta > 91)
sum(eta < 61)
(12+23)/145
```

In questi casi sarebbe preferibile rinunciare a rappresentare i dati con il consueto istogramma, ma passare alla rappresentazione per mezzo del 'diagramma a scatola e baffi' di John Tukey, ovvero il **boxplot**:

```
boxplot(eta)
summary(eta)
```

E' facile capire cosa rappresentino i baffetti e la scatola assieme alla 'riga grossa' in mezzo alla scatola se si va a leggere l'output del comando `summary(eta)`. La **mediana** di circa 81 anni significa che metà dei pazienti arruolati nello studio avevano meno di 81 anni, e l'altra metà di essi era più anziana. Il **primo quartile** vale circa 70 anni: dunque il 25% di essi non superava quell'età. Il **terzo quartile** vale circa 86 anni. I pallini che vediamo nella parte inferiore del grafico rappresentano i dati **outlier**, i dati cioè che si discostano *parecchio* (ossia più di una volta e mezza) dalla ampiezza interquartile (ossia `1.5 * (97.92 - 69.99)`).

Dunque, quando la distribuzione non è simmetrica, la media e la deviazione standard non servono più a nulla? Beh, rimangono comunque due

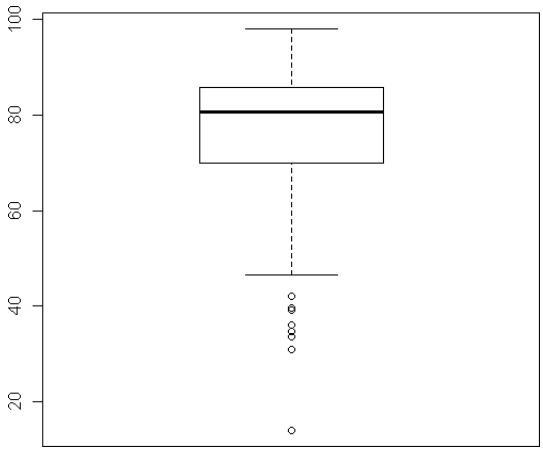

Figura 1.2: Boxplot della `eta` del dataset `candida`.

indici importanti che ci consentono di trarre alcune informazioni sul dataset: il matematico russo Pafnutij Čebyšëv ha dimostrato che *per qualsiasi* set di dati, se sono note la media μ e la deviazione standard σ, allora per ogni k è vero che:

$$P(|x - \mu| \geq k \cdot \sigma) \leq \frac{1}{k^2}$$

Il significato probabilistico di questa disugualianza, detto in termini semplici, è che (ad esempio se $k = 2$) i valori che distano dalla media per più di $k = 2$ deviate non possono essere più del 25 per cento. Cioè, al massimo il 75 per cento dei dati è compreso nell'intervallo $[\mu - 2\sigma, \mu + 2\sigma]$. Si tratta di una stima 'grossolana', che però non si può migliorare in mancanza di altre informazioni; ecco che però se disponiamo di un'informazione ulteriore, quella della normalità dei dati, le cose vanno già molto meglio - perché, lo ricordavamo prima, il 68 per cento circa dei dati è compreso nel range di una deviata (vedi nuovamente Figura 1.3).

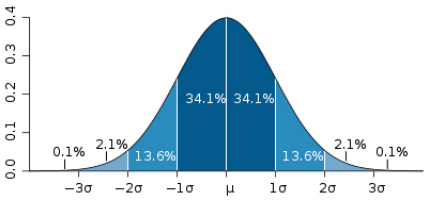

Figura 1.3: Verificate la proprietà della normale, per esempio con il comando `pnorm(1) - pnorm(-1)`. L'immagine è tratta da Wikipedia, dall'indirizzo `http://en.wikipedia.org/wiki/68-95-99.7_rule`.

1.1.3 .. ed inoltre:

.. ed inoltre, questa è l'occasione buona per mettere a fuoco quattro argomenti su cui spesso non vi è chiarezza, o vi sono delle credenze errate:

- l'entropia di Shannon
- la standardizzazione
- il QQ plot
- lo standard error

L'entropia di Shannon

Dal National Center for Biotechnology Information[4] ci siamo scaricati il dataset `mecp2`, relativo alla proteina denominata 'methyl CpG binding protein 2', coinvolta in una malattia neurologica detta Sindrome di Rett:

```
rm(list = ls())
www <- "http://www.dmi.units.it/~borelli/dataset/mecp2.txt"
mecp2 <- read.table(www, header = TRUE)
attach(mecp2)
str(mecp2)
head(mecp2)
names(mecp2)
summary(mRNA)
```

Siccome il dataset contiene solo una variabile fattore, nella quale sono presenti quattro livelli, non ha senso preoccuparsi di né di medie (non possiamo sommare lettere!) né di mediane (non ha senso dire che la Citosina venga prima della Timina, se non in ordine alfabetico); dunque nemmeno di deviazioni standard e di quartili. Come misura di centralità possiamo utilizzare la **moda**, ossia il dato con la maggior frequenza (assoluta o relativa):

```
table(mRNA) / length(mRNA)
table(mRNA)[[2]] / length(mRNA)
```

Esiste anche una misura di eterogeneità detta **entropia di Shannon**[5] la quale fornisce un'indicazione se i dati siano 'equamente distribuiti', oppure se siano 'polarizzati':

[4] `http://www.ncbi.nlm.nih.gov/nuccore/383873279?report=fasta`
[5] `http://en.wikipedia.org/wiki/Entropy_(information_theory)`

$$H = \sum_{i=1}^{4} f_i \cdot \log_2(\frac{1}{f_i})$$

Per calcolare l'entropia, possiamo sfruttare le capacità del linguaggio di programmazione R, utilizzando un ciclo `for`:

```
entropia <- 0
for (i in 1:4)  {
   frequenza <- table(mRNA)[[i]] / length(mRNA)
   entropia <-  entropia + frequenza * log2( 1 / frequenza )
   }

entropia
```

Otteniamo un valore circa uguale a 1.998, che è vicino a 2 (ossia il valore massimo di entropia che si ottiene quando abbiamo a che fare con 4 eventi ciascuno di probabilità 1/4), ma ben distante dallo 0 (ossia la situazione in cui solo uno degli eventi si realizza con certezza, e gli altri hanno probabilità nulla).

Standardizzare una variabile

Capita talvolta che si debbano analizzare due (o più) variabili che hanno scale di misura diverse, e che in una di esse ad esempio i numeri siano 'molto grandi' rispetto all'altra. Questo fatto potrebbe comportare delle difficoltà nella stima di alcuni parametri statistici, soprattutto se il calcolo di tali parametri comporta che nelle *routine* interne di R vi siano degli algoritmi iterativi. Si può utilizzare comodamente la funzione `scale`, oppure fare lo cose da sé seguendo la formulina riportata da Wikipedia[6]. Facciamo una verifica sul dataset `tms`:

```
rm(list = ls())
www <- "http://www.dmi.units.it/~borelli/dataset/tms.txt"
tms <- read.table(www, header = TRUE)
attach(tms)
boxplot(amplitude, intensity)

rsamplitude <- ( amplitude - mean(amplitude) ) / sd(amplitude)
rsintensity <- ( intensity - mean(intensity) ) / sd(intensity)
boxplot(rsamplitude, rsintensity)
```

[6] http://en.wikipedia.org/wiki/Standard_score

Fate attenzione che la standardizzazione opera solo un cambiamento di scala, come accade quando passiamo dai gradi centigradi europei ai gradi Fahrenheit statunitensi; non viene intaccata la 'forma' della distribuzione dei dati (i.e., non è che se si ha a che fare con dati non distribuiti normalmente la standardizzazione li rende magicamente normali; per questa faccenda conviene affidarsi alla `library(MASS)` ed al suo comando `boxcox`: ne riparleremo nel Capitolo 5.

Il QQplot

Adesso che abbiamo capito cosa siano i quartili, possiamo spiegare velocemente il senso del QQ plot. Mettiamo in ordine crescente le età del dataset `candida`:

```
rm(list = ls())
www <- "http://www.dmi.units.it/~borelli/dataset/candida.txt"
candida <- read.table(www, header = TRUE)
attach(candida)
etaordine <- sort(eta)
```

e scopriamo con il comando `length(etaordine)` che abbiamo a che fare con 145 dati. Suddividiamo il grafico della curva normale standard in 145 'fettine', in modo tale che queste fettine abbiano la medesima area: facciamo questo perché immaginiamo che i nostri 145 siano stati estratti a caso da una variabile aleatoria normale, e quindi imbattersi nell'uno o nell'altro deve essere un evento equiprobabile, come pescare la pallina blu o la pallina verde in mezzo ad altre 143 palline rosse. Il trucco è considerare i numeri che vanno da 1 a 145, `1:145`, 'spostarli di un pochino a sinistra' (`(1:145)-0.5`), dividere il tutto per 145, `((1:145)-0.5)/145`, in modo da avere a che fare con una sequenza di numeri compresi nell'intervallo che va da zero ad uno. Infine, usare tale sequenza per trovare i punti sull'asse delle ascisse (ossia i **quantili**) che individuano le 'fettine' del grafico della normale standard di area uguale:

```
qnorm(((1:145)-0.5)/145)
```

Adesso è immediato verificare che il comando:

```
plot( qnorm(((1:145)-0.5)/145), etaordine )
```

produce un grafico che è esattamente il QQplot che si sarebbe ottenuto con `qqnorm(eta)` .

Lo standard error

Nei due paragrafi precedenti abbiamo visto che la media va d'accordo con la deviazione standard, e che la mediana va d'accordo con i quartili. Ogni misura di centralità va a braccetto con una misura di dispersione. E in molti paper si vede più di qualche volta che gli autori, quando hanno a che fare con dati che sono espressi in percentuale, utilizzino - come se fosse un indice di dispersione - lo **standard error**.

Ad esempio, Kaur, Samant, Pramanik et al. nel loro articolo *Silencing of directional migration in roundabout4 knockdown endothelial cells*[7] si occupano di alcune molecole - guida denominate 'Roundabouts'. Nella loro figura 3, seppure in tre pannelli diversi, appaiono dati conteggiati e dati espressi in percentuale, e le **barre d'errore** talvolta rappresentano la deviazione standard e talvolta lo standard error: questo genera molta confusione per il lettore. Lieske, Thoby-Brisson, Ramirez et al. in un celebre articolo di neuroscienze della respirazione apparso su Nature[8], *Reconfiguration of the neural network controlling multiple breathing patterns: eupnea, sighs and gasps*, affermano semplicemente che nel loro lavoro 'Statistical values are given as mean ± s.e.'. Lo stesso accade nella tabella 4 di Johnson, Chen, Limburg et al. (*Accuracy of CT Colonography for Detection of Large Adenomas and Cancers*) pubblicata sul New England Journal of Medicine[9].

Si tratta di scelte discutibili. Martin Bland [3] raccomanda:

> *The mean and standard error are often written as* 4.062 ± 0.089. *This is rather misleading [...] This practice is not recommended.*

Nella sezione dedicata alla statistica descrittiva è bene riassumere i dati con un indice di centralità ed un indice di dispersione opportuni. Quindi, per dati distribuiti normalmente, media e deviazione standard sono ottimi. Per dati distribuiti non normalmente, mediana e quartili sono altrettanto ottimi. Martin Bland, per essere indulgente, 'scusa' gli autori che confondono la deviazione standard con lo standard error ricordando un aspetto teorico della statistica matematica, ossia che lo standard error è in effetti una deviazione standard: quella delle media campionaria.

Tralasciando i risultati teorici, noi vogliamo qui illustrare il fatto del perché lo standard error non sia l'oggetto 'giusto' per la statistica descrittiva,

[7] `http://www.ncbi.nlm.nih.gov/pubmed/18980679`
[8] `http://www.ncbi.nlm.nih.gov/pubmed/10816317`
[9] `http://www.ncbi.nlm.nih.gov/pubmed/18799557`

Tabella 1.1: Un milione di lanci (simulazione) negli esperimenti lancio di una monetina per dieci, cento, mille e diecimila volte (prove ripetute): la media e la deviazione standard aumentano, lo standard error rimane fisso.

Esperimento	`mean`	`sd`	`N`	`sd/sqrt(N)`
mega10	5.00	1.58	10	0.50
mega100	50.00	5.00	100	0.50
mega1000	500.00	15.81	1000	0.50
mega10000	4999.95	50.02	10000	0.50

ma per la **statistica inferenziale**, argomento dei prossimi capitoli. Immaginiamo quattro esperimenti distinti: il lancio di una moneta per dieci volte consecutive; per cento volte consecutive; per mille volte, e per diecimila volte consecutive. Simuliamo un milione di prove ripetute, per ciascuno di questi quattro esperimenti:

```
mega10  <- rbinom(1000000, size = 10, prob = 0.5)
mega100  <- rbinom(1000000, size = 100, prob = 0.5)
mega1000  <- rbinom(1000000, size = 1000, prob = 0.5)
mega10000  <- rbinom(1000000, size = 10000, prob = 0.5)
```

Nella Tabella 1.1 si legge ciò che la Figura 1.4 mette bene in evidenza: quando aumentano le prove ripetute del medesimo esperimento (da dieci a diecimila lanci), certamente la media e la deviazione standard cambiano. Ma questo invece non accade per lo standard error, che rappresenta quindi una ottima **misura di inaffidabilità** della stima puntuale della media, ma non è una misura di dispersione, perché - in proporzione - la variabilità diviene sempre più contenuta, come si vede dalla Figura 1.4. Per convincersi, si pensi se è probabile che lanciando dieci volte la moneta escano tre teste; o se lanciando diecimila volte la moneta, escano tremila teste. Eppure, non abbiamo dubbi che in dieci lanci mediamente avremo cinque teste; e in diecimila lanci avremo mediamente cinquemila teste).

1.1.4 Laboratorio

1. Innanzitutto, accertatevi se nel vostro R avete già installato il pacchetto *R Commander* digitando il comando `library(Rcmdr)`. Se vi appare una segnalazione di errore, allora dovete scaricarvi dalla rete il pacchetto aggiuntivo. Si fa così:

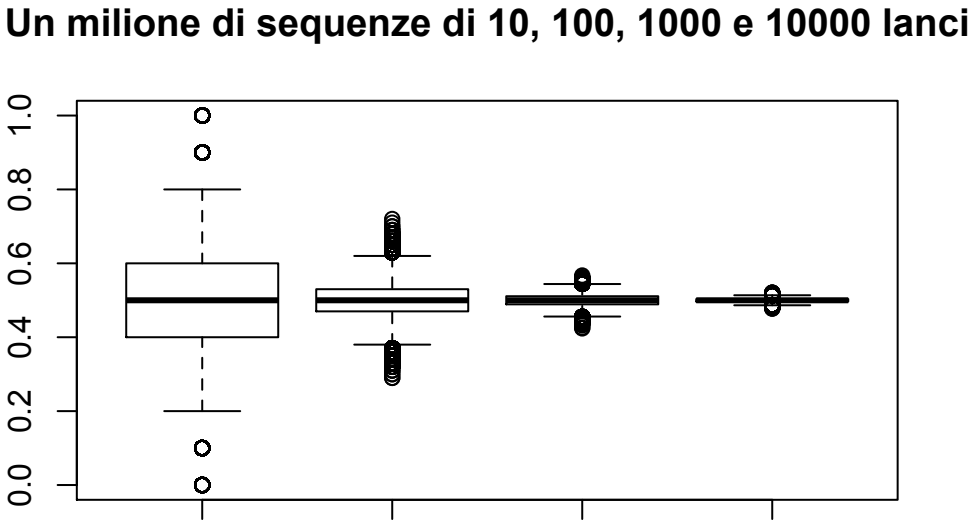

Figura 1.4: Boxplots degli esperimenti `mega10`, `mega100`, `mega1000` e `mega10000` espressi in percentuale.

- Dalla barra del menu, seguite il percorso Pacchetti : Installa pacchetti ...
- Selezionate il CRAN mirror più vicino a voi, ad esempio Italy (Padua) e date OK
- Aspettate qualche istante sino a quando appare la cospicua finestra Packages. Scorretela in basso, sino alla voce Rcmdr, e date OK
- Da questo momento in poi rispondete sempre OK alle eventuali richieste del software; quando si chiudono le finestre di dialogo ridate il comando `library(Rcmdr)` nella console di R. Abbiate pazienza, perché si può trattare di un'operazione che può durare anche una decina di minuti, giacché di default verranno installati molti pacchetti aggiuntivi.

2. Ripetete quanto abbiamo visto sin qui usando l'interfaccia grafica *R Commander*. Dopo averla avviata come abbiamo appena detto, im-

portate dal web il dataset `metabolismo`. Per fare questo, è sufficiente scegliere il percorso:

DATI > IMPORTA DATI > DA FILE DI TESTO O BLOCCO APPUNTI O URL

ed attivare il pulsante INTERNET URL, incollando nella successiva finestra di dialogo l'indirizzo:

`http://www.dmi.units.it/~borelli/dataset/metabolismo.txt`

Esplorate il dataset con il percorso STATISTICHE > INFORMAZIONI RIASSUNTIVE e provate a visualizzare con un boxplot la VARIABILE *pesoT0* con il comando GRAFICI > GRAFICO A SCATOLA E BAFFI ..

3. Scoprite come si possa simulare il lancio di una monetina generando dei numeri casuali con *R Commander*.

Capitolo 2

Cenni di inferenza statistica

2.1 I test statistici più consueti nei disegni Cross-Section

> *.. sono una Dottoranda in Medicina Sperimentale, e il laboratorio in cui mi trovo si occupa dello studio di cellule staminali cardiache adulte, prevalentemente in vitro. Inoltre negli ultimi due anni ho avuto l'opportunità di lavorare con cellule staminali murine con l'obiettivo di indurne il differenziamento cardiaco. Fino ad ora, nella normale attività di laboratorio, mi sono trovata ad analizzare i miei dati semplicemente con il T-Test. So che esistono metodi diversi che possono essere utilizzati, ma non ho nessuna conoscenza al riguardo..* (Vittoria I., socia ANBI)

Riprendiamo brevemente alcune nozioni della statistica inferenziale univariata. Ci ricordiamo che abbiamo a disposizione due modi di operare, che non sempre forniscono risposte equivalenti, e quindi ci potrebbero condurre a prendere delle decisioni erronee se mal applicati. Il primo metodo di inferenza statistica è quello detto **parametrico**, che si basa sulla conoscenza di alcuni parametri (tipicamente la media, la deviazione standard, i gradi di libertà) di alcune celebri variabili aleatorie (la normale o gaussiana, la binomiale, la t di Student, ...). Il secondo approccio è detto **non parametrico** perché, *a priori*, non abbiamo conoscenza delle variabili aleatorie che descrivono il fenomeno che stiamo studiando: questo secondo tipo di approccio è quello che preferiamo applicare quando i dati non sono distribuiti normalmente (mentre i metodi parametrici sono preferiti quando si ipotizza che i dati di cui si dispone sono distribuiti normalmente).

Dunque, la prima cosa da fare è verificare la **normalità dei dati**; possiamo a tale scopo utilizzare preliminarmente il 'grafico quantile quantile', il QQ plot del capitolo precedente. Ecco due esempi in cui percepiamo da subito, in maniera molto intuitiva, la normalità e la non normalità dei dati:

```
www <- "http://www.dmi.units.it/~borelli/
    dataset/studentiannoscorso.txt"
studentimedicina <- read.table(www , header = TRUE)
attach(studentimedicina)
names(studentimedicina)

par(mfrow = c(1,2))
qqnorm(statura)
qqline(statura)
qqnorm(anno)
qqline(anno)
par(mfrow = c(1,1))
```

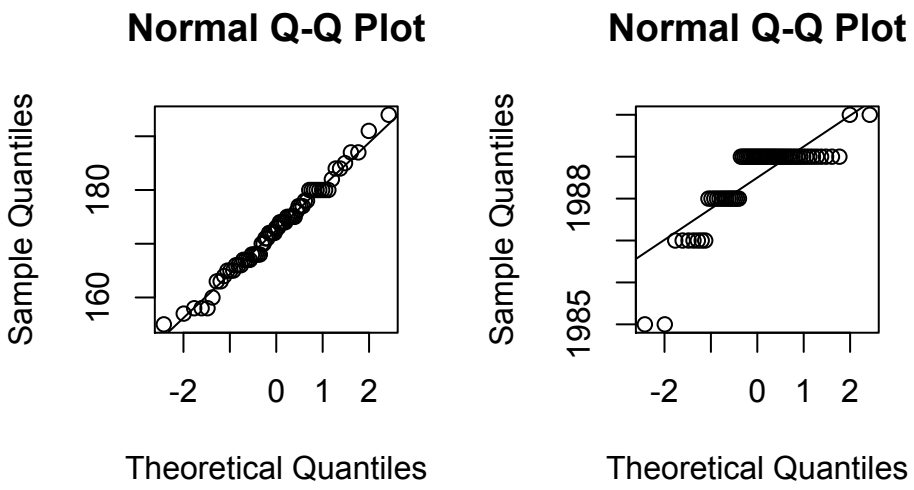

Figura 2.1: QQplot della `statura` e dell'`anno` di nascita del dataset `studentimedicina`.

Nel pannello di sinistra della Figura 2.1 i punti sono praticamente allineati lungo la retta che congiunge il primo ed il terzo quartile della distribuzione, e questo è un ottimo segnale di normalità; nel secondo pannello a destra invece la faccenda è parecchio diversa. Se vogliamo essere più rigorosi, abbiamo a disposizione un test statistico formale, il cosiddetto **test di Shapiro e Wilk**:

```
shapiro.test(statura)
shapiro.test(anno)
```

Siccome l'**ipotesi nulla** che si assume in questo test è che la distribuzione dei dati sia normale, allora osserviamo che nel primo caso il p-value è 'molto grande' (ossia 0.785) e dunque possiamo essere ragionevolmente certi di avere a che fare con dati distribuiti normalmente; in questo caso quindi l'approccio parametrico sarà da preferirsi. Nel secondo caso, siccome il p-value è 'molto piccolo', dell'ordine di dieci elevato alla meno dieci, possiamo essere praticamente certi del fatto che rifiutando l'ipotesi di normalità sarà altamente improbabile 'prendere una cantonata' (i.e. essere erroneamente convinti di avere a che fare con dati non normali), e dovremo quindi preferire l'approccio non parametrico[1].

Come tutti sanno, per ragioni storiche si fissa al 5% la soglia di accettazione o di rifiuto del test, anche se molte linee guida fanno delle ragionevoli considerazioni contro questo approccio acritico (per esempio, [11]).

Ora, continuando, siccome ci siamo accorti grossolanamente che i ragazzi sono più alti delle ragazze:

```
summary(statura[ genere == "m" ])
summary(statura[ genere == "f" ])
tapply(statura, genere, mean)
```

vogliamo provare a verificare con un test formale se vi sia effettiva differenza in senso statistico tra la statura dei maschi e quella delle femmine, oppure se la variazione che abbiamo osservato sia casuale, 'dovuta a mero accidente' come direbbero gli ammiratori della filosofia aristotelica. Stante la normalità dei dati, utilizziamo l'approccio parametrico. Dobbiamo innanzitutto accertarci che i dati non soffrano di **eteroschedasticità**: ossia, che la dispersione (i.e. la varianza) dei dati relativi agli studenti maschi sia paragonabile a quella delle femmine.

[1] Seguendo l'approccio di Casella e Berger [6] e di Parmeggiani e Inoue [22] quando rifiuteremo l'ipotesi nulla vorrà dire che accetteremo l'ipotesi alternativa.

