

# **Metodi di Biostatistica**

**per le Scuole di Dottorato dell'Università Magna Græcia di Catanzaro**

**Massimo Borelli**

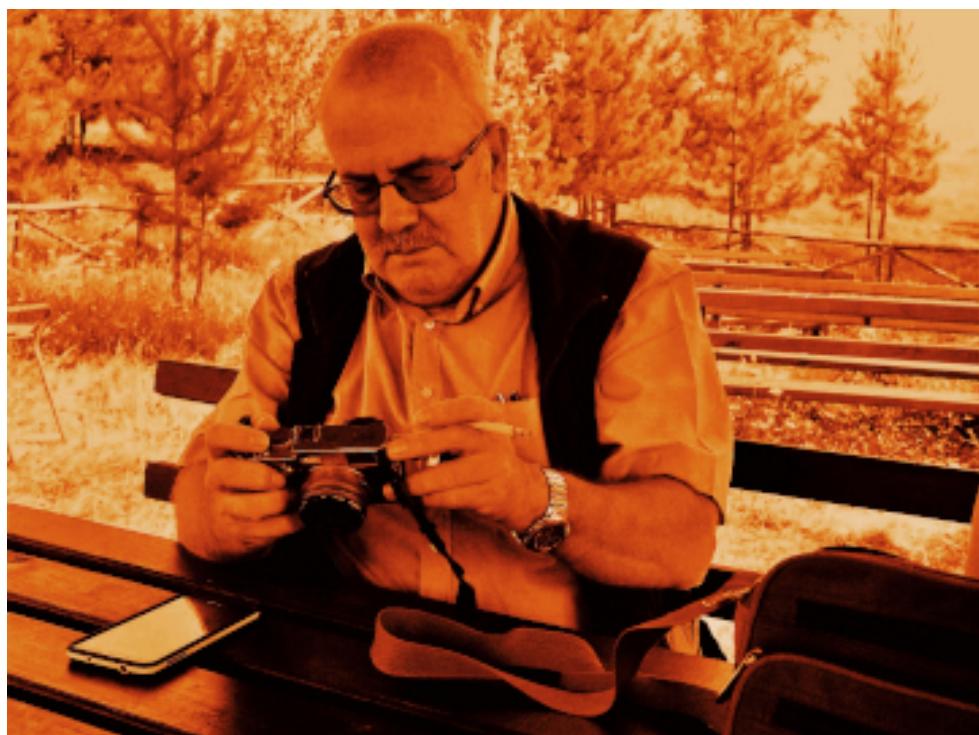
Copyright © 2019 Massimo Borelli

PUBLISHED BY PUBLISHER

<http://www.biostatisticaumg.it>

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*First printing, June 2019*



il professore Gianni Morrone





## Prefazione

### La vera finalità di questo libro

La finalità di questo libro è unica: vogliamo rendere omaggio al professor Gianni Morrone, che dal 2013 diresse le Scuole di Dottorato "Life sciences and technologies" dell'Università Magna Græcia di Catanzaro, dedicandosi in particolare alla sezione in Oncologia Molecolare, Traslazionale e delle Tecnologie Medico-Chirurgiche Innovative. Gianni è stato uno scienziato con moltissime qualità ed io non sono di certo la persona più adatta ad enumerarle, perché ho conosciuto Gianni appena nel 2012; ma quello che mi sento qui di ricordare è la passione che Gianni dimostrava di dedicare agli specializzandi, ai dottorandi, ai giovani ricercatori, cercando di assicurare loro le migliori attenzioni su tutti i punti che coinvolgessero una didattica di qualità, una ricerca di base di qualità, ed alle ricadute concrete della ricerca in termini di cure e di metodologie diagnostiche innovative.

Questo volume che iniziate a leggere è poco più di nulla: è un sassolino arancione che si unisce al tesoro di gemme preziose che egli ha lasciato alle sue allieve ed ai suoi allievi. In realtà il colore arancione è un richiamo al Calvados, uno dei liquori preferiti dal professore; in esergo ad ogni capitolo troverete delle notevoli fotografie, scattate da Debora De Bartolo, anatomopatologa e dottore di ricerca del XXX ciclo della nostra UMG PhD Schools; naturalmente la ringrazio di questa grande cortesia.

Oltre ai ricordi personali ed alle belle foto, il libro raccoglie ed illustra una certa quantità di metodi statistici di tipo descrittivo, esplorativo, inferenziale e di modellazione che possono essere di interesse per ogni Dottoranda/o che si occupi di questioni attinenti alle Scienze della Vita: dall'ingegnere clinico, al medico, al farmacologo, al biologo, al veterinario, al chimico, al biotecnologo, al tecnologo farmaceutico. È singolare infatti come la statistica sia una scienza che – per usare un concetto caro a Claudio Magris – è al tempo stesso canto e controcanto: una scienza esatta fondata sulla matematica che risolve questioni proprie, che però accompagna in tono sommesso le scoperte delle altre scienze, incorniciandole ed impreziosendole (e traendo da queste nuovi spunti e nuovi *incipit*).

Il libro è confezionato come un e-book, che certamente si potrebbe stampare, ma che per propria natura si legge meglio seduti davanti al computer, connessi alla rete per approfondire i vari punti attraverso dei rimandi evidenziati con il simbolo web:



Questo libro possiede un sito web di riferimento, dove tra l'altro sono archiviati i dataset che andremo via via a studiare: <http://www.biostatisticaumg.it>. Il libro invece è stato composto in L<sup>A</sup>T<sub>E</sub>X, sfruttando – con gratitudine – un template di Mathias Legrand, disponibile all'indirizzo <https://www.latextemplates.com/cat/books>.

Alla fine di ogni capitolo viene riportata una selezione di esercizi, la gran parte dei quali sono tratti da alcuni testi di statistica che risultano essere fondamentali per quelli studiosi che, come me, sono affascinati da questa disciplina. Queste attività si devono svolgere in autonomia, sedendosi al proprio computer ed utilizzando i software di cui vi andiamo a parlare immediatamente.

### La scelta del software: Python oppure R?

Viene il momento in cui si desidera pubblicare la propria ricerca. Ed è quello il momento in cui ci si deve ricordare che i dati ottenuti nel vostro esperimento sono stati 'centrali' per il successo della vostra pubblicazione; è giusto quindi che essi siano parte integrante del paper, che siano liberamente consultabili, che il software e gli algoritmi che avete utilizzato per analizzarli siano descritti con sufficiente dettaglio e, possibilmente, con il 'codice sorgente' in modo da poter assicurare la riproducibilità [32] dell'analisi statistica. Non seguendo queste indicazioni automaticamente vengono poste delle restrizioni alla vostra ricerca nei confronti della comunità scientifica; restrizioni che potrebbero persino influire sulla decisione finale della rivista, soprattutto se si trattasse di uno dei cosiddetti *top journal* nel vostro settore di ricerca.



L'EMBO, con sede ad Heidelberg, è un'organizzazione che promuove l'eccellenza nella ricerca scientifica: il professor Morrone vi trascorse molti periodi di ricerca, e molti nostri studenti di Dottorato continuano a recarsi lì. Prima di pubblicare la vostra ricerca, ricordatevi di leggere con attenzione le loro Linee Guida: <https://emboj.embopress.org/authorguide>

Naturalmente, ogni laboratorio sceglie ed utilizza i software che preferisce; ma se vogliamo assicurare la fruizione al pubblico più ampio dei nostri dati e delle nostre analisi conviene adottare – o quantomeno, affiancare al proprio software – quella che è da un paio di decenni diventata la *lingua franca* della statistica: è il linguaggio R. Si tratta di un ambiente di programmazione *open source*, nato verso la fine degli anni '90 presso l'Università di Auckland [55] e particolarmente adatto alla statistica ed alla rappresentazione grafica dei dati, che può essere scaricato gratuitamente da uno dei numerosi *mirror* del CRAN (the Comprehensive R Archive Network) sparsi nel mondo e nel cloud. R funziona praticamente su tutti i sistemi operativi: UNIX/Linux, Windows e MacOS. Se volete fare qualche semplice calcolo a scopo didattico con il vostro tablet, non potendo di norma scaricare applicativi che non siano delle app, potete utilizzare gli *snippets* di RDrr: <https://rdrr.io/snippets/>. La maggior parte degli utenti, invece di lavorare direttamente nella Console di R, preferisce utilizzare l'ambiente di sviluppo integrato (IDE, Integrated Development Environment) denominato R Studio. I principianti talvolta trovano 'più amichevole' – e bastante per i loro scopi – l'interfaccia grafica denominata R Commander. C'è anche da aggiungere che, in rete, si reperiscono numerose risorse per imparare a conoscere ed usare R: e-book, blog, video tutorial e persino MOOC gratuiti da parte di prestigiose università.



Per scaricare la vostra copia di R accedete al mirror a voi più comodo: <https://cran.r-project.org/mirrors.html>. Dopo aver installato R, potete scaricarvi R Studio dal sito <https://www.rstudio.com/>. Se lo volete, potrete dare un'occhiata ad R Commander, installandolo da <https://www.rcommander.com/>. Per quanto riguarda la documentazione, un libro introduttivo gratuito è offerto da Hadley Wickham e Garrett Grolemund *R for data science* [60], consultabile dal sito <https://r4ds.had.co.nz/>. Oppure, il libro di Kim Seefeld ed Ernst Linder, *Statistics Using R with Biological Examples*, disponibile all'indirizzo [https://cran.r-project.org/doc/contrib/Seefeld\\_StatsRBio.pdf](https://cran.r-project.org/doc/contrib/Seefeld_StatsRBio.pdf). Tra le pagine web utili ricordiamo:

- [http://ncss-tech.github.io/stats\\_for\\_soil\\_survey/chapters/](http://ncss-tech.github.io/stats_for_soil_survey/chapters/)
- <http://www.sthda.com/english/wiki/r-software>
- Quick-R, <https://www.statmethods.net/>

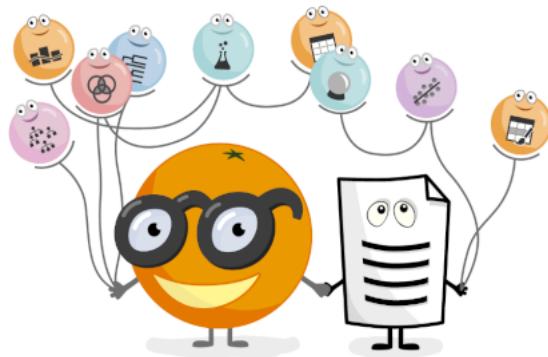
Molti video tutorial vengono pubblicati su YouTube; trovate se qualcosa vi piace con la query [https://www.youtube.com/results?search\\_query=R+tutorial](https://www.youtube.com/results?search_query=R+tutorial). Se poi volete conoscere altri dettagli sulla 'vita' di R potete leggere l'articolo di Carlos Alberto Gómez Grajales, *Created by statisticians for statisticians: How R took the world of statistics by storm*: <http://www.statisticsviews.com/view/index.html>. Se invece volete scoprire perché RDRR si chiama così, guardatevi un minutino di cartone animato dei Simpsons: [https://www.youtube.com/watch?v=gSFd\\_2oJgak](https://www.youtube.com/watch?v=gSFd_2oJgak).

Agli antipodi della Nuova Zelanda, in Slovenia, sempre verso la fine degli anni '90 vedeva la luce [18] una *library* di algoritmi di *machine learning* e di procedure collegate, quali il pre-processing, il campionamento, e la manipolazione di dati. Gli autori, dopo alcuni tentativi iniziali in C++ sono passati a Python, un linguaggio intuitivo e tanto semplice da imparare quanto robustamente impiegato nella didattica, nella ricerca e nell'industria, dotandolo di un'interfaccia visuale estremamente intuitiva; era nato Orange, <https://orange.biolab.si/>. Orange oggi è un sistema open-source di visualizzazione dei dati, di machine learning e di data mining, che può venir usato sia come libreria Python che come front-end di programmazione visuale.

## Data Mining Fruitful and Fun

Open source machine learning and data visualization for novice and expert.  
Interactive data analysis workflows with a large toolbox.

[Download Orange](#)



**web**

La graziosa immagine qui sopra è tratta dal sito di Orange: <https://orange.biolab.si/>. Nello stesso sito troviamo un'ampia pagina di documentazione, che tra l'altro comprende i video tutorial (<https://www.youtube.com/channel/UC1KKWBe2SCAEyv7ZNGhIe4g>) condotti da Ajda Pretnar dell'Università di Lubiana.

## Imparare la biostatistica sui dati reali

In apertura di libro voglio dirvi che uno dei miei maestri, il professor Sergio Invernizzi, raccomanda sempre di studiare gli esempi che siano tratti dalla realtà. In questo volume accadrà proprio questo: i dati di cui ci occuperemo riguardano per lo più ricerche che sono state condotte da dottorande e dottorandi delle differenti Scuole dirette dal professor Gianni Morrone presso l'Università *Magna Graecia* di Catanzaro. Per farvi qualche esempio, il dataset ZNF521 di Emanuela Chiarella e di Stefania Scicchitano raccoglie la sequenza della proteina Zinc Finger, principale oggetto di studio del laboratorio di hematopoiesi molecolare e biologia delle cellule staminali fondato dal professor Morrone[7]; dal medesimo laboratorio provengono le p.c.r. real time quantitative del dataset hedgehog di Valeria Lucchino. Teresa Calimeri invece è l'autrice del dataset mieloma in cui ci si occupa di come progredisca *in vivo* questa patologia su un particolare scaffold tridimensionale. I dataset raul di Annalisa Di Cello e salpinge di Roberta Venturella si occupano di questioni

ginecologiche, mentre nel campo della anatomia patologica Debora De Bartolo ha studiato il dataset *tossicologia*. Abbiamo anche un esempio in ambito endocrinologico, il dataset *gdm*, che riguarda il diabete gestazionale studiato da Eusebio Chieffari.

Vi sono anche dataset che provengono da lavori condotti presso l'Università di Trieste: in ambito odontoiatrico *analgesia*, anestesiologico *percussive*, ginecologico *roma*, pediatrico *epilessie*, ematologico *cholesterol*. C'è anche un dataset futile ma reale, *fresher*, che raccoglie dati biometrici di una coorte di miei ex-studenti del primo anno di medicina.

Infine si utilizzano anche dei dataset che hanno avuto un ruolo 'storico' nello studio della statistica: ad esempio i fiorellini *iris*, l'inquinamento in *airquality*, le eccezioni di *anscombe* ed i dati degli esperimenti condotti da Student ad inizio '900 che lo hanno condotto a formulare il celebre test. C'è un unico dataset che ha dei dati completamente *fake*: riguarda il peso di due fantastiche gemelle ballerine di da-da-un-pa. L'ho inventato di sana pianta solo perché mi faceva ridere.

# Indice

	<b>Prima Parte</b>
<b>1</b>	<b>Descrivere i dati .....</b> <span style="float: right;">15</span>
1.1	<b>Misure di tendenza centrale</b> <span style="float: right;">16</span>
1.1.1	Facciamo conoscenza con R ..... <span style="float: right;">17</span>
1.2	<b>Misure di dispersione</b> <span style="float: right;">21</span>
1.2.1	Misure di dispersione con R ..... <span style="float: right;">23</span>
1.3	<b>Descrivere i dati nei design cross-section</b> <span style="float: right;">25</span>
1.3.1	Facciamo conoscenza con Orange ..... <span style="float: right;">26</span>
1.3.2	Il boxplot con R ..... <span style="float: right;">27</span>
1.3.3	Due presupposti fondamentali ..... <span style="float: right;">28</span>
1.4	<b>Descrivere i dati nei design a misure ripetute</b> <span style="float: right;">29</span>
1.4.1	Calcolare i dati di relative gene expression con R ..... <span style="float: right;">29</span>
1.4.2	L'equazione più pericolosa, parte prima ..... <span style="float: right;">30</span>
1.4.3	Sovrapporre due grafici con R ..... <span style="float: right;">32</span>
1.5	<b>Esercizi ed attività di approfondimento</b> <span style="float: right;">32</span>
<b>2</b>	<b>Simulare i dati sperimentali con R .....</b> <span style="float: right;">35</span>
2.1	<b>R è un linguaggio di programmazione</b> <span style="float: right;">35</span>
2.1.1	Usare il ciclo for e la decisione if ..... <span style="float: right;">36</span>
2.1.2	Creare le proprie funzioni ..... <span style="float: right;">40</span>
2.2	<b>Gli eventi casuali con R</b> <span style="float: right;">41</span>
2.2.1	Randomizzare i pazienti ..... <span style="float: right;">42</span>
2.2.2	Simulare una mutazione genica ..... <span style="float: right;">45</span>

<b>2.3</b>	<b>Le variabili aleatorie con R</b>	<b>47</b>
2.3.1	Jacob Bernoulli e gli eventi dicotomici . . . . .	47
2.3.2	L'equazione più pericolosa, parte seconda . . . . .	52
2.3.3	Siméon Poisson e la conta degli eventi . . . . .	53
2.3.4	Carl Gauss, o della normalità . . . . .	54
2.3.5	L'equazione più pericolosa, parte terza (ed ultima) . . . . .	60
<b>2.4</b>	<b>Riepilogone del capitolone</b>	<b>61</b>
<b>2.5</b>	<b>Esercizi ed attività di approfondimento</b>	<b>62</b>