```
var(statura[ genere == "m" ])
var(statura[ genere == "f" ])
```

L'output dei due comandi ci fornisce due numeri non perfettamente uguali, 37.8 contro 35.2. Ci chiediamo dunque se questi due numeri siano lo 'stesso numero' in senso statistico, oppure no. Utilizziamo dunque il **test F della varianza**, di Fisher e Snedecor:

```
var.test(statura[ genere == "m" ], statura[ genere == "f" ])
```

oppure

```
var.test(statura ~ genere)
```

Riportiamo qui di seguito l'output che si ottiene:

```
        F test to compare two variances

data:  statura[genere == "m"] and statura[genere == "f"]
F = 1.0728, num df = 32, denom df = 31, p-value = 0.8464
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5256948 2.1806114
sample estimates:
ratio of variances
          1.072840
```

Il p-value pari a 0.846 è molto elevato e ci fa decidere sul fatto che non vi sia differenza in senso statistico tra le due varianze. Infatti, il rapporto tra le due varianze (`ratio of variances 1.073`) fa differire i due numeri appena di un 7 per cento. Si noti che alla **stima puntuale** 1.073 potremmo preferire la **stima intervallare** (con un grado di fiducia del 95 per cento) la quale afferma che il rapporto delle varianze è un numero compreso nell'intervallo $(0.526; 2.181)$; il fatto che il valore 1 appartenga a tale intervallo è equivalente al fatto che il p-value non sia inferiore al 5%.

Dunque siamo in presenza di dati normali ed omoschedastici: possiamo a buon diritto applicare il celebre **test *t* di Student** (o di Student-Welch, nel caso in cui le dimensioni dei due gruppi siano differenti) per verificare se le medie siano o no diverse in senso statistico.

```
t.test(statura[ genere == "m" ], statura[ genere == "f" ])
```

oppure

```
t.test(statura ~ genere)
```

Come si vede dall'output, siamo in presenza di un p-value piccolissimo, dell'ordine di dieci elevato alla meno dieci. Non vi è ragionevole dubbio dunque che gli studenti maschi di questo dataset siano, mediamente, di statura diversa da quella delle ragazze. Ad esser rigorosi, quello che abbiamo appena eseguito è un **test a due code**, nel senso che ci permette di dire che le medie delle due stature sono diverse tra loro, ma non di dire chi sia più grande di chi. Per sapere però se i maschi siano più alti delle femmine avremmo dovuto effettuare un test ad una coda (e tenere per buono un livello di significatività del 2.5 per cento invece che del consueto 5 per cento):

```
t.test(statura ~ genere, alternative = "less")
```

Qui in realtà essendo il p-value così piccolo non possiamo assolutamente incorrere nel cosiddetto *errore di terzo tipo* ed effettivamente quest'ultimo test è superfluo. Nell'output del test, oltre a vedere riportate le stime puntuali delle **medie campionarie** (178.3 e 166.8), troviamo anche la stima intervallare (erroneamente detta *intervallo di confidenza*; erroneamente giacché in inglese *confidence* significa 'fiducia') della differenza tra le due medie; ricordiamo che questo, malgrado molti lo pensino, non è l'intervallo che contiene in 95 casi su 100 la 'vera differenza' tra le medie. Esso è invece l'insieme dei possibili valori del parametro incognito per i quali la differenza tra il parametro stesso e la sua stima campionaria non è significativamente diversa da zero (con $p < 0.05$). Per approfondire, consigliamo ad esempio [21].

Al contrario, abbiamo già visto che l'`anno` di nascita degli studenti non è distribuito normalmente. Per testare l'omoschedasticità dell'anzianità dei nostri studenti versus il loro `genere`, disponiamo del non così celebre test di Fligner e Killeen, [8] :

```
fligner.test(anno ~ genere)
```

Non c'è differenza in dispersione poiché il p-value vale 0.606, e quindi procediamo con il test dei ranghi di Wilcoxon, noto anche con il nome di test di Mann e Whitney[2]. Come abbiamo capito, possiamo usare indifferentemente la seguente sintassi:

```
wilcox.test(anno[genere == "f"] , anno[genere == "m"])
```

oppure questa:

```
wilcox.test(anno ~ genere)
```

Qui il test ci lascia un po' in dubbio: il p-value 0.095 non è così lontano dalla 'mitica' soglia del 5 per cento. Inoltre, il messaggio d'errore (*impossibile calcolare p-value esatto in presenza di ties*) ci avverte che alcuni (anzi, molti) studenti sono nati nello stesso anno e quindi è impossibile ordinarli in una sequenza senza che ci siano dei 'pari merito', *ties*, come nelle gare sportive in cui due o più atleti si classificano *ex aequo*. Questo potrebbe comportare una non accurata stima del p-value. Esiste però la possibilità di installare il pacchetto `coin`, ed attivarlo con il comando:

```
library(coin)
```

per eseguire un test esatto con il comando:

```
wilcox_test(anno ~ genere , distribution = "exact")
detach(studentimedicina)
rm(studentimedicina)
```

Fissato dunque, come tradizione, un livello soglia di significatività del 5 per cento, dichiariamo che non vi è una differenza significativa tra l'anno di nascita dei ragazzi e quello delle ragazze, stante il p-value = 0.093.

2.1.1 APPROFONDIMENTO: *When variances are different, do not compare the means!*

Molto spesso nei paper che leggiamo, nella sezione Analisi Statistica, gli autori ci dicono che per i dati distribuiti normalmente hanno usato il test t di Student (mentre per quelli non normali il test di Wilcoxon / Mann-Whitney). Ma M. Crawley [9] fa un'osservazione molto importante per evitare di cadere

[2] C'è un po' di confusione a cosa serva il test di Wilcoxon ed a cosa quello di Mann e Whitney. Su `Wikipedia` però le cose sono spiegate molto bene. E poi, siccome essi operativamente sono indistinguibili, noi ci adeguiamo e non li distinguiamo.

in un possibile errore. Si considerino infatti le concentrazioni di ozono misurate giorno dopo giorno in tre diverse serre, rappresentate complessivamente nei tre boxplot di Figura 2.2:

```
gardenA <- c(3, 4, 4, 3, 2, 3, 1, 3, 5, 2)
gardenB <- c(5, 5, 6, 7, 4, 4, 3, 5, 6, 5)
gardenC <- c(3, 3, 2, 1, 10, 4, 3, 11, 3, 10)
mean(gardenA)
mean(gardenB)
mean(gardenC)
boxplot(gardenA, gardenB, gardenC)
```

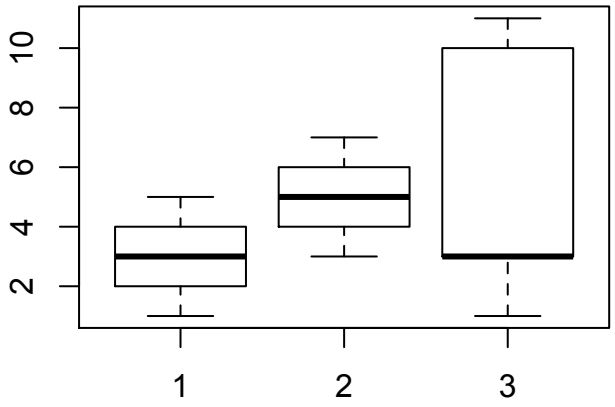

Figura 2.2: Boxplot dell'ozono misurato in tre serre (da Crawley, [9]).

Mediamente, l'esposizione della seconda e della terza serra si attesta attorno a 5, e il t test afferma (p-value = 1) che non vi è differenza tra le loro medie:

```
t.test(gardenB, gardenC)
```

Ma da un punto di vista biologico le cose non vanno affatto bene: la lattuga deperisce per dosi di ozono prossime o superiori a 10, come accade nella serra C - ma non nella B. Ci siamo dimenticati di verificare l'ipotesi

di omoschedasticità (e il boxplot mette chiaramente in evidenza che le .. *variances are different*! Dunque effettuiamo il test della varianza, scoprendo quello che il test t non ci aveva fatto capire:

```
var.test(gardenB, gardenC)
```

Vedremo nel Capitolo 4 come supplire ad un problema che affligge `var.test` in presenza di dati non normali.

2.1.2 APPROFONDIMENTO: Perché si deve preferire l'approccio non parametrico ai dati non distribuiti normalmente?

Consideriamo questa serie di dati artificiali:

```
x1 <- c(2.1827399, 0.6612782, 2.6547994, 10.3405919, 0.7018143,
 3.3023397, 3.1768196, 5.2301088, 4.7231181, 3.3830004, 1.6510160,
 1.0344448, 2.0438494, 7.6171924, 3.4512514, 1.5351440, 0.9625423,
 0.3800261, 5.8930994, 2.3911871, 3.4583686, 11.2182470, 0.4984094,
 18.2966276, 2.6927398, 2.8090413, 1.1376068, 2.8501929, 3.1493031,
 3.6198942, 7.3194375, 0.9101756, 2.4577082, 2.8716387, 4.3151650,
 8.9974668, 1.3442910, 3.0754040, 7.9701409, 1.7676621, 1.2284825,
 1.0322363, 0.3097715, 2.6687727, 1.3383917, 7.7901044, 0.5477570,
 1.1895382, 5.4992737, 19.1438758)

x2 <- c(12.1347169, 2.8135280, 6.0888779, -3.5941390, 3.2262050,
 11.0531101, 4.8366262, 9.0598673, 3.8335178, 12.1224004, 7.9146186,
 1.4004370, 0.5835565, 6.6081925, 10.9355714, 6.7194532, 4.5103941,
 15.6580892, 10.3699346, -8.3829560, 4.5334939, 12.2116074, 1.2127565,
 6.2460818, 2.3934101, 8.4225512, 13.7136161, 5.6485410, 4.2983026,
 7.9318666, 2.4385717, 7.7371111, 6.2890272, 4.6535362, 11.0538054,
 7.7223592, -0.5551599, 0.1950408, 7.9276520, -3.6371784, 9.4464075,
 -2.1047708, -2.7603878, 7.5134609, 7.5472778, -0.9999441, 4.4192166,
 1.4388831, -1.2378510, 3.7481823)
```

Il test di Fisher/Snedecor e il test t di Student ci farebbero supporre che non vi sia differenza statistica tra le due serie di dati:

```
var.test(x1, x2)
t.test(x1, x2)
```

E invece, il vettore `x1` non è distribuito normalmente, e che le due distribuzioni siano diverse viene rivelato dal test dei ranghi (come si poteva anche supporre dagli istogrammi):

```
shapiro.test(x1)
wilcox.test(x1, x2)
par(mfrow = c(1,2)); hist(x1); hist(x2); par(mfrow = c(1,1))
```

2.2 I test statistici più consueti nei disegni longitudinali

Consideriamo il dataset `pdd` relativo ad alcuni bambini affetti da *Pervasive developmental disorder*, e vogliamo vedere se vi sia differenza in senso statistico tra una certa quantità misurata in due tempi diversi, `grsft1h` e `grsft2h`. Si tratta dunque di un design longitudinale, nel senso che su ogni soggetto sono state raccolte misure in diverse occasioni temporali. Innanzitutto, importiamo il dataset:

```
www <- "http://www.dmi.units.it/~borelli/dataset/pdd.txt"
pdd <- read.table(www, header = TRUE)
attach(pdd)
str(pdd)
```

Cerchiamo di capire se i dati sono distribuiti normalmente, non tenendo conto del genere dei partecipanti all'esperimento:

```
par(mfrow = c(1,2))
qqnorm(grsft1h)
qqline(grsft1h)
qqnorm(grsft2h)
qqline(grsft2h)
par(mfrow = c(1,1))
shapiro.test(grsft1h)$p.value
shapiro.test(grsft2h)$p.value
```

No, riteniamo di non aver a che fare con dati normali: lo vediamo dalla Figura 2.3 e dai test di Shapiro e Wilk (rispettivamente, p-value = 0.005 e p-value 0.017). Dunque, per testare l'omoschedasticità non utilizziamo `var.test`, ma preferiamo il test di Fligner e Killeen:

```
fligner.test(grsft1h, grsft2h)
```

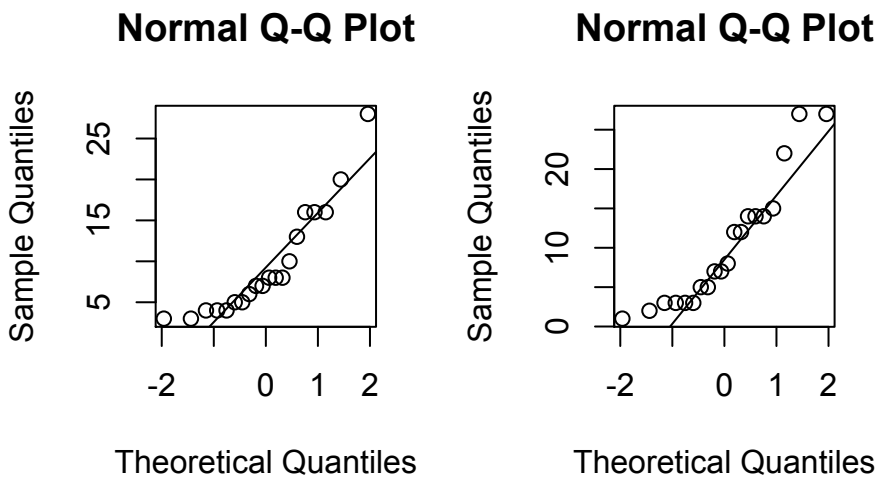

Figura 2.3: I grafici quantile-quantile rispetto alla distribuzione normale delle variabili `grsft1h` e `grsft2h` del dataset `pdd`.

Il p-value ci rassicura abbastanza (0.166) e quindi procediamo a saggiare se vi siano differenze nell'indice di centralità. Usiamo quindi il test di Wilcoxon nella **versione accoppiata**, ossia specificando l'opzione `paired = TRUE`:

```
wilcox.test(grsft1h, grsft2h, paired = TRUE)
boxplot(grsft1h, grsft2h, notch = TRUE)
```

Il p-value 0.666 suggerisce che non ci sono differenze tra il prima e il dopo dell'esperimento, e lo vediamo anche dai boxplot che presentano sia i trattini della mediana che i *notches* ('intagli') praticamente allineati[3]. A titolo informativo, ricordando che non sarebbe stato corretto usare il test `t`, vogliamo mostrare a scopo didattico che se ci 'si dimentica' di mettere l'opzione `paired = TRUE` le cose cambiano:

```
t.test(grsft1h, grsft2h, paired = TRUE)
t.test(grsft1h, grsft2h)
```

Nel primo caso, quello corretto (per modo di dire, perché i dati non sono normali) abbiamo un p-value = 0.598 calcolato su 19 gradi di libertà (`df = 19`): infatti nel dataset ci sono 20 persone, ossia 20 informazioni al

[3] `http://en.wikipedia.org/wiki/Box_plot#Variations`

tempo 1; ma siccome il test t ha bisogno di calcolarsi la media del campione, una informazione 'viene consumata' e ne rimangono 19 libere. Nel secondo caso, quello sbagliato, il numero di gradi di libertà è circa 37 (si tratta di un'approssimazione ad un famoso problema statistico tuttora insoluto[4]): si capisce che c'è qualcosa che non torna perché 20 dati non ci possono fornire 37 informazioni.

Aggiungiamo ancora un'osservazione: i boxplot evidentemente non ci raccontano 'la storia personale' di ciascun soggetto, ma semplicemente fotografano la situazione complessiva del campione all'istante 1 ed all'istante 2. Sarebbe meglio, ad esempio, rappresentare questo esperimento longitudinale sul piano cartesiano come in Figura 2.4:

```
plot(NA, xlim = c(-2, 30), ylim = c(-2, 30), xlab = "grsft1h",
ylab = "grsft2h")
text(grsft1h, grsft2h, soggetto)
abline(0,1, lty = 2)
```

Concludiamo mettendo l'accento su un fatto importante: il test t, anche nella versione accoppiata, fissa solo la sua attenzione sul fatto che le medie non siano diverse, ma non riesce a scopire '*se è <u>veramente</u> successo qualcosa*'. Spieghiamoci con questo esempio semplice: generiamo 30 numeri casuali distribuiti normalmente, e ordiniamo in maniera crescente nel vettore `x` e decrescente nel vettore `y`. La media ovviamente non cambia, e infatti il t test ha `p-value = 1`; ma pensando ad un design longitudinale i valori per ogni soggetto sarebbero molto diversi, come evidenziato nel `plot`:

```
numericasuali <- rnorm(30)
x <- sort(numericasuali)
y <- sort(numericasuali, decreasing = TRUE)
plot(x,y)
abline(0,1, lty = 2)
t.test(x,y, paired = TRUE)
```

Da ultimo, ricordiamoci che i test accoppiati sono applicabili solo se le misure vengono ripetute in due occasioni temporali, e non di più. Per disegni longitudinali più complessi abbiamo a disposizione tecniche molto più sofisticate (i.e. l'uso dei *random effects*, come vedremo alla fine di questa dispensa).

[4] `http://en.wikipedia.org/wiki/Behrens%E2%80%93Fisher_problem`

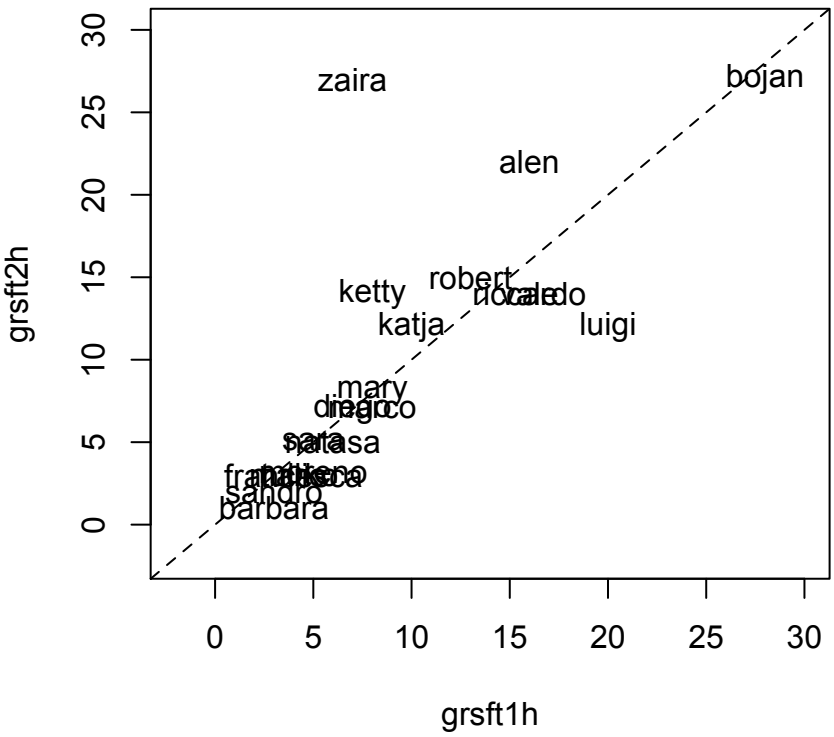

Figura 2.4: Un `plot` bidimensionale che mette in luce la variazione delle variabili `grsft1h` e `grsft2h` del dataset `pdd`.

2.2.1 Laboratorio

1. Importate in R Commander il dataset `metabolismo`:

   `http://www.dmi.units.it/~borelli/dataset/metabolismo.txt`

   e testate (scegliendo il test adeguato) se il `pesoT0` *baseline* sia diverso nei due gruppi sottoposti al diverso `trattamento`. A tale proposito, servitevi del menu STATISTICHE. Rappresentate i dati con un grafico opportuno.

2. Sempre con R Commander e sempre con il dataset `metabolismo`, riuscite a scoprire se il `pesoT3` è statisticamente diverso dal `pesoT0`, nei due gruppi sottoposti a diverso `trattamento`?

Capitolo 3

Stabilire associazioni tra due variabili

3.1 I test statistici di associazione

3.1.1 Variabili fattore e variabili numeriche

Quando si ha a che fare con coppie di variabili fattore (i.e. categoriche), ci si ricorda di un altro celebre test che si insegna sempre nei corsi di base di Statistica e che serve a misurare il 'grado di associazione' che intercorre tra le due variabili. Si tratta del **test di indipendenza del Chi Quadrato** di Karl Pearson. Con i seguenti comandi creiamo una **tavola di contingenza**, una sua rappresentazione in Figura 3.1 ed un consuntivo di tale tabella:

```
www <- "http://www.dmi.units.it/~borelli/
   dataset/studentiannoscorso.txt"
studentimedicina <- read.table( www , header = TRUE )
attach(studentimedicina)
tavola <- table(fumo, genere)
tavola
plot(tavola, main="fumo vs. genere",col=c("pink","lightblue"))
summary(tavola)
```

Possiamo interpretare quel p-value 0.590 come un'indicazione di indipendenza statistica tra i due eventi, 'essere fumatori' ed 'essere maschi'. In altri termini, conoscere il `genere` dei nostri studenti non ci dà informazione per scoprire se essi abbiano o meno il dannoso vizio del `fumo`. O, ancora, il `genere` non è un predittore del `fumo`.

Ricordiamoci sempre però che il test del Chi Quadrato è un **test asintotico**, nel senso che i risultati sono maggiormente attendibili se 'la dimensione

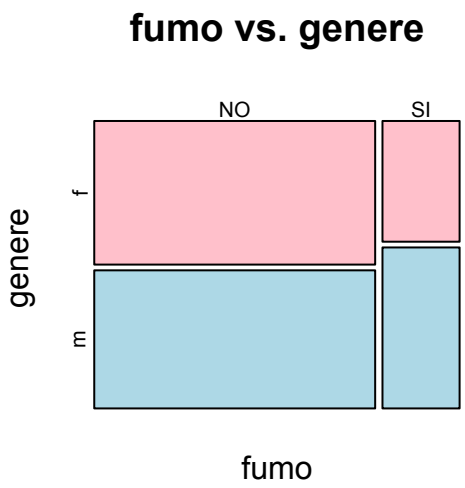

Figura 3.1: Il `plot` del `fumo` e del `genere` del dataset `studentimedicina`. Il fatto che i rettangoli rosa e quelli azzurri siano 'quasi' proporzionali, ossia il fatto che le due 'strade bianche' che li dividono si incrociano 'quasi' perfettamente, ci fa intuire che il fumo e il genere non siano variabili associate.

del campione è grande' (il che per un matematico vuol dire tutto e nulla, ma per uno statistico vuol dire che nelle celle della tavola di contingenza dovrebbero apparire numeri di almeno qualche decina). Se abbiamo a che fare con un campione di dimensione esigua è preferibile ricorrere al test esatto di Fisher:

```
fisher.test(tavola)
```

Qui il p-value ammonta a 0.764 e ritroviamo nell'output anche la nota misura di rischio detta *odds ratio*, la quale non si discosta in maniera significativa da uno (stima puntuale 1.38, stima intervallare al 95 per cento da 0.36 a 5.58). Se volete un rapido refresh sull'odds ratio e su altre misure di associazione e di rischio, vi suggerisco di dare un'occhiata alle *review* numero 3, numero 6 e numero 8 scaricabili da `http://ccforum.com/series/CC_Medical`.

Se invece ci si deve occupare di due variabili numeriche, come la `statura` ed il `peso` delle persone, di solito non ha senso dicotomizzare i dati suddividendoli in quelli sopra e quelli sotto la media, per creare una tavola di contingenza (se non per le ragioni che illustriamo nel prossimo paragrafo).

In questo caso, di solito i ricercatori ricorrono al test di correlazione lineare (anche se oggi questo approccio è superato dalla possibilità di proporre un modello statistico completo):

```
cor.test(statura, peso)
```

e qui non vi è dubbio che `statura` e `peso` siano correlati tra loro (p-value $<$ 0.001). Attenzione, questo test però presuppone che tra la `statura` ed il `peso` sia soggiacente una relazione lineare; ma la questione non è banale, perchè non ci possiamo affidare ad una semplice osservazione del grafico (Figura 3.2) che si ottiene con il comando `plot(statura,peso)` .

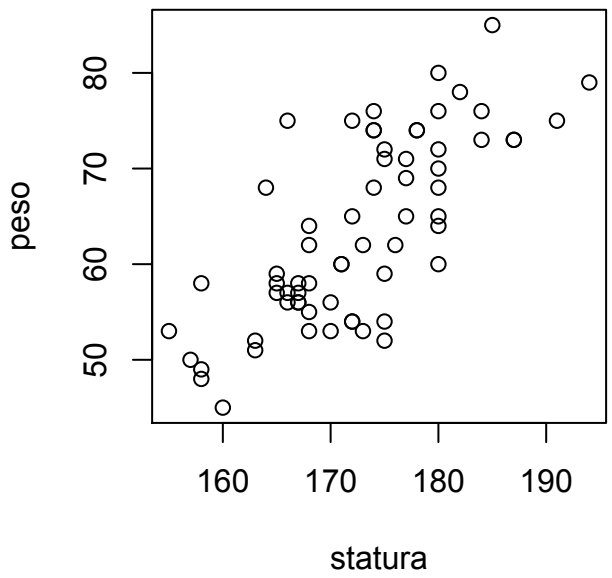

Figura 3.2: Il `plot` della `statura` e del `peso` del dataset `studentimedicina`.

Se non abbiamo indizi sull'ipotesi di linearità, possiamo analizzare la correlazione in maniera non parametrica usando per esempio il metodo di Kendall:

```
cor.test(statura, peso, method = "kendall")
```

e ri-scopriamo che i due caratteri sono correlati tra loro, stante il p-value minore di 0.001.

Talvolta alcuni hanno a che fare con più tabelle di contingenza: nel dataset tratto dagli studi di Daniele possiamo creare due tavole di contingenza, a seconda del `Gruppo`(`Caso` oppure `Controllo`):

> *.. sto cercando di capire se alcuni SNP possano predisporre al cancro al fegato: ogni paziente può avere tre genotipi diversi per ogni SNP, e vorrei capire quindi se un genotipo o un allele in particolare possano essere un fattore di rischio per la popolazione..*
> (Daniele B., studente Ph.D.)

Qui c'è da dire che le analisi statistiche relative ai microarray sono affascinanti per la loro complessità e innovazione; raccomando la lettura di Drăghici [12] a tale proposito. Per fare un'analisi iniziale, importiamo innanzitutto il dataset:

```
www <- "http://www.dmi.units.it/~borelli/dataset/hcc.txt"
hcc <- read.table( www , header = TRUE )
attach(hcc)
table (Genotipo1, Genotipo2, Gruppo)
```

Il test di 'Cochran-Mantel-Haenszel' basato sulla variabile aleatoria Chi Quadrato verifica se vi sia indipendenza tra i due **strati** dello studio:

```
mantelhaen.test(Genotipo1, Genotipo2, Gruppo)
detach(hcc)
rm(hcc)
```

Ricordiamoci però che la presenza di zeri nella tabella, come in questo caso, 'fa a pugni' con la natura asintotica del test. La library `coin` dispone di un test esatto (`independence_test`), che fornisce p-value più attendibili. Rimane però qui la limitazione di non poter inserire delle covariate, come il `Sesso` dei soggetti e la loro `Eta`; vedremo di superare questa difficoltà con l'aiuto dei **modelli lineari generalizzati**.

3.1.2 Scegliere un cut-off appropriato

Spesso si ha a che fare per esempio con il problema di surrogare l'informazione che provenga da un golden standard (per esempio, una biopsia su cellule presunte neoplastiche) mediante un test diagnostico che sia meno invasivo, o meno costoso, o più rapido (per esempio, EIA / ELISA). Tutto ciò coinvolge di fatto la scelta adeguata del **cut-off** a partire dal quale dichiarare negativi o positivi al test i soggetti esaminati. Di solito è un'operazione che si fa 'a

naso', oppure scegliendo dei quantili legati alla dispersione (per esempio, due deviazioni standard), o alla centralità (come nel caso della *MoM*, i multipli delle mediana). Vi vogliamo mostrare qui che possiamo disporre di tecniche visuali (e algebriche) molto efficaci.

Da quando è avvenuto l'attacco di Pearl Harbor[1], la curva ROC è diventato lo strumento più comodo che consenta di individuare, sia visivamente che in maniera numerica, quale possa essere il 'migliore' cut-off per un test diagnostico, in maniera da ottimizzarne l'affidabilità riducendo il numero complessivo dei casi falsi positivi e/o dei falsi negativi.

Riflettiamo, a tale proposito, se la `statura` possa essere un proxy valido per individuare il `genere` di uno studente. Di solito, infatti, i ragazzi sono mediamente più alti delle ragazze.

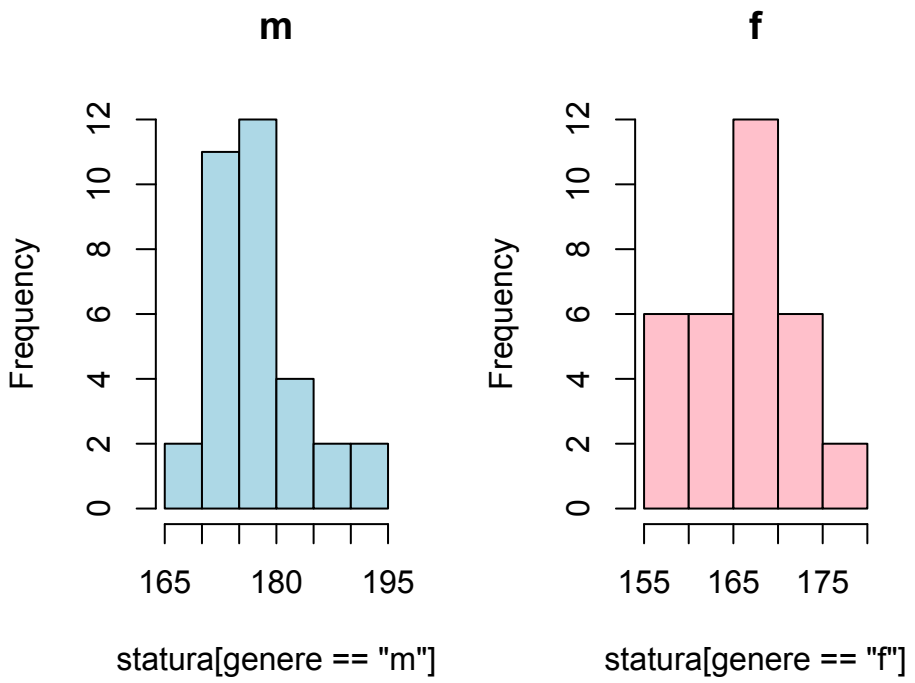

Figura 3.3: Istogrammi della `statura` suddivisi per `genere` nel dataset `studentimedicina`.

```
par( mfrow = c(1,2) )
hist( statura [genere =="m"] , col = "lightblue", main = "m")
```

[1] http://en.wikipedia.org/wiki/Receiver_operating_characteristic#History

```
hist( statura [genere =="f"] , col = "pink" , main = "f")
par( mfrow = c(1,1) )
```

Gli istogrammi in Figura 3.3 mostrano ciò che abbiamo già verificato con il test t di Student nel modulo precedente, ossia che vi è una differenza significativa di altezza tra i maschi e le femmine. La domanda è: quale è il cut-off 'migliore' della `statura` per riuscire a discriminare il `genere`? Essenzialmente, qui, abbiamo a che fare con una variabile numerica che si vorrebbe trasformare in categorica (i.e. 'alto' o 'basso' di statura) in modo da mimare il comportamento della variabile `genere` (i.e. 'm' o 'f').

Importiamo la libreria ROCR e disegniamo la curva ROC:

```
library(ROCR)
predizione <- prediction(statura, genere)
risultato <- performance(predizione, "sens", "spec")
plot(risultato, colorize=TRUE)
```

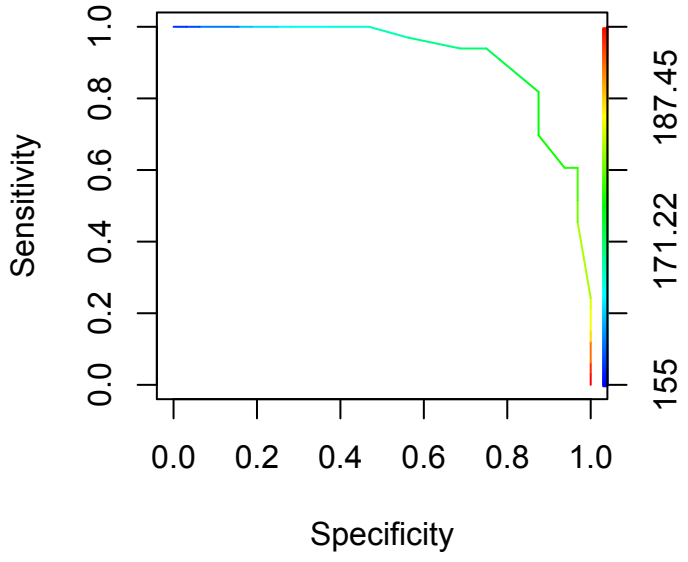

Figura 3.4: La curva ROC della `statura` versus il `genere` nel dataset `studentimedicina`.

La curva convessa in falsi colori visualizzata nella Figura 3.4 e la relativa scala colorata riportata nell'asse di destra ci fa capire che è conveniente

scegliere per l'altezza un cut-off di quelli colorati in verde, in modo da avere un giusto equilibrio tra la sensibilità e la specificità (i.e. minimizzare complessivamente i falsi negativi ed i falsi positivi). Proviamo con tentativo, scegliendo come cut-off 175.

```
xtabs( ~ genere + as.factor(statura < 175))
```

Con questa scelta, avremmo ottenuto $4 + 10 = 14$ errori al test, e precisamente 4 studenti femmine che hanno statura maggiore al metro e settantacinque, e 10 maschi più bassi dell'uno e settantacinque.

Potremmo ora continuare per tentativi variando il cut-off, ma è più conveniente 'inventare' una funzione, che chiamiamo `migliorecutoff` la quale, andando a leggere all'interno delle liste di dati di `predizione` e `risultato`, ci restituisca il punto geometricamente più vicino ('teorema di Pitagora') all'angolo superiore destro del grafico, nel quale la sensibilità e la specificità sono entrambe uguali ad uno (nessun falso negativo e nessun falso positivo).

```
migliorecutoff <- function(perf)
{
Posizionecutoff <- which(
(risultato@x.values[[1]]-
risultato@y.values[[1]]) == min((risultato@x.values[[1]]-
risultato@y.values[[1]])
[risultato@x.values[[1]] - risultato@y.values[[1]]>0]))
return(risultato@alpha.values[[1]][Posizionecutoff])
}
```

Applicando la funzione otteniamo che il cut-off migliore nel nostro caso è 174:

```
migliorecutoff(risultato)
```

Saremmo potuti giungere a un risultato quasi analogo rinunciando ad usare la curva ROC e sfruttando invece il grafico 3.5 della densità condizionata:

```
cdplot(genere ~ statura)
lines(c(165, 200), c(0.5, 0.5), col = "violet")
lines(c(172, 172), c(0, 0.6), col = "orange")
detach(studentimedicina)
rm(studentimedicina)
```

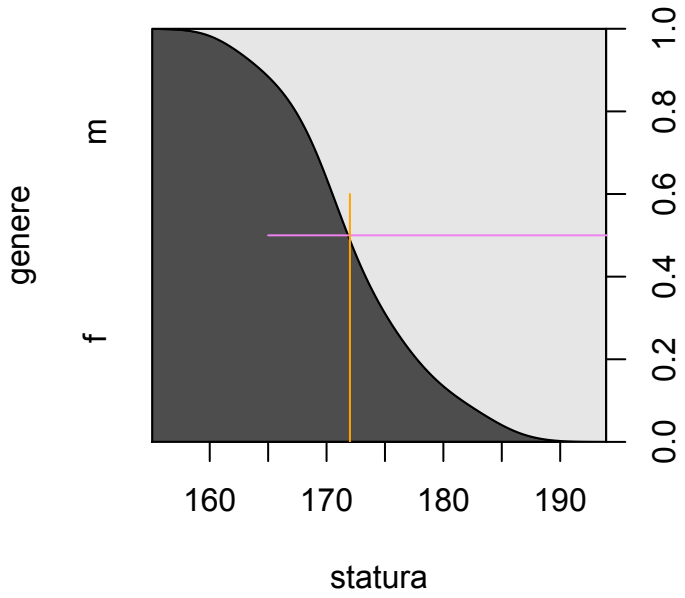

Figura 3.5: Il grafico della densità condizionata della `statura` versus il `genere` nel dataset `studentimedicina`. Le linee viola ed arancione aiutano ad individuare il cut-off relativo alla probabilità 0.5

Vediamo che in questo caso che il 'valore di incertezza' del 50 per cento si colloca attorno alla statura di 172 centimetri. E' istruttivo infine visualizzare con il grafico 3.6 il fatto che sensibilità (colore violetto) e specificità (colore arancione) di un test sono due oggetti antagonisti: al crescere dell'uno, l'altro cala:

```
plot(risultato@alpha.values[[1]], risultato@x.values[[1]],
col = "orange", "l")
lines(risultato@alpha.values[[1]], risultato@y.values[[1]],
col = "violet")
detach(studentimedicina)
rm(studentimedicina)
```

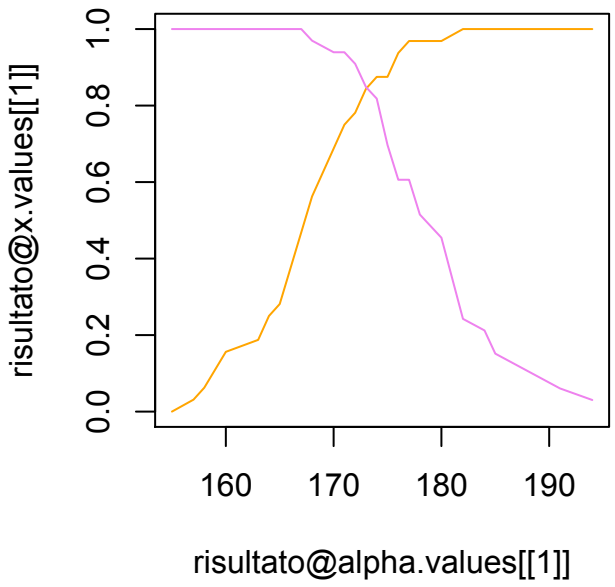

Figura 3.6: Il grafico della sensibilità e della specificità della `statura` pensata come un *proxy* del `genere` nel dataset `studentimedicina`.

3.1.3 APPROFONDIMENTO: Il p-value dipende dalla dimensione del campione

Abbiamo sentito molte volte dire: '*eh sì, effettivamente qui la significatività statistica non c'è, ma basterebbe aumentare la dimensione del campione..*'. In queste parole si nasconde il più comune *pitfall* che, a nostro avviso, insidia chi non ha ben compreso la differenza che c'è tra *effect size* e *p-value*. Il p value non dice se un trattamento è o non è efficace, o quanto lo sia. Il p-value ci dice solo se possiamo assumere che, al di là di un ragionevole dubbio, il trattamento abbia avuto un effetto. Questo ovviamente riguarda tutti i tipi di test, ma ne parliamo adesso perché abbiamo introdotto il test di correlazione e possiamo meglio capire questo esempio.

```
www <- "http://www.dmi.units.it/~borelli/dataset/heart.txt"
heart <- read.table( www , header = TRUE )
attach(heart)
dimensione <- length(ETA)
```

```
head(heart)
```

Il dataset `heart` fornisce informazioni sui cardiogrammi di mille e trecento pazienti: una dimensione campionaria considerevole! Supponiamo di essere interessati a sapere se `ETA` e `FC` (frequenza cardiaca) siano correlate linearmente:

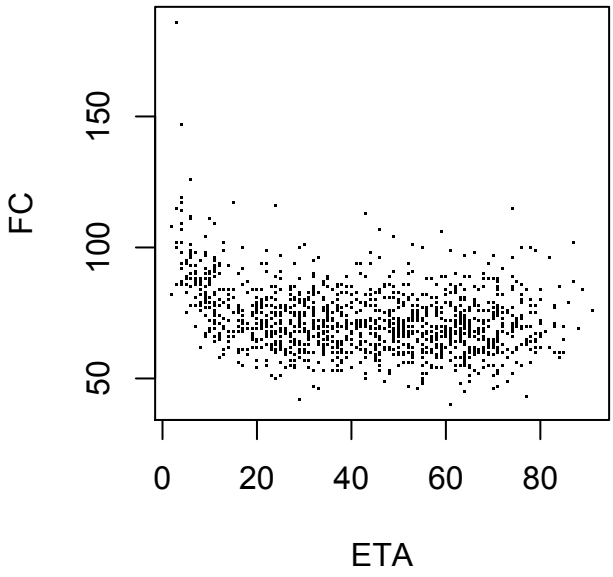

Figura 3.7: Il grafico della `ETA` come predittore della frequenza cardiaca `FC` nel dataset `heart`.

```
plot(ETA, FC, pch = ".")
cor.test(ETA, FC)$p.value
```

In effetti il grafico della Figura 3.7 ci fa percepire un andamento decrescente di `FC` durante la giovinezza, e il p-value bassissimo supporta la nostra idea. Ma supponiamo di avere condotto uno studio più limitato, diciamo di cento pazienti - che comunque sembra essere un *sample size* ragionevole. Provate a copiare ed incollare alcune volte i comandi che seguono:

```
sottoinsieme_casuale_indici <-  sample (1:dimensione, 100 )
```

```
eta <- ETA[sottoinsieme_casuale_indici]
fc <- FC[sottoinsieme_casuale_indici]
plot(ETA, FC, pch = ".")
points (eta, fc, col = "red", pch = 20)
cor.test(eta, fc)
```

Vedrete che il p-value che otterrete per ogni sottocampione varierà in maniera del tutto spropositata, 'traballando' da valori prossimi allo zero a valori praticamente vicini all'uno. Torneremo sulla questione in un prossimo modulo.

```
detach(heart)
rm(heart)
```

3.1.4 APPROFONDIMENTO: Scegliere la dimensione corretta del campione

> *.. da questa Summer School mi aspetto di reimpadonirmi dei principali concetti di statistica per poter scrivere correttamente un lavoro e/o un progetto, partendo banalmente dalla scelta della numerosità campionaria che io e colleghi mettiamo sempre un po' a caso ..* (Roberta M., Ph.D., Assegnista di Ricerca)

Se abbiamo capito che il p-value varia (diminuisce verso lo zero, cioè) aumentando la dimensione dei dati che si hanno a disposizione (i.e. la dimensione del campione), allora si capisce perché è opportuno cercare di fissare a priori (soprattutto nei *clinical trial*) una dimensione sufficientemente grande, in modo che il test statistico abbia una ragionevole **potenza**, i.e. che sappia riconoscere la falsità dell'ipotesi nulla quando questa è effettivamente falsa.

Per capirci, facciamo un esempio. Per l'ordinamento giudiziario italiano l'ipotesi nulla è 'l'imputato è innocente', che si suppone vera fino a prova contraria (ed il pubblico ministero è l'ufficio che deve cercare di esibire questa prova). Mettiamo però di essere stati testimoni del fatto, ed avere visto con i nostri occhi che l'imputato è colpevole, però di non poter dimostrarlo con alcuna prova. Quello che speriamo tanto è che il giudice non commetta l'errore β, ossia quello di non riconoscere la falsità dell'ipotesi nulla; ossia, che abbia 'la potenza' $1-\beta$ di riconoscere la falsità dell'innocenza dell'imputato.

Bene: la scelta della dimensione del campione coinvolge tutte queste quantità. Purtroppo, per calcolarla, possiamo per lo più solo sfruttare le conoscenze nel caso gaussiano della distribuzione normale, come accade nel test t di Student con la funzione `power.t.test`. E non solo! Abbiamo anche

bisogno di avere un'idea, basata per esempio su uno studio pilota, di quali siano i valori medi che incontreremo dei due gruppi che andremo a testare con il test t.

Supponiamo dunque, come nel dataset `kmg`, che vi siano due categorie di donne con una possibile `mutazione` a livello genetico: per le prime, in media, occorrono 0.67 cicli di fecondazione assistita per avere una gravidanza favorevole; per le seconde, 0.9. Dunque, la differenza tra le due medie è `delta = 0.23`:

```
www <- "http://www.dmi.units.it/~borelli/dataset/kmg.txt"
kmg <- read.table( www , header = TRUE )
attach(kmg)
mean(nciclolivebirth[mutazione =="nmut"], na.rm = TRUE)
mean(nciclolivebirth[mutazione =="mut"], na.rm = TRUE)
```

Cautelativamente, cerchiamo la peggiore dispersione (quantificata dalla deviazione standard) tra i due gruppi, e vediamo che essa vale `sd = 1.27`:

```
sd(nciclolivebirth[mutazione =="nmut"], na.rm = TRUE)
sd(nciclolivebirth[mutazione =="mut"], na.rm = TRUE)
```

Quante donne dobbiamo arruolare in un ipotetico *clinical trial* se prefissiamo un livello di significatività (errore α) `sig.level = 0.05` ed una potenza $(1 - \beta)$ `power = 0.80`? La risposta è 480 donne per gruppo:

```
power.t.test(n = NULL, delta = 0.23, sd = 1.27,
sig.level = 0.05, power = 0.80)
detach(kmg)
rm(kmg)
```

3.1.5 Laboratorio

1. Considerate il dataset `vaevt`:

   ```
   www <- "http://www.dmi.units.it/~borelli/dataset/vaevt.txt"
   dativaevt <- read.table( www , header = TRUE)
   attach(dativaevt)
   ```

   ed indagate se sia possibile trovare dei cut-off su `basale`, `ventiquattro` o `quarantotto` in modo tale da ottenere un predittore ragionevolmente affidabile per `outcome` (ricordando che ciò che si vuole evitare è l'evento infausto `outcome` uguale a `0`).

Capitolo 4

Differenze tra gruppi

4.1 L'analisi della varianza negli studi cross-section

L'*Anova* è una tecnica statistica molto celebre che da un lato rappresenta una naturale generalizzazione del test t di Student, dall'altro è una reinterpretazione, in termini di **modello lineare**, del concetto di retta di regressione. Ed infatti, con R, l'Anova si effettua per mezzo del comando `lm`, acronimo di '**l**inear **m**odel'. Sebbene didatticamente sarebbe più proficuo partire dal concetto di retta di regressione per poi introdurre - come conseguenza - l'Anova, l'esperienza ci mostra che per un biotecnologo l'Anova è (quasi) il pane quotidiano (e non solo per i biotecnologi):

> *.. sono biologo evoluzionista e mi occupo principalmente di genomica evoluzionistica, e per il tipo di lavoro che devo eseguire, ovvero l'analisi di trascrittomi, devo individuare gruppi di campioni e/o gruppi di geni che presentino un'espressione significativamente diversa tra loro ..* (Fabrizio G., Ph.D. post doc)

Dunque, partiamo subito con un esempio introduttivo, riprendendo le cose viste nel capitolo precedente.

4.1.1 Il caso introduttivo a due livelli

Riesaminiamo il dataset `studentiannoscorso` e ci chiediamo se vi sia differenza (in senso statistico) di `peso` tra ragazzi e ragazze. Ossia, ci chiediamo se il `genere` possa essere un predittore del `peso` (o, detto ancora in altri termini, se il `genere` abbia un effetto sul `peso`). E' chiaro che per fare questo sarebbe naturale ricorrere al test t di Student, ma in questo paragrafo vogliamo introdurre l'uso del comando `lm` di R:

```
www <- 'http://www.dmi.units.it/~borelli/
   dataset/studentiannoscorso.txt'
studentimedicina <- read.table(www, header = TRUE)
attach(studentimedicina)
names(studentimedicina)
modello <- lm (peso ~ genere)
summary(modello)
```

L'output che otteniamo questa volta `summary` è molto corposo, e a capirlo bene ci vuole un po' di esperienza - ma non preoccupiamoci, la acquisiremo nel prossimo modulo, quello della retta di regressione. L'occhio però cade immediatamente su delle meravigliose stelline: il `genere` *c'entra* con il `peso`. Il dubbio però ci assale se disegniamo la *retta di regressione* che abbiamo appena trovato con il comando `lm`:

```
plot(genere, peso)
abline(modello)
```

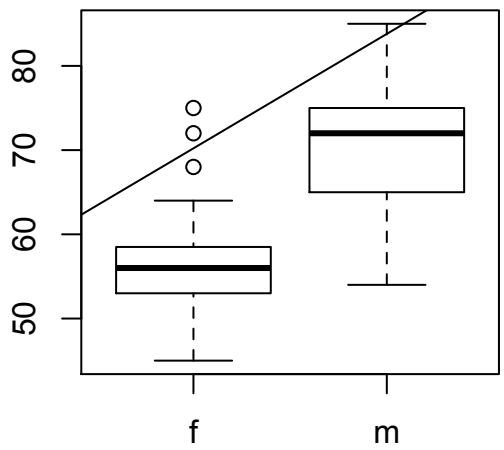

Figura 4.1: Il `plot` del `peso` versus il `genere` del dataset `studentimedicina`: evidentemente qualcosa non va.

Cosa c'è che non va? Il `genere` è una variabile **fattore** a due **livelli**:

```
levels(genere)
```

e perciò bisogna abbandonare questo tipo di rappresentazione grafica fuorviante e cercare invece di interpretare con un po' di fantasia i coefficienti della retta di regressione nell'output del comando `summary` (e chi ha colpo d'occhio si sarà già accorto di un apparente refuso di R, ossia della strana parola `generem` che è apparsa lì nel mezzo). Il trucco in realtà è molto semplice. Pensiamo all'equazione della retta che abbiamo imparato a scuola, $y = mx+q$, e immaginiamo che la variabile x possa assumere solo i valori 0 ed 1, come capita per i bit nella teoria dell'informazione, o come capita ... negli interruttori della luce. Se $x = 0$, allora $y = m \cdot 0 + q = q = 56.625$, mentre se invece x vale 1, allora $y = m \cdot 1 + q = m + q = 13.587 + 56.625 = 70.212$.

Questa è esattamente la convenzione che si utilizza per interpretare l'output del comando `lm` di R nel caso di predittori categorici. Si considerano i livelli del fattore in ordine alfabetico:

```
levels(genere)[1]
levels(genere)[2]
```

e per il primo livello di `genere` (cioè le femmine), corrispondente ad $x = 0$, il modello si riconduce all'intercetta $y = q = 56.625$, mentre per il secondo livello, corrispondente ad $x = 1$, il modello diviene $y = m + q = 70.212$.

Ecco dunque svelato perché nelle righe dei `Coefficients` il nostro R ha scritto '`generem`' e non '`genere`': perché è sottointeso l'utilizzo dell'ordine alfabetico (o meglio, dell'ordine ASCII) per indicare i livelli delle variabili fattore, e l'intercetta rappresenta in maniera sottointesa '`generef`'.

Rimane ora da capire 'che senso abbiano' i numeri 56.6 e 70.2, ma tutto si spiega immediatamente se ci ricordiamo che l'*Anova* rappresenta la naturale generalizzazione del test t di Student:

```
t.test(peso ~ genere)
mean(peso[genere =='f'])
mean(peso[genere =='m'])
```

Dunque, l'*Anova* ci ridice la stessa cosa che ci dice il test di Student-Welch, ossia che nel dataset `studentimedicina` vi è una differenza tra le medie dei pesi delle studentesse femmine e degli studenti maschi, e che tale differenza di quasi 14 chilogrammi (nel gergo degli statistici, l'**effect size**) è altamente significativa, ossia che vi è un effetto altamente significativo del fattore `genere` sulla variabile `peso`:

```
t.test(peso ~ genere)$p.value
summary(modello)$coefficients[8]
```

A parte piccoli errori di arrotondamento, il p-value quasi nullo (`p-value: 2.627e-11`) ci permette di essere fiduciosi nel rifiutare l'ipotesi, supposta vera, che non vi sia differenza di peso tra maschi e femmine.

Attenzione: l'Anova (come il test t) ha delle ipotesi matematiche sottostanti, che devono essere verificate. Il modo più semplice per farlo è effettuare la **diagnostica** del modello, per mezzo di due grafici che ci permetteranno di valutare sia la **normalità dei residui** che la loro **omoschedasticità** (i.e. medesima dispersione) per mezzo di questi comandi:

```
par(mfrow = c(1, 2))
qqnorm(residuals(modello))
qqline(residuals(modello))
plot(jitter(fitted(modello)), jitter(residuals(modello)))
par(mfrow = c(1, 1))
```

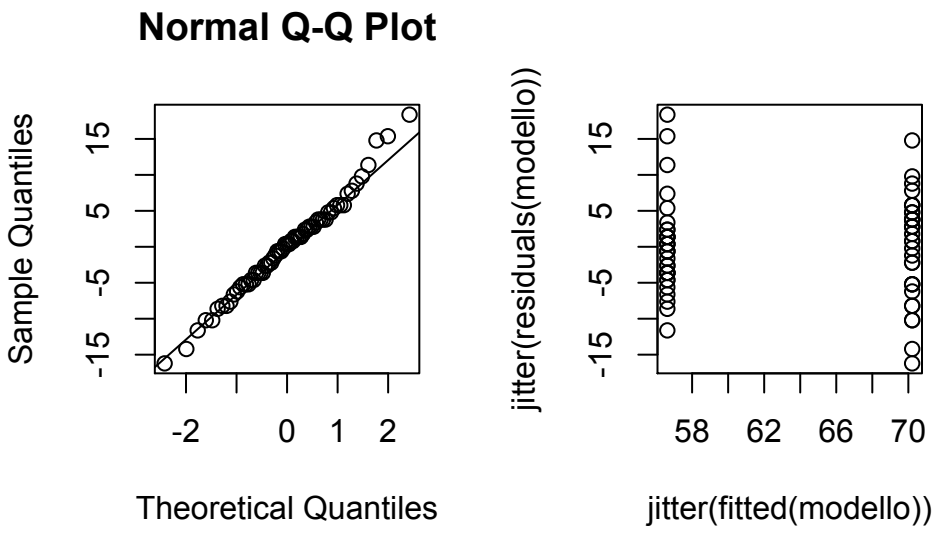

Figura 4.2: I due `plot` della diagnostica del `modello` lineare `peso` $\sim$ `genere`. I residui sembrano adagiarsi quasi perfettamente lungo la diagonale nel QQ plot dei residui, e anche la dispersione dei residui tra maschi e femmine non sembra essere difforme.

4.1.2 APPROFONDIMENTO: analisi delle medie o analisi della varianza?

E' interessante riflettere su una curiosità etimologica: come mai si dice 'analisi della varianza', se invece il test t valuta la differenza tra le medie? E' istruttivo seguire il ragionamento di Crawley, [9]. Iniziamo a rappresentare con un **time-plot** il `peso` degli studenti, indipendentemente dal loro `genere`, aggiungendo la retta del peso medio e le distanze di ciascun punto da tale retta:

```
plot(1:length(peso), peso, xlab='identificatore', ylab='peso')
abline(mean(peso), 0)
for(i in 1:length(peso)) lines(c(i, i), c(mean(peso), peso[i]))
```

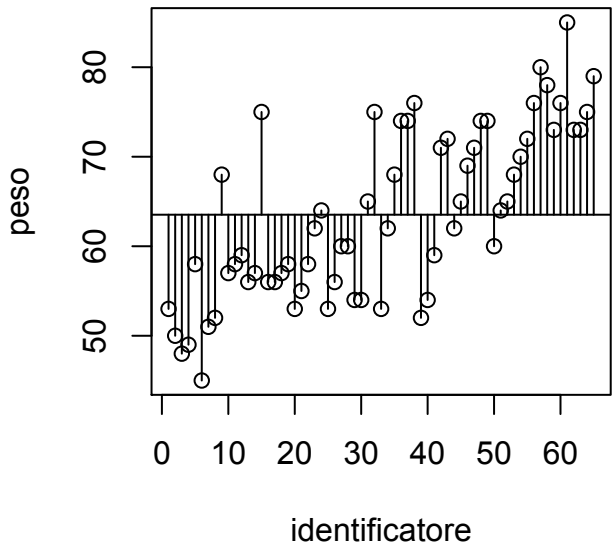

Figura 4.3: Il time-`plot` del `peso` degli studenti.

Nel grafico si osserva molta dispersione attorno alla linea nera orizzontale, che può essere quantificata calcolando la **devianza** del modello lineare (la somma dei quadrati delle lunghezze di tutti i segmentini neri), che storicamente viene indicata con il simbolo SSY, *sum of squares of y*:

```
(SSY <- sum((peso - mean(peso))^2))
```

Ricordiamoci che la devianza SSY = 5896.2 è 'parente prossima' della varianza:

```
var(peso)*64
```

Abbiamo dovuto moltiplicare per 64 (e non per 65, quanti sono i studenti del dataset) ricordandoci delle classiche formule della deviazione standard e della varianza nel caso si abbia a che fare con popolazioni oppure con campioni. Cosa succede ora se introduciamo l'effetto di `genere`? Succede che la devianza complessiva, intesa come somma delle due devianze attribuite ai livelli `m` ed `f` diminuisce:

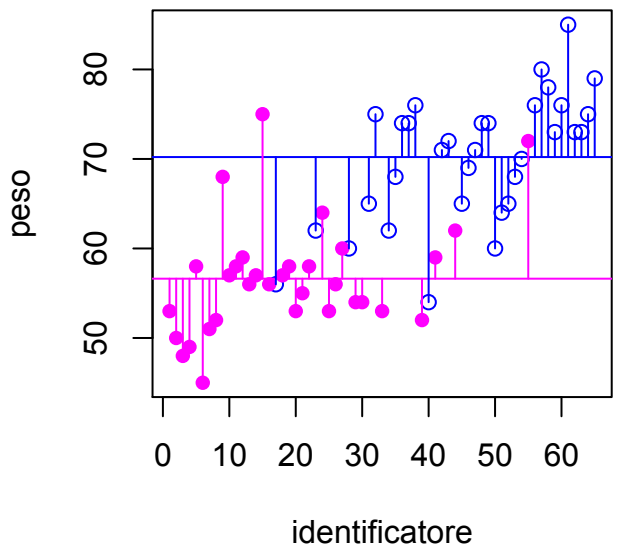

Figura 4.4: Il time-`plot` del `peso` degli studenti distinto per `genere`: in magenta le ragazze, in azzurro i ragazzi.

```
plot(1: length(peso), peso, type='n', xlab = 'identificatore',
ylab ='peso')
points(which(genere=='m'), peso[genere=='m'],
pch=1, col='blue')
points(which(genere=='f'), peso[genere=='f'],
pch=16, col='magenta')
```

```
abline(mean(peso[genere=='m']), 0, col='blue')
abline(mean(peso[genere=='f']), 0, col='magenta')
maschi <- which(genere=='m')
for (i in 1: length(peso[genere=='m']))
lines(c(maschi[i], maschi[i]),
c(mean(peso[genere =='m']), peso[genere =='m'][i]),
col='blue')
femmine <- which(genere=='f')
for (i in 1: length(peso[genere=='f']))
lines(c(femmine[i], femmine[i]),
c(mean(peso[genere =='f']), peso[genere =='f'][i]),
col='magenta')
SSYM <- sum((peso[genere=='m']-mean(peso[genere=='m']))^2)
SSYF <- sum((peso[genere=='f']-mean(peso[genere=='f']))^2)
(SSE <- SSYM + SSYF)
```

Cosa possiamo concludere da tutto ciò? Se il peso medio dei maschi e delle femmine fossero stati uguali, le due linee magenta ed azzurra sarebbero rimaste sovrapposte come nel primo grafico, e i residui avrebbero conservato una dispersione (devianza) molto ampia, SSY = 5896. Al contrario, visto che il peso medio dei maschi e delle femmine esibiscono una differenza sostanziale, la devianza si abbatte considerevolmente. In altri termini, il `genere` riesce a spiegare quasi la metà della variabilità che c'è nel `peso`, lasciando nei residui una devianza SSE pari a 2897:

```
summary(modello)$r.squared
1 - summary(modello)$r.squared
SSE / SSY
```

Siccome il dataset contiene il `peso` di 65 soggetti, e noi conosciamo il peso medio generale delle studentesse (56.6) e quello degli studenti maschi (70.2), ossia disponiamo di due informazioni indipendenti tra loro, vi sono ancora 63 informazioni 'libere' relativamente al `peso`: abbiamo infatti 63 gradi di libertà nei residui:

```
summary(modello)$df[2]
```

Ora, partendo dalla devianza residua SSE e dalla devianza spiegata (SSY - SSE), possiamo considerare le cosiddette 'medie dei quadrati', dividendo le devianze per i loro gradi di libertà:

```
SSE / 63
(SSY - SSE) / 1
```

Questi due numeri appaiono molto diversi tra loro, e questo può essere testato formalmente considerando il loro rapporto:

```
((SSY - SSE) / 1) / (SSE / 63)
```

e confrontandolo con la variabile aleatoria F di Fisher - Snedecor a (1; 63) gradi di libertà:

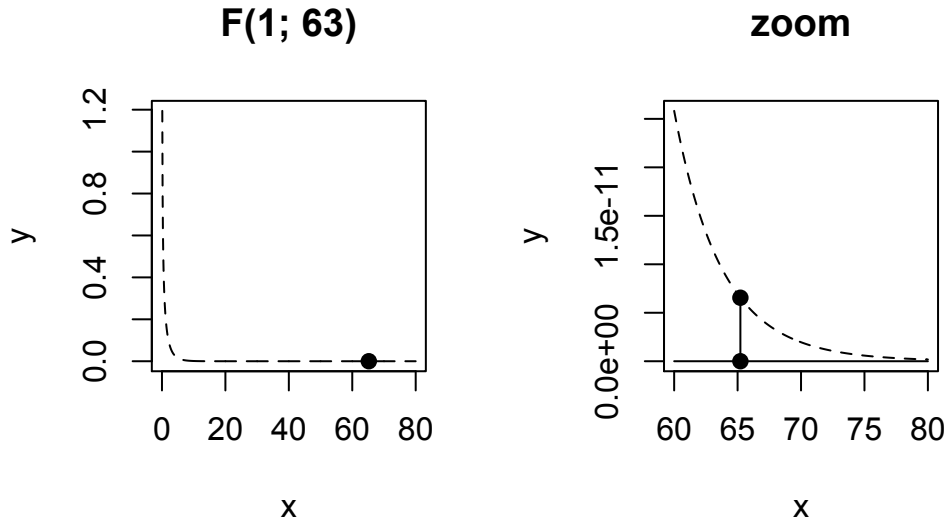

Figura 4.5: La variabile aleatoria F ed il quantile 65.2: si intuisce che l'area del grafico a destra del quantile è estremamente bassa.

```
par(mfrow = c(1, 2))
x <- (0:800)/10
y <- df(x, 1, 63)
plot(x, y, 'n', main ="F(1; 63)")
lines(x, y, lty = 2)
points (65.22, 0, pch = 19)
x <- (600:800)/10
y <- df(x, 1, 63)
plot(x, y, ylim = c(0, max(y)), 'n', main ="zoom")
lines(x, y, lty = 2)
```

```
points (65.22, 0, pch = 19)
points (65.22, df(65.22, 1, 63), pch = 19)
lines (c(65.22, 65.22), c(0, df(65.22, 1, 63)))
lines (c(60, 80), c(0, 0))
par(mfrow = c(1, 1))
```

la probabilità di ottenere un rapporto superiore al quantile 65.2 (cioè, l'area della coda destra della distribuzione) è veramente molto piccola:

```
1-pf(65.22, 1, 63)
summary(modello)$coefficients[8]
t.test(peso ~ genere)$p.value
```

Ecco dunque interpretato il senso del p-value del modello lineare nella accezione dell'Anova. Ciascuna delle quantità che abbiamo sin qui calcolato appare nella cosiddetta 'tavola Anova' associata al modello lineare:

```
summary.aov (modello)
```

La 'tavola Anova' rappresenta ormai un approccio *fuori moda* all'analisi dei dati, legato ai vecchi tempi in cui il computer non esisteva: essa infatti non mette in rilievo l'*effect size*, che è in effetti ciò che importa ad un ricercatore. Che si tratti però di due approcci essenzialmente equivalenti lo si vede utilizzando il comando `aov` :

```
modelloAnova <-  aov(peso ~ genere)
summary(modelloAnova)
summary.lm(modelloAnova)
```

4.1.3 Il caso di un predittore a più livelli

Vediamo ora il caso veramente interessante, ossia cosa possa accadere se in un predittore ci sono tre o più livelli in gioco: in questo caso infatti il test t di Student non si può applicare. Consideriamo l'esempio dell'attività sportiva (`sport`) che ciascuno studente svolge e se questo influenzi il `peso`.

```
levels(sport)
boxplot(peso[sport =='poco'], peso[sport =='saltuario'],
peso[sport =='tanto'])
modellos <- lm (peso ~ sport)
summary(modellos)
```

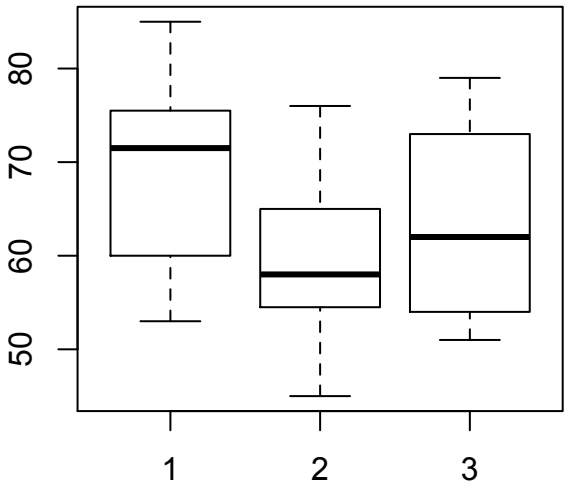

Figura 4.6: I boxplot del `peso` versus il grado di attività `sportiva` praticata.

Sembrerebbe dunque che chi faccia `sport`, in maniera saltuaria o in maniera assidua, mediamente abbia un peso statisticamente diverso da chi non ne fa (rispettivamente di circa 9.7 e di 6.4 chilogrammi in meno dei non sportivi). Ma attenzione a non confondersi con le stelline: esse ci dicono che `sporttanto` è significativamente (p = 0.023) diverso da `sportpoco`, e che `sportsaltuario` è diverso da `sportpoco` in maniera altamente significativa (p = 0.0007). Ma il software non ci dice nulla se `sportsaltuario` e `sporttanto` siano o non siano diversi tra loro in senso statistico.

Scoprire quest'ultima cosa ha a che fare con la procedura dei **multiple comparison** o, detto alla latina, dei confronti **post hoc**. Ci sarebbe molto da dire a tale proposito (Crawley [9], Faraway [14], Bretz et al. [4]), ma ci limitiamo ad indicare un metodo ideato dal geniale chimico e matematico John Wilder Tukey denominato 'Differenza Onestamente Significativa'. Possiamo infatti creare e visualizzare un insieme di intervalli di fiducia che ci mostrano se, con fiducia del 95%, vi sia o meno differenza di effetto tra i vari livelli della variabile considerata.

```
aovpesosport <- aov (peso ~ sport)
```

```
plot(TukeyHSD(aovpesosport))
```

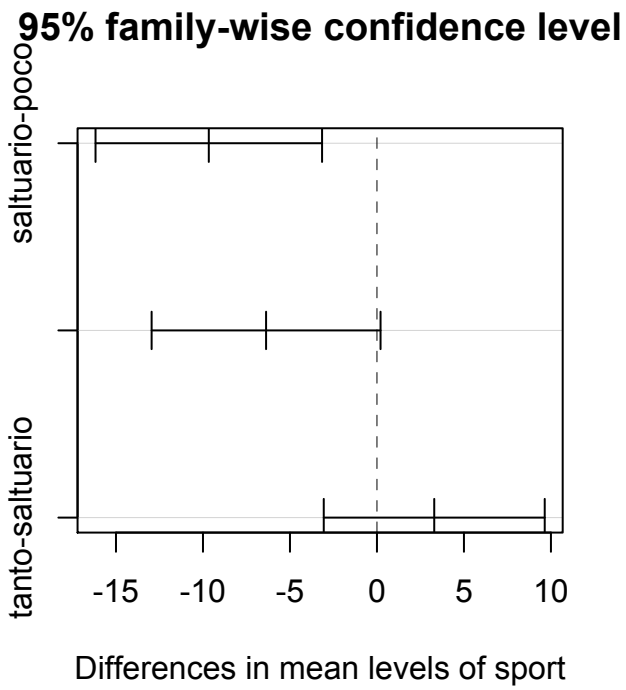

Figura 4.7: Il plot di `TukeyHSD`.

Il fatto che il primo intervallo di fiducia, '`saltuario-poco`', sia discosto dalla linea tratteggiata dello zero, ci indica che tra tali due gruppi di studenti il `peso` è significativamente diverso. Al contrario, il terzo intervallo di fiducia, '`tanto-saltuario`' sormonta la linea tratteggiata; in parole povere, per quanto riguarda questo campione di studenti, fare `sport` (tanto, o saltuariamente) potrebbe essere la stessa cosa ed avere lo stesso effetto sul `peso`. Lo possiamo anche leggere dalla tabella:

```
TukeyHSD(aovpesosport)
```

Proviamo dunque a semplificare il nostro modello raggruppando tra loro i livelli `saltuario` e `tanto`, e ripetere l'analisi. In altre parole, creiamo una nuova variabile più semplice, dicotomica, `sportSN`, che inizialmente è una copia della variabile `sport`, e poi ne modifichiamo i livelli in maniera tale che a `tanto` e a `saltuario` corrisponda il nuovo livello `SI`, mentre a `poco` corrisponda `NO`:

```
sportSN <- sport
levels (sportSN)[1] <- 'NO'
levels (sportSN)[2] <- 'SI'
levels (sportSN)[3] <- 'SI'
par(mfrow = c(1, 2))
pie(table(sport), main ='sport')
pie(table(sportSN), main ='sportSN')
par(mfrow = c(1, 2))
```

Ripetiamo dunque l'analisi:

```
modellosridotto <- lm (peso ~ sportSN)
summary(modellosridotto)
```

Molto bene, stelline in quantità! Accertiamoci con il comando `anova`[1] di non aver perso troppa informazione passando dal `modellos` precedente al `modellosridotto`:

```
anova(modellos, modellosridotto)
```

Con un p-value di 0.21§8 deduciamo che non vi è differenza statistica significativa tra i due modelli, e dunque possiamo preferire la versione semplificata del fattore `sportSN`. C'è da dire che esiste anche una regola pratica per vedere a colpo d'occhio se vi siano differenze significative tra i livelli di un fattore: è sufficiente confrontare le stime dei coefficienti del modello con i loro standard error (1.98, 2.71, 2.74):

```
summary(modellos)$coef
```

Basta verificare che le stime degli effect size:

```
(beta1 <- summary(modellos)$coef[2])
(beta2 <- summary(modellos)$coef[3])
```

siano distanti tra loro per almeno il doppio dello strandard error:

```
(se <-  summary(modellos)$coef[5])
abs(beta1 - beta2) > 2 * se
```

Siccome questo non accade abbiamo una conferma empirica del fatto che nel nostro dataset non vi è differenza tra `sportsaltuario` e `sporttanto`.

[1] Sì, mannaggia, si chiama `anova`, ma non ha quasi niente a che fare con l'Anova.. Che poca fantasia che c'hanno 'sti statistici, eh? ..

4.1.4 APPROFONDIMENTO: Post-hoc in presenza di un *gold standard*

Il test della differenza significativa onesta di Tukey è ottimo se vogliamo comparare differenze tra diversi livelli di un fattore e non abbiamo ragione di preferire uno dei livelli rispetto agli altri. Ma se ad esempio abbiamo informazioni diagnostiche sullo stato di salute in ambito oncologico di n pazienti rispettivamente mediante immagini mammografiche, ecografiche e di prelievo istologico, evidentemente il *gold standard*, ossia l'istologia, è da assumersi come riferimento certo su cui 'tarare' gli altri responsi. Il test di Dunnett in tal caso è da preferirsi. Vediamolo con un esempio classico (Bretz et al. [4]), servendoci della library `multcomp`.

```
library(multcomp)
summary(recovery)
plot(recovery)
```

In questo dataset `minutes` rappresenta il tempo postoperatorio impiegato dal paziente per ristabilire una condizione termica corporea ottimale per mezzo di speciali coperte riscaldate `b1`, `b2` e `b3` versus le coperte standard dell'ospedale, `b0`. Dai boxplot traiamo l'idea che effettivamente le coperte `b2` siano 'meglio' di quelle di riferimento, `b0`. Il problema consueto dei test post-hoc è quello di voler controllare il tasso di errore, (il **familywise error rate**), ossia vogliamo '*correggere per la molteplicità*' (e molto spesso viene applicata la correzione di Bonferroni, che si sa essere in generale troppo conservativa).

```
recovery.aov <- aov(minutes ~ blanket, data = recovery)
summary.lm(recovery.aov)
```

Con un'Anova classica, i p-value che otteniamo sono 'troppo ottimistici'. La library `multcomp` implementa la funzione `glht` (*general linear hypotheses test*) che permette di specificare quale tipo di contrasto (`mcp`, *matrix contrast parameters*) utilizzare nella funzione lineare di interesse (`linfct`, *linear interest function*). Valutiamo dunque il modello di Dunnett:

```
recovery.dunnett <- glht(recovery.aov,
linfct = mcp(blanket = "Dunnett"),
alternative = "less")
summary(recovery.dunnett)
```

Se si raffronta l'output di `summary(recovery.dunnett)` con quello di `summary.lm(recovery.aov)` si nota la differenza, e se in precedenza avevamo un'indicazione che anche le coperte di tipo `b3` potessero avere un certo interesse, con il test di Dunnett questa tesi va scemando. A titolo di curiosità, si osservi che la correzione di Bonferroni non tenendo conto della correlazione esistente tra le tre ipotesi nulle sarebbe risultato meno potente del metodo di Dunnett:

```
summary(recovery.dunnett, test = adjusted(type = "bonferroni"))
```

4.2 Anova a due vie

In questo paragrafo trattiamo due esempi di **two-way Anova**, cioè di analisi della varianza con due predittori: vogliamo capire se due variabili categoriche, ad esempio `genere` e `fumo`, oppure `genere` e `sport`, siano dei validi predittori statistici del `peso`. Iniziamo con il primo esempio.

```
modellogf <- lm (peso ~ genere + fumo)
summary(modellogf)
```

Il fatto che `fumoSI` non abbia le stelline ci fa capire che la covariata `fumo` non è significativa e che non è opportuno considerarla nel modello: verrebbe da dire che 'fumare non fa dimagrire!', se non ci ricordassimo che nei modelli lineari non è corretto pensare che i predittori siano sempre in rapporto di causa-effetto con la risposta.

Riprendiamo invece il modello 'ridotto' dell'attività sportiva e vediamo cosa accade:

```
sportSN <- sport
levels (sportSN)[1] <- 'NO'
levels (sportSN)[2] <- 'SI'
levels (sportSN)[3] <- 'SI'
modellogs <- lm (peso ~ genere + sportSN)
summary(modellogs)
```

Questo modello additivo ci dice che, mediamente, una ragazza non sportiva pesa 61.8 chilogrammi, che un ragazzo non sportivo pesa mediamente 61.8 + 13.2 = 75.0 Kg, e che indipendentemente dal loro genere l'attività sportiva diminuirà il loro peso di 7.2 Kg, portandolo rispettivamente a 61.8 - 7.2 = 54.6 Kg per le femmine e 61.8 + 13.2 - 7.2 = 67.8 Kg per i maschi. A questo punto però è naturale chiedersi: chi ci garantisce che l'effetto

dell'attività sportiva non influenzi in maniera diversa il `peso` dei maschi e delle femmine? In altri termini, chi ci dice che non vi sia un'**interazione** tra l'effetto di `genere` e l'effetto di `sportSN` sul `peso`?

Per esplorare questo modello di '**analisi della varianza a due vie con interazione**' abbiamo a disposizione due sintassi, secondo la notazione di Wilkinson e Rogers, tra loro equivalenti:

```
lm (peso ~ genere + sportSN + genere : sportSN)
lm (peso ~ genere * sportSN)
```

Ecco il `summary`:

```
modellogsinter <- lm (peso ~ genere * sportSN)
summary(modellogsinter)
```

A quanto risulta, il termine di interazione tra il `genere` e l'attività `sportiva` non è significativo, ossia la differenza di `peso` dovuta all'attività sportiva non è diversa tra i due sessi. Dunque, forti dell'insegnamento di William of Ockham[2] (*entia non sunt multiplicanda praeter necessitatem*), eliminiamo il termine di interazione del tutto ridondante:

```
anova(modellogsinter, modellogs)
```

Avremmo potuto già prevedere una risposta negativa in tal senso, guardando il **plot di interazione**:

```
interaction.plot(genere, sportSN, peso)
```

Se la diminuzione di `peso` legata all'attività sportiva fosse stata associata al `genere`, le due rette avrebbero avuto delle pendenze diverse l'una dall'altra, mentre in questo caso esse ci appaiono praticamente parallele. Inoltre, con il comando `plot.design` possiamo anche vedere che l'effetto di `sportSN` è meno rilevante di quello attribuibile al `genere`:

```
plot.design(peso ~ genere + sportSN)
```

4.2.1 APPROFONDIMENTO: l'Anova e la matrice del modello lineare

Possiamo anche utilizzare il comando `model.matrix` per conoscere, per ogni record del dataset, quali siano i termini coinvolti nel modello. Per esempio, il comando:

[2] `http://it.wikipedia.org/wiki/Rasoio_di_Occam`

```
model.matrix(modellogsinter) [2, ]
```

ci fa capire che il secondo (2) record si riferisce ad una studentessa femmina (`generem 0`), che pratica lo sport (`sportSNSI 1`), e che sappiamo avere un peso medio pari a

```
1 * 61.222 + 0 * 14.323 + 1 * (-6.396) + 0 * (-1.604)
```

Per coloro che conoscono il senso della parola **prodotto scalare**, tutto apparirà più chiaro digitando questi due comandi:

```
model.matrix(modellogsinter)[2, ]  %*% modellogsinter$coef
sum(model.matrix(modellogsinter)[2, ] * modellogsinter$coef)
```

4.3 Le ipotesi matematiche dell'Anova

Non bisogna mai dimenticare (ma talvolta, anzi, spesso, ahimé, succede ..) che la analisi della varianza richiede che siano soddisfatte alcune ipotesi matematiche (Rohatgi, [24]):

1. i dati devono provenire da variabili aleatorie normali;
2. la varianza delle distribuzioni deve essere la stessa per tutti i livelli;
3. (e molto molto importante negli studi longitudinali) i dati devono essere indipendenti.

Cosa si fa se i dati sono non normali? Dunque, ci sono varie soluzioni praticabili. Abbiamo innanzitutto a disposizione l'approccio non parametrico basato sui ranghi, con il test di Kruskal e Wallis (Conover, [7]):

```
kruskal.test(peso ~ sport)
```

Naturalmente il `p-value = 0.005` dice esclusivamente che almeno uno dei livelli differisce da qualcuno degli altri livelli, e null'altro. Per quanto riguarda l'omoschedasticità, abbiamo a disposizione sia il test di Bartlett:

```
bartlett.test(peso ~ sportSN)
bartlett.test(peso ~ genere)
```

e ancor meglio il test di Levene, robusto alla violazione della normalità, che è contentuto nella libreria `car`:

```
library(car)
leveneTest(peso ~ sportSN)
leveneTest(peso ~ genere)
```

Dunque, non ci sono problemi di eteroschedasticità. Invece, potremmo dire di aver fatto i conti senza l'oste: infatti l'ipotesi basilare della normalità del `peso` viene a cadere (`p-value = 0.022`):

```
shapiro.test(peso)
```

e pertanto, abbiamo (volutamente, a scopo didattico) commesso un errore molto frequente: ci siamo dimenticati di verificare le ipotesi matematiche prima di iniziare l'analisi statistica. Possiamo forse rimediare in qualche modo?

Possiamo scegliere qui due strade. La prima è: cerchiamo di cambiare la variabile di risposta, per esempio considerando non il `peso` degli studenti ma il loro body mass index `bmi`:

```
bmi <- 10000 * peso / (statura ^2)
```

e, dopo aver controllato le ipotesi necessarie, verificare se `genere` e `sport` siano predittori significativi. Vi lasciamo per esercizio questo passaggio, che vi darà grande soddisfazione.

La seconda strada è quella di provare a modificare la 'forma' della distribuzione dei dati, applicando una trasformazione monotona (radice quadrata, logaritmo, esponenziale, ..) La cosa più proficua da fare è utilizzare la **trasformazione di Box e Cox**, implementata nella library MASS (Venables, Ripley, 2003). Si tratta in sintesi di andare a cercare per quale esponente è necessario elevare il `peso` per ottenere il 'migliore' modello statistico, in termini del principio di massima verosimiglianza.

```
library(MASS)
boxcox (peso ~ genere + sportSN)
```

Il grafico che otteniamo ci fa intuire che un esponente compreso tra 0 ed 1/2 potrebbe fare al caso nostro. Vediamo la situazione più da vicino:

```
boxcox (peso ~ genere + sportSN, lambda = seq(0, 0.5, 0.01))
```

Quindi, si potrebbe provare a considerare come esponente il valore di 0.25 (i.e., la radice quarta)

```
peso2 <- peso ^ 0.25
```

Vi lasciamo per esercizio la verifica di ciò che accade in questo caso. Infine, una strategia che in certi casi può rivelarsi vincente, consiste nello splittare la variabile risposta non normale (nel nostro caso `peso`) in una somma di variabili aleatorie normali: questa tecnica è detta **mixture analysis**. Facciamo finta di non sapere ciò che invece sappiamo benissimo: che il peso delle ragazze sia in genere minore di quello dei ragazzi, e che dunque donne e uomini siano due distinte popolazioni. Carichiamo la library `mclust` e vediamo che il 'miglior' modello è quello a due componenti:

```
library(mclust)
pesosplit <- Mclust(peso)
pesosplit
pesosplit$parameters
```

Il pacchetto assume che si tratti di due distribuzioni normali combinate tra loro, la prima di media 56.3 e l'altra di media 73.0, di varianza comune 22.0, e pesate rispettivamente per il 57% e per il 43%. Lo si può bene visualizzare con questo grafico, in raffronto all'istogramma del `peso`:

```
x<- (450:850)/10
y <- 0.5655504 * dnorm(x, 56.25831, sqrt(22.00815)) +
0.4344496 * dnorm(x, 72.98008, sqrt(22.00815))
par(mfrow = c(1, 2))
plot(x, y)
hist(peso)
```

Si noti peraltro che `Mclust` ha trovato esattamente quello che noi sapevamo già in partenza:

```
mean(peso[genere == "f"])
mean(peso[genere == "m"])
```

4.4 In conclusione: perché parliamo di modello Anova?

Perché, appunto, abbiamo più e più volte usato la parola **modello** invece di dire semplicemente che la Anova è un test statistico? L'idea che soggiace ad esempio al comando `lm (peso` $\sim$ `sport)` è che il fattore `sport`, in generale, abbia un effetto sul peso medio μ_0 del campione; osiia questo possa venir modificato, in più o in meno, dai vari livelli del fattore `sport` :$\mu_0 + \beta_i, i = 1, 2, 3$. Naturalmente si tratta di 'discorsi medi', ed è dunque prevedibile che

il modello abbia in sé anche degli errori, o meglio dei residui, legati a ciascun soggetto preso in esame: $\mu_0 + \beta_i + \epsilon_j$.

In realtà, l'output standard di R è leggermente diverso. Nel caso `lm (peso ~ sport)` il `summary` ci rivela che il peso medio di chi fa `sport` a livello `poco` è $\mu = 69.1$ e che chi fa `sport` a livello `saltuario` o `tanto` ha effetti ulteriori rispettivamente $\beta_2 = -9.67$ e $\beta_2 = -6.37$

4.5 Laboratorio

1. Avendo a disposizione l'esperienza acquisita nel dataset `analgesia`, se voi foste un medico odontoiatra con quale farmaco ritenete sarebbe meglio effettuare la terapia del dolore per i vostri pazienti? E fareste prescrizioni differenti tra uomini e donne?

```
www <- 'http://www.dmi.units.it/~borelli/
    dataset/analgesia.txt'
analgesia <- read.table(www, header = TRUE)
attach(analgesia)
```

Capitolo 5

Calibrare in R un insieme di dati con una retta di regressione

5.1 La retta di regressione negli esperimenti longitudinali

Iniziamo questo modulo con il dire da ben principio cosa **non si deve fare**. La Figura 5.1 è tratta dal lavoro di tesi della nostra amica, consigliere del direttivo ANBI, Elisa D'Este[1].

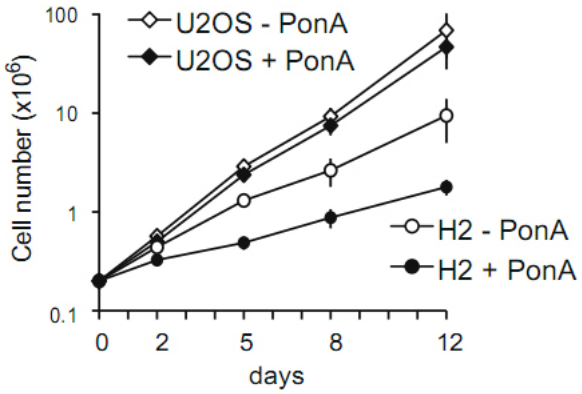

Figura 5.1: Differenti curve di crescita dell'espressione del gene C16orf35 .

Come vedete i dati appaiono allinearsi lungo quattro rette di pendenza via via maggiore. E' giusto modellare questi dati con delle rette di regressione?

[1] `http://www.ncbi.nlm.nih.gov/pubmed/19666006`

Beh, non proprio: infatti, i dati che le leggiamo al dodicesimo giorno **non sono indipendenti** da quelli dell'ottavo e del quinto e del secondo giorno. Nei disegni longitudinali le cose sono più difficoltose, e ne riparleremo a fine corso.

5.2 La retta di regressione negli esperimenti cross-section

Partiamo allora non da una calibrazione di dati, ma da un problema statistico più generale. Nel modulo precedente abbiamo visto che l'Anova si effettua con il potente comando `lm` di R. Potente, perché in effetti esso è il comando del modello lineare, che è una generalizzazione del concetto di retta di regressione. Il modello lineare è in effetti un argomento che varrebbe la pena di studiare in maniera seria, perché è un po' la base di moltissimi *framework* statistici; i libri che mi sento di consigliare sono [26] (per una più approfondita introduzione matematica al modello lineare) e [9, 14, 15] per un approccio approfondito mediante R.

In termini 'liceali', abbiamo capito che si tratta di trovare un opportuno coefficiente angolare m ed un'opportuna intercetta (o quota) q in modo tale che la retta $y = m \cdot x + q$ attraversi la nuvola di punti disegnata sul piano cartesiano xy nel 'miglior modo possibile'. Tale miglior modo possibile si ottiene usualmente utilizzando un procedimento detto *ai minimi quadrati*, nel senso che si cerca di minimizzare, complessivamente, le distanze dei *residui*, ossia gli 'scostamenti' da ciascun punto della nube di coordinate (x_i, y_i) rispetto alla sua proiezione verticale sulla retta di regressione $y = m \cdot x + q$. In altri termini, per ogni x_i si definisce come residuo ϵ_i la distanza che intercorre tra la ascissa del punto, y_i, e la sua proiezione sulla retta, $m \cdot x_i + q$. Abbiamo visto questo concetto nel modulo precedente quando abbiamo introdotto l'effetto di `genere` sul `peso`, con la retta azzurra e la retta rosa.

Con R, oltre a trovare immediatamente tali m e q, abbiamo anche una quantità di strumenti che ci consente di valutare l'affidabilità e la 'qualità' del modello che noi abbiamo formulato. Iniziamo a vedere come vanno le cose se vogliamo ipotizzare che la `statura` dei nostri studenti sia un predittore del loro `peso`.

```
www <- "http://www.dmi.units.it/~borelli/
   dataset/studentiannoscorso.txt"
studentimedicina <- read.table( www , header = TRUE )
attach(studentimedicina)
names(studentimedicina)
```

```
modello <- lm (peso ~ statura)
plot (statura, peso)
abline(modello, lty = 3)
```

Con il termine `modello` abbiamo 'battezzato' la retta di regressione, e l'abbiamo tracciata con il comando `abline`, in maniera tratteggiata (`lty = 3`). Vediamo i coefficienti della retta di regressione:

```
modello
```

Ecco qui di seguito riportato l'output del software:

```
Call:
lm(formula = peso ~ statura)

Coefficients:
(Intercept)      statura
    -83.891        0.854
```

Cosa significa? Abbiamo determinato una stima puntuale della intercetta q e del coefficiente angolare m. In parole povere, il `peso` del nostro studente i-esimo è legato alla sua `statura` dalla relazione:

$$peso_i = 0.854 \cdot statura_i - 83.891 + \epsilon_i$$

ed ϵ_i è il **residuo** che viene attribuito a ciascun soggetto in maniera da poter giustificare quell'errore in più o in meno che il modello lineare non riesce a 'spiegare perfettamente'. La teoria matematica ci richiede delle particolari ipotesi su tali residui, che costituiscono la **componente stocastica** del modello lineare: essi devono essere dei numeri casuali distribuiti normalmente, indipendenti tra loro, con media nulla (il che significa che non ci sono errori sistematici nelle misure che stiamo analizzando), e con una dispersione (deviazione standard) σ costante.

Dicevamo che R fornisce la possibilità di giudicare criticamente la adeguatezza della regressione, e questo si fa innanzitutto con il commando `summary`:

```
summary(modello)
```

Riportiamo qui di seguito l'output, che contiene moltissime informazioni, e che sarebbe opportuno riuscire ad essere in grado di interpretarle tutte, o quasi tutte, nella maniera corretta.