## II

## Seconda Parte

<b>3</b>	<b>C'era una volta il p-value . . . . .</b>	<b>69</b>
3.1	<b>Il risultato è statisticamente significativo. E dunque?</b>	69
3.2	<b>Come nacque il t test</b>	71
3.2.1	il comando <code>t.test</code> di R . . . . .	72
3.2.2	il test t di Student tra due campioni . . . . .	74
3.3	<b>Ascesa e declino del p-value</b>	76
3.4	<b>La retta di regressione</b>	78
3.4.1	Covarianza e correlazione . . . . .	78
3.4.2	L'idea di Francis Galton . . . . .	79
3.5	<b>Esercizi ed attività di approfondimento</b>	81
<b>4</b>	<b>Che cos'è un modello lineare . . . . .</b>	<b>83</b>
4.1	<b>I dettagli da conoscere</b>	84
4.1.1	I residui di un modello lineare . . . . .	86
4.1.2	La devianza di un modello lineare . . . . .	87
4.1.3	La componente aleatoria di un modello lineare . . . . .	88
4.1.4	Il modello nullo è importante . . . . .	89
4.1.5	Hirotugu Akaike, un nome da ricordare per sempre . . . . .	91
4.1.6	La diagnostica del modello lineare . . . . .	94
4.1.7	Lineare non è sinonimo di rettilineo . . . . .	97
4.1.8	Anche il t test è un modello lineare . . . . .	98
4.2	<b>Ancova: unire i predittori numerici ai fattori</b>	100
4.3	<b>Facciamo il punto della situazione</b>	102
4.4	<b>Anova: la generalizzazione del t test</b>	104
4.5	<b>La meta finale: condurre un'analisi multivariabile</b>	108
4.5.1	Interpretare il modello minimale adeguato . . . . .	111
4.6	<b>Riassuntone del capitolone</b>	112
4.7	<b>Perle di saggezza: tecniche di linearizzazione</b>	113
4.7.1	Il modello iperbolico. . . . .	113
4.7.2	Il modello esponenziale. . . . .	114
4.7.3	Il modello <i>maxima function</i> . . . . .	115
4.7.4	Il modello potenza. . . . .	116
4.7.5	Il modello logistico. . . . .	118
4.8	<b>Esercizi ed attività di approfondimento</b>	120

<b>5</b>	<b>I modelli lineari generalizzati</b>	<b>121</b>
5.1	<b>I dettagli da conoscere</b>	<b>122</b>
5.1.1	La funzione di collegamento . . . . .	122
5.1.2	Ad ogni variabile aleatoria, la sua funzione di link . . . . .	123
5.1.3	Interpretare una regressione logistica . . . . .	123
5.1.4	Problemi con lo standard error . . . . .	126
5.1.5	La sovradispersione . . . . .	128
5.2	<b>La meta finale: valutare l'accuratezza del modello logistico</b>	<b>129</b>
5.2.1	La curva ROC . . . . .	132
5.3	<b>Esercizi ed attività di approfondimento</b>	<b>133</b>

### III

## Terza Parte

<b>6</b>	<b>Misure ripetute</b>	<b>137</b>
6.1	<b>Lo strano caso delle gemelle Alice ed Ellen</b>	<b>138</b>
6.1.1	Tutta colpa di Student? . . . . .	139
6.2	<b>Il modello lineare ad effetti misti</b>	<b>141</b>
6.2.1	Un esempio introduttivo . . . . .	141
6.2.2	Come impostare il dataset . . . . .	142
6.2.3	L'analisi non appropriata . . . . .	143
6.2.4	Interpretare il modello ad effetti misti . . . . .	144
6.3	<b>Intermezzo ectopico, inattuale ed (abbastanza) opzionale</b>	<b>147</b>
6.4	<b>La selezione di un modello ad effetti misti</b>	<b>148</b>
6.4.1	Selezione con il Criterio di Informazione di Akaike . . . . .	148
6.4.2	Selezione con il 'parametric bootstrap' . . . . .	149
6.5	<b>La Anova repeated measures, i modelli misti e la qRT-PCR</b>	<b>151</b>
6.6	<b>Esercizi ed attività di approfondimento</b>	<b>153</b>
<b>7</b>	<b>Sopravvivenza</b>	<b>155</b>
7.1	<b>I dati di tipo Time-to-Event</b>	<b>155</b>
7.1.1	Rischio competitivo ed i modelli multistato . . . . .	156
7.2	<b>La cornice matematica</b>	<b>157</b>
7.2.1	Le definizioni . . . . .	157
7.2.2	L'approccio non parametrico . . . . .	159
7.2.3	L'approccio parametrico . . . . .	159
7.3	<b>Le curve di sopravvivenza con R</b>	<b>162</b>
7.3.1	L'approccio non parametrico . . . . .	163
7.3.2	L'approccio parametrico . . . . .	164
7.4	<b>I modelli semiparametrici</b>	<b>166</b>
7.4.1	L'aspetto matematico . . . . .	166
7.4.2	Interpretare un modello di Cox . . . . .	168
7.4.3	Selezione del modello . . . . .	168
7.5	<b>Esercizi ed attività di approfondimento</b>	<b>169</b>

<b>Bibliografia</b>	.....	<b>171</b>
<b>Articoli</b>		<b>171</b>
<b>Libri</b>		<b>173</b>
<b>Risorse Web</b>		<b>174</b>
<b>Index</b>	.....	<b>175</b>



# Prima Parte

<b>1</b>	<b>Descrivere i dati .....</b>	<b>15</b>
1.1	Misure di tendenza centrale	
1.2	Misure di dispersione	
1.3	Descrivere i dati nei design cross-section	
1.4	Descrivere i dati nei design a misure ripetute	
1.5	Esercizi ed attività di approfondimento	
<b>2</b>	<b>Simulare i dati sperimentali con R ....</b>	<b>35</b>
2.1	R è un linguaggio di programmazione	
2.2	Gli eventi casuali con R	
2.3	Le variabili aleatorie con R	
2.4	Riepilogone del capitolo	
2.5	Esercizi ed attività di approfondimento	





## 1. Descrivere i dati

**Problema 1.1** ... abbiamo iniziato a scrivere il nostro primo articolo, e ci è stato chiesto di preparare la classica Tabella 1, quella in cui si deve fare la descrittiva del campione. E non sappiamo bene come si debba procedere ...

Iniziamo questo libro affrontando un problema che sembra presentare molte possibili soluzioni, talune anche in apparente contraddizione. Vi proponiamo di passare in rassegna alcuni esempi che sono stati tratti da articoli pubblicati dai nostri Dottori di Ricerca della Università Magna Græcia di Catanzaro.

1. Roberta Venturella [57] nell'articolo intitolato *3 to 5 years later: long-term effects of prophylactic bilateral salpingectomy on ovarian function* si occupa di comparare la funzione ovarica misurata con OvAge<sup>TM</sup> raffrontandala con le età delle pazienti arruolate nello studio adottando la convenzione Mean Values  $\pm$  SD. Ad esempio, l'età delle pazienti al momento dell'intervento chirurgico era  $45.85 \pm 2.40$
2. Emanuela Chiarella [14] chiarisce un importante ruolo biologico della proteina Zinc Finger nel paper *ZNF521 Represses Osteoblastic Differentiation in Human Adipose-Derived Stem Cells*, e per descrivere gli esperimenti preferisce adottare la convenzione Means + SD.
3. Invece se leggiamo il paper di Maria Teresa De Angelis, *Short-term retinoic acid treatment sustains pluripotency and suppresses differentiation of human induced pluripotent stem cells* [17], vediamo che la convenzione adottata è quella di mean  $\pm$  standard error of the mean (SEM).
4. Maria Vittoria Caruso [13] si occupa nel suo paper *Influence of IABP-Induced Abdominal Occlusions on Aortic Hemodynamics: A Patient-Specific Computational Evaluation* di modelli computazionali fluidodinamici, citando tra l'altro un lavoro di altri studiosi e riportando che *Furthermore, they also discovered that the distance between LSA and CT was  $241 \pm 23\text{mm}$ ,* senza fornire al lettore precise indicazioni sul ruolo - dato per scontato - delle due quantità.
5. Annalisa Di Cello [19] per descrivere i livelli degli isoenzimi LDH1 ed LDH3 nei sarcomi uterini nell'articolo *A more accurate method to interpret lactate dehydrogenase isoenzymes' results in patients with uterine masses* utilizza la convenzione Mean [Median].

6. Infine, Paolo Zaffino [63], *Radiotherapy of hodgkin and non-hodgkin lymphoma: a nonrigid image-based registration method for automatic localization of prechemotherapy gross tumor volume*, nelle analisi delle immagini tomografiche descrive certi coefficienti di similarità con il sistema mediana  $\pm$  quartili.

I pochi esempi che abbiamo presentato mostrano dunque un'ampia varietà di scelte operative, e si potrebbe chiedere se tali scelte si collochino in un ambito di preferenza soggettiva, o addirittura arbitraria, oppure se esista una scelta 'più giusta' delle altre. Vedremo che la risposta in assoluto manca, ma se terremo in considerazione il tipo di ricerca che stiamo effettuando avremo modo di ridurre sensibilmente questa apparente arbitrarietà, distinguendo se si tratti di analisi dei dati condotti in un design sperimentale di tipo cross-section (come ad esempio accade in uno studio retrospettivo sulla prevalenza di una patologia in determinati gruppi di pazienti) oppure se abbiamo effettuato delle misure ripetute (come accade nei replicati biologici o nei replicati tecnici, o negli studi di proliferazione cellulare). Dovremo inoltre valutare di quali tipi di dato statistico ci stiamo occupando (ad esempio percentuali di soggetti o misure con strumenti di laboratorio). Questo capitolo richiamerà gli strumenti di cui possiamo disporre per fornire, alla fine del capitolo, una risposta definitiva al problema.

## 1.1 Misure di tendenza centrale

Conosciamo senza dubbio i concetti di **media**, **mediana** e **moda**: sono queste le **misure di tendenza centrale** (dette anche **indici di posizione**) più utilizzate nella descrizione dei dati provenienti dai nostri esperimenti.



La pagina Central Tendency di Wikipedia fornisce un rapido richiamo a numerose altre misure di tendenza centrale: [https://en.wikipedia.org/wiki/Central\\_tendency](https://en.wikipedia.org/wiki/Central_tendency).

Molti autori raccomandano di utilizzare le misure di tendenza centrale anche in ragione del tipo di dati che si stanno prendendo in considerazione. Tuttavia in letteratura sussistono varie possibili classificazioni riguardanti il 'tipo' di dato statistico. Ad esempio, alcuni autori [6] preferiscono distinguere i dati **qualitativi** (ad esempio quando un paziente può essere assegnato senza confusione a gruppi disgiunti, tra i quali non è necessario che intercorra una relazione di tipo numerico: per esempio, il genere: femmina o maschio; il gruppo sanguigno A, B, AB oppure 0), i dati **quantitativi discreti** (tipicamente, elementi che vengono contati ad uno ad uno con numeri interi) e **quantitativi continui** (evidentemente, tutti i casi rimanenti). Tuttavia, vi propongo di riflettere su un quesito come questo.

**Quesito 1.1.1** Se il donatore di sangue Massimo Borelli riceve nel referto l'indicazione che i trigliceridi ammontano ad 89, e non 89.0 mg/dL, ci conviene interpretare questo dato quantitativo come discreto o continuo?

Per continuare, riflettiamo sulla scala A.S.A. che valuta il rischio anestesiologico in base alle condizioni del paziente, e che assume i valori I, II, III, IV e V (modificabili ciascuno con la E di 'emergency'): si tratta di una scala solamente qualitativa, oppure l'ordine conta? Ecco perché alcuni autori [20] considerano 1) i dati categorici, 2) i dati ordinali (come appunto il rischio anestesiologico) e 3) i dati quantitativi a loro volta suddivisi in continui e discreti.

Altri autori ancora [29] introducono il concetto di **scale di misura**, suddividendole in quattro tipi fondamentali: la scala **nominale** (ad esempio nell'epidemiologia, parliamo di individui Suscettibili, Infettivi e Rimossi), la scala **ordinale**, quella **intervallare** (come ad esempio la scala di temperatura Celsius, che ha uno zero convenzionale ed in cui ha senso solamente parlare di differenze di temperatura) ed infine la scala **a rapporti** (come ad esempio la scala di temperatura Kelvin, che ha

uno zero assoluto ed in cui ha senso dire che un corpo a  $200^{\circ}K$  ha una temperatura doppia di un corpo a  $100^{\circ}K$ ).

**Quesito 1.1.2** L'indice di massa corporea BMI (<https://meshb.nlm.nih.gov/record/ui?ui=D015992>) individua negli adulti quattro categorie: below 18.5 (underweight), 18.5-24.9 (normal), 25.0-29.9 (overweight), 30.0 and above (obese). Di che tipo di dato stiamo parlando?

Vediamo ora come i nostri software affrontano, in maniera piuttosto draconiana, la questione.

### 1.1.1 Facciamo conoscenza con R

#### Importare un dataset esistente

Il software R opera una classificazione molto basilare, utilizzando il nome **fattore** (suddiviso in due o più **livelli**) per i dati di tipo categorico, elencati in diversi gruppi; questi ultimi possono essere ordinati secondo il criterio prescelto dall'utente – altrimenti, di default si utilizza l'ordine alfabetico. Dunque in R il primo tipo di variabili si chiama **factor**, ed i suoi due o più **levels** possono eventualmente essere **ordered** dall'utente (vedremo che Orange invece chiama **categorical** questo tipo di dati). Entrambi i software semplicemente utilizzano la dizione **numeric** per i casi rimanenti.

Proviamo ad utilizzare per la prima volta R Studio, o se preferite R. Dal menu *File* apriamo un nuovo **script** (nuovo documento) e prendiamo conoscenza con il dataset più petalo che esista al mondo. Si tratta del dataset **iris** utilizzato da uno dei padri della statistica, Ronald Fisher [24], che lo aveva tratto da un lavoro di Edgar Anderson [3]. Si tratta semplicemente di un dataset didattico, che viene riportato in tutti i software – anche in Orange, ovviamente – per esercitarsi e per testare la funzionalità di nuove procedure.

Nella finestra del nuovo script, dopo aver premuto per comodità un poche di volte il tasto invio ed essersi posizionati circa a metà della finestra, digitiamo:

```
iris
```

Ora selezioniamo il testo, e 'schiacciamo' il pulsante *Run* (i vecchietti come me ed i nerd invece di usare il mouse o il touchpad trovano molto comodo e rapido utilizzare i comandi direttamente dalla tastiera). Scorrendo la finestra della Console capiamo che si tratta di un dataset di 150 righe e 5 colonne, denominate rispettivamente Sepal.Length, Sepal.Width, Petal.Length, Petal.Width e Species. Le prime quattro colonne contengono dati numerici, mentre l'ultima colonna elenca tre diversi gruppi (le specie, appunto) di fiori. Il comando **str** serve proprio per descrivere la struttura del dataset specificando quante righe vi sono (obs., ossia **osservazioni**), quante colonne (ossia, **variabili**) e di che tipo esse siano, numeric oppure factor. Vediamolo:

```
str(iris)
```

```
'data.frame': 150 obs. of 5 variables:
 \$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 \$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 \$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 \$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 \$ Species     : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 ...
```

#### Determinare le misure di tendenza centrale

Iniziamo con le variabili numeriche. Se vogliamo considerare Petal.Length – la lunghezza dei petali – e calcolarne la media oppure la mediana, abbiamo le funzioni **mean** e **median**. Prima però dobbiamo 'attivare' il dataset all'interno del percorso di ricerca dei file con il comando **attach**:

```
attach(iris)
mean(Petal.Length)
median(Petal.Length)

> attach(iris)
> mean(Petal.Length)
[1] 3.758
> median(Petal.Length)
[1] 4.35
```

Passiamo ora alle variabili categoriche, i fattori; nel nostro caso, abbiamo solamente `Species`. Effettivamente R dispone di una funzione `mode`, che però non calcola la moda di una variabile, ma ne indica il tipo (`numeric`, oppure `factor`). Invece useremo la funzione `table`, che crea una tabella di **frequenze assolute** della variabile in esame:

```
table(Species)

> table(Species)
Species
  setosa versicolor virginica
    50          50          50
```

**Vocabolario 1.1 — dataset bilanciato.** Il dataset `iris` si dice **bilanciato** (in inglese **balanced**) in quanto i dati dei tre gruppi presi in esame sono stati osservati rispettando in ciascuno la medesima frequenza assoluta.

Supponiamo adesso di voler conoscere le medie, o le mediane, delle lunghezze dei petali distinguendoli tra le tre diverse specie, `setosa`, `versicolor` e `virginica`; useremo la funzione `tapply` che consente di applicare una funzione (per esempio `mean`, oppure `median`) creando una semplice tabella:

```
tapply(Petal.Length, Species, mean)
tapply(Petal.Length, Species, median)

> tapply(Petal.Length, Species, mean)
  setosa versicolor virginica
    1.462      4.260      5.552
> tapply(Petal.Length, Species, median)
  setosa versicolor virginica
    1.50      4.35      5.55
```

### Manipolare un dataset

Nel gergo dell'informatica, tutte quelle azioni che coinvolgono il riordino dei dati, anche attraverso la selezione di certe righe o di certe colonne, aggiungendone, o spostandole, o togliendone alcune di esse, viene indicato con il termine generico di **manipolazione** di un dataset. E dunque, prima di proseguire con i concetti della statistica di base, vi suggeriamo di provare a fare in maniera autonoma alcuni esercizi di manipolazione dei dataset: continuiamo dapprima a lavorare con i graziosi `iris`, e poi ci occuperemo di ozono e della qualità dell'aria che respiriamo.

**Esercizio 1.1** Cercate di capire da soli cosa succede se eseguite, ad una ad una, le istruzioni qui di seguito elencate. In particolare, cercate di cogliere il senso delle funzioni `head`, `tail` e `names`.

```
iris[1,]
```

```
iris[1:6,]
head(iris)
iris[145:150,]
tail(iris)
iris[,1]
iris[,c(3,4,5)]
iris[,3:5]
names(iris)
```

Adesso continuiamo ad investigare sui livelli di un fattore e sul tipo di variabile in esame.