```
Call:
lm(formula = peso ~ statura)

Residuals:
    Min      1Q  Median      3Q     Max
-13.546  -4.209  -1.569   4.431  17.139

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -83.89056   16.67708   -5.03 4.34e-06 ***
statura       0.85392    0.09649    8.85 1.18e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.459 on 63 degrees of freedom
Multiple R-squared: 0.5542,     Adjusted R-squared: 0.5471
F-statistic: 78.31 on 1 and 63 DF,  p-value: 1.18e-12
```

Per iniziare, osserviamo che la nostra nuvola di punti ha un **coefficiente di determinazione** R^2 (`Multiple R-squared`) pari a 0.5542, che non è vicino ad 1 come tutti vorremmo, ma che non è neanche vicino allo 0, come nessuno vuole; ricordiamo che R^2 è il coefficiente di correlazione lineare di Pearson ρ elevato al quadrato:

```
summary(modello)$r.squared
cor(statura, peso)^2
```

Ci viene anche fornita una stima puntuale del valore di σ, la deviazione standard dei residui (`Residual standard error`, 6.459) modellati sulla variabile aleatoria $N(0, \sigma)$:

```
summary(modello)$sigma
```

Ciò che però colpisce sempre l'occhio del ricercatore sono le stelline, `***`. Nella nostra analisi, le stelline ci suggeriscono che la `statura` è un predittore

altamente significativo del `peso`; nel senso geometrico che la retta di regressione ha una pendenza m e un'intercetta q 'vere e proprie', ossia entrambi i numeri sono diversi da zero in senso statistico: non può quindi darsi (con un grado di fiducia del 95 per cento, si intende) che uno di questi due parametri sia trascurabile, e che la retta di regressione possa ricondursi alla forma $y = mx$ (una retta che passa per l'origine degli assi), oppure $y = q$ (una retta orizzontale).

5.2.1 APPROFONDIMENTO: Come *nascono* le stelline?

Concentriamoci per esempio sulla stima della pendenza della retta di regressione e sulla misura dell'inaffidabilità di tale stima, il cosiddetto errore standard della media (a tale proposito, vedi [26]). Il quoziente `m/se` tra questi due valori rappresenta il **consuntivo** del test (*test statistic*, erroneamente: 'statistica test') t di Student:

```
(m <- summary(modello)$coef[[2]])
(se <- summary(modello)$coef[[4]])
(m / se)
(tstudent <- summary(modello)$coef[[6]])
```

Nella variabile aleatoria `t` di Student (a 63 gradi di libertà, visto che ci sono 65 studenti ma ho già dovuto 'consumare' 2 informazioni per stimare m e q) imbattersi in un quantile pari ad 8.85 significa aver azzeccato un evento di probabilità molto molto rara; il consuntivo t sta dunque nella coda destra della distribuzione:

```
(pvalue <- summary(modello)$coef[[8]])
2*(1 - pt( tstudent, 63))
```

Con un p-value così basso, possiamo rifiutare a cuor leggero l'ipotesi nulla del test di Student, ossia che il coefficiente m non sia diverso da zero. Lo vediamo forse meglio anche con questi disegni:

```
x <- (-400:1000)/100
ygauss <- dnorm(x)
ystudent <- dt(x, 63)
zoom <- (800:1000)/100
zoomgauss <- dnorm(zoom)
zoomstudent <- dt(zoom, 63)
areastudent <- pt(x, 63)
areazoomstudent <- pt(zoom, 63)
```

```
par(mfrow = c(2,2))
plot(x,ygauss, "l", col = "violet", lty = 2, lwd = 3,
main = "variabili aleatorie", ylab = "v. aleatorie")
lines(x,ystudent, col = "orange", lty = 3, lwd = 4)
legend(1.8, 0.38, col = c("violet", "orange"),
c("standard normal", "student df 63"),
lty = c(2,3), lwd = 3:4)
points (8.85, dt(8.85, 63), pch = 19)
plot(zoom, zoomstudent, col = "orange", "l", lty = 3,
lwd = 4, main = "ingrandimento")
legend(4, 0.3, col = "orange",  "student df 63",
lty = 3, lwd = 4)
points (8.85, dt(8.85, 63), pch = 19)
plot(x, areastudent, "l", col = "orange", lty = 3,
lwd = 4, main = "probabilità (area)")
points (8.85, pt(8.85, 63), pch = 19)
plot(zoom, 1- areazoomstudent, "l", col = "orange",
lty = 3, lwd = 4, main = "area coda (1 - prob)")
points (8.85, 1 - pt(8.85, 63), pch = 19)
```

5.3 La diagnostica del modello lineare

> *.. cosa facciamo noi degli outlier? .. mah.. di solito nel nostro laboratorio ce li fanno buttare via dallo studio ..* (Giorgia G., biotecnologa specialista studente Ph.D.)

Bene. Fin qui abbiamo interpretato le stime del modello. Adesso dobbiamo eseguire la **diagnostica** del modello, perché non è mica detto che una nuvola di punti sia sempre e solo 'calibrabile' con una retta. In ogni modello di regressione dobbiamo prestare attenzione ai **punti separati** dalla nuvola, perché questi potrebbero avere una **forza di leva** tale da riuscire a influenzare i coefficienti della retta, riuscendo a 'spostarla'. E' un argomento che è stato messo benissimo in luce da F. Anscombe, con il suo *quartetto*[2].

Per esempio, consideriamo la studentessa numero 15:

```
genere[15]
statura[15]
peso[15]
```

[2] http://en.wikipedia.org/wiki/Anscombe's_quartet

Il nostro modello di regressione non distingue però tra maschi e femmine, e prevede genericamente che per un'altezza di 166 centimetri sia verosimile pesare attorno ai 58 chili:

```
0.8539245 * statura[15] - 83.89056
modello$fitted[[15]]
```

Pertanto, nel caso particolare della nostra studentessa, il modello ha un residuo ϵ_{15} che ci potrebbe apparire piuttosto elevato:

```
peso[15] - (0.8539245 * statura[15] - 83.89056)
modello$residual[[15]]
```

Ci chiediamo: se questa studentessa non fosse stata presente nel dataset, come sarebbe cambiata la retta di regressione? Di tanto o di poco?

```
modello_senza15 <- lm (peso ~ statura,
studentimedicina[c(1:14, 16:65),] )
modello_senza15
```

I coefficienti della regressione si sono modificati, ma non è facile giudicare in senso critico se lo spostamento sia di lieve o rilevante entità. E' proprio per questo motivo che sono state implementate in R numerose istruzioni per effettuare la diagnostica del modello lineare, che in pratica si riconduce ad esaminare visualmente questi quattro grafici:

```
par(mfrow = c(2,2))
plot(modello)
```

Il primo grafico raffigura i residui ϵ_i. La retta grigia tratteggiata indica il valore teorico medio di zero (infatti, in teoria, la media degli errori dovrebbe essere nulla). La curva rossa è uno **smoother**, ossia una curva 'pesata localmente' che riassume l'andamento della nube dei residui: essa appare ragionevolmente piatta, e dunque non abbiamo a preoccuparci della ipotesi che la media sia pari a zero. La nube dei punti inoltre appare sufficientemente caotica (non ha cioè una forma di torta, o di cuneo, o qualche particolare curvatura), e questo ci fornisce un'indicazione affidabile del fatto che i residui abbiano deviazione standard σ costante. Il grafico però ci suggerisce di stare attenti agli studenti numero 1, numero 64 e numero 65 perché i loro residui potrebbero essere 'particolarmente elevati' rispetto agli altri.

Nel secondo pannello, il QQplot ci rassicura: non dobbiamo temere per la normalità dei residui. Il terzo pannello riflette quanto già visto nel primo

-e lo trascuriamo-, mentre dal quarto grafico deduciamo che non ci sono punti isolati nella nube che abbiano una rilevante forza di leva sulla retta di regressione (e la distanza di Cook[3] è l'oggetto matematico adatto per trattare questo aspetto). Quindi la studentessa numero 15 era stata da noi 'ingiustamente accusata' di essere differente dal punto di vista morfometrico dai suoi colleghi.

5.4 Quando una covariata non è un predittore?

Naturalmente, non è mica sempre detto che i coefficienti della regressione siano così 'ricchi di stelline', come nell'esempio precedente. Per esempio, consideriamo il *body mass index* degli studenti:

```
bmi <- 10000 * peso /(statura^2)
```

e convinciamoci che voler predire linearmente il `bmi` per mezzo della `statura` non è una idea molto astuta (il che è più che ovvio: il *body mass index* non cresce all'aumentare della statura, ma decresce, e in maniera quadratica)

```
modelloinutile <- lm (bmi ~ statura)
summary(modelloinutile)
```

Vediamo che la `statura` adesso non è più un predittore significativo (non ci sono le stelline, e infatti il p-value è 'grande' e vale 0.22, proprio perché il consuntivo t è 'piccolo', non tanto distante dallo zero) e a questo punto ci convinciamo che avremmo sicuramente fatto meglio, prima di fare la regressione, a dare un'occhiata alla nuvola di punti e capire che essi non avevano alcuna 'forma' rettilinea:

```
plot(statura, bmi)
abline(modelloinutile)
```

5.4.1 APPROFONDIMENTO: la parola *lineare* non vuol dire *rettilineo*

Facciamo un precisazione molto, molto importante: il termine modello **lineare** non sta a significare che la nuvola di punti debba necessariamente essere (approssimativamente) *rettilinea*. Il concetto di 'linearità' è inteso nella sua accezione matematica (senza essere troppo pesanti, una funzione è lineare ad esempio se $f(x + 2y) = f(x) + 2f(y)$), e riguarda i coefficienti della regressione. Osserviamo il

[3] `http://en.wikipedia.org/wiki/Cook's_distance`

```
plot(massima, minima)
```

Qui la nube di punti sembra esibire una curvatura, e potrebbe essere quindi opportuno aggiungere dei termini di secondo grado: potrebbe darsi che una *parabola di regressione* sia più adeguata di una retta per modellare il fenomeno? Proviamo a vedere:

```
summary ( lm (minima ~ massima + I(massima^2) ) )
```

Beh, no. Di stelline non se ne vedono. Tuttavia è istruttivo ragionare sul fatto che anche questa parabola di regressione **è** un modello lineare. E' vero che a scuola ci hanno insegnato che la parabola ha equazione matematica $y = ax^2 + bx + c$, ma in questo caso bisogna capire che le 'incognite' che vogliamo conoscere sono a, b e c (rispettivamente 0.003, -0.341 e 67.731), che sono di primo grado, donde la linearità!

5.4.2 Laboratorio

1. Considerate il dataset `heart`:

   ```
   www <- "http://www.dmi.units.it/~borelli/dataset/heart.txt"
   heart <- read.table( www , header = TRUE )
   attach(heart)
   head(heart)
   ```

   ed indagate se il `QTC` possa venir descritto con un modello lineare sull'`ETA` dei pazienti. Ci sono dei commenti da fare a tale proposito?

Capitolo 6

Calibrare in R una serie nonlineare di dati con una 'curva di regressione'

6.1 Stime dei parametri di modelli completamente nonlineari

Nell'articolo [18], Umberto Lucangelo ha misurato determinati volumi di aria `w` in un modello polmonare ventilato ad alta frequenza rispetto al volume `u` presente in un compartimento ventilato in maniera convenzionale, in condizioni sperimentali diverse caratterizzate da due parametri fisici, la resistenza `rr` e la elastanza `ee`:

```
u <- c(2.018181818, 1.835125448, 1.402826855, 1.214788732,
1.144827586, 0.959459459, 0.777027027, 0.753289474,
0.687096774)
w <- c(1.615482234, 0.968397291, 0.570221753, 1.568181818,
0.939325843, 0.555208333, 1.540145985, 0.926247289,
0.555443548)
rr <- c(5, 20, 50, 5, 20, 50, 5, 20, 50)
ee <- c(35, 35, 35, 50, 50, 50, 100, 100, 100)
plot(u,w, col = factor(rr), pch = 19 + trunc(ee/16))
```

Nel grafico, i punti di colore uguale sono caratterizzati dalla medesima resistenza `rr`, mentre i punti della medesima forma hanno la stessa elastanza `ee`. In termini matematici, fissato `u`, al crescere di `rr` diminuisce `w`, ed al crescere di `ee` (e, forse, con lo zampino di `rr`) la funzione $w(u)$ diviene 'meno

convessa'. Per modellare questo fenomeno si potrebbe provare a cercare una funzione $w = w(u; ee, rr)$ della seguente forma:

$$w = \alpha + \frac{\beta}{\texttt{rr}} + \gamma \cdot \texttt{rr} \cdot \texttt{ee} \cdot u^{\delta}$$

Il problema qui è quello di stimare i parametri $\alpha, \beta, \gamma, \delta$ che appaiono incardinati in maniera nonlineare, visto che compaiono contemporaneamente nella formula addizione, moltiplicazione ed elevamento a potenza. A tale proposito possiamo utilizzare la funzione iterativa `nls` (*nonlinear least-squares*). Occorre però specificare alcuni valori di partenza per determinare i parametri ignoti:

```
curva <- nls(w ~ alfa + beta / rr + gamma *
( rr * ee * u ^ delta), start =
list(alfa = 1.1, beta = 3.2,
gamma = -0.01, delta = 1.3), trace = TRUE)
summary(curva)
```

Dopo sei ricorsioni viene raggiunta una precisione di circa un milionesimo sulle stime ottenute, e dal punto di vista statistico i coefficienti sono significativi (i.e. non sono nulli con fiducia del 95%).

$$w = 1.005 + \frac{3.195}{\texttt{rr}} - 0.00018 \cdot \texttt{rr} \cdot \texttt{ee} \cdot u^{1.236}$$

6.2 Stime dei parametri di modelli riconducibili a modelli lineari

Vi sono dei fenomeni che possono venir modellati da leggi nonlineari, le quali - con delle opportune trasformazioni algebriche - possono venir riformulate modificando le variabili in modo tale che la nuova relazione che le lega sia di tipo lineare (cfr. [17]). Si tratta essenzialmente dei modelli:

1. iperbolico
2. esponenziale
3. *maxima function*
4. potenza
5. logistico
6. sigmoidale

6.2.1 Il modello iperbolico.

Un esempio biochimico di legge iperbolica è dato dalla *cinetica di Michaelis-Menten*, che descrive l'andamento della velocità di una reazione catalizzata da enzimi[1]. In tal caso la risposta y (il tasso di reazione enzimatica) dipende dalla concentrazione x tramite la legge:

$$y = \frac{A \cdot x}{B + x}$$

Supponiamo dunque che siano dati x e y:

```
x <- (3:40)/10
y <- c( 0.71, 0.91, 1.11, 0.60, 0.48, 1.29, 0.97, 1.23, 1.12,
1.05, 1.63, 1.44, 1.23, 1.45, 1.41, 2.00, 2.04, 1.28,
1.56, 1.51, 1.87, 1.75, 1.66, 2.23, 1.75, 2.02, 1.81,
2.30, 2.19, 1.76, 1.51, 2.36, 2.35, 1.67, 1.66, 1.98,
1.66, 1.83)
```

La trasformazione algebrica opportuna che rende lineare la relazione è data dal rapporto delle variabili, x/y:

```
trasforma <- x/y
modello <- lm (trasforma ~ x)
modello
par(mfrow = c(1,2))
plot(x,y)
plot(x, trasforma)
abline(modello)
```

Ora i parametri m e q del `modello` di regressione appena trovato sono legati ai parametri incogniti A e B dalle seguenti relazioni:

```
( A <- 1 / modello$coefficients[[2]] )
( B <- modello$coefficients[[1]] / modello$coefficients[[2]] )
```

6.2.2 Il modello esponenziale.

Il Tecnezio 99m è un tracciante radioattivo utilizzato ad esempio nella scintigrafia ossea. Il decadimento radioattivo di questo elemento si modella con una legge nonlineare del tipo $y = A \cdot \exp(B \cdot x)$. Supponiamo di avere questa serie di dati:

[1] `http://en.wikipedia.org/wiki/Michaelis%E2%80%93Menten_kinetics`

```
x <- 1:24
y <- c(4.355, 3.844, 3.540, 3.016, 2.856, 2.433, 2.123,
2.029, 1.737, 1.583, 1.379, 1.201, 1.172, 1.002, 0.842,
0.781, 0.679, 0.712, 0.645, 0.426, 0.420, 0.355, 0.378, 0.308)
plot(x,y)
```

Considerando il logaritmo della y, il problema diviene lineare:

```
trasforma <- log(y)
modello <- lm (trasforma ~ x)
modello
par(mfrow = c(1,2))
plot(x,y)
plot(x, trasforma)
abline(modello)
```

Ed anche ora dai parametri m e q della retta di regressione `modello` possiamo ricalcolare i parametri incogniti A e B con le seguenti relazioni:

```
( A <- exp(modello$coefficients[[1]]) )
( B <- modello$coefficients[[2]] )
```

Si osservi che da B si può ricavare anche l'emivita stimata del tracciante mediante la formuletta

```
log(0.5) / B
```

che nel nostro caso vale circa 6.03 ore (in letteratura[2] il valore riportato è 6.01).

6.2.3 Il modello *maxima function.*

Il dataset `analgesia`, di tipo cross-section, in realtà è tratto da uno studio longitudinale, e la 'curva del dolore' di un certa paziente trattata con il `farmaco` di tipo `ketorolac`, misurata in 7 occasioni nell'arco di 72 ore dalla fine dell'intervento e valutata in una scala soggettiva da 1 a 10 è risultata essere la seguente:

```
x <- c(1, 6, 12, 24, 36, 48, 72)
y <- c(1, 7, 8, 4, 2, 1, 1)
plot(x,y, "b")
```

[2] `http://en.wikipedia.org/wiki/Technetium`

Questa curva può venir modellata con una legge nonlineare del tipo $y = A \cdot x \exp(B \cdot x)$. Il cambio di variabile da operare sulla y per rendere il problema lineare è il seguente:

```
trasforma <- log(y/x)
modello <- lm (trasforma ~ x)
modello
par(mfrow = c(1,2))
plot(x,y, "b")
plot(x, trasforma)
abline(modello)
```

I parametri m e q della retta di regressione permettono di calcolare i parametri incogniti A e B con le medesime relazioni viste nei casi precedenti:

```
( A <- exp(modello$coefficients[[1]]) )
( B <- modello$coefficients[[2]] )
```

6.2.4 Il modello potenza.

Le fasi iniziali della crescita cellulare in una lesione tumorale possono essere modellate dall'*equazione di von Bertalanffy*[3] del tipo $y = K + A \cdot x^B$. Supponiamo che questi siano i dati, registrati in unità di tempo arbitrarie:

```
x <- 1:7
y <- c(1.2, 1.3, 1.7, 2.8, 5.5, 10.9, 20.2)
plot(x,y, "b")
```

Per modellare questa curva, in cui compaiono tre parametri incogniti(K, A e B) non possiamo sperare di ottenere tutte le risposte da una retta di regressione (che ha solo due parametri, la pendenza e l'intercetta). Dunque, iniziamo a stimare il valore di K. Siccome per $x = 1$ l'equazione diventa $y = K + A \cdot 1^B = K + A$, questo significa che $K + A = 1.2$. Per $x = 0$ non conosciamo il valore di $y = K$, ma possiamo presumere che 'assomigli' ad 1.2, e ne sia leggermente inferiore. Stimiamo dunque che K valga 1.1. Operiamo perciò due cambi di variabile, su x e su y:

```
stimaK <- 1.1
ytrasforma <- log(y - stimaK)
xtrasforma <- log(x)
```

[3] http://en.wikipedia.org/wiki/Ludwig_von_Bertalanffy

```
modello <- lm (ytrasforma ~ xtrasforma)
modello
par(mfrow = c(1,2))
plot(x,y, "b")
plot(xtrasforma, ytrasforma)
abline(modello)
```

Per ritrovare i coefficienti della relazione, operiamo le trasformazioni inverse:

```
( A <- exp(modello$coefficients[[1]]) )
( B <- modello$coefficients[[2]] )
```

Guardiamo con un grafico se i valori di A e B ottenuti dalla nostra stima di K (0.0503528 e 2.784975) ci forniscono una soluzione affidabile:

```
plot(x,y, "b")
prova <- stimaK + A * x ^ B
points(x, prova, col ="red")
```

Come si vede, i punti di colore rosso non sono affatto soddisfacenti: questo significa che la nostra stima di K non era adeguata. Potremmo ritentare l'analisi con altri valori di K leggermente più grandi o leggermente più piccoli; oppure, conviene utilizzare la nostra stima di K e i valori corrispondenti di A e B come valori `start` per la funzione `nls` che abbiamo introdotto nel paragrafo precedente:

```
curva <- nls(y ~ kappa + A * x ^ B,
start =  list(kappa = 1.1, A = 0.050, B = 2.785),
trace = TRUE)
summary(curva)
```

Otteniamo che $K = 1.209$, $A = 0.00365$ e $B = 4.398$, ed il grafico ora si che è soddisfacente:

```
plot(x,y, "b")
stimanls <- 1.209 + 0.00365 * x ^ 4.398
points(x, stimanls, col ="blue")
```

6.2.5 Il modello logistico.

Il modello di crescita delle popolazioni proposto nel 1838 da Pierre-François Verhulst si può rappresentare nella forma $y = \frac{K}{1+A\cdot\exp(B\cdot x)}$. Anche in questo caso i parametri sono tre, e la nonlinearità obbliga a fornire una stima di partenza di K; siccome di solito il coefficiente B è negativo, per x 'molto grande' i valori di y tendono a K. Consideriamo ([19]) la curva:

```
x <- 1:7
y <- c(199.5,  220.6,  292.0,  341.0,  396.3,  423.3,  436.5)
plot(x,y, "b")
```

Il grafico ci suggerisce di provare a stimare $K = 500$; operiamo quindi il seguente cambio di variabile sulla risposta y:

```
stimaK <- 500
trasforma <- log( stimaK / y - 1)
modello <- lm (trasforma ~ x)
modello
par(mfrow = c(1,2))
plot(x,y, "b")
plot(x, trasforma)
abline(modello)
```

Come in precedenza, otteniamo stime dei parametri A e B (rispettivamente 2.52 e -0.43), che appaiono essere abbastanza in accordo con i dati.

```
( A <- exp(modello$coefficients[[1]]) )
( B <- modello$coefficients[[2]] )
plot(x,y, "b")
prova <- stimaK / ( 1 + A * exp(B * x) )
points(x, prova, col ="red")
```

Volendo, anche in questo caso possiamo migliorare le stime con la funzione `nls`:

```
curva <- nls(y ~ kappa / ( 1 + A * exp(B * x) ),
start =  list(kappa = 500, A = 2.52, B = -0.43),
trace = TRUE)
summary(curva)
plot(x,y, "b")
stimanls <- 499.3 / ( 1 + 2.615 * exp( -0.436 * x) )
points(x, stimanls, col ="blue")
```

6.2.6 Il modello sigmoidale.

Concludiamo questa rassegna provando a *fittare* i dati precedenti non con il modello logistico di Verhulst, ma con una sigmoide: $y = \frac{K}{1+A \cdot x^B}$. Il cambio di variabile è analogo a quello della potenza e della logistica combinate assieme:

```
x <- 1:7
y <- c(199.5,  220.6,  292.0,  341.0,  396.3,  423.3,  436.5)
stimaK <- 500
ytrasforma <- log( stimaK / y - 1)
xtrasforma <- log(x)
modello <- lm (ytrasforma ~ xtrasforma)
modello
par(mfrow = c(1,2))
plot(x,y, "b")
plot(xtrasforma, ytrasforma)
abline(modello)
( A <- exp(modello$coefficients[[1]]) )
( B <- modello$coefficients[[2]] )
```

Otteniamo delle stime dei parametri A e B (rispettivamente 2.20 e -1.29), ma il grafico nel pannello destro ci dice che siamo completamente fuori strada, a causa della errata scelta del parametro K. Non ci resta che utilizzare `nls`:

```
curva <- nls(y ~ kappa /( 1 + A * x^B),
start = list(kappa = 500, A = 2.20, B = -1.29),
trace = TRUE)
```

Purtroppo però otteniamo un messaggio di errore, la routine non converge ad una soluzione stabile. Si potrebbe ora provare a modificare i parametri di partenza, opure provare a cercare di analizzare per quali ragioni questo accada.

Capitolo 7

Ancova: alla ricerca del modello lineare minimale adeguato in R

7.1 La *Ancova*

Il termine *Ancova* è acronimo di *analysis of covariance*; ma sebbene in statistica la covarianza sia un oggetto che esprime la 'variazione comune' di due variabili (che in R si calcola con il comando `cov` e che è algebricamente legata alla correlazione lineare di Pearson `cor`), in termini di modellazione la *Ancova* altro non è che una 'unione' della Anova alla retta di regressione. Spieghiamoci meglio: abbiamo già visto l'esempio dell'attività sportiva (`sport`) (variabile **fattore** di tipo categorico a tre livelli) che influenza il `peso` degli studenti, introducendo la Anova:

```
www <- 'http://www.dmi.units.it/~borelli/
   dataset/studentiannoscorso.txt'
studentimedicina <- read.table(www, header = TRUE)
attach(studentimedicina)
modellos <- lm (peso ~ sport)
summary(modellos)
```

Nel modulo successivo abbiamo visto che la `statura` (variabile **numerica** di natura continua) dei nostri studenti è anche un predittore del loro `peso`:

```
modelloa <- lm (peso ~ statura)
summary(modelloa)
```

La *Ancova* è la tecnica che consente di analizzare congiuntamente predittori fattore e predittori numerici. Iniziamo con l'esempio più semplice, distinguendo la retta di regressione del peso vs. la statura nel caso degli studenti maschi, in colore blu, e delle studentesse femmine, in color violetto:

```
plot(statura, peso, pch =".")
points(statura[genere == "m"], peso[genere == "m"],
col = "blue", pch = 19)
points(statura[genere == "f"], peso[genere == "f"],
col = "violet", pch = 19)
uominiregr <- lm (peso[genere == "m"] ~ statura[genere == "m"])
abline(uominiregr, col = "blue", lty = 3, lwd = 3)
donneregr <- lm (peso[genere == "f"] ~ statura[genere == "f"])
abline(donneregr, col = "violet", lty = 2, lwd = 2)
```

Vediamo che le due rette non sono parallele, e quella dei maschi ha una pendenza superiore a quella delle femmine. Inoltre, la retta blu ha un'intercetta (negativa) inferiore a quella della violetta:

```
uominiregr
donneregr
```

Dal punto di vista statistico ci potremmo imbattere in varie casistiche: potrebbe innanzitutto darsi che le due rette abbiano realmente due pendenze m diverse in senso statistico. In tal caso, si dice che il predittore `genere` possiede una **interazione** (simbolo: '`*`') con il predittore `statura`; oppure, potrebbe darsi che le le due rette sono (in senso statistico) parallele, e dunque l'interazione tra il `genere` e la `statura` non è significativa; siamo dunque in presenza solo di un modello **additivo** (simbolo: '`+`'). O ancora, potrebbe darsi che le rette hanno pendenze diverse in senso statistico ma la loro intercetta è la stessa, nel senso che il `genere` non dà origine ad una differenza significativa di q.

Vediamo cosa succede nel caso concreto:

```
modelloancova <- lm (peso ~ genere * statura)
summary(modelloancova)
```

Interpretiamo l'output. Il fatto che `statura` abbia una stima di 0.4505 (con un p-value significativo di 0.015) vuol dire che la retta di colore violetto, pertinente alle femmine, ha una pendenza 'vera', diversa da zero. Ossia con un grado di fiducia elevato, superiore al 95%, siamo portati a ritenere che il

peso aumenti in ragione della statura, ed un aumento -per le ragazze- di 1 cm di altezza comporti un aumento di massa corporea di circa .45 chilogrammi.

Vediamo però che il **termine di interazione** `generem:statura`, che vale 0.1925, non è significativo. Cosa vuol dire? Vuol dire che la retta di colore blu ha un coefficiente angolare uguale a $0.4505 + 0.1925 = 0.643$. Tuttavia l'inaffidabilità della stima, misurata dallo standard error 0.2493, è molto elevata; pertanto il consuntivo $t = 0.772$ non è affatto discosto dallo zero, e dunque un valore di probabilità del 44.3% ci fa decidere che la pendenza della retta blu non sia diversa , in senso statistico, dalla pendenza della retta rosa. In conclusione, per i ragazzi, un aumento di 1 cm di altezza provoca un aumento di peso, in media, di circa 0.64 chilogrammi; ma non abbiamo prove sufficienti per affermare che i numeri 0.45 e 0.64 siano diversi tra loro.

Dunque, il termine di interazione non ha senso che venga considerato, e forti dell'insegnamento di Guglielmo di Occam (*Numquam ponenda est pluralitas sine necessitate*), consideriamo un modello più parsimonioso:

```
modelloancova2 <- lm (peso ~ genere + statura)
summary(modelloancova2)
```

In questo modello le rette si ipotizzano parallele, ossia la crescita di peso nei maschi e nelle femmine è di 0.55 chili per ogni centimentro di statura (diciamolo bene: ci attendiamo che uno studente che è più alto di 1 centimetro di un altro studente -indifferentemente se maschio o femmina- pesi circa 0.55 chili più dell'altro). Ma le rette di regressione hanno delle quote diverse: per le ragazze $y = 0.55 \cdot x - 35.37$, per i ragazzi $y = 0.55 \cdot x - 35.37 + 7.20$, ossia $y = 0.55 \cdot x - 28.17$. Il fatto che la stima su `generem` sia significativa vuol dire che i ragazzi partono da una quota diversa da quella delle ragazze; ma non siamo certi (p-value 0.093) che sia opportuno mettere una quota q di partenza per le ragazze.

Facciamo ancora un tentativo: due rette che entrambe partano dall'origine (ossia, non abbiano alcuna intercetta q, donde il simbolo `-1`, oppure `+0`, nella sintassi di R) ma che abbiano pendenza diversa a seconda del `genere`:

```
modelloancova3 <- lm (peso ~ genere : statura - 1)
summary(modelloancova3)
```

Molto interessante! Abbiamo un modello ricchissimo di stelline. L'unica cosa di cui ci dobbiamo peoccupare ora è di '*non aver buttato via troppa informazione*': ossia, che il modello numero 3 sia 'troppo povero' rispetto al secondo modello. A tale scopo, effettuiamo una **analisi della devianza**, che con una discutibile scelta etimologica si effettua con il comando `anova`:

```
anova(modelloancova2, modelloancova3)
```

Il responso della analisi della devianza è un p-value, 0.173, che per tradizione non ci consente di rifiutare l'ipotesi nulla. In altri termini, i due modelli non sono diversi tra di loro in senso statistico, e perciò preferiamo il `modelloancova3` proprio perché ha 63 gradi di libertà rispetto ai 62 del suo antagonista (ossia, il `modelloancova3` descrive i dati con 2 soli parametri invece dei 3 necessari al `modelloancova2`: *frustra fit per plura quod potest fieri per pauciora*. La diagnostica del modello si effettua, come abbiamo visto in precedenza, con il comando `plot(modelloancova3)`.

7.1.1 APPROFONDIMENTO: nell'Ancova l'ordine è importante

Ci sarebbe da usare una certa cautela nell'Ancova, perché .. diciamolo con parole imprecise: nei modelli con interazione non vale la proprietà commutativa. In altri termini:

```
uno <- lm (peso ~ genere * statura)
due <- lm (peso ~ statura * genere)
```

operano una ripartizione della varianza dei residui in maniera diversa. Ce ne accorgiamo dalle vecchie tavole Anova:

```
summary.aov(uno)
summary.aov(due)
```

Nel primo modello, ad esempio, il p-value del predittore `genere` ($4.388 \cdot 10^{-13}$) è infimo, mentre nel secondo modello esso vale 0.0009: si tratta comunque di un valore molto piccolo, ma in altri esempi questa differenza potrebbe essere cruciale e ci metterebbe nell'imbarazzo di non sapere 'a chi dare ragione'.

7.1.2 APPROFONDIMENTO: il modello lineare e l'algebra

Per chi conosca un po' il linguaggio dell'algebra, possiamo affrontare il discorso della stima dei coefficienti della regressione analizzando ad esempio questo modello:

```
modello <- lm (peso ~ statura + scarpe)
summary(modello)
```

Qui noi vogliamo esprimere la risposta Y (`peso`) in termini dei due predittori X1 (`statura`) ed X2 (`scarpe`) utilizzando il modello lineare:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

dove i β sono tre parametri incogniti ed ϵ rappresenta genericamente l'errore. In forma matriciale scriviamo:

$$Y = X\beta + \epsilon$$

ed intendiamo che:

- $Y = (Y_1, \cdots, Y_{65})^T$ è il vettore 65-dimensionale della statura
- $\epsilon = (\epsilon_1, \cdots, \epsilon_{65})^T$ è il vettore 65-dimensionale degli errori
- $\beta = (\beta_0, \beta_1, \beta_2)^T$ è il vettore tridimensionale dei coefficienti della regressione

mentre X è una matrice composta da 65 righe e 3 colonne, in cui la prima colonna è composta da tanti 1 per rappresentare l'intercetta β_0, la seconda colonna contiene il vettore `statura` e la terza colonna contiene il vettore del numero di `scarpe`:

$$X = \begin{pmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ \vdots & \vdots & \vdots \\ 1 & X_{65,1} & X_{65,2} \end{pmatrix}$$

Un teorema (di Gauss e Markov) assicura che ci sono 'molti vantaggi matematici' nell'andare alla ricerca di tre opportuni parametri $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)^T$ che siano la **stima ai minimi quadrati** del vettore incognito $\beta = (\beta_0, \beta_1, \beta_2)^T$. A tale scopo, i matematici sanno dimostrare che per trovare chi siano tali $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)^T$ (cioè, i numeri che R ci fornisce) è sufficiente risolvere il **sistema di equazioni normali**:

$$X^T \cdot X\hat{\beta} = X^T \cdot Y$$

Ora, se la matrice quadrata $X^T \cdot X$ è invertibile (i.e. il suo determinante è diverso da zero), si ha che, calcolandone l'inversa $(X^T \cdot X)^{-1}$:

$$(X^T \cdot X)^{-1}(X^T \cdot X)\hat{\beta} = (X^T \cdot X)^{-1} X^T \cdot Y$$

$$\hat{\beta} = (X^T \cdot X)^{-1} X^T \cdot Y$$

$$X\hat{\beta} = X(X^T \cdot X)^{-1} X^T \cdot Y$$

$$\hat{Y} = \hat{X} \cdot Y$$

e perciò i valori previsti (in gergo, 'fittati') del modello lineare si ottengono a partire dal vettore Y proiettandolo ortogonalmente sullo spazio lineare generato dalle colonne della matrice X per mezzo della **matrice cappuccio**, $X(X^T \cdot X)^{-1} X^T$. La componente stocastica del modello, ossia il vettore degli errori $\epsilon = (\epsilon_1, \cdots, \epsilon_{65})^T$, rimane perciò stimata per mezzo del **vettore dei residui**:

$$\hat{\epsilon} = Y - \hat{Y} = (\mathbb{I} - \hat{X}) \cdot Y$$

Verifichiamo il tutto con R. Creiamo innanzitutto la matrice X del modello lineare:

```
( X <- model.matrix( ~ statura + scarpe ) )
```

e 'creiamo' (i.e. 'cloniamo') la variabile risposta

```
y <- peso
```

Ora creiamo la matrice quadrata di dimensione tre per tre, $X^T \cdot X$, per mezzo del prodotto matriciale **righe per colonne**, `%*%` , e dell'operatore di trasposizione `t( )`, che scambia le righe della matrice con le colonne:

```
( t(X) %*% X )
```

Chi sono i numeri che vediamo nella matrice $X^T \cdot X$? Confrontiamo l'output di questi comandi:

```
( t(X) %*% X )[2,2]
( t(X) %*% X )[2,3]
sum (statura * statura)
sum (statura * scarpe)
```

Accertiamoci ora che questa matrice quadrata sia invertibile, ossia che abbia determinante non nullo:

```
det( t(X) %*% X ) != 0
```

Procediamo utilizzando la funzione `solve` per calcolare la matrice inversa $(X^T \cdot X)^{-1}$:

```
(inversa <- solve(t(X)%*%X))
```

Otteniamo infine le stime dei tre coefficienti della regressione, $\hat{\beta} = (X^T \cdot X)^{-1} X^T \cdot Y$:

```
(inversa %*% t(X) %*% y)
```

in perfetto accordo con il `summary` del `modello`, e mediante la *matrice cappuccio* di 65 righe e 65 colonne, otteniamo anche i valori predetti dalla regressione, i *fitted values*:

```
Xcappuccio <-  X %*% inversa %*% t(X)
Xcappuccio %*% y
```

Vale la pena osservare che mediante la funzione `crossprod` che, date due matrici A e B ne calcola il prodotto $A^T \times B$, il sistema di equazioni normali si può anche risolvere in questo modo:

```
solve( crossprod(X, X) , crossprod(X, y))
```

Infine, si può verificare che la componente stocastica ha valore atteso nullo (o quasi ..)

```
mean( y - Xcappuccio %*% y )
```

e deviazione standard pari (o quasi ..) allo **residual standard error**:

```
sd(y - Xcappuccio %*% y) * 65/64
summary(modello)$sigma
```

che equivale a:

```
sqrt(deviance(modello3) / df.residual(modello3))
```

Vogliamo concludere questo approfondimento ricordando che gli elementi della diagonale della matrice 65 per 65 `Xcappuccio` che noi abbiamo creato 'a mano':

```
diag(Xcappuccio)
```

sono invece richiamabili direttamente dal `modello` anche con questi comandi:

```
influence(modello)$hat
hat(X)
```

si chiamano in gergo i **leverages**. La loro somma (i.e. la **traccia** della matrice $\hat{X}$) equivale al numero di parametri del `modello`:

```
sum(diag(Xcappuccio))
```

I leverages sono gli strumenti adeguati per affrontare la diagnostica del modello lineare (per approfondimenti, si veda il capitolo 7 di [14]).

Capitolo 8

La regressione logistica e la regressione di Poisson con R negli studi cross-section

8.1 La regressione logistica

Umberto Lucangelo nel suo paper [20] si occupa di un certo indice capnografico, detto $\frac{V_{AE}}{V_T}$, e della sua possibile predittività di un outcome sfavorevole in caso di peggioramento di tale indice per i pazienti ricoverati in terapia intensiva. Il dataset `vaevt` in effetti è di tipo longitudinale, ma in questo caso ci interessa l'evoluzione della situazione (`deltavaevt`) sulla base delle prime 48 ore di ricovero. Nel `plot` si evidenzia chiaramente che è poco probabile che una persona con il `deltavaevt > 0` abbia `outcome` sfavorevole (in questo esempio, il decesso è codificato con 0 e la sopravvivenza con 1).

```
www <- "http://www.dmi.units.it/~borelli/dataset/vaevt.txt"
vaevt <- read.table( www , header = TRUE )
attach(vaevt)
str(vaevt)
deltavaevt <- quarantotto - basale
deltapafi <- pafi48 - pafi0
plot(deltavaevt, outcome, col = "violet", pch = 19)
```

Alcuni autori in letteratura attribuiscono un ruolo predittivo anche a `deltapafi`, anche se questo non appare così evidente in queste due figure:

```
layout( matrix(1:2, ncol = 2))
cdplot( factor(outcome) ~ deltavaevt)
```

```
cdplot( factor(outcome) ~ deltapafi)

plot( deltapafi ~ deltavaevt, col = "white" )
points( deltapafi[outcome =="1"] ~ deltavaevt[outcome=="1"],
 col = "darkgreen", pch = 22 )
points( deltapafi[outcome =="0"] ~ deltavaevt[outcome=="0"],
 col = "red", pch = 19 )
```

Ritorniamo al primo grafico, ed aggiungiamogli la retta di regressione: vediamo chiaramente che un modello lineare, raffigurato in colore rosso, appare del tutto fuori luogo:

```
plot(deltavaevt, outcome, col = "violet", pch = 19)
sbagliato <- lm(outcome ~ deltavaevt)
abline(sbagliato, col = "red", lwd = 3, lty = 2)
```

Questo accade perché la variabile 'risposta' è di tipo **binario** (successo/insuccesso), e in questo caso non è la retta di regressione ma la **regressione logistica** la soluzione più adeguata: per un rapido richiamo potete consultare `http://ccforum.com/content/9/1/112`. La regressione logistica è un esempio di una classe più generale di modelli che va sotto il nome di **generalized linear models**, modelli lineari generalizzati (per maggiori dettagli: [**?**, 9, 15, 1]), noti anche con il nomignolo di *glim*. In un glim, ci sono tre argomenti:

1. la **componente aleatoria** $Y = (Y_1, Y_2, \ldots, Y_N)$, che è la variabile di risposta (in questo caso, `outcome`);
2. la **componente sistematica**, detta anche il **predittore lineare** - proprio a ricordare la loro stretta relazione con i modelli lineari;
3. la **funzione di link**, ossia una funzione $g(\cdot)$ che collega la componente aleatoria a quella sistematica in una maniera precisa (e tralasciamo i dettagli per semplicità).

Per definire la funzione di link, nel caso binomiale, l'idea è che se un certo evento ha probabilità $0 < p < 1$ di verificarsi, allora il rapporto $p/(1-p)$ tra la probabilità dell'evento e la probabilità del suo complementare viene denominato *odds*. L'odds è un numero prossimo a zero se p è molto bassa; al contrario, l'odds è molto grande se p è vicina ad 1. Il logaritmo di un *odds* viene detto in gergo *logit* ed, al variare della probabilità p, il grafico del *logit* esplode, tra meno infinito e più infinito. Ma invertendo il grafico, ossia

esplicitando la probabilità p in funzione del *logit*, la funzione che si ottiene ha chiaramente valori compresi tra zero ed uno, e questa curva ha equazione $p = \frac{exp(y)}{1+exp(y)}$.

La cosa *user-frienfly* è che con R si possono trascurare questi dettagli matematici, ed andare subito 'al sodo' con il comando `glm`; l'unica accortezza è quella di specificare al software che si tratta di eventi binomiali (il link *logit* è sottinteso):

```
modello1 <- glm(outcome ~ deltavaevt, family = binomial())
summary(modello1)
```

Il `deltavaevt` è un predittore significativo dell'`outcome`. Non lo è invece l'intercetta; quindi la togliamo di mezzo per risparmiare un parametro:

```
modello2 <- glm(outcome ~ deltavaevt - 1 , family = binomial())
summary(modello2)
```

Abbiamo rinunciato ad un parametro, e quindi il modello è diventato 'un po' più povero'. Ma per essere certi di non aver fatto danni, vogliamo verificare che il `modello2` non sia troppo parsimonioso, conducendo un'analisi della devianza come abbiamo visto nel modulo precedente dedicato alla modellazione lineare. Qui, la variabile aleatoria corretta da utilizzare per confrontare i due modelli non è la distribuzione normale, ma la distribuzione del Chi Quadrato:

```
anova(modello2, modello1, test = "Chisq" )
```

Eh, insomma, un p-value di 0.097 non è significativo (però si potrebbe fare anche un discorso maggiormente conservativo..). Per esercizio, potreste provare a controllare che il modello `outcome` $\sim$ `deltavaevt + deltapafi - 1` è ridondante, come avevamo intuito nei grafici iniziali.

8.1.1 Come interpretare i parametri della regressione logistica

Il `modello2` è un modello molto semplice, poiché possiede un unico predittore (`deltavaevt`) il cui coefficiente vale 10.45. Come interpretiamo graficamente, e numericamente, questa quantità? Se fossimo in presenza di un modello lineare, esso sarebbe il coefficiente angolare (la pendenza) della retta di regressione di equazione $y = mx+q$. Qui, sappiamo che abbiamo a che fare con una componente sistematica: il predittore lineare, del tipo $a + bx$; siccome nel modello abbiamo tolto l'intercetta, allora $a = 0$ e $b = 10.45$:

```
a <- 0
b <- modello2$coef[[1]]
```

A questo punto, possiamo anche determinare la 'curva di regressione' $p = \frac{exp(a+bx)}{1+exp(a+bx)}$ che descrive la probabilità di avere un outcome favorevole in funzione del `deltavaevt`:

```
probabilita <- exp(a+b*deltavaevt)/(1+exp(a+b*deltavaevt))
probabilita
x <- (100 * min(deltavaevt)) : (100 * max(deltavaevt)) / 100
lineacurva <-  exp (0 + b * x) /  (1 + exp(0 + b * x))
plot(deltavaevt , outcome)
points(deltavaevt , probabilita, col = "red")
points(x, lineacurva, col = "red", pch =".")
lines(c(min(deltavaevt), max(deltavaevt)),
c(0.5, 0.5), col = "blue")
```

La curva punteggiata di colore rosso descrive la `probabilita` di sopravvivere al variare del `deltavaevt`: per esempio, un paziente che in quarantotto ore migliora la sua ossigenazione con un `deltavaevt` pari a 0.10 ha una probabilità di circa il 75% di sopravvivenza.

8.1.2 Il problema della sovradispersione

C'è da tenere presente il fatto che nel comando `lm` la `family` di variabili aleatorie è la `gaussian`, e la funzione di *link* è la `identity`: queste precisazioni sono sottointese, perché nel comando `lm` esse vengono assunte per default. In una variabile aleatoria gaussiana, la media μ e la deviazione standard σ sono due parametri che non sono necessariamente legati tra loro; al contrario, in una variabile aleatoria binomiale (o meglio, in un processo bernoulliano) il valore atteso $n \cdot p$ e la deviazione standard $n \cdot p \cdot (1-p)$ sono chiaramente legate tra loro ($\sigma = \mu \cdot (1-p)$). Da questa argomentazione si può avere un cenno circa la ragione per la quale nel `summary` di un `glm` si debba tenere d'occhio la `Residual deviance`, che non deve superare i gradi di libertà dei residui:

```
summary(modello2)$deviance
summary(modello2)$df[2]
```

Nel nostro modello la devianza residua 41.8 supera i 35 gradi di libertà, ed è pertanto opportuno 'far modificare' da R la distribuzione su cui modellare i residui:

```
modello2bis <- glm( outcome ~ deltavaevt - 1,
 family = quasibinomial())
summary(modello2bis)
```

Non ci accorgiamo immediatamente di cosa sia successo, ma se confrontiamo tra loro le stime dei parametri:

```
summary(modello2bis)$coef
summary(modello2)$coef
detach(vaevt)
rm(vaevt)
```

osserviamo che non viene modificato il β della regressione (10.45), ma si modifica l'inaffidabilità della stima, ossia lo standard error (che conseguentemente modifica il t-value ed il p-value).

8.2 La regressione di Poisson

In certi casi la risposta di uno studio è formata da *count*, ossia da numeri interi che quantificano l'occorrenza di un certo evento. Ad esempio, Pierpaolo Busan ([5]) ha condotto uno studio-pilota longitudinale 'randomizzato' e 'placebo-controllato' per stabilire il ruolo di un farmaco inibitore del reuptake della serotonina (la paroxetina) nella gestione della balbuzie. Focalizziamo la nostra attenzione sul periodo a 12 mesi, per ricondurci al caso di uno studio cross-section.