**Esercizio 1.2** Provate ad eseguire, ad una ad una, le seguenti istruzioni. Scoprirete alla fine due importanti **costanti booleane** di R.

```
levels(Species)
levels(Species)[2]
is.numeric(Species)
is.factor(Species)
is.factor(Petal.Length)
```

Vediamo ora come possiamo ordinare i dati in R oppure selezionarli in base ai nostri criteri (in pratica, quello che nei fogli elettronici come MS Excel™ viene definito 'applicare un filtro ai dati').

**Esercizio 1.3** Provate ad eseguire, ad una ad una, le seguenti istruzioni e discutetene l'output.

```
iris[order(Sepal.Length), ]
iris[order(Sepal.Width), ]
iris[order(Sepal.Length, Sepal.Width), ]
iris[rev(order(Sepal.Length)), ]
iris[Species == "virginica",]
iris[(Species == "virginica") & (Sepal.Length == 6.3),]
iris[(Species == "virginica") & (Sepal.Length != 6.3),]
```

Adesso approfondiamo l'utilizzo dei comandi `attach` e `detach`. Proviamo a 'staccare' il dataset `iris` e vediamo come si comporta la funzione `length` che valuta la dimensione di un dataset, o di una variabile.

```
length(iris)
length(Petal.Length)
detach(iris)
length(iris)
length(Petal.Length)

> length(iris)
[1] 5
> length(Petal.Length)
[1] 150
```

```
> detach(iris)
> length(iris)
[1] 5
> length(Petal.Length)
Error: object 'Petal.Length' not found
```

Vediamo che quando 'stacchiamo' con `detach` il dataset, R riesce ancora a contare quante colonne compongono il dataset, ma non riesce più a leggere né i nomi delle variabili né tantomeno il loro contenuto. Forse questa particolarità vi potrà sembrare uno svantaggio od una scomodità. Al contrario, per le persone come Paolo Zaffino e Sasi Scaramuzzino [63] oppure MaVi Caruso [13], le quali si occupano di analizzare dati di *imaging digitale* che agevolmente raggiungono dimensioni di gigabyte, o terabyte – all'Elettra di Trieste in una mattinata di immagini con la luce di sincrotrone si accumulano addirittura petabyte di dati –, non possono illudersi di poter caricare nella memoria ram del loro portatile tutto quel popo di informazione. In questi particolari casi, la funzione `with` risolve in maniera agevole il problema di andare a leggere i dati all'interno di un dataset 'senza doverlo caricare in R' ma lasciandolo in remoto, là dove è archiviato:

```
with(iris, length(Petal.Length))

> with(iris, length(Petal.Length))
[1] 150
```

Abbandoniamo momentaneamente il dataset `iris`, che era un esempio di **dataset completo** (un dataset in cui tutte le colonne (che nei vari gerghi scientifici vengono anche chiamate 'campi', o 'istanze') e tutte le righe (ossia 'records') hanno un valore noto. Al contrario, prendiamo in considerazione il dataset `airquality` che raccoglie alcuni dati della qualità dell'aria misurati a New York durante la primavera e l'estate del 1973.

Come tutti i software, anche R dispone di un 'help in linea'. Il punto di domanda ? attiva infatti un menu di descrizione sintetica del comando o dell'oggetto in questione; osservate in particolare cosa è successo durante i giorni del 5 e del 6 maggio, nei quali alcuni valori non sono stati registrati.

```
? airquality
attach(airquality)
head(airquality)

> ? airquality
> attach(airquality)
> head(airquality)
   Ozone Solar.R Wind Temp Month Day
1    41      190  7.4   67     5    1
2    36      118  8.0   72     5    2
3    12      149 12.6   74     5    3
4    18      313 11.5   62     5    4
5    NA       NA 14.3   56     5    5
6    28      NA 14.9   66     5    6
```

**Vocabolario 1.2 — dataset incompleti.** Un dataset si dice **incompleto** quando osserviamo alcuni **missing data** o **valori mancanti**, che in R (ed in Orange) vengono codificati con il simbolo `NA`, acronimo di Not Available.

R è un software 'particolarmente abile' nel gestire i missing values, i quali però creano seri problemi negli ambiti tipici del machine learning, in cui Orange primeggia per immediatezza e semplicità di utilizzo. Provate dunque a svolgere questo esercizio che mostra come andare ad eliminare le righe del dataset in cui appaiono valori mancanti (e, ricordiamolo, non devono mai essere codificati 'lasciando una cella vuota': prendete l'abitudine di digitare la sigla NA in maiuscolo).

**Esercizio 1.4** Provate ad eseguire ciascuno di questi comandi e discutetene l'output. Cercate in particolare di capire con precisione il significato della funzione which.

```
na.omit(airquality)
complete.cases(airquality)
which(complete.cases(airquality) == TRUE)
```

■

### Usare R come una calcolatrice scientifica

Ovviamente R esegue i calcoli algebrici proponendo le risposte più appropriate dal punto di vista matematico, anche nei casi più 'spigolosi'. Ad esempio, vengono valutate le quantità infinite ed i casi indeterminati:

```
1/0
0/0

> 1/0
[1] Inf
> 0/0
[1] NaN
```

**Esercizio 1.5** Considerate il numero  $e$  ed osservate come funzionano gli arrotondamenti ed i troncamenti in R.

```
exp(1)
round(exp(1), 2)
floor(exp(1))
floor(-exp(1))
trunc(exp(1))
trunc(-exp(1))
```

■

Vedremo nel prossimo capitolo come si possa usare R non solo come una calcolatrice statistica o algebrica, ma come un vero e proprio linguaggio di programmazione. Si tratta di una potenzialità enorme, anche se chi si occupa per professione o per ricerca di programmazione, di solito preferisce sviluppare il proprio codice con altri interpreti (ad esempio Python) o compilatori (ad esempio C++).

## 1.2 Misure di dispersione

Partirò dal presupposto che tutti coloro i quali stanno leggendo questo libro abbiano familiarità con il concetto secondo il quale tramite la **deviazione standard** si riesca a quantificare la **variabilità** (o uno dei suoi sinonimi, **eterogeneità, dispersione**) che caratterizza i dati che stiamo osservando –

'piccola' deviazione standard: dati concentrati attorno alla media; 'grande' deviazione standard: dati sparpagliati, anche lontani dalla media. Tuttavia, vale la pena precisare un aspetto importante che riguarda la definizione algebrica della deviazione standard. Forse qualcuna/o si sarà già accorto del fatto che quando si prova a determinare questo indice con un foglio di calcolo, nel menu a video appaiono (almeno) due possibili scelte. Questo è quello che accade ad esempio in Open Office Calc™, in cui viene richiesto di scegliere tra la deviazione standard di un **campione** (DEV.ST) o quella di una **popolazione** (DEV.ST.POP):



Per fare chiarezza, ricordiamoci che tutti i nostri esperimenti si svolgono su campioni, ossia un sottoinsieme di elementi tratti da un più ampio insieme detto, appunto, popolazione. Osserviamo attentamente i denominatori delle formule che definiscono le due diverse deviazioni standard:

$$\text{popolazione: } \sqrt{\frac{\sum_i^n (x_i - m)^2}{n}}, \text{ campione: } \sqrt{\frac{\sum_i^n (x_i - m)^2}{n-1}}$$

I libri di statistica matematica (ad esempio [48]) spiegano perfettamente perché questo sia opportuno, richiamando i concetti di 'stimatore statistico corretto' e 'stimatore statistico coerente'. Per capirlo facilmente vi invento una storiella. Supponiamo che Massimino Borellino al liceo abbia ricevuto rispettivamente il voto  $x$  ed il voto  $y$  nel primo e nel secondo compito di latino. Supponiamo che la mamma di Massimino scopra che il voto medio dei due compiti è 6. Riuscirebbe la mamma di Massimino a dedurre il voto  $x$  ed il voto  $y$ ? No, perché disporrebbe di una sola informazione rispetto ai due voti ignoti. Viceversa, supponiamo che la mamma di Massimino scopra dal registro elettronico che il voto medio dei due compiti è 6 e che la deviazione standard è 1.41. Riuscirebbe la mamma a scoprire i voti  $x$  ed il voto  $y$ ? Eh sì, eh:

$$\begin{aligned}
 x = ? & \quad y = ? \\
 \frac{x+y}{2} = 6 & \quad \sqrt{\frac{(x-6)^2 + (y-6)^2}{2-1}} = 1.41 \\
 \frac{x+y}{2} = 6 & \quad \sqrt{2(x-6)^2} = 1.41 \\
 y = 12 - x & \quad \sqrt{2} \cdot (x-6) = 1.41 \\
 \sqrt{(x-6)^2 + (6-x)^2} = 1.41 & \quad x-6 = 1
 \end{aligned}$$

Dunque, ricapitolando. Se conosciamo la media abbiamo 1 informazione sull'andamento scolastico di Massimino; quindi per conoscere gli  $n = 2$  voti ce ne manca ancora  $(n-1) = 2-1 = 1$  di informazione. Ecco l'intuizione: quel denominatore  $(n-1)$  rappresenta quante informazioni ci mancano per conoscere compiutamente tutti i valori della variabile in esame, partendo dalla loro media. Ecco perché gli statistici di inizio secolo hanno trovato appropriato mutuare il termine di **gradi di libertà** dal gergo della meccanica e della fisica in generale. Re-importiamo il dataset **iris**, e verifichiamo 'a mano' il corretto calcolo della deviazione standard di un campione per mezzo della funzione **sd** (la deviazione standard di una popolazione non è implementata in R):

```

attach(iris)
sqrt(sum((Petal.Length - mean(Petal.Length))^2)/150)
sqrt(sum((Petal.Length - mean(Petal.Length))^2)/149)
sd(Petal.Length)

```

```
> attach(iris)
> sqrt(sum((Petal.Length - mean(Petal.Length))^2)/150)
[1] 1.759404
> sqrt(sum((Petal.Length - mean(Petal.Length))^2)/149)
[1] 1.765298
> sd(Petal.Length)
[1] 1.765298
```

**Quesito 1.2.1** Vi viene in mente un modo per verificare, con R, che la deviazione standard risulta essere definita come la radice quadrata della varianza (che si calcola con la funzione var)? No? Allora provate a rileggere a cosa servano i comandi `sd` e `sqrt` qui sopra.

### 1.2.1 Misure di dispersione con R

#### La funzione `summary`

Impariamo a conoscere una funzione molto versatile di R, che ci tornerà comoda in tutto il prosieguo del libro, la funzione `summary`. Se la applichiamo ad una variabile fattore otteniamo il medesimo output che avevamo ottenuto con la funzione `table` nella sezione 1.1.1:

```
summary(Species)

> summary(Species)
  setosa  versicolor  virginica
      50          50          50
```

Ma se la applichiamo ad una variabile numerica, ecco che l'output è molto più interessante:

```
summary(Petal.Length)

> summary(Petal.Length)
  Min. 1st Qu. Median   Mean 3rd Qu.   Max.
1.000 1.600 4.350 3.758 5.100 6.900
```

Oltre alla media ed alla mediana che già conoscevamo, il `summary` vi rivela sia il valore minimo che il valore massimo dei `Petal.Length`. Vengono inoltre calcolati anche il primo ed il terzo **quartile** della distribuzione, il che ci permette immediatamente di ottenere due nuove misure di dispersione: il **range**, ossia l'intervallo minimo e massimo di valori della variabile in esame, e l'**intervallo interquartile** (IQR, interquartile range), la differenza dei due quartili. Questo ci ricorda che la deviazione standard non è l'unica misura di dispersione a nostra disposizione, ma possiamo prendere in considerazione varie scelte.



La pagina Statistical Dispersion di Wikipedia fornisce un rapido richiamo a numerose altre misure di variabilità: [https://en.wikipedia.org/wiki/Statistical\\_dispersion](https://en.wikipedia.org/wiki/Statistical_dispersion).

#### Importare un dataset dalla rete

Come abbiamo visto con `iris` e con `airquality`, all'interno di R troviamo una collezione di decine di dataset che possono tornare comodi per fare degli esempi didattici. Ma nelle nostre ricerche utilizziamo dati veri, nostri. Ci conviene per il momento soprassedere riguardo al modo in cui si importa in R un vostro foglio di dati che avete conservato in una qualche cartella del

computer. Ne ripareremo alla fine del capitolo, nella sezione 2.5 dedicata agli esercizi ed alle attività di approfondimento.

Parliamo ora di colesterolo. Vogliamo pertanto importare in R il dataset `cholesterol` che raccoglie i dati di alcuni donatori di sangue della provincia di Trieste. Il dataset è pubblicato nel sito web del nostro corso di biostatistica:

<http://www.biostatisticaumg.it/dataset/cholesterol.csv>

Osservate innanzitutto il **formato .csv** del dataset, specificato dall'estensione del file, acronimo di 'comma separated value'.



È essenziale farsi una piccola cultura di base sui formati con cui vengono salvati i dati per poter essere scambiati. Qui vediamo cosa sono i file .csv, [https://en.wikipedia.org/wiki/Comma-separated\\_values](https://en.wikipedia.org/wiki/Comma-separated_values). Teniamo presente che il carattere 'virgola' si può usare come separatore perché per default nel mondo anglosassone i numeri decimali vengono rappresentati con il punto (ossia  $\pi \approx 3.14$ ) e non come in Italia con la virgola (ossia  $\pi \approx 3,14$ ). Questo implica che se volete condividere il vostro foglio elettronico ed all'interno avete delle misure decimali, dovrete assicurarvi di non fare pasticci, agendo sul *Formato delle Celle*. Qui potete trovare alcuni tutorial per come esportare il vostro foglio dati in formato .csv, in funzione del foglio di calcolo che preferite usare:

- [https://www.youtube.com/results?search\\_query=csv+export+excel](https://www.youtube.com/results?search_query=csv+export+excel)
- [https://www.youtube.com/results?search\\_query=csv+export+openoffice](https://www.youtube.com/results?search_query=csv+export+openoffice)
- [https://www.youtube.com/results?search\\_query=csv+export+libreoffice](https://www.youtube.com/results?search_query=csv+export+libreoffice)

Per importare il dataset, ci serviremo della funzione `read.csv`, specificando che il dataset contiene una quantità di dati elencati a partire dalla seconda riga, mentre la prima riga viene denominata header ed è riservata alle intestazioni delle colonne (come già abbiamo visto nei dataset `iris` e `airquality`). Basterà specificare a quale indirizzo di rete (ossia, l'url della risorsa Internet) il software vada a puntare, e poi dare l'attach. Con la funzione `tail` infine mostriamo le ultime sei righe del dataset; deduciamo dunque immediatamente che esso è composto da 1025 righe e 4 colonne:

```
indirizzo = "http://www.biostatisticaumg.it/dataset/cholesterol.csv"
cholesterol = read.csv(indirizzo, header = TRUE)
attach(cholesterol)
tail(cholesterol)

> indirizzo = "http://www.biostatisticaumg.it/dataset/cholesterol.csv"
> cholesterol = read.csv(indirizzo, header = TRUE)
> attach(cholesterol)
> tail(cholesterol)
   idanag sex TOTchol HDLchol
1020 id1060787   f     186      57
1021 id1060796   m     146      48
1022 id1060802   m     163      51
1023 id1060884   m     235      72
1024 id1060888   f     193      74
1025 id1061003   m     151      60
```

Le 1025 righe relative a donatori femmine e maschi sono contrassegnate da un identificatore anagrafico, anonimo ma univoco, `idanag`: nel gergo degli sviluppatori di database questo identificatore viene detto di solito **chiave primaria**. Questo identificatore permette di scoprire che non è appropriato dire che siano riportati i dati del colesterolo di 1025 donatori; infatti esiste almeno una persona che è andata a donare sangue due volte:

```
max(table(idanag))

> max(table(idanag))
[1] 2
```

In effetti possiamo scoprire che ci sono esattamente quattro persone che sono andate a donare sangue due volte, ed i valori di colesterolo relativi alla loro seconda donazione sono stati raccolti rispettivamente nella 519-esima, 556-esima, 618-esima e 966-esima riga del dataset:

```
which(table(idanag) == max(table(idanag)))

> which(table(idanag) == max(table(idanag)))
id524157 id531317 id543326 id948356
      519      556      618      966
```

### I quantili di una distribuzione

Alcune patologie, come per esempio proprio l'ipercolesterolemia, per convenzione vanno a considerare come 'patologici' i valori che eccedono un determinato percentile di quelli della popolazione in esame. Proviamo dunque a calcolare il 95 percentile del TOTcho1 tratto dal nostro campione.

```
quantile(TOTchol, 0.95)

> quantile(TOTchol, 0.95)
95 percent
267
```

Da dove esce questo 267 e come funziona questo calcolo? Si considerano tutti i 1025 valori di TOTchol e li si riordina, ad esempio, in modo crescente, dal minimo al massimo,  $x_1 \leq x_2 \leq \dots \leq x_{1025}$ , con la funzione `sort`. Siccome il 95 per cento di 1025 vale 973.75, andiamo a vedere chi sia il 974-esimo termine,  $x_1 \leq x_2 \leq \dots \leq x_{974} \leq \dots \leq x_{1025}$ , della sequenza riordinata:

```
sort(TOTchol)[974]

> sort(TOTchol)[974]
[1] 267
```

**Esercizio 1.6** Verificate che il 513-esimo elemento di `sort(TOTchol)`, ossia il cinquantesimo percentile, è proprio la mediana di `TOTcho1`. E verificate anche che il venticinquesimo ed il settantacinquesimo percentile sono proprio il primo ed il terzo quartile che avevamo trovato nel `summary`. ▀

## 1.3 Descrivere i dati nei design cross-section

Siamo quasi pronti per iniziare a dare una (parziale) risposta al problema 1.1 che abbiamo presentato all'inizio di questo capitolo. Ma prima di procedere, riflettete un istante su questi due quesiti.

**Quesito 1.3.1** Tre amici aprono i loro portafogli, contano il denaro in loro possesso e ci dicono che il valore **medio** è 43.71 euro. Cosa possiamo dedurre?

**Quesito 1.3.2** Tre amici aprono i loro portafogli, contano il denaro in loro possesso e ci dicono che il valore **mediano** è 43.71 euro. Cosa possiamo dedurre?

La lingua britannica possiede un aggettivo che calza perfettamente alla situazione: *uninformative*. Siamo così abituati a tenere in considerazione la media dei voti della pagella, del reddito nazionale, della pressione arteriosa, da non accorgerci del fatto che la media non 'trasmette alcuna informazione' sui dati che stiamo prendendo in esame; anzi, la media, in generale, non è nemmeno uno dei dati che abbiamo considerato nel nostro campione: è dunque assolutamente plausibile che nel primo quesito nessuno dei tre amici abbia esattamente quella somma in portafoglio. Nella seconda situazione invece abbiamo un'informazione certa: il secondo – per ricchezza – dei tre amici ha esattamente quella somma in portafoglio; e gli altri due amici, plausibilmente, hanno uno di meno, e l'altro di più, soldi in portafoglio (o, per combinazione, uno – o addirittura entrambi – la stessa somma di denaro).

Adesso, ricordiamo che qui ci stiamo occupando di dati che generalmente provengono da studi di tipo **cross sectional**, ossia **trasversali**: in un preciso istante ciascuno dei soggetti che compongono il nostro campione viene osservato e vengono raccolti i dati di nostro interesse. E noi, nella famosa 'Tabella 1' dobbiamo riassumerli al lettore, trasmettendo più informazioni possibili nella maniera più economica possibile. Nella prossima sezione parleremo meglio degli esperimenti con misure ripetute, come ad esempio quando facciamo analisi con la qRT-PCR, raccogliendo i  $C_T$  in duplicato o in triplicato, tecnico o biologico; oppure, quando facciamo colture cellulari, e ne misuriamo la crescita giorno dopo giorno; o infine, quando seguiamo nei mesi l'aumento ponderale delle nostre assistite in gravidanza.

Seguiamo innanzitutto un principio generale (nel Capitolo 4 vi prometto che spiegherò meglio il senso di questi aggettivi):

- 'approccio parametrico': la media 'va d'accordo' con la deviazione standard
- 'approccio non parametrico': la mediana 'va d'accordo' con i quartili ed il range

Accertata dunque la 'non informatività' della media, sarebbe logico ripiegare sempre sulla scelta mediana/quartili; i quali, tra l'altro forniscono informazione sulla **asimmetria** dei dati (in gergo, **skewness**). E' anche vero però che in presenza di dati simmetrici ed unimodali abbiamo condizioni favorevoli per usare tutto un corpus di teorie matematiche sviluppate nei secoli dei secoli. Ne ripareremo meglio alla fine della sezione 2.3.4.

Il consiglio finale che mi sento di dare è quello di scegliere quali indici usare solo dopo aver effettuato un'analisi esplorativa del dataset; cosa che riesce molto facile da fare con Orange. Vediamo come.

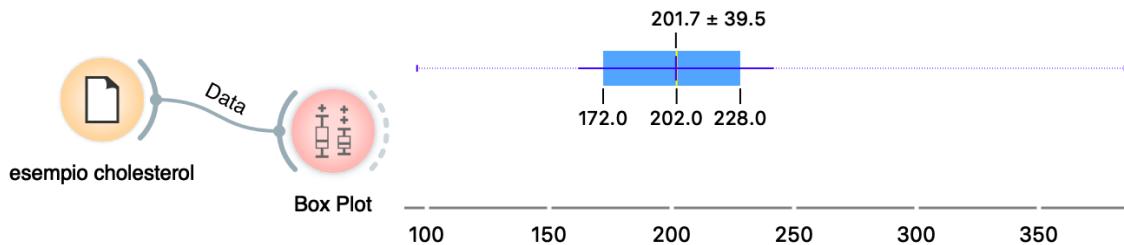
### 1.3.1 Facciamo conoscenza con Orange

Non vale nemmeno la pena pensare di mettersi a scrivere *come* si faccia ad usare Orange, essendo quest'ultimo un software totalmente 'visual'. Guardate invece questo video:



e potrete intuire quanto sia semplice trascinare i **widget** colorati nel **canvas** bianco per creare dei **workflow**. Con il primo workflow possiamo caricare il dataset **iris** e creare un **diagramma a barre** (ossia, un **barplot**) che conta le frequenze assolute dei tre livelli del fattore **Species**. Con il secondo workflow carichiamo direttamente dalla rete il dataset **cholesterol** ed otteniamo delle statistiche descrittive su **LDHtot** per mezzo del widget **Box Plot**.

Parliamo un pochino ora di questo grafico azzurro, il cui vero nome sarebbe **box and whiskers plot**, come deciso dal suo ideatore, il geniale John Tukey. Vengono riportati i quartili (172.0 e 228.0) e la mediana (202.0); il range viene tratteggiato con una linea azzurra, dal minimo 96.0 al massimo



385.0 (che però non vengono riportati). Non fanno parte del boxplot invece le informazioni della zona alta, ossia la media 201.7 e la deviazione standard 39.5, giustapposte con la simbologia ' $\pm$ ', le quali evidenziano un intervallo blu scuro su cui dovremo ritornare in seguito. Orange sembra fornire la soluzione più brillante al problema 1.1: scrivi tutto quello che sai, usa sempre tutti gli indici.

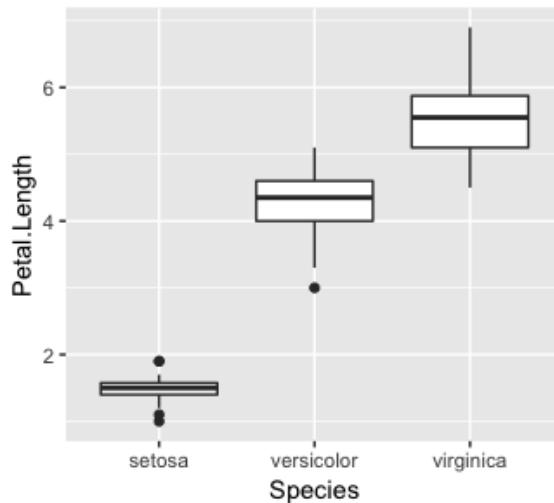
### 1.3.2 Il boxplot con R

Con R è altrettanto semplice disegnare un boxplot. Per esempio, ritorniamo al dataset `iris` e disegniamo i boxplot di `Petal.Length` ripartiti nelle tre diverse `Species`. Per farlo avremo bisogno del carattere speciale 'accento circonflesso', 'tilde'. Ecco il comando per R:

```
boxplot(Petal.Length ~ Species)
```

Se siete degli utenti di R Studio, troverete grande soddisfazione estetica nell'utilizzare il pacchetto `ggplot2`, che sta diventando uno standard *de facto* nella grafica con R. Il comando base è lievemente più evoluto:

```
library(ggplot2)
ggplot(iris, aes(x=Species, y=Petal.Length)) + geom_boxplot()
```



Osservate un dettaglio importante. Vedete che in un paio di boxplot i baffi finiscono evidenziando alcuni pallini? Questo succede perché siamo in presenza di alcuni punti 'isolati dagli altri' che vengono definiti **outlier**. Essi rispondono al fatto di essere 'lontani' per più di una volta e mezza la distanza interquartile, rispettivamente al di sotto di  $Q1 - 1.5 \cdot IRQ$  o al di sopra di  $Q3 + 1.5 \cdot IRQ$ . Ossia, calcolando  $IRQ = Q3 - Q1$ , si amplifica tale  $IRQ$  del 50%,  $1.5 \cdot IRQ$ , e si cerca se ci siano dei punti  $x_j$  tali che  $x_j < Q1 - 1.5 \cdot IRQ$  oppure  $x_j > Q3 + 1.5 \cdot IRQ$ : in tal caso, tali punti  $x_j$  vengono raffigurati

con un pallino. *Vox populi* afferma che fu durante un pranzo nella mensa universitaria che lo stesso John Tukey proponesse questa definizione; ed alla domanda del perché fosse opportuno amplificare *IRQ* di una volta e mezza, egli rispondesse che una sola volta gli pareva troppo poco, ma due gli pareva troppo.

### 1.3.3 Due presupposti fondamentali

Abbiamo appena visto come si realizza un boxplot con i nostri software preferiti, R ed Orange. Sicuramente il boxplot è un grafico che è molto informativo e che è adatto a descrivere i dati in molte situazioni. Tuttavia, ci sono moltissime altre possibilità per visualizzare i dati, e ciascun grafico ha certamente i suoi punti di forza ed i suoi punti di debolezza. Per non parlare dell'importanza di eseguire grafici con un aspetto professionale, gradevoli dal punto di vista estetico, e presentabili a tutto il pubblico, comprese le persone con possibili deficit visivi o quelle che sono dovute sedersi in fondo all'aula magna perché non c'era più posto davanti. Ecco dunque il primo dei due presupposti fondamentali di chi vuole esporre le proprie ricerche: operare una scelta consapevole sul grafico opportuno da presentare.



Su questo primo punto vi invitiamo certamente a leggere con attenzione le raccomandazioni di Claus Wilke nel suo e-book *Fundamentals of Data Visualization* [61], <https://serialmentor.com/dataviz/> per visualizzare quantità, distribuzioni, proporzioni, associazioni, serie temporali e molti molti altri tipi di dato, evitando i possibili tranello in cui spesso si incorre. Il libro utilizza il pacchetto `ggplot2`, il quale è veramente molto accattivante e molto professionale, ma richiede un po' di tempo per impararne. Date un'occhiata alla sua galleria di immagini <http://www.ggplot2-exts.org/gallery/> per convincervi che vale la pena impararlo ad usare, ad esempio da <https://ggplot2.tidyverse.org/>, oppure, leggendo il libro di Hadley Wickham e Garrett Grolemund [60], <https://r4ds.had.co.nz/>.

Il secondo punto importante che viene dato per scontato, ma scontato non è, riguarda la fase di 'raccolta dei dati', o meglio, la fase di predisposizione del dataset: i dati che andremo ad analizzare provengono da tabelle di database esportati in un foglio elettronico, oppure siamo stati noi a digitali, oppure ce li hanno forniti e noi non sappiamo neppure chi li abbia inseriti, magari codificando il genere maschile/femminile con la codifica 0 / 1, dimenticando sinanco di prepararci una legenda. Siccome l'argomento è corposo, per non perdere il filo del discorso lo completiamo alla fine del capitolo con degli esercizi ad hoc.

## 1.4 Descrivere i dati nei design a misure ripetute

Partiamo da un esempio concreto: Stefania Scalise e Maria Teresa De Angelis [17] effettuano routinariamente delle qRT-PCR, ad esempio sui geni delle proteine p28 e p53; qui gli esperimenti sono stati condotti in duplicato tecnico e triplicato biologico. Nella tabella che segue consideriamo i cicli soglia  $C_T$  dei geni pensando al primo come quello di interesse ed al secondo come gene house-keeping. I dati sono stati raccolti in formato .xls in lingua italiana, ossia con la virgola come separatore decimale, ma qui li vediamo riscritti in forma di dataset di R, con il punto decimale:

<b>Id</b>	<b>Well</b>	<b>BiolRep</b>	<b>TechRep</b>	<b>CtGene</b>	<b>CtHK</b>
1	p28	a	1	22.76	17.20
2	p28	a	2	22.71	17.09
3	p28	b	1	22.46	16.95
4	p28	b	2	22.37	16.91
5	p28	c	1	22.69	17.09
6	p28	c	2	22.66	16.96
7	p53	d	1	24.65	17.08
8	p53	d	2	24.57	17.07
9	p53	e	1	24.54	17.26
10	p53	e	2	24.61	17.19
11	p53	f	1	24.76	17.33
12	p53	f	2	24.63	17.30

Il problema 1.1 di descrivere i dati, in forma numerica o grafica, è molto sottile (per non dire farraginoso). Discutiamolo passo per passo (ma se i passaggi tecnici vi annoiano, passate subito alle sezione 1.4.2).

### 1.4.1 Calcolare i dati di relative gene expression con R

#### Creare un dataset con R

Se abbiamo una quantità di dati numerici in un foglio .xls che vogliamo importare in R creandoci un dataset, possiamo utilizzare un piccolo trucco: fare un passaggio intermedio in un editor di testo, in modo da 'aggiustare' la questione della virgola decimale. Vediamo questo video:



Adesso che i numeri decimali sono sistemati, possiamo creare la variabile Well usando l'istruzione rep, che in questo esempio permette di ripetere per 6 volte la parola p28, e poi altre 6 volte la parola p53. La variabile TechRep si crea ripetendo la coppia di numeri 1 e 2 per sei volte, mentre BiolRep crea la sequenza di lettere da a ad f in una maniera fantasiosa, che vi invitiamo a cercare di capire da soli. Infine il comando data.frame organizza le variabili in un dataset di nome geneexpression:

```
Well = factor(c(rep("p28", 6), rep("p53", 6)))
BiolRep = factor(sort(rep(letters[1:6], 2)))
TechRep = rep(1:2, 6)
CtGene = c(22.7585, 22.7101, 22.4559, 22.3718, 22.6911, 22.658, 24.6510,
          24.5663, 24.5423, 24.6085, 24.7599, 24.6339)
CtHK = c(17.1981, 17.0887, 16.9521, 16.9078, 17.088, 16.9579, 17.0813,
        17.0689, 17.2591, 17.1872, 17.3280, 17.3011)
```

```
geneexpression = data.frame(Well, BiolRep, TechRep, CtGene, CtHK)
geneexpression
```

### Separare una variabile in base ad un fattore con R

La teoria in vigore [38] prevede che si determini il  $\Delta C_T$ , ossia la differenza tra CtGene e CtHK, e poi si 'normalizzi' rispetto al calibratore, ossia la media di CtGene di p28. A questo proposito possiamo usare la funzione `split`, che restituisce una **lista** di vettori, in questo caso due giacché Well è un fattore a due livelli. Calcolando poi la media del primo, `calibra`, stimiamo l'`amountoftarget` come previsto (ed eventualmente possiamo calcolare anche la media sui replicati tecnici):

```
deltaCt = CtGene - CtHK
calibra = mean(split(deltaCt, Well)[[1]])
deltadeltaCt = deltaCt - calibra
amountoftarget = 2^(-deltadeltaCt)
round(amountoftarget, 3)
round(tapply(numtarget, BiolRep, mean), 3)

> deltaCt = CtGene - CtHK
> calibra = mean(split(deltaCt, Well)[[1]])
> deltatadeltaCt = deltaCt - calibra
> amountoftarget = 2^(-deltadeltaCt)
> round(numtarget, 3)
[1] 1.010 0.969 1.051 1.080 0.981 0.917 0.251 0.264 0.306 0.278
[11] 0.276 0.296
> round(tapply(numtarget, BiolRep, mean), 3)
      a      b      c      d      e      f
0.990 1.066 0.949 0.257 0.292 0.286
```

**Esercizio 1.7** Verificate passo per passo come 'funzionano' questi comandi:

```
letters[1:6]
rep(letters[1:6],2)
sort(rep(letters[1:6],2))

split(deltaCt, Well)
split(deltaCt, Well)[1]
split(deltaCt, Well)[[1]]
```

### 1.4.2 L'equazione più pericolosa, parte prima

Riprendiamo i dati di Maria Teresa e Stefania, dopo averne calcolato l'espressione genica relativa. I valori di espressione genica relativa del primo gruppo sono così discosti dal secondo che qui la statistica più che descrittiva sembra avere un ruolo 'cosmetico'. Tuttavia ci chiediamo: dobbiamo usare un indice di dispersione come la deviazione standard, oppure come l'intervallo interquartile, per riassumere la variabilità di questi dati? Oppure, in questo caso, se vogliamo rispondere alla domanda del problema 1.1 noi non siamo interessati a descrivere quanta variabilità abbiano questi dati; noi vogliamo stabilire con quale tolleranza, con quale **affidabilità** (i.e. **reliability**) stiamo stimando un valore teorico, il valore medio 'vero' della popolazione, che non ci è noto ma di cui possiamo osservare solo alcuni valori sperimentali.

Il titolo di questo paragrafo in realtà è la citazione di un articolo di Howard Wainer, [59], *The most dangerous equation*, ripreso poi anche da Tu e Gilthorpe, *The most dangerous hospital or the most dangerous equation?*. Se avete un pochino di tempo leggetevi, io li ho trovati molto interessanti.



Howard Wainer: [https://www.researchgate.net/publication/255612702\\_The\\_Most\\_Dangerous\\_Equation](https://www.researchgate.net/publication/255612702_The_Most_Dangerous_Equation)

Yu-Kang Tu, Mark Gilthorpe: <https://bmchealthservres.biomedcentral.com/articles/10.1186/1472-6963-7-185>

Nel prossimo capitolo ci occuperemo della simulazione di dati, e riprenderemo la questione nella sezione 2.3.2 mostrando come emerge nella teoria matematica il concetto di **errore standard della media** (i.e. lo **standard error of the mean**), il quale è un indice che algebricamente dipende anche dalla dimensione del campione,  $n$ , in maniera del tutto ragionevole: quadruplicando la dimensione del campione si dimezza l'incertezza che noi abbiamo nel ritenere valida la stima ottenuta dalle misure sperimentali.

$$SEM = \frac{\sigma}{\sqrt{n}}$$

Il fatto di decidere sulla affidabilità della media invece che sulla dispersione dei dati è connesso al fatto che, di solito, gli esperimenti biologici che vengono effettuati sono complessi e costosi e quindi, per ragioni del tutto pratiche, disponiamo di campioni con dimensioni (**sample size**)  $n$  esigue (e discuteremo meglio in seguito di quali limitazioni comporti questo fatto). Ecco dunque che quando vogliamo rappresentare con un grafico i dati di questa natura, è raccomandabile che ciascuno dei dati grezzi venga raffigurato individualmente, e non solo tramite le statistiche descrittive [21]. Consideriamo a questo proposito il grafico denominato **dynamite plot**, nel quale i dati vengono riassunti esibendo la loro media in una colonnina, sormontata da una 'barretta d'errore' che tipicamente è l'errore standard della media del campione. Ad esempio, Emanuela Chiarella [14] lo utilizza nella figura 1 del suo paper:

<https://www.mdpi.com/1422-0067/19/12/4095/htm>

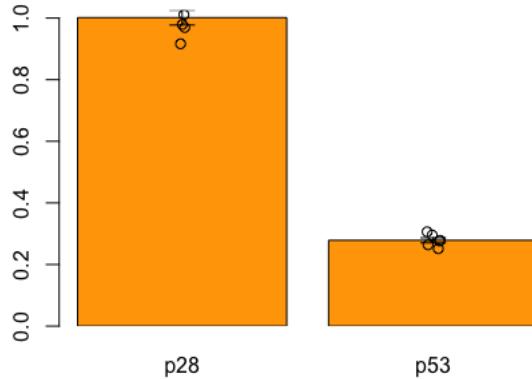
Da un lato, vi sono dei forti detrattori sull'impiego di tale grafico; tanto per elencarne alcuni:

- il professor Tatsuki Koyama, che nel suo poster [35] mostra chiaramente come due dynamite plot apparentemente uguali celino completamente la struttura e la dimensione dei campioni presi in esame
- la comunità degli sviluppatori di R, che non ha mai creato una funzione di base (se non in alcune library aggiuntive) per disegnare il dynamite plot
- la comunità degli sviluppatori di Orange, che a quanto pare non ha creato nemmeno library aggiuntive per farlo.

Dall'altro lato, osserviamo che – non solo Emanuela – ma praticamente tutti i paper in ambito biologico utilizzano questa rappresentazione per i dati effettuati in laboratorio ripetendo le misure sul campione.

Come uscirne? La mia raccomandazione è:

1. se il nostro design sperimentale non prevede l'utilizzo di misure ripetute, **non** utilizzare mai il dynamite plot
2. se dobbiamo raffigurare esperimenti con misure ripetute, allora sovrapponiamo al dynamite plot anche un **dot plot**, i cui pallini evidenzino i dati grezzi. Se volete farlo con R, ecco qui di seguito la descrizione di come procedere.



### 1.4.3 Sovrapporre due grafici con R

Riprendiamo i dati che abbiamo visto poco fa nella sezione 1.4.1. Supponiamo che Stefania e Maria Teresa vogliono rappresentare amountoftarget con un dynamite plot, in base al fattore Well a due livelli p28 e p53, come vediamo qui sopra. Sarà innanzitutto necessario scaricare ed installare il pacchetto aggiuntivo di nome `sciplot`. Lo si può fare dalla barra dei menu, oppure direttamente dalla Console di R con il comando:

```
install.packages("sciplot")
```

Adesso dovremo attivare il pacchetto con il comando `library`, e creare il grafico dinamite con il comando `bargraph.CI`. Successivamente dovremo fare un po' di bricolage: usando la funzione `points` sovrapporremo 6 punti sulla prima barra e sei punti sulla seconda; la 'mezzeria' della barra ha approssimativamente coordinata 0.7, mentre la seconda approssimativamente 1.9. Per evitare che i punti si sovrappongano li 'disturberemo' applicando una piccola perturbazione numerica decimale casuale con il comando `jitter`, sia rispetto all'asse `x` che rispetto all'asse `y`:

```
library(sciplot)
bargraph.CI(x.factor = Well, response = amountoftarget,
             data = finale, col = "orange")
points(jitter(rep(0.7,6)), jitter(amountoftarget[Well == "p28"]))
points(jitter(rep(1.9,6)), jitter(amountoftarget[Well == "p53"]))
```



Se desiderate utilizzare invece `ggplot2` in R Studio, allora è meglio lasciare la parola ad Hadley Wickham: <https://rpubs.com/hadley/ggplot2-layers>

## 1.5 Esercizi ed attività di approfondimento

■ **Attività 1.1 — misure di tendenza centrale e di dispersione.** ([49, pagina 14] ) Considerate il campione dei giorni intercorsi tra il penultimo e l'ultimo periodo mestruale di 500 giovani donne. La colonna delle frequenze riporta il numero di donne che ha riferito rispettivamente quel periodo.

Sapreste determinare 'a mente' la moda? E sempre 'a mente' la mediana? Ed il primo e terzo quartile? E sempre 'a mente', o 'con carta e matita', sapreste dire se ci sono forse delle ragazze

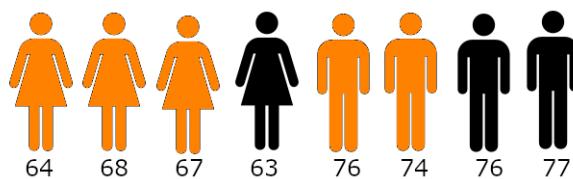
Periodo	Frequenza	Periodo	Frequenza	Periodo	Frequenza
24	5	29	96	34	7
25	10	30	63	35	3
26	28	31	24	36	2
27	64	32	9	37	1
28	185	33	2	38	1

outlier? Provate poi ad utilizzare la funzione `rep()` per implementare in R un vettore denominato `periodo`, e controllate con le funzioni `table` e `summary` le due soluzioni da voi proposte. ■

■ **Attività 1.2 — creare ed importare un dataset.** Un mio caro amico di Trieste, il professor Umberto Lucangelo, aveva suggerito ad una laureanda in Odontoiatria uno studio sull’effetto analgesico di tre farmaci. Importate in R il dataset `analgesia.txt` con i comandi seguenti, osservando che si tratta di un dataset in formato `.txt`, il cui carattere di separazione è il segno di tabulazione (e non la virgola) e pertanto utilizziamo il comando `read.table` invece di `read.csv`.

```
www = "http://www.biostatisticaumg.it/dataset/analgesia.txt"
analgesia = read.table(www, header = TRUE)
attach(analgesia)
```

- Quante righe e quante colonne ha il dataset `analgesia`? Scopritelo con `str`.
- Cosa otteniamo con il comando `table(sex)`: le frequenze assolute o relative dei pazienti?
- Cosa otteniamo con il comando `prop.table(sex)`?
- Cosa otteniamo con il comando `prop.table(table(sex))`?
- Mia moglie, mia mamma e mia suocera affermano che i maschi sopportano meno il dolore delle femmine. Provate a disegnare una coppia di boxplot del `dolore6h` rispetto a `sex`.
- Quanti sono i pazienti maschi che hanno assunto il tramadol? Scopritelo con `table`.
- Quanti sono i pazienti maschi con (`dolore6h >= 5`) ? Scopritelo con `table`.



■ **Attività 1.3 — creare ed importare un dataset.** State iniziando uno studio inerente i disturbi dell’alimentazione, ed avete un campione iniziale di otto soggetti, maschi e femmine, alcuni dei quali possiedono una certa mutazione genetica (soggetti arancione della figura qui in alto). I numeri che vedete raffigurati rappresentano il peso di ciascun soggetto. Impostate i dati in un foglio di calcolo elettronico (MS Excel™, Open Office Calc, Google Sheets, Libre Office Calc, ...), salvatelo, ed importatelo in R. Calcolate usando `tapply` la deviazione standard dei pesi dei pazienti con mutazione e di quelli senza mutazione.

*Suggerimento: vi tornerà utile la funzione `file.choose()`; scoprite come funziona cercando qualche esempio in rete.* ■

■ **Attività 1.4 — struttura di un dataset.** Il foglio elettronico è uno strumento molto comodo per raccogliere i dati. Tuttavia, trascrivere dei dati non significa automaticamente aver creato un dataset.

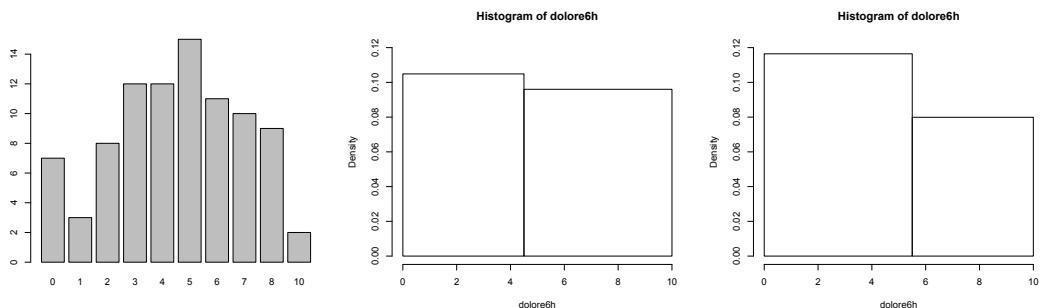
Date un'occhiata alle due schermate sottostanti, osservate che entrambe descrivono correttamente l'esperimento dell'attività precedente, ma questi dati non possono venire importati *sic stantibus* in un qualsiasi software oggi esistente di analisi dei dati. ■

	A	B	C
1	peso	MUT	WT
2	F	64, 68, 67	63
3	M	74, 76	76, 77
4			

	A	B
1	F	M
2	64	76
3	68	74
4	67	76
5	63	77
6		

■ **Attività 1.5 — istogramma.** Parliamo un poco della diversità di significato che hanno i grafici a barre dagli histogrammi, riferendoci ancora al dataset analgesia.



```
par(mfrow = c(1,3))
barplot(table(dolore6h))
hist(dolore6h, breaks = c(0, 4.5, 10), ylim = c(0, 0.12))
hist(dolore6h, breaks = c(0, 5.5, 10), ylim = c(0, 0.12))
```

A sinistra vediamo un grafico a barre, da cui deduciamo immediatamente che il dolore6h modale del nostro campione è 5. Capite che in un diagramma a barre l'informazione essenziale è fornita dalla *altezza* delle colonnine grigie, le quali sono proporzionali alle *frequenze assolute* di dolore6h:

```
table(dolore6h)
```

Ora invece osservate cosa succede con i comandi `table(dolore6h < 5)` e `table(dolore6h > 5)`: riuscite a collegare i numeri che avete ottenuto (32, 42, .. , ..) con la *forma* dei rettangoli bianchi degli histogrammi? E se vi chiedessimo di calcolare, 'con il righello' e 'con carta e matita', le aree dei rettangoli bianchi (*base per altezza*), che valori otterreste? Completate dunque da soli la frase: *in un istogramma l'informazione essenziale è fornita dalle . . . . delle colonne, le quali sono proporzionali alle frequenze . . . . di dolore6h*. In definitiva, in un diagramma a barre non importa la larghezza delle colonne, ma in un istogramma sì. Cercate a tale proposito nella vostra biblioteca universitaria (o in qualche angolo nascosto nella rete) se riuscite a trovare il manuale di Venables e Ripley [56] e guardate la loro figura 5.8 di pagina 127: vedete come la scelta dei **cut-off** (i.e. dei `breaks`) dell'istogramma riesce a cambiare, e di molto, la forma dei rettangoli? ■



## 2. Simulare i dati sperimentali con R

Ai tempi in cui il professor Morrone ed io eravamo studenti, sui nostri libri di testo le locuzioni *in vivo* ed *in vitro* si contrapponevano, ma non venivano mai affiancate nelle loro definizione dal neologismo *in silico*: il fatto che da un paio di decenni ormai i computer offrano potenze di calcolo un tempo inimmaginabili, e che discipline come la bioinformatica o la system biology siano esplose, richiede che i software che utilizziamo ci consentano di sviluppare righe di codice per trovare soluzioni ad hoc per i nostri problemi, o che simulino per noi dati artificiali da confrontare con i nostri dati di laboratorio. La simulazione di dati è inoltre alla base di alcune importanti tecniche di analisi statistica che vengono definite ad esempio con i termini metodi bootstrap, metodi Monte Carlo o metodi di ricampionamento.

### Di cosa si parlerà in questo Capitolo?

Il Capitolo 2 di questo libro non è un capitolino, ma un capitolone: risulta essere certamente il più 'sgradevole' ed è quello che richiede il maggior sforzo matematico nella lettura. Ma io mi rendo perfettamente conto del fatto che quasi sempre un ricercatore ha una domanda precisa, ben circostanziata e desidera ottenere una risposta immediata, efficace ed operativa, non avendo il tempo o l'energia necessaria per investigare sui dettagli matematici ed informatici della faccenda. I quali però sono importanti per una corretta comprensione del tutto.

Vi propongo allora questa strategia draconiana. Se ce la fate, proseguite la lettura del Capitolo pagina per pagina. Se non ce la fate, saltate a piè pari tutte queste sezioni e ci rivediamo fra una trentina di pagine direttamente al Riepilogone del Capitolone, alla sezione 2.4.

### 2.1 R è un linguaggio di programmazione

Innanzitutto, R è un vero e potente linguaggio di programmazione object-oriented, particolarmente adatto alla programmazione scientifica [34]. In particolare, riesce ad eseguire calcoli in ambito bioinformatico molto rapidamente usando l'approccio vector based, che adesso andiamo ad illustrare con un esempio semplice che è stato uno dei cavalli di battaglia della ricerca di Gianni Morrone e della sua amata moglie, Heather Bond [7].

**Problema 2.1** .. abbiamo qui la sequenza della proteina Zinc Finger 521, e vorremmo calcolarci lo strand complementare, in maniera semplice.

### 2.1.1 Usare il ciclo for e la decisione if

Per comodità, siamo andati a prenderci la sequenza pubblicata in <https://www.ebi.ac.uk/ena/data/view/CAD57322> e la abbiamo trasformata in uno dei nostri dataset. Importiamola in R, osservando che i dati sono stati originariamente salvati in formato .txt e non .csv come in precedenza, e dunque usiamo il comando `read.table` invece che `read.csv`:

```
www = "http://www.biostatisticaumg.it/dataset/ZNF521.txt"
zincfinger = read.table(www, header = TRUE)
attach(zincfinger)
head(zincfinger)
tail(zincfinger)
table(Q96K83)

> www = "http://www.biostatisticaumg.it/dataset/ZNF521.txt"
> zincfinger = read.table(www, header = TRUE)
> attach(zincfinger)
> head(zincfinger)
Q96K83
1      A
2      T
3      G
4      T
5      C
6      T
> tail(zincfinger)
Q96K83
3931     A
3932     G
3933     T
3934     T
3935     A
3936     G
> table(Q96K83)
Q96K83
A      C      G      T
1136 1012  903   885
```

Osserviamo che il dataset `zincfinger` è composto da 3936 righe (i nucleotidi, con Adenosina più frequente di Timidina) e da un'unica colonna di nome `Q96K83`. Per semplicità, iniziamo a lavorare con una breve sequenza di basi, 21 per la precisione, che troviamo a partire dalla millesima posizione:

```
esempio = Q96K83[1000:1020]
esempio

> esempio = Q96K83[1000:1020]
> esempio
```

```
[1] C A A C C G G A G T C A T G C A A T C A C
Levels: A C G T
```

Partendo ora da questa sequenza di **esempio** vogliamo ottenere una sequenza di 21 lettere **revcompl**, complementare ed inversa:

```
C A A C C G G A G T C A T G C A A T C A C
G T G A T T G C A T G A C T C C G G T T G
```

Possiamo farlo in due tempi, iniziando a creare la sequenza inversa: bisogna che il nucleotide al posto 1 venga occupato da quello del posto 21, quello al posto 2 da quello al posto 20, quello al posto 3 da quello al posto 19, eccetera.

```
C A A C C G G A G T C A T G C A A T C A C
C A C T A A C G T A C T G A G G C C A A C
```

Accorgiamoci di una semplice relazione algebrica che caratterizza questi indici: se spostiamo la prima lettera al ventunesimo posto,  $1 + 21 = 22$ ; ed anche per la seconda al ventesimo,  $2 + 20 = 22$ ; ed anche  $3 + 19 = 22$ . Potremmo dunque dire che per creare la sequenza inversa, bisogna che il nucleotide al posto 1 venga occupato da quello del posto  $22 - 1$ , quello al posto 2 da quello al posto  $22 - 2$ , quello al posto 3 da quello al posto  $22 - 3$ , e così via. In altre parole se partiamo ricopiando **esempio** in **rovescio**, **rovescio** = **esempio** e se indichiamo con **N** (= 21) la lunghezza **length** del vettore **esempio**, vogliamo che:

```
rovescio = esempio
N = length(esempio)
rovescio[1] = esempio[(N + 1) - 1]
rovescio[2] = esempio[(N + 1) - 2]
rovescio[3] = esempio[(N + 1) - 3]
```

e così via. Ma per non dover scrivere a mano queste istruzioni per altre diciotto volte, utilizziamo un'istruzione di **iterazione**, il **ciclo for**. Vediamo che in esso compaiono due istruzioni: tra le parentesi tonde, l'indice **i** che viene fatto variare da 1 ad **N**; tra le parentesi graffe un'istruzione (o più istruzioni) che dipende dall'indice **i**.

```
rovescio = esempio
N = length(esempio)
for( i in 1:N )
{ rovescio[i] = esempio[(N + 1) - i] }
rovescio

> rovescio = esempio
> N = length(esempio)
> for( i in 1:N )
+ { rovescio[i] = esempio[(N + 1) - i] }
> rovescio
[1] C A C T A A C G T A C T G A G G C C A A C
Levels: A C G T
```

Per il momento lasciamo da parte **esempio**, consideriamo **rovescio**, ricopiamolo in **revcompl** e calcoliamo il complementare di ciascuna delle basi della sequenza inversa. Questo si può fare direttamente all'interno del ciclo **for** con una serie di decisioni, o meglio di **istruzioni condizionate if**: la prima decisione potrebbe essere quella di cambiare la A, se presente, con una T – notate l'uso delle parentesi tonde e quadre come nel ciclo **for**:

```
if (rovescio[i] == "A"){ revcompl[i] = "T" }
```

e via via le altre tre:

```
if (rovescio[i] == "C"){ revcompl[i] = "G" }
if (rovescio[i] == "G"){ revcompl[i] = "C" }
if (rovescio[i] == "T"){ revcompl[i] = "A" }
```

Il programma è dunque pronto. Ma, prima di eseguirlo, sfruttiamo la funzione `proc.time()` che 'cronometra' quanto tempo impiega R per eseguire tutte le istruzioni. Con il nostro portatile, risultano 0.028 secondi.

```
inizio = proc.time()
rovescio = esempio
revcompl = rovescio
N = length(esempio)
for( i in 1:N )
{
  rovescio[i] = esempio[(N + 1) - i]
  if (rovescio[i] == "A"){ revcompl[i] = "T" }
  if (rovescio[i] == "C"){ revcompl[i] = "G" }
  if (rovescio[i] == "G"){ revcompl[i] = "C" }
  if (rovescio[i] == "T"){ revcompl[i] = "A" }
}
revcompl
proc.time() - inizio

> inizio = proc.time()
> rovescio = esempio
> revcompl = rovescio
> N = length(esempio)
> for( i in 1:N )
+ {
+   rovescio[i] = esempio[(N + 1) - i]
+   if (rovescio[i] == "A"){ revcompl[i] = "T" }
+   if (rovescio[i] == "C"){ revcompl[i] = "G" }
+   if (rovescio[i] == "G"){ revcompl[i] = "C" }
+   if (rovescio[i] == "T"){ revcompl[i] = "A" }
+ }
> revcompl
[1] G T G A T T G C A T G A C T C C G G T T G
Levels: A C G T
> proc.time() - inizio
  user  system elapsed
 0.018   0.002   0.028
```

**Esercizio 2.1** Modificate il programma appena scritto, facendolo agire non sull'esempio ma su tutta la proteina zinc finger Q96K83 e controllate quanto tempo-macchina viene impiegato (a noi occorrono 0.347 secondi). ■

**Esercizio 2.2** Provate ad inventare un programma che calcoli con un ciclo `for` la deviazione standard di un vettore numerico (per esempio `Petal.Length` di `iris`) utilizzando la definizione esplicita del paragrafo 1.2.

### La programmazione vector-based

Vediamo ora un metodo nettamente più efficiente per risolvere la questione: invece di ripetere per molte volte una medesima istruzione con un ciclo `for`, le azioni vengono svolte 'in blocco' sul vettore Q96K83. Occorre anche qui un po' di astuzia; innanzitutto R dispone di una funzione, `rev()`, che rovescia un vettore dalla fine all'inizio:

```
rovescio = rev(esempio)
rovescio

> rovescio = rev(esempio)
> rovescio
[1] C A C T A A C G T A C T G A G G G C C A A C
Levels: A C G T
```

Il vettore `rovescio` è ancora un fattore a quattro livelli, A, C, G e T. Vediamo cosa succede se chiediamo di cambiare il suo **tipo**, da `factor` a `numeric`:

```
as.numeric(rovescio)

> as.numeric(rovescio)
[1] 2 1 2 4 1 1 2 3 4 1 2 4 3 1 3 3 2 2 1 1 2
```

Come vedete, le lettere dei nucleotidi sono state convertite in numeri ed i numeri rappresentano appunto i livelli del fattore `rovescio`. Ora, facciamo una osservazione algebrica molto utile: con l'operazione di complementazione succede che il livello 1 si scambia con il livello 4, il livello 2 si scambia con il livello 3, il 3 si scambia con il 2 ed infine il 4 con l'1. Se notate ora che la somma di queste coppie di numeri fa sempre 5, con questa astuzia possiamo immediatamente generalizzare: per trovare lo strand complementare basta scambiare il livello `i` di `rovescio` con il livello `5 - i`:

```
5 - as.numeric(rovescio)
```

Utilizziamo dunque questa astuzia nella nostra istruzione di complementazione e poi riconvertiamo i numeri nei livelli A, C, G e T precedenti con il comando che agisce sui livelli del fattore:

```
levels(rovescio) = levels(esempio)
```

Il programma completo dunque diviene:

```
inizio = proc.time()
revcompl = as.factor(5 - as.numeric(rev(esempio)))
levels(revcompl) = levels(esempio)
revcompl
proc.time() - inizio
```

```
> inizio = proc.time()
> revcompl = as.factor(5 - as.numeric(rev(esempio)))
> levels(revcompl) = levels(esempio)
> revcompl
[1] G T G A T T G C A T G A C T C C G G T T G
Levels: A C G T
> proc.time() - inizio
  user  system elapsed
0.002  0.001  0.007
```

**Esercizio 2.3** Per dare una ottima soluzione al problema 2.1 provate a modificare il programma appena scritto, facendolo agire non sull'esempio ma su tutta la proteina zinc finger Q96K83 e controllate quanto tempo-macchina viene impiegato (a noi sono occorsi 0.017 secondi contro i precedenti 0.347 secondi: venti volte di meno). ■

**Esercizio 2.4** Per esercizio, provate ad inventare un programma che calcoli in maniera vector-based la deviazione standard di un vettore numerico (per esempio Petal.Length di iris) utilizzando la definizione esplicita del paragrafo 1.2. ■

### 2.1.2 Creare le proprie funzioni

Sino ad ora nel libro avete trovato nominate le parole *funzione* e *comando* in maniera del tutto equivalente per indicare le istruzioni del linguaggio R. In effetti è molto comodo per chi programma poter scrivere delle **funzioni definite dall'utente**: le righe di codice ne guadagnano in leggibilità e sono riutilizzabili quando necessario. Per esempio, trasformiamo il programma che abbiamo appena mostrato sulla trascrizione della proteina ZNF521 in una funzione che determini lo strand complementare di qualsiasi sequenza di nucleotidi.

Definiremo dunque una funzione di nome `revcom()` in questo modo:

```
revcom = function(x)
{
}
```

La parola riservata `function` specifica che all'interno delle parentesi tonde viene posta una **variabile locale** (o più, separate da virgolette), che assumerà il valore dell'argomento passato dalla funzione `revcom` quando verrà – come si dice in gerco – chiamata. Nello spazio tra le parentesi graffe invece inseriamo il codice che abbiamo appena creato, usando un'altra variabile locale `y` per fornire l'output:

```
revcom = function(x)
{
  y = as.factor(5 - as.numeric(rev(x)))
  levels(y) = levels(x)
  return(y)
}
```

Adesso è sufficiente digitare `revcom(Q96K83)` per ottenere lo strand complementare di mRNA, o di qualsiasi altra sequenza di nucleotidi.

```

revcom = function(x)
{
  y = as.factor(5 - as.numeric(rev(x)))
  levels(y) = levels(x)
  return(y)
}
revcom(Q96K83)

> revcom = function(x)
+ {
+   y = as.factor(5 - as.numeric(rev(x)))
+   levels(y) = levels(x)
+   return(y)
+ }
> revcom(Q96K83)
[1] C T A A C T G C T G T G T G G G T C A T T G T A T G A T T C
[32] T G C A G C T C T G T T G G A A G A A A A A C T T C T G T G
...
[993] T A T G C G T G
[ reached getOption("max.print") -- omitted 2936 entries ]
Levels: A C G T

```

L'output è considerevolmente lungo (abbiamo omesso molte righe infatti) ed alla fine R Studio non ci rivela più di mille nucleotidi. Quindi, bisogna fare una modifica: conservare il risultato delle funzione `revcom()` in un vettore `rovescio`, creare un dataset ad esempio di nome `risposta` con la funzione `data.frame` e salvarlo sul disco del computer ad esempio in formato testo con il comando `write.csv`. La funzione `getwd()` vi aiuterà a ritrovare la cartella (la **working directory**) dove il file `trascZincFinger.txt` è stato salvato.

```

rovescio = revcom(Q96K83)
risposta = data.frame(rovescio)
write.csv(risposta, file = "trascZincFinger.txt")
getwd()

> rovescio = revcom(Q96K83)
> risposta = data.frame(rovescio)
> write.csv(risposta, file = "trascZincFinger.txt")
> getwd()
[1] "/Users/Massimino/Documents/"

```

## 2.2 Gli eventi casuali con R

Parliamo ora degli aspetti che riguardano la possibilità di eseguire esperimenti *in silico*, ossia simulando in maniera opportuna i 'dati della Natura'. Si tratta di dati che si presentano ai nostri occhi in maniera casuale; ma nella loro casualità possiedono dei vincoli, delle caratteristiche che li rendono in un qualche senso 'tipici'. Da secoli i matematici studiano e modellano il comportamento di dadi e di monetine, come paradigmi della casualità. Ora, sia chiaro che casualità ed imprevedibilità non sono concetti equivalenti. Osservate infatti questa semplice sequenza di numeri:

7, 0, 5, 10, 3, ?

Se vi chiedessero di provare a completare la sequenza, a quanti di voi verrebbe in mente di suggerire 8? Certamente, molte altre risposte sarebbero accettabili; tuttavia noi avevamo in mente le ore dell'orologio ed abbiamo considerato come *seme* le 7 del mattino: andando avanti costantemente di 5 ore siamo arrivati alle 12 (ossia, le 'zero'), alle 5 del pomeriggio, alle 10 di sera, alle 3 di notte, ed alle 8 del mattino successivo. Evidentemente questa non è affatto una sequenza casuale, ma è **deterministica**. Ecco un modo per realizzarla con R:

```
orologio = seq(from = 7, by = 5, length.out = 6 )
orologio
orologio %% 12

> orologio = seq(from = 7, by = 5, length.out = 6 )
> orologio
[1] 7 12 17 22 27 32
> orologio %% 12
[1] 7 0 5 10 3 8
```

Ripetiamo, si trattava di una sequenza che sembrava essere, all'inizio, difficilmente prevedibile. Ebbene i linguaggi di programmazione come R hanno dei metodi molto efficienti per generare sequenze di numeri praticamente imprevedibili; questi ultimi vengono detti **numeri pseudocasuali**.



Per chi fosse interessato, in <https://random.org/> si sfruttano le perturbazioni radio generate dai fenomeni atmosferici, come per esempio le scariche elettriche temporalesche (il cosiddetto rumore atmosferico), per ottenere dei numeri 'veramente casuali' e non generati da alcun algoritmo algebrico.

### 2.2.1 Randomizzare i pazienti

**Problema 2.2** ... stiamo per iniziare una sperimentazione in doppio cieco e dobbiamo randomizzare 30 pazienti. Come potremmo farlo con il computer?

Cominciamo a vedere l'impiego della funzione `sample` di R, la quale estrae in ordine pseudo-casuale gli elementi di un vettore, con un esempio molto semplice. Se per scherzo volessimo giocare a tombola, potremmo estrarre le prime 5 palline in questo modo, creando una sequenza di numeri interi che partono da 1 e si fermano a 90, a passo unitario:

```
tombola = seq(from = 1, to = 90, by = 1)
sample(tombola, 5)

> tombola = 1:90
> sample(tombola, 5)
[1] 69 14 62 31 43
```

**Quesito 2.2.1** Nel gioco della tombola di volta in volta, in maniera imprevedibile, sortiscono dei numeri. Anche nel gioco della roulette nelle case da gioco i numeri si susseguono in maniera imprevedibile. Riuscite a cogliere delle analogie, o delle differenze, tra i due giochi?

Naturalmente, nel gioco della tombola le palline vengono estratte dal sacchetto una alla volta, e fino a che il gioco non finisce esse non vengono più riposte nel sacchetto, o come si dice anche, 'reimbussolate', 'rimpiazzate'. Al contrario, quando lanciamo una monetina e giochiamo ripetutamente a testa o croce, i due eventi aleatori si possono ripetere ad ogni lancio. Si parla in tal caso di estrazioni con rimpiazzo; con R possiamo inserire in `sample` l'opzione `replace = TRUE`:

```
monetina = sample(c(0,1), size = 5, replace = TRUE)
monetina

> monetina = sample(c(0,1), size = 5, replace = TRUE)
> monetina
[1] 1 1 0 0 1
```

Concretamente, qui abbiamo preso in esame un ipotetico sacchetto che contiene il simbolo 0 ed il simbolo 1; abbiamo chiesto al software di estrarre imprevedibilmente uno dei due simboli, scriverlo a video, reinserirlo nel sacchetto, e di volta in volta ripetere questa procedura per cinque volte. È quella che in matematica si chiama **distribuzione binomiale**, relativa alla **variabile aleatoria di Bernoulli**: ne riparliamo fra poco. Attenzione, però: se voi provate ad eseguire questi comandi sicuramente non otterrete la sequenza che abbiamo ottenuto qui sopra. Per farlo, decidiamo di utilizzare lo stesso seme per generare i numeri pseudo-casuali, inserendolo nella funzione `set.seed`. E questo è molto utile nei nostri lavori di ricerca quando pubblichiamo il codice che abbiamo utilizzato per assicurare al lettore la **riproducibilità** dell'esperimento.

L'esempio della monetina fornisce allora una soluzione semplice al problema 2.2: se noi abbiamo due bracci sperimentali, diciamo controllo = 0 e trattamento = 1, potremmo utilizzare il codice appena visto. Ma guardate questa stranezza. Decidiamo di prendere cinque dadi, e lanciarli, in modo da generare casualmente il seme. Per caso, abbiamo ottenuto 4, 6, 5, 2 ed 1:



Utilizziamo questo numero come seme per 'randomizzare' i 30 pazienti. Osservate che la sequenza casuale generata (e vi assicuriamo che non c'è alcun imbroglio da parte nostra) se per caso ci fosse capitata in una serata alla roulette del Casino ci avrebbe immediatamente portato a pensare a qualche trucco, a qualche frode; estrarre da un sacchetto una pallina bianca per tre volte ed una pallina nera per ventisette volte, e non dubitare che nel sacchetto ci siano (molte) di più nere che bianche.. strano, no?

```
set.seed(46521)
sample(c(0,1), size = 30, replace = TRUE)

> set.seed(46521)
> sample(c(0,1), size = 30, replace = TRUE)
[1] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
```

Per non correre rischi come questi, alcuni metodologi (ad esempio [43, pagina 340]) propongono un interessante approccio a questo problema, la **randomizzazione adattiva**: una tecnica che cerca di controllare, almeno nella fase iniziale della randomizzazione, sequenze 'troppo lunghe' di attribuzione allo stesso braccio sperimentale. La strategia è molto semplice: si parte idealmente con un sacchetto che contiene una pallina bianca ed una pallina nera; si estrae la prima pallina (ad esempio, bianca), la si reimbussola e si aggiunge una pallina del colore opposto (nell'esempio, alla seconda estrazione il sacchetto conterebbe una bianca e due nere, aumentando la probabilità di far uscire l'evento complementare, la nera). Ecco un possibile codice:

```

set.seed(46521)
urna = c(0,1)
sequenza = NULL
for(i in 1:30)
{
  estrazione = sample(urna, size = 1)
  sequenza = c(sequenza, estrazione)
  urna = c(urna, 1-estrazione)
}
sequenza
table(urna)

> set.seed(46521)
> urna = c(0,1)
> sequenza = NULL
> for(i in 1:30)
+ {
+   estrazione = sample(urna, size = 1)
+   sequenza = c(sequenza, estrazione)
+   urna = c(urna, 1-estrazione)
+ }
> sequenza
[1] 1 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 0 1 0 0 1 1
> table(urna)
urna
 0   1
 18 14

```

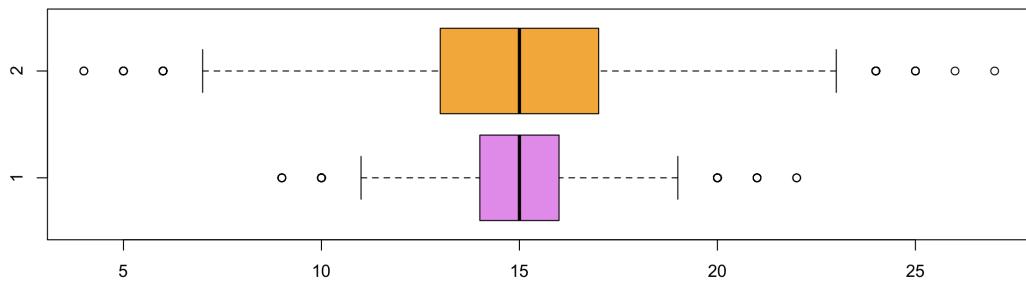


Figura 2.1: Randomizzazione adattiva: il boxplot arancione mostra che simulando per centomila volte il lancio di trenta monetine, non è improbabile che appaiano sequenze che hanno un numero di teste (o croci) anche di molto superiori a venti (o inferiori a dieci). Nel boxplot viola, raffigurante centomila 'randomizzazioni adattive', questo non accade.

Descriviamo il codice passo per passo. Si inizia con l'urna contenente la pallina bianca, 0, e la pallina nera, 1; in partenza, inoltre, abbiamo una sequenza vuota (ossia NULL) di pazienti 'randomizzati'. Si ripetono poi per 30 volte queste istruzioni:

1. dalla urna si ottiene con `sample` un'estrazione casuale di `size = 1`;
2. l'estrazione ottenuta va ad aggiornare la sequenza, aggiungendosi in coda;
3. nell'urna viene aggiunta una pallina di colore complementare a quella appena estratta; infatti se `estrazione` vale 0, allora `1 - estrazione` vale 1; e viceversa.

Innanzitutto si osservi che alla fine nell’urna risultano esserci 18 palline 0 contro 14 palline 1, ed è del tutto ragionevole ritenere che lo ‘sbilanciamento’ tra le palline nell’urna vada via via a ridursi, con l’aumentare della dimensione dei pazienti da ‘randomizzare’. Per quanto riguarda invece la sequenza che determina la randomizzazione dei pazienti, essa risulta essere ‘ben mescolata’ e composta da 13 pazienti 0 contro 17 pazienti 1. La figura 2.1 vi mostra che questa situazione ‘bilanciata’ è tipica della randomizzazione adattiva, nella quale le sequenze tendono ad avere frequenze degli eventi non troppo eterogenee tra loro.

## 2.2.2 Simulare una mutazione genica

**Problema 2.3** .. e come potremmo simulare uno ‘snip’ nella proteina Zinc Finger 521?

Riprendiamo in esame il dataset `zincfinger` della sezione precedente; siamo ora interessati a simulare un polimorfismo a singolo nucleotide, modificando casualmente uno dei nucleotidi. Scegliamo innanzitutto una posizione casuale del vettore `Q96K83`:

```
N = length(Q96K83)
posizioneacaso = sample(1:N, size = 1)
posizioneacaso

> N = length(Q96K83)
> posizioneacaso = sample(1:N, size = 1)
> posizioneacaso
[1] 850
```

Nel nostro esempio, abbiamo ottenuto la 850-esima posizione; andiamo dunque a leggere l’850-esimo nucleotide di `Q96K83`:

```
Q96K83[posizioneacaso]

> Q96K83[posizioneacaso]
[1] G
Levels: A C G T
```

Per essere più comprensibili, convertiamo quel nucleotide `T` in formato numerico:

```
as.numeric(Q96K83[posizioneacaso])

> as.numeric(Q96K83[posizioneacaso])
[1] 3
```

La mutazione casuale dovrà trasformare quel numero 3 in uno qualsiasi degli altri numeri non presenti, ossia 1, 2, e 4; ricordandoci dei concetti di base dell’algebra, dobbiamo andare a prendere in considerazione l’insieme complementare di quel numero 3, e sceglierne un elemento a caso. Il complementare in R si ottiene con l’operazione di differenza di insiemi, `setdiff`:

```
glialtritre = setdiff(1:4, as.numeric(Q96K83[posizioneacaso]))
glialtritre

> glialtritre = setdiff(1:4, as.numeric(Q96K83[posizioneacaso]))
> glialtritre
[1] 1 2 4
```

e da tale insieme complementare `glialtritre` estraiamo un elemento a caso:

```
mutazione = sample(glialtritre, 1)
mutazione

> mutazione = sample(glialtritre, 1)
> mutazione
[1] 4
```

Siamo pronti. Adesso che abbiamo la `mutazione`, 4 invece di 3, la ri-convertiamo in nucleotide con il comando `levels`:

```
levels(Q96K83)[mutazione]

> levels(Q96K83)[mutazione]
[1] "T"
```

e sostituiamo questa lettera T al posto della G che si trovava nella 850-esima posizione:

```
Q96K83[posizioneacaso] ## prima
Q96K83[posizioneacaso] = levels(Q96K83)[mutazione]
Q96K83[posizioneacaso] ## dopo

> Q96K83[posizioneacaso] ## prima
[1] G
Levels: A C G T
> Q96K83[posizioneacaso] = levels(Q96K83)[mutazione]
> Q96K83[posizioneacaso] ## dopo
[1] T
Levels: A C G T
```

Ora abbiamo tutti gli ingredienti per scrivere la nostra funzione che crea di volta in volta una mutazione casuale; la chiameremo `snip`:

```
snip = function(x)
{
  posizioneacaso = sample(1:length(x), 1)
  print("posizione a caso")
  print(posizioneacaso)
  print("mutazione")
  print(x[posizioneacaso])
  glialtritre = setdiff(1:4, as.numeric(x[posizioneacaso]))
  mutazione = sample(glialtritre, 1)
  x[posizioneacaso] = levels(x)[mutazione]
  return(x[posizioneacaso])
}
```

Vediamola in azione; digitiamo `snip(Q96K83)`:

```
snip(Q96K83)
```

```

> snip = function(x)
+ {
+   posizioneacaso = sample(1:length(x), 1)
+   print("posizione a caso")
+   print(posizioneacaso)
+   print("mutazione")
+   print(x[posizioneacaso])
+   glialtritre = setdiff(1:4, as.numeric(x[posizioneacaso]))
+   mutazione = sample(glialtritre, 1)
+   x[posizioneacaso] = levels(x)[mutazione]
+   return(x[posizioneacaso])
+ }
> snip(Q96K83)
[1] "posizione a caso"
[1] 850
[1] "mutazione"
[1] G
Levels: A C G T
[1] T
Levels: A C G T

```

## 2.3 Le variabili aleatorie con R

Noialtri matematici siam persone che amano l'astratto più di Vasilij Kandinskij ed Immanuel Kant messi assieme; ma quando ci mettiamo a parlare nel nostro linguaggio riusciamo a far inorridire il resto del mondo in pochi secondi. Leggete questa definizione (originariamente in inglese, la traduzione è mia) [48, pagina 82]:

Sia  $\Omega$  uno spazio campionario con una classe di eventi  $\zeta$ . Ogni regola  $X$  che assegna ad ogni  $\omega \in \Omega$  un numero reale  $X(\omega)$  è detta **variabile aleatoria**.

Ecco, ci siamo capiti: dobbiamo affrontare un tema che risulta essere di capitale importanza per il prosieguo del testo, ma che purtroppo parte da un background ostico, a dir poco. Eppure al giorno d'oggi l'**approccio bayesiano** alla biostatistica sta guadagnando sempre maggior favore (e ne parleremo a tempo debito) e lì il concetto di variabile aleatoria è pervasivo, anzi: fondamentale. Ma non possiamo pretendere che biologi, biotecnologhe, chimici, chirurghe, e via via in ordine alfabetico sino ad urologhe e veterinari, trovino il tempo e la voglia necessari per affrontare i dettagli matematici della questione. Quindi, nel seguito di questa sezione andremo a conoscere il significato delle idee di tre importanti matematici dei secoli passati, ma interpretandole nei nostri ambiti scientifici.

### 2.3.1 Jacob Bernoulli e gli eventi dicotomici

*Il mondo si divide in due:  
quelli che conoscono Gianni Morrone,  
e quelli che non conoscono Gianni Morrone.*

*Levi Beverly  
University of Louisville School of Medicine*

Ecco, a proposito di eventi dicotomici, no? Riprendiamo gli argomenti che abbiamo introdotto nella sezione 2.2.1, in particolare pensiamo all'esempio del lancio di una monetina, la quale presenta

una probabilità  $p = \frac{1}{2}$  di dare origine all'evento testa, ed  $1 - p = 1 - \frac{1}{2} = \frac{1}{2}$  all'evento croce (.. approssimativamente! Infatti quando ero un giovane insegnante supplente, lanciando una monetina in classe per fare gli esperimenti del calcolo delle probabilità la monetina imprevedibilmente cadde a terra rimanendo verticale, in equilibrio sul bordo, tra lo stupore di tutta la classe; purtroppo a quel tempo il telefono cellulare non era ancora stato inventato, ed ovviamente non avevo alcuna macchina fotografica con me, quindi non abbiamo alcuna documentazione a suffragio della mia affermazione e pertanto siete liberi di dubitarne della veridicità – se questa circostanza fosse capitata al professor Morrone, il quale senza fallo aveva con sé almeno un paio di macchine fotografiche, oggi avremmo la foto da mostrarvi).

Puntiamo i nostri riflettori sul matematico svizzero Jacob Bernoulli, autore del libro *Ars Conjectandi*, caposaldo della teoria della probabilità, esponente della grande famiglia fiamminga di scienziati e letterati, vissuto nella seconda metà del 1600 a Basilea, che è noto anche con i nomi James, o Jacques: la variabile aleatoria di Bernoulli si può schematizzare in questo modo:

$$\begin{pmatrix} 0 & 1 \\ 1-p & p \end{pmatrix}$$

Nella prima riga vengono elencati gli eventi possibili, e nella seconda riga le probabilità teoriche con cui essi si manifestano. Come esempio guida consideriamo il dataset `raul` (acronimo di risk assessment in uterine lesions) tratto dalla ricerca di Annalisa Di Cello [19]:

```
www = "http://www.biostatisticaumg.it/dataset/raul.csv"
raul = read.csv(www, header = TRUE)
attach(raul)
names(raul)
table(Esito)
length(Esito)
table(Esito)/length(Esito)

> www = "http://www.biostatisticaumg.it/dataset/raul.csv"
> raul = read.csv(www, header = TRUE)
> attach(raul)
> names(raul)
[1] "Anni"      "Ca125"     "Ldhuno"    "Ldhtre"    "Esito"
[6] "Diagnosi"
> names(raul)
[1] "Anni"      "Ca125"     "Ldhuno"    "Ldhtre"    "Esito"     "Diagnosi"
[6] "Esito"
> table(Esito)
Esito
benigno maligno
 1568      42
> length(Esito)
[1] 1610
> table(Esito)/length(Esito)
Esito
benigno maligno
0.97391304 0.02608696
```

Nel dataset sono raccolte informazioni sulle età e sui valori degli isoenzimi lattato-deidrogenasi di 42 pazienti affette da sarcoma uterino e 1568 pazienti affette da lesioni uterine benigne (adenomiosi e fibromi). Abbiamo quindi una **stima frequentista** della probabilità  $p = 0.02$  (arrotondiamo

per difetto quel 0.026, perché, a scopi didattici, abbiamo eliminato quasi un migliaio di altre pazienti benigne) con la quale si manifesta la patologia maligna tra le 1610 pazienti affette da una ben determinata sintomatologia; la variabile aleatoria bernoulliana che descrive l'outcome è pertanto (0 benigno, 1 maligno):

$$\begin{pmatrix} 0 & 1 \\ 0.98 & 0.02 \end{pmatrix}$$

**Problema 2.4** *Reclutando le prossime dieci pazienti, quale potrebbe essere la probabilità che non vi sia alcun sarcoma tra di esse?*

**Problema 2.5** *Mi sarebbe utile riuscire a stimare quante pazienti dovrò arruolare nello studio prospettico per poter osservare una trentina di sarcomi uterini.*

Questi due problemi tipici si affrontano proprio utilizzando la variabile aleatoria bernoulliana. Vediamo come.

### La distribuzione binomiale

Quando effettuiamo delle prove ripetute con la variabile aleatoria bernoulliana, si parla di distribuzione binomiale degli eventi. Un modo per affrontare empiricamente il problema 2.4 potrebbe essere quello di usare la funzione `sample`, estraendo casualmente centomila pazienti, attribuendo in partenza le opportune probabilità  $p$  degli eventi:

```
caso = sample(c(0,1), size = 100000, prob = c(0.98, 0.02), replace = TRUE)
```

Un'idea operativa potrebbe essere ora quella di ri-arrangiare i centomila eventi casuali in una tabellona (nome matematico: **matrice**) di diecimila righe e dieci colonne, e di fare per diecimila volte la somma dei valori contenuti in ciascuna riga. Questo si può fare con R, anche se i comandi sono un po' tecnici:

```
tabellona = matrix(caso, nrow = 10000, ncol = 10)
somme = apply(tabellona, MARGIN = 1, FUN = sum)
table(somme)

> tabellona = matrix(caso, nrow = 10000, ncol = 10)
> somme = apply(tabellona, MARGIN = 1, FUN = sum)
> table(somme)
somme
 0    1    2    3    4 
8158 1669 167   5   1
```

Cosa osserviamo? La simulazione ci mostra che potremmo ipotizzare una risposta empirica al problema 2.4, stimando attorno all'81.6% la probabilità di non imbattersi in alcun sarcoma arruolando dieci pazienti sintomatiche. Ebbene, Jacob Bernoulli avrebbe saputo rivelarvi una formula esatta per il calcolo teorico di queste probabilità, e tale formula è implementata nella funzione `dbinom` di R (il nome `dbinom` è eloquente: **densità** di probabilità della distribuzione **binomiale**)

```
teoriche = dbinom(0:10, size = 10, prob = 0.02)
round(teoriche, 3)[1:5]
```

```
> teoriche = dbinom(0:10, size = 10, prob = 0.02)
> round(teoriche, 3)[1:5]
[1] 0.817 0.167 0.015 0.001 0.000
```

Spieghiamo i comandi. Se arruolassimo `size = 10` nostre pazienti, ciascuna delle quali ha una `prob = 0.02` di avere una neoplasia uterina maligna, potremmo essere interessati a conoscere quale sia la probabilità che nessuna di esse abbia una neoplasia; oppure che una di esse ce l'abbia; oppure due, tre, eccetera, eccetera, sino a conoscere la probabilità che tutte e dieci abbiano il tumore. Indichiamo questa sequenza, i cosiddetti **quantili della distribuzione**, con il simbolo `0:10`. Adesso, vogliamo calcolare queste probabilità teoriche, e lo facciamo con la densità della binomiale, `dbinom`. Ci basta conoscere le prime tre cifre decimali dopo la virgola, e per questo le arrotondiamo con `round`. E ci accontentiamo di visualizzare le prime cinque di queste undici probabilità, per confrontarle con la soluzione empirica che `sample` ci aveva fornito nella tabellona, e per questo scriviamo `[1:5]`.



Se vi interessano le formule della probabilità degli eventi binomiali, le trovate ad esempio su Wikipedia [https://it.wikipedia.org/wiki/Distribuzione\\_binomiale](https://it.wikipedia.org/wiki/Distribuzione_binomiale)

Ora vale la pena perdere qualche minuto di tempo per capire come si legano tra loro la densità di probabilità (talvolta detta anche **probabilità di massa**) `dbinom` e la probabilità cumulativa (ovvero la **funzione di distribuzione**) `pbinom`. Facciamo un esempio semplicissimo, il lancio di una monetina ripetuto per due volte, pensandolo in maniera equivalente a quello di lanciare due monetine uguali per una sola volta: la probabilità di avere testa e testa è  $1/4 = 0.25$ , così come croce e croce; invece la probabilità di fare una testa ed una croce, oppure una croce ed una testa, è  $2/4 = 0.50$ . Interpretiamo queste affermazioni con i comandi di R:

**Esercizio 2.5** Provate passo per passo ad eseguire questi comandi e spiegate il loro output:

```
dbinom(x = 0, size = 2, prob = 0.5)
pbinom(q = 0, size = 2, prob = 0.5)

dbinom(x = 1, size = 2, prob = 0.5)
dbinom(x = 0, size = 2, prob = 0.5) + dbinom(x = 1, size = 2, prob = 0.5)
pbinom(q = 1, size = 2, prob = 0.5)

dbinom(x = 2, size = 2, prob = 0.5)
dbinom(x = 0, size = 2, prob = 0.5) + dbinom(x = 1, size = 2, prob = 0.5)
+ dbinom(x = 2, size = 2, prob = 0.5)
pbinom(q = 2, size = 2, prob = 0.5)
```

### La distribuzione binomiale negativa

Negli studi clinici prospettici è essenziale, sia dal punto di vista etico che da quello economico, fissare a priori un criterio di arresto dello studio in modo tale da garantire di non arruolare troppi pazienti – senza però correre il rischio di averne arruolati troppo pochi ed avere grande inaffidabilità nelle stime statistiche di interesse. Il problema 2.5 riguarda proprio questo aspetto, il quale si affronta sempre sfruttando la variabile aleatoria bernoulliana, in quella che viene definita distribuzione di Blaise Pascal, nel caso discreto, o **binomiale negativa** nel caso continuo. Attualmente le distribuzioni aleatorie di questo tipo sono di interesse, ad esempio, nell’analisi differenziale di espressione dei geni [47].



Potete dare un'occhiata su Wikipedia [https://it.wikipedia.org/wiki/Distribuzione\\_di\\_Pascal](https://it.wikipedia.org/wiki/Distribuzione_di_Pascal) per farvi qualche idea iniziale.

Da un punto di vista matematico, siamo interessati al numero di insuccessi che si debbono ottenere prima di giungere all'ennesimo successo. Da un punto di vista semantico è raccapriccante che si interpreti un sarcoma maligno come un 'successo'; ma dal punto di vista lessicale lo si deve interpretare come il participio del verbo 'succedere', sinonimo di 'accadere', 'capitare': vogliamo stimare quante pazienti benigne dobbiamo arruolare prima di aver annoverato trenta sarcomi. Possiamo anche qui cercare di simulare il problema, adattando un'idea di Michael Crawley [16, pagine 787-788]:

```
set.seed(1234)
caso = sample(c(0,1), size = 100000, prob = c(0.98, 0.02), replace = TRUE)
sarcomi = which(caso == 1)
sarcomi[1:30]

> set.seed(1234)
> caso = sample(c(0,1), size = 100000, prob = c(0.98, 0.02), replace = TRUE)
> sarcomi = which(caso == 1)
> sarcomi[1:30]
[1] 39 123 142 149 156 158 192 199 210 240 245 302 308 355 389
[16] 445 453 536 562 726 730 739 868 899 918 932 971 985 988 993
```

Dopo aver fissato con `set.seed(1234)` il generatore di numeri casuali, per assicurare la riproducibilità del codice, riprendiamo il comando che abbiamo usato nella sezione precedente per simulare centomila pazienti con una delle due patologie, a caso. Ci chiediamo, con la funzione `which`, quali siano le pazienti con sarcoma, per le quali `caso == 1`. Creiamounque il vettore `sarcomi` che contiene le posizioni in cui il vettore `caso` è uguale ad uno. Visualizziamo ora le prime 30 posizioni: la prima risulta essere la paziente numero 39, mentre la trentesima è la 993-esima. Sappiamo che cambiando seme cambia anche la sequenza casuale delle pazienti, e pertanto proviamo a ripetere con un ciclo `for` per mille volte questa simulazione:

```
numpazienti = numeric(1000)
for(i in 1:1000)
{
  caso = sample(c(0,1), size = 100000, prob = c(0.98, 0.02), replace = TRUE)
  sarcomi = which(caso == 1)
  numpazienti[i] = sarcomi[30]
}
summary(numpazienti)

> numpazienti = numeric(1000)
> for(i in 1:1000)
+ {
+   caso = sample(c(0,1), size = 100000, prob = c(0.98, 0.02), replace = TRUE)
+   sarcomi = which(caso == 1)
+   numpazienti[i] = sarcomi[30]
+ }
> summary(numpazienti)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  781    1276    1461    1484    1668    2439
```

Dal `summary` osserviamo per esempio che il valore mediano, ossia il 50% percentile della distribuzione casuale, vale 1461. Ebbene, vediamo che la funzione `qnbnom`, la quale determina il **quantile della binomiale negativa**, propone la medesima risposta:

```
qnbnom(p = 0.50, size = 30, prob = 0.02)
> qnbnom(p = 0.50, size = 30, prob = 0.02)
[1] 1454
```

Viceversa, nella nostra prima simulazione, avevamo ottenuto come stima il numero di 993 pazienti, che risulta essere poco meno del terzo percentile della distribuzione. Lo vediamo con la **probabilità della binomiale negativa**:

```
pnbnom(q = 993, size = 30, prob = 0.02)
> pnbnom(q = 993, size = 30, prob = 0.02)
[1] 0.02689502
```

**Esercizio 2.6** Per avere un grado di fiducia del 90 per cento di osservare una trentina di sarcomi, quante pazienti dovremmo arruolare nello studio?

```
qnbnom(p = 0.90, size = 30, prob = 0.02)
```

■

### 2.3.2 L'equazione più pericolosa, parte seconda

Facciamo ancora qualche simulazione; supponiamo di volere fare `n = 15` esperimenti relativi al lancio contemporaneo di `size = 2` monetine (vedi Esercizio 2.5), e di sommare il numero di teste uscite in ciascuno dei quindici esperimenti. Abbiamo a disposizione la funzione `rbinom`, che genera numeri casuali distribuiti binomialmente (ove `r` sta per `random`):

```
set.seed(234)
rbinom(n = 15, size = 2, prob = 0.5)
> rbinom(n = 15, size = 2, prob = 0.5)
[1] 1 2 0 2 0 1 2 1 2 1 1 1 1 1 0
```

Vediamo che qualche volta si sono verificate 2 teste, altre volte 0 teste; per lo più, 1 testa. Bene; ritorniamo alla questione lasciata in sospeso nella sezione 1.4.2, circa la definizione dell'errore standard della media. Ci riproponiamo ora di fare quattro esperimenti con `rbinom`, ripetendo per ciascuno di essi un milione (sì, un milione) di lanci:

1. lancio contemporaneo di dieci monetine
2. lancio contemporaneo di cento monetine
3. lancio contemporaneo di mille monetine
4. lancio contemporaneo di diecimila monetine

Fatto questo, creiamo una tabella, nella quale inizialmente indichiamo le medie e le deviazioni standard dei quattro esperimenti.

```
set.seed(3456)
monete10    = rbinom(n = 10^6, size = 10,    prob = 0.5)
monete100   = rbinom(n = 10^6, size = 100,   prob = 0.5)
monete1000  = rbinom(n = 10^6, size = 1000,  prob = 0.5)
monete10000 = rbinom(n = 10^6, size = 10000, prob = 0.5)
```

Dalle tre colonne di sinistra abbiamo già una prima scoperta, che forse non ci aspettavamo: se decuplichiamo il numero di monetine, anche la media si decuplica, ma non la deviazione standard; quest'ultima non cresce con un fattore 10, bensì con il fattore  $\sqrt{10} = 3.16..$  (infatti  $50/15.8 = 15.8/5 = 5/1.6 = 3.16..$ ). Ma la scoperta delle scoperte si realizza quando nel dividere le deviazioni standard per la radice quadrata del numero di monete nell'esperimento si trova un numero che non cambia mai:

esperimento	<i>n</i>	media	dev. st. $\sigma$	$\sqrt{n}$	s.e. $\sigma/\sqrt{n}$
monete10	10	5.00	1.60	3.16	0.50
monete100	100	50.00	5.00	10	0.50
monete1000	1000	500.00	15.80	31.62	0.50
monete10000	10000	5000.00	50.00	100	0.50

Colpo di scena: abbiamo reperito una quantità,  $\sigma/\sqrt{n}$ , che 'non si muove' quando aumentiamo la quantità di volte con la quale effettuiamo il nostro esperimento. Allora possiamo ritenere che l'errore standard sia l'oggetto algebrico 'giusto' per misurare l'affidabilità delle nostre stime statistiche, poiché esso appare essere una caratteristica dell'esperimento non legata alla quantità di volte per le quali l'esperimento viene ripetuto. Ci fermiamo per il momento qui, anche se la questione ha ancora degli aspetti da scoprire che discuteremo nella sezione 2.3.5.

**Quesito 2.3.1** Dopo che avrete svolto l'Attività 2.5, ritornate a pensare alla tabella qui sopra, e scoprite la ragione del fatto che, in questi esperimenti,  $\frac{\sigma}{\sqrt{n}} = 0.5$ .

### 2.3.3 Siméon Poisson e la conta degli eventi

È ora la volta di un matematico francese, Siméon Denis Poisson: la sua variabile aleatoria troneggia in una quantità di pacchetti di R, quali ad esempio dpcR per l'analisi della reazione a catena della polimerasi digitale, o di edgeR, il pacchetto per l'analisi empirica dell'espressione genica.



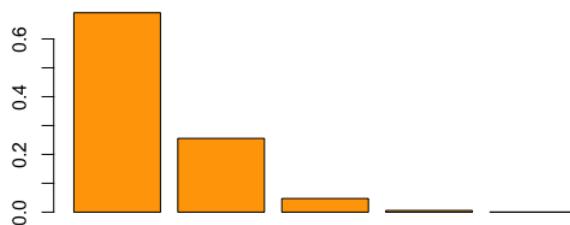
Se vi interessa approfondire questi due discorsi, iniziate a dare un'occhiata da qui: <https://cran.r-project.org/web/packages/dpcR/dpcR.pdf> e <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2796818/>

E pensare che in origine la variabile aleatoria di Poisson veniva definita come la 'variabile aleatoria degli eventi rari' perché si può idealmente ottenere dalla variabile binomiale ripetuta per un numero di volte  $n$  molto elevato, e con una probabilità di successo  $p$  molto piccola (esempio tipico [48]: calcolare la probabilità che qualche centinaio di fanti della prima guerra mondiale armati di moschetto, sparando contemporaneamente, colpiscono il velivolo che li sorvola). Vogliamo invece qui citare un esempio di dosimetria di radiazione e di probabilità di controllo locale del tumore in radioterapia oncologica [33, pagina 85]. Si immagini che una dose di radiazione provochi una certa quantità di 'colpi letali' distribuiti casualmente sulla popolazione di cellule tumorali. Ad esempio, alcune cellule ricevono un 'colpo letale' e muoiono; ma alcune cellule non muoiono e necessitano ancora di un altro 'colpo letale' ed un altro ancora, e così via. Supponiamo che ad una dose di radiazione sufficiente a provocare, in media, un 'colpo letale' sopravvivano in media il 37 per cento di cellule clonogeniche ( $\lambda = 0.37$ ). Possiamo ideare un modello di tumore composto da  $n = 36$  cellule clonogeniche con la funzione rpois (come nella sezione precedente, r sta per **random**), arrangiando i valori ottenuti in una **matrice quadrata** e confrontando la frequenza degli esiti con le probabilità teoriche di dpois:

```
colpiletali = rpois(n = 36, lambda = 0.37)
```

```
matrix(colpiletali, nrow = 6)
barplot(dpois(x = 0:4, lambda = 0.37), col = "orange")
```

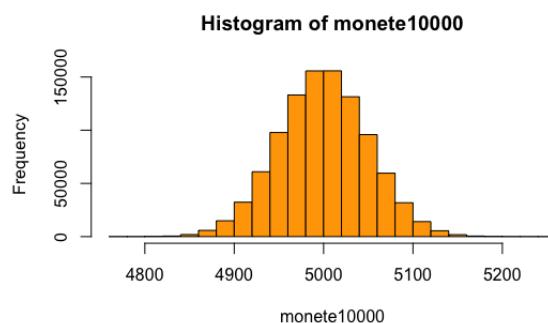
0	0	2	1	4	0
3	0	0	1	0	0
0	0	1	2	1	1
0	1	0	3	0	0
1	0	1	0	0	2
0	0	0	1	0	1



### 2.3.4 Carl Gauss, o della normalità

Ora qui, ci dovremmo alzare tutti in piedi, al cospetto del matematico più importante di tutti i tempi: Johann Friedrich Carl Gauß. La sua variabile aleatoria è veramente rilevante e sbuca fuori da ogni angolo. Vi ricordate l'esempio delle monetine lanciate per una milionata di volte? Ebbene, date un'occhiata all'istogramma e ditemi se non è stupefacente vedere una campana così perfetta.

```
monete10000 = rbinom(n = 10^6, size = 10000, prob = 0.5)
hist(monete10000, col = "orange")
```



E se pensate che ci sia qualche 'imbroglio', magari legato alla simmetria della monetina ed al fatto che  $p = 1 - p = 0.5$  proviamo a svolgere l'Esercizio 2.7, in cui la simmetria non c'entra affatto. E così avremo a disposizione un'ulteriore verifica dell'importante *Central Limit Theorem* di Lindeberg e Lévy, in cui si dimostra che pur partendo da una variabile aleatoria qualunque (non

solo la bernoulliana e la Poisson, ma proprio qualunque!), l'insieme delle medie che si genera ripetendo millanta volte l'esperimento tende a disporarsi secondo la gaussiana.

**Esercizio 2.7** Riprendete il dataset cholesterol del primo capitolo, ed affiancate all'istogramma viola dei valori del colesterolo HDL dei donatori l'istogramma arancione delle medie di diecimila campioni di cento donatori estratti a caso. Distinguete l'asimmetria dei dati dalla perfetta forma a campana delle medie.

```
indirizzo = "http://www.biostatisticaumg.it/dataset/cholesterol.csv"
cholesterol = read.csv(indirizzo, header = TRUE)
attach(cholesterol)
medieHDLchol = numeric(10000)
for (i in 1:10000) {medieHDLchol[i] = mean(sample(HDLchol, size = 100))}
par(mfrow = c(1,2))
hist(HDLchol, col = "violet")
hist(medieHDLchol, col = "orange")
```



Andate a cercare sul sito di Philipp Plewa l'animazione dedicata al The Central Limit Theorem: <https://bl.ocks.org/pmpblewa>. Fantastica, non è vero? La si vede ancor meglio in una delle pagine interne del Chapter 3, Probability Distributions, Discrete and Continuous <https://seeing-theory.brown.edu/>. Inoltre, avremo modo nelle prossime pagine di celebrare anche il nome del cugino di Charles Darwin, sir Francis Galton. Tuttavia qui vi invitiamo a scoprire il suo gioco del *Quincunx* (antica moneta romana): <https://www.mathsisfun.com/data/quincunx.html>. Volendo, potete vedervelo anche nel vostro R:

```
install.packages("visualization")
library(animation)
quincunx()
```

### La densità della normale

Adesso vogliamo visualizzare per bene l'andamento della gaussiana, e possiamo farlo con il comando grafico `curve`, specificando che ci interessa usare una finestra grafica che parte `from = -3` ed arriva `to = 3`. Vogliamo dunque realizzare il disegno della **densità della normale standard**, `dnorm`, che per definizione ha media 0 e deviazione standard 1, e che di solito sui libri viene indicata con il simbolo Z:

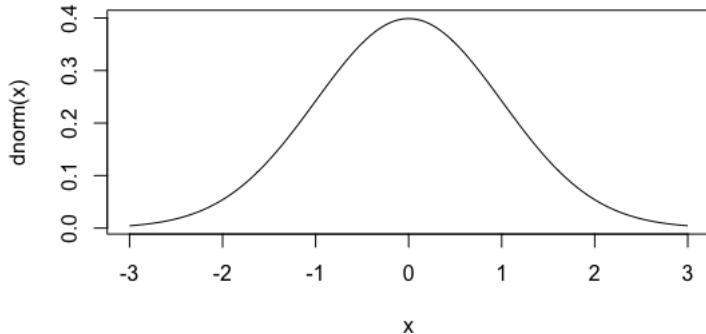
```
curve(dnorm, from = -3, to = 3)
```

**Esercizio 2.8** La variabile aleatoria normale standard è simmetrica rispetto all'origine: il ramo destro della curva è speculare a quello sinistro.

```
dnorm(-1) == dnorm(1)
dnorm(-2.3456) == dnorm(2.3456)
```

### La probabilità della normale

La simmetria della gaussiana ha delle importanti e note ripercussioni per quanto riguarda le aree sotto la curva che in un senso geometrico raffigurano le **probabilità** degli eventi. Ci occupiamo



dunque di pnorm, la **funzione di distribuzione cumulativa** della normale standard, che spesso sui libri talvolta viene indicata con  $\Phi$ , talvolta con  $erf$  (= error function):

```
pnorm(-1)
1 - pnorm(1)
pnorm(1) - pnorm(-1)

> pnorm(-1)
[1] 0.1586553
> 1 - pnorm(1)
[1] 0.1586553
> pnorm(1) - pnorm(-1)
[1] 0.6826895
```

Confrontiamo questi valori con il grafico riportato in rete da Wikipedia:

[https://commons.wikimedia.org/wiki/File:Standard\\_deviations\\_diagram.svg](https://commons.wikimedia.org/wiki/File:Standard_deviations_diagram.svg)

e vediamo che l'area blu compresa tra -1 ed 1 vale proprio  $34.1\% + 34.1\% = 68.2\%$  dell'area totale.

**Esercizio 2.9** Verificate che l'area compresa tra -2 e 2, e tra -3 e 3, valgono rispettivamente il 95.4% ed il 99.7%. ■

### La normale è speciale

Attenzione: questa faccenda del 68%, 95% e 99% è una caratteristica esclusiva della normale, con le altre variabili aleatorie le cose non vanno mica così bene. Riflettiamo su questi due quesiti:

**Quesito 2.3.2** (Bland [6, pagina 120]) Il picco di flusso espiratorio (PEFR) delle ragazze di undici anni si distribuisce normalmente con media 300 litri/minuti e deviazione standard 20 litri/minuto.

**Quesito 2.3.3** (Pärna et al. [42, pagina 10]) Il consumo di alcol in Estonia nel 1994 si attestava su una media di 128 grammi/settimana con una deviazione standard 147 grammi/settimana.

Nel primo caso l'ipotesi di normalità ci autorizza a dedurre che circa il 68 per cento delle ragazzine di undici anni abbiano un PEFR compreso tra 280 e 320; e che circa il 16 per cento di loro possa avere un PEFR minore di 280 litri/minuto. Ma nel secondo caso saremmo in errore a

dedurre che nel 1994 in Estonia circa il 68 per cento delle persone considerate nello studio esibiva un consumo di alcol compreso tra -19 grammi/ settimana e 275 grammi/settimana; anche perché poi sarebbe stato veramente interessante scoprire come facesse il corpo di quel 16 per cento di loro a *produrre* (e non *consumare*, visto il segno negativo) alcol durante la settimana, risolvendo ovviamente per sempre i problemi energetici dell'umanità ... :-)

Cionondimeno, la media e la deviazione standard sono sempre legate tra loro, in una relazione probabilistica valida a prescindere dalla distribuzione aleatoria dei dati. Il matematico russo Pafnutij Čebišëv (facile da leggere, eh?) ha dimostrato l'esistenza di una disegualanza generica che

[https://en.wikipedia.org/wiki/Chebyshev%27s\\_inequality](https://en.wikipedia.org/wiki/Chebyshev%27s_inequality)

### Numeri casuali normali

Abbiamo ancora due notizie da tenere a mente: la prima è che se 'amplifichiamo' la normale standard  $Z$  moltiplicandola per un numero positivo  $\sigma$ , lei si allarga e si abbassa (e la sua deviazione standard da 1 diviene proprio  $\sigma$ ) ma rimane ancora una normale; la seconda è che se 'trasliamo' la normale standard  $Z$  addizionando un numero qualsiasi  $\mu$ , lei rimane ancora una normale, della stessa forma, ma sposta verso destra (o verso sinistra se  $\mu$  fosse negativo) la sua media portandola da 0 a  $\mu$ . Quindi, se ad esempio il peso dei neonati si distribuisce normalmente con peso medio 3500 grammi e deviazione standard 240 grammi, potremmo scegliere due modi del tutto equivalenti per simulare con `rnorm` (ormai abbiamo già capito: random **normale**) il peso di quattro nascituri:

```
set.seed(987)
mu = 3500
sigma = 240
mu + sigma * rnorm(n = 4)
set.seed(987)
rnorm(n = 4, mean = mu, sd = sigma)

> set.seed(987)
> mu = 3500
> sigma = 240
> mu + sigma * rnorm(n = 4)
[1] 3486.317 3564.799 3709.873 3454.317
> set.seed(987)
> rnorm(n = 4, mean = mu, sd = sigma)
[1] 3486.317 3564.799 3709.873 3454.317
```

Equivalentemente, se partiamo come nell'esempio del peso dei neonati da una variabile aleatoria  $X$  che ha la sua media  $\mu$  e la sua deviazione standard  $\sigma$ , il procedimento di **standardizzazione** va a considerare la trasformazione algebrica che trasla la  $X$  e le cambia scala in modo tale