```
www <- "http://www.dmi.units.it/~borelli/dataset/paroxetine.txt"
paroxetine <- read.table(www, header = TRUE)
attach(paroxetine)
trattamento <- treatment[period == 12]
SSI <- SSIproportion[period == 12]
```

Ciò che ci interessa sapere è se lo *score* relativo alla quantità di ripetizioni di suoni e sillabe misurate dallo Stuttering Severity Index `SSI` sia legato al `trattamento`:

```
marginal <- glm(SSI ~ trattamento, family = poisson)
summary(marginal)
detach(paroxetine)
rm(paroxetine)
detach(trattamento)
```

```
detach(SSI)
rm(trattamento)
rm(SSI)
```

Siccome nella regressione di tipo Poisson il *link* è la funzione esponenziale, `exp(2.5649-0.4855)` e `exp(2.5649)` rappresentano le 'stime medie' di `SSI` rispettivamente nei due gruppi di diverso `trattamento`. Si osservi che il modello soffre di una elevata sovradispersione, e dunque l'opzione `family = quasipoisson` fornisce un modello più attendibile.

8.3 Laboratorio

1. Considerate il dataset `hcc` relativo agli 'snip' di Daniele:

   ```
   www <- "http://www.dmi.units.it/~borelli/dataset/hcc.txt"
   hcc <- read.table( www , header = TRUE )
   attach(hcc)
   head(hcc)
   table (Genotipo1, Genotipo2, Gruppo)
   ```

   e verificate quali tra le covariate `Genotipo1`, `Genotipo2`, `Sesso` ed `Eta` siano predittori significativi del `Gruppo`.

Capitolo 9

Curve di sopravvivenza con R

9.1 La distribuzione dei *time-to-event* non è gaussiana

In questo modulo ci occupiamo di un problema statistico che sarebbe riduttivo definire di tipo 'cross-section', stante il fatto che ci occupiamo della **analisi di sopravvivenza**, ossia della modellazione dei tempi in cui si realizza un determinato evento (che tipicamente negli studi biomedici è il decesso di un paziente, o il tempo impiegato per raggiungere un predeterminato valore soglia). C'è da dire che l'analisi di sopravvivenza è 'più difficile' della regressione, perché è più complicato stabilire cosa siano i residui (con conseguenti difficoltà per effettuare la diagnostica del modello). Inoltre, siamo fuori dal campo gaussiano: non possiamo servirci della variabile aleatoria normale per modellare i tempi di attesa. Come ci mostra bene Crawley, [10], facciamo finta di avere un'urna con una pallina nera e nove palline bianche; ed un'urna con due palline nere e otto palline bianche. L'esperimento consiste nel contare quante estrazioni sono necessarie affinché esca una pallina nera, reimbussolando di volta in volta la pallina (bianca) estratta, e ripetere questo esperimento per 50 volte: ci interessa cioé descrivere la distribuzione dei tempi di attesa di un evento con probabilità 1/10 e di uno con probabilità 2/10:

```
coda1 <- numeric(50)
coda2 <- numeric(50)
for (i in 1:50)
{
casuale <- runif(100)
coda1[i] <- which(casuale <= 0.1)[1]
```

```
coda2[i] <- which(casuale <= 0.2)[1]
}

boxplot(coda1, coda2)
```

Dai due boxplot si osserva che la distribuzione dei tempi di attesa non è gaussiana, ma è molto sfrangiata, con una grande 'skewness'. Ecco perché il `t.test` di certo non è adeguato per cercare differenze in questo frangente.

9.2 L'analisi di sopravvivenza

Avremo bisogno di installare il pacchetto `survival` per analizzare il dataset `breastcancer`, in cui l'interesse del chirurgo (e dell'azienda ospedaliera) è quello di avere la minore `los` (*length of stay*) in reparto:

```
library(survival)
www <- 'http://www.dmi.units.it/~borelli/
    dataset/breastcancer.txt'
breastcancer <- read.table(www, header = TRUE)
attach(breastcancer)
str(breastcancer)
head(breastcancer)
summary(breastcancer)
```

Innanzitutto, osserviamo che nel nostro dataset non vi sono **dati censurati**, ossia tutti i pazienti sono stati dimessi:

```
discharge
Surv(los, discharge)
```

Questo non succede sempre, anzi! Ad esempio, nel dataset `fertility` ci sono alcune coppie che dopo un `ciclo` di fecondazioni assistite hanno, oppure non hanno (e lo vediamo dal `+` posposto), avuto una gravidanza (`status`):

```
fertility <- 'http://www.dmi.units.it/~borelli/
    dataset/fertility.txt'
fertility <- read.table(www, header = TRUE)
attach(fertility)
Surv(ciclo, status)
detach(fertility)
rm(fertility)
```

Con il comando `survfit` possiamo calcolare gli elementi delle curve di sopravvivenza nei gruppi trattati con i bisturi dei tipi '*diart*' ed '*ultra*' mediante lo stimatore (nonparametrico) di Kaplan e Meier, e raffigurare le famose curve:

```
kaplanmeier <- survfit(Surv(los, discharge) ~ treatment)
summary(kaplanmeier)
plot(kaplanmeier, lty=c(5,3), col = c('violet', 'orange'),
xlab='los', lwd = 2)
legend( 15, 0.7, levels(treatment), lty=c(5,3),
col = c('violet', 'orange') , lwd = 2)
```

La domanda di ricerca più immediata è quella di scoprire se vi sia differenza tra le due curve, in senso statistico; a tale scopo disponiamo del test non parametrico denominato *log-rank* (e che è detto anche test di Mantel-Haenszel):

```
( logrank <- survdiff(Surv(los, discharge) ~ treatment) )
```

Dall'output risulta evidente che nei tempi di dimissione dei due gruppi la differenza è altamente significativa, e questo a prescindere dalle diverse covariate che sono presenti nel dataset. Volendo introdurre nell'analisi anche l'effetto delle variabili *age* e *drainage*, il log-rank test non è più utilizzabile, ma si può ricorrere al modello parametrico di sir David Roxbee Cox[1], dapprima introducendo l'età del paziente:

```
modello1 <- survreg(Surv(los, discharge) ~ age + treatment)
summary(modello1)
```

A quanto sembra l'età, `age`, ha un ruolo nel modello statistico (p-value = 0.0216, ossia 2.16e-02). Per interpretare correttamente i coefficienti dell'output, pensiamo che per un paziente di età `age` operato con il bisturi di tipo `diart`, il tempo atteso t di degenza postoperatoria è dato da $t = exp(1.5362+0.0105 \cdot age)$, mentre per un paziente operato con il bisturi di tipo `ultra`, la degenza attesa si riduce: $t = exp(1.5362-0.6486+0.0105 \cdot age)$, ossia $t = exp(0.8876+0.0105 \cdot age)$. Vediamo però cosa succede se aggiungiamo l'effetto relativo ai volumi di drenaggio:

```
modello2 <- survreg(Surv(los, discharge) ~ age
+ drainage + treatment)
summary(modello2)
```

[1] http://en.wikipedia.org/wiki/David_Cox_(statistician)

Un altro effetto altamente significativo: maggiori volumi di drenaggio implicano un aumento dei tempi di degenza. La covariata *age* ora però ha perso significatività (p-value 0.0676) e quindi per parsimonia la togliamo dall'analisi. Sarebbe interessante scoprire se i volumi di drenaggio differiscano nei due trattamenti, ossia se il termine di interazione '`drainage:treatment`' sia significativo:

```
modello3 <- survreg(Surv(los, discharge) ~ drainage*treatment)
summary(modello3)
```

Così non è, stante il p-value = 0.0676. Dunque, il nostro modello finale è puramente additivo:

```
modello4 <- survreg(Surv(los, discharge) ~ drainage+treatment)
summary(modello4)
detach(breastcancer)
rm(breastcancer)
```

9.3 Laboratorio

1. Consideriamo un sottoinsieme del dataset `watermaze` ([2]) relativo ad esperimenti sulla working memory dei topi nell'esperimento di tipo Morris Water Maze:

   ```
   www <- "http://www.dmi.units.it/~borelli/
       dataset/watermaze.txt"
   watermaze <- read.table(www, header = TRUE)
   attach(watermaze)
   gruppo <- group[day == "2"]
   topo <- subject[day == "2"]
   tempo <- time[day == "2"]
   censurato <- status[day == "2"]
   ```

   C'è qualche `gruppo` che si comporta in modo diverso dagli altri?

Capitolo 10

Studi longitudinali e misure ripetute

10.1 La questione delle misure ripetute nei design longitudinali

Partiamo, anzi, ri-partiamo da quanto abbiamo lasciato in sospeso nel modulo dedicato alla retta di regressione, nel caso degli esperimenti longitudinali: avevamo parlato del paper di Elisa D'Este[1], e di questa figura:

```
www <- "http://www.dmi.units.it/~borelli/dataset/elisadeste.txt"
elisadeste <- read.table( www , header = TRUE )
attach(elisadeste)
str(elisadeste)
elisadeste[c(1:3, 10:13, 25:28), ]
logcelln  <- (log(numerocellule)) ^ 0.25
library(lattice)
xyplot(logcelln ~ giorni | linea * ponasterone, type = "b")
```

La figura in effetti suggerisce un comportamento ragionevolemente 'lineare' dei dati, in tutti e quattro i pannelli. Quindi, di primo acchito, ci potrebbe venire in mente di analizzare i dati con un modello lineare di tipo Ancova con interazione, e la quantità di stelline potrebbe darci l'ebbrezza di aver trovato '*qualcosa di importante dal punto di vista della statistica*':

```
modellosbagliato <- lm (logcelln ~ giorni * linea * ponasterone)
summary(modellosbagliato)
```

[1] http://www.ncbi.nlm.nih.gov/pubmed/19666006

Attenzione, però! C'è una frase che deve metterci immediatamente in allarme:

```
Residual standard error:  0.005432 on 20 degrees of freedom
```

Come faccio avere 20 gradi di libertà se in effetti ha fatto solo quattro esperimenti? Beh, il problema è che avendo il dataset 28 righe (7 occasioni temporali per 4 esperimenti) mi sto imbattendo in un tipico caso di **pseudoreplication**: le misure contenute nel dataset non sono tutte indipendenti le une dalle altre. Per controprova, buttiamo via le letture del sedicesimo giorno: se si trattasse di dati indipendenti, dovremmo accorgerci di qualche cambiamento nel `summary`:

```
soloquattordici <- subset(elisadeste, giorni != "16")
detach(elisadeste)
rm(elisadeste)
attach(soloquattordici)
logcelln  <- (log(numerocellule)) ^ 0.25
controprova <- lm (logcelln ~ giorni * linea * ponasterone)
summary(controprova)
detach(soloquattordici)
rm(soloquattordici)
```

.. e, invece, praticamente non cambia un bel nulla. Questo significa che i dati possiedono una correlazione interna di cui dobbiamo cercare di tener conto nella modellazione, non affidandoci spensieratamente al comando `lm`. Prendiamo dunque in considerazione un esempio ancor più rilevante per dimensione campionaria. Il dataset `percussive` riporta i dati di uno studio longitudinale ([19]) che Umberto Lucangelo ha condotto paragonando l'efficacia della ventilazione ad alta frequenza (hfpv, *high frequency percussive ventilation*) versus la ventilazione convenzionale in pazienti affetti da gravi sindromi respiratorie.

```
www <- "http://www.dmi.units.it/~borelli/dataset/percussive.txt"
percussive <- read.table( www , header = TRUE )
attach(percussive)
library(lattice)
xyplot(pafi ~ time | treatment, type = "b",
groups = subject, xlab = "time (hour)",
ylab = expression(PaO[2] / FiO[2]))
modellolineare <- lm(pafi ~ time*treatment)
summary(modellolineare)
```

Siccome siamo in presenza di un dataset bilanciato (i.e., tutti i soggetti hanno esattamente quattro letture, non ci sono missing value), molti autori ricorrono alla tecnica di analisi nota con il nome di **Anova repeated measure**:

```
anovaRM <- aov(pafi ~ time * treatment
+ Error(factor(subject)/time))
summary(anovaRM)
```

dalla quale si potrebbero ottenere stime più attendibili sui p-value, se non fosse che spesso ci si dimentica di verificare se i dati siano distribuiti normalmente (e così non è!), e che la dispersione (o come si dice utilizzando il `mauchly.test`, la **sfericità**) dei dati sia costante. Si tratta comunque di una argomento che ormai sta diventando *old fashioned*, soppiantato dalla **modellazione ad effetti misti**. Con l'ANBI ci siamo già occupati nel 2011 di questi argomenti, e vi rimandiamo per un approfondimento alla nostra dispensa:

`http://www.dmi.units.it/~borelli/mixedfrailty/index.html`

Brevemente, possiamo dire che il modello 'migliore' per fittare i dati di Umberto Lucangelo è:

```
library(lme4)
mixedpafi2 <- lmer(pafi ~ time*treatment+(1 | subject)
      +(time+0 | subject))
summary(mixedpafi2)
```

Senza entrare nei dettagli, osserviamo che non c'è alcuna discrepanza tra le stime (i **fixed effect**) ottenuti in `mixedpafi2` ed i coefficienti del `modellolineare`. Al contrario, l'incertezza delle stime si è ridotta, nel senso che gli standard error del *mixed model* sono più contenuti, e conseguentemente i consuntivi *t value* sono più discosti dallo zero.

Dobbiamo dire che una prima difficoltà da superare è già quella della 'stesura' del dataset. Per esercizio, riosservate la figura:

```
xyplot(pafi ~ time | treatment, type = "b",
groups = subject, xlab = "time (hour)",
ylab = expression(PaO[2] / FiO[2]))
```

e con la vostra fantasia immaginatevi un foglio elettronico vuoto innnanzi a voi; provate a pensare come avreste compilato il foglio con i dati dell'esperimento. Fatto ciò, guardate invece come è stato costruito il dataset `percussive`:

```
percussive[1:5,]
percussive[34:38,]
percussive[68:72,]
percussive[276:280,]
```

10.2 Laboratorio intermedio

Angiogenesis is considered a hallmark of multiple myeloma (MM) progression. In the present study, we evaluated the morphological and functional features of endothelial cells (ECs) derived from bone marrow (BM) of patients affected by MM (MMECs). We found that MMECs compared with normal BM ECs (BMECs) showed increased expression of syndecan-1. Silencing of syndecan-1 expression by RNA interference technique decreased in vitro EC survival, proliferation and organization in capillary-like structures. In vivo, in severe combined immunodeficient mice, syndecan-1 silencing inhibited MMEC organization into patent vessels. When overexpressed in human umbilical vein ECs and BMECs, syndecan-1 induced in vitro and in vivo angiogenic effects. Flow-cytometric analysis of MMECs silenced for syndecan-1 expression indicated a decreased membrane expression of vascular endothelial growth factor (VEGF) receptor-2 (VEGFR-2). Immunoprecipitation and confocal analysis showed colocalization of VEGFR-2 with syndecan-1. Absence of nuclear translocation of VEGFR-2 in syndecan-1-knockdown cells together with the shift from perinuclear localization to recycling compartments suggest a role of syndecan-1 in modulation of VEGFR-2 localization. This correlated with an in vitro decreased VEGF-induced invasion and motility. These results suggest that syndecan-1 may contribute to the highly angiogenic phenotype of MMECs by promoting EC proliferation, survival and modulating VEGF-VEGFR-2 signalling.
(Sara L., Ph.D., `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3356559/`)

1. Sara ci ha gentilmente messo a disposizione uno di questi suoi dataset, all'indirizzo:

   `http://www.dmi.units.it/~borelli/anbi/2012/sara_syndecan.xls`

   La domanda è: riuscite a trasformare il foglio di dati Excel di Sara in un dataset che sia adatto all'analisi con il pacchetto `lme4`?

10.3 La questione delle misure ripetute negli studi cross-section

Negli studi di tipo cross-section le misure ripetute sono tipiche (esperimenti in duplicato, in triplicato, ..) qualora si cerchi di '*eliminare*' la variabilità delle misure dovute ad errori connessi alla tecnica dell'esperimento. E' questo il caso ad esempio di Valentina:

> *.. Brevemente, si tratta di un saggio colorimetrico per la rilevazione della attività metabolica cellulare in cui un indicatore di ossidazione/riduzione cambia colore in risposta alla crescita cellulare. Il saggio prevede che i pozzetti contenenti le cellule trattate con l'agente citotossico da valutare vengano lette allo spettrofotometro a due lunghezze d'onda diverse. Ho effettuato il saggio per valutare l'attività citotossica del mio virus su diverse linee cellulari..* (Valentina G., Ph.D.)

```
www <- "http://www.dmi.units.it/~borelli/dataset/resazurin.txt"
resazurin <- read.table(www, header = TRUE)
attach(resazurin)
names(resazurin)
str(resazurin)
head(resazurin)
```

Per semplicità, limitiamo l'analisi alla lunghezza d'onda `lambda == 570`

```
pias <- piastra[lambda == "570"]
asso <- assorbanza[lambda == "570"]
colt <- coltura[lambda == "570"]
infe <- infezione[lambda == "570"]
```

Vediamo il significato di un modello ad effetti misti che descriva i dati:

```
modello1 <- lmer(asso ~ colt + infe + (1 | pias))
summary(modello1)
```

La variabile `colt` ha tre livelli: `BT474`, `MDAMB453` e `medium`, così come la variabile `infe` (`assente`, `moi10` e `moi2`). Le stime dei coefficienti 'si leggono' come abbiamo già visto nel modulo dedicato alla modellazione lineare. In particolare, la assorbanza media sulla linea cellulare `BT474` in assenza di infezione vale 0.649, un numero certamente non nullo in senso statistico (*t-value* 14.019). La possibile variabilità su questa stima è descritta da una

variabile aleatoria normale $N(\mu, \sigma)$ di media nulla ($\mu = 0$) e deviazione standard $\sigma = 0.050$ (`Random effects: pias`).

Per capire bene il senso di questa informazione, consideriamo i valori di assorbanza degli otto pozzetti relativi alla quarta riga della piastra, in cui viene rivelata la linea cellulare `BT474` senza infezione virale:

```
quartariga <- asso[25:32]
```

L'idea è che questi otto valori (raffigurati nel grafico seguente con dei pallini rossi sull'asse delle ascisse) potrebbero essere paragonati a otto numeri estratti a caso dalla variabile aleatoria normale di media 0.649 e deviazione standard 0.050:

```
x <- seq(from = 0.50, to = 0.80, by = 0.001)
y <- dnorm(x, 0.649, 0.050)
plot(x,y, "l")
points (quartariga, rep(0,8), col = "red")
```

In altri termini, gli 8 valori misurati nei diversi pozzetti rappresentano un possibile campione casuale estratto da una popolazione gaussiana $N(0.649, 0.050)$. Non dovremmo dunque stupirci se qualcuno volesse replicare lo studio di Valentina, ed ottenesse invece questi 8 valori:

```
rnorm(8, 0.649, 0.050) + rnorm(8, 0, 0.074)
```

dove 0.074 è la variabilità dei residui, non spiegata dal modello.

```
detach(resazurin)
rm(resazurin)
```

10.3.1 APPROFONDIMENTO: Come '*si scrive*' in un paper tutto ciò?

L'uso dei mixed model, per quanto potenti e versatili, non è ancora così diffuso nella comunità scientifica, e quindi spesso vi è imbarazzo nei *referees* ad affrontare la recensione degli articoli (talvolta, con penose conseguenze per gli estensori del paper). Non vi è ancora una certa unanimità sul *come* si debbano riportare nel paper queste analisi. Da parte nostra, ci sembra che l'articolo *open acces* di Archana Singh-Manoux et al. ([25]), offra un possibile esempio da seguire.

Bibliografia

[1] Agresti A. (2007). *An Introduction to Categorical Data Analysis*. 2nd Edition, Wiley.

[2] Antonini V., Marrazzo A., Kleiner G., Coradazzi M., Ronsisvalle S., Prezzavento O., Ronsisvalle G., Leanza G. (2011). *Anti-amnesic and neuroprotective actions of the sigma-1 receptor agonist (-)-MR22 in rats with selective cholinergic lesion and amyloid infusion.* J. Alzheimers Dis. 24(3), 569-586.

[3] Bland M. (2000). *An introduction to medical statistics*, Oxford University Press.

[4] Bretz F., Hothorn T., Westfall P. (2010). *Multiple Comparisons Using R*, CRC Press.

[5] Busan P., Battaglini P.P., Borelli M., Evaristo P., Monti F., Pelamatti G. (2009). *Investigating the efficacy of paroxetine in developmental stuttering.* Clin. Neuropharmacol. 32(4), 183-188.

[6] Casella G., Berger R. L. (2002). *Statistical Inference*, Duxbury.

[7] Conover W. J. (1999). *Practical Nonparametric Statistics*, Wiley.

[8] Conover W. J., Johnson M. E., Johnson M. M. (1981). *A Comparative Study of Tests for Homogeneity of Variances, with Applications to the Outer Continental Shelf Bidding Data.* Technometrics, 23(4), 351-361.

[9] Crawley M. J. (2005). *Statistics: An Introduction using R*, Wiley.

[10] Crawley M. J. (2007). *The R Book*, Wiley.

[11] Curran-Everett D., Benos D.J. (2004). *Guidelines for reporting statistics in journals published by the American Physiological Society.* Adv Physiol Educ, 28(3), 85-87. http://advan.physiology.org/content/28/3/85.long

[12] Drăghici S. (2011). *Statistics and Data Analysis for Microarrays Using R and Bioconductor*, Chapman and Hall/CRC.

[13] Espa G. Micciolo R. (2012). *Analisi esplorativa dei dati con R*, Apogeo.

[14] Faraway J. (2004). *Linear Models with R*, Chapman Hall/CRC.

[15] Faraway J. (2005). *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*, Chapman Hall/CRC.

[16] Faraway J. (2010). *Changes to the Mixed Effects Model chapters in ELM*, `http://www.maths.bath.ac.uk/∼jjf23/ELM/mixchange.pdf`.

[17] Invernizzi S., Rinaldi M., Sgarro A. (2000). *Moduli di Matematica e Statistica*, Zanichelli.

[18] Lucangelo U., Accardo A., Bernardi A., Ferluga M., Borelli M., Antonaglia V., Riscica F., Zin W.A. (2010). *Gas distribution in a two-compartment model ventilated in high-frequency percussive and pressure-controlled modes.* Intensive Care Med. 36(12), 2125-31.

[19] Lucangelo U., Zin W.A., Fontanesi L., Antonaglia V., Peratoner A., Ferluga M., Marras E., Borelli M., Ciccolini M., Berlot G. (2011). *Early Short-Term Application of High-Frequency Percussive Ventilation Improves Gas Exchange in Hypoxemic Patients.* Respiration, DOI: 10.1159/000334403.

[20] Lucangelo U., Bernabè F., Vatua S., Degrassi G., Villagrà A., Fernandez R., Romero P.V., Saura P., Borelli M., Blanch L. (2008). *Prognostic value of different dead space indices in mechanically ventilated patients with acute lung injury and ARDS.* Chest, 133(1), 62-71.

[21] Montomoli C., Nichelatti M. (2008). *La statistica bayesiana in medicina - parte II: le principali applicazioni e l'inferenza.* Giornale italiano di nefrologia, 25(4), 422-431.

[22] Parmigiani G., Inoue L. (2009). *Decision Theory: Principles and Approaches.* Wiley

[23] R Development Core Team (2009). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria.

[24] Rohatgi V. K. (1984). *Statistical Inference*, Wiley.

[25] Singh-Manoux A., Kivimaki M., Glymour M.M., Elbaz A., Berr C., Ebmeier K.P., Ferrie J.E., Dugravot A. (2011). *Timing of onset of cognitive decline: results from Whitehall II prospective cohort study.* BMJ, 344:d7622.

[26] Weisberg S. (2005). *Applied Linear Regression*, Wiley.

[27] West B.T., Welch K.B., Galecki A.T. (2007). *Linear Mixed Model*, Chapman Hall/CRC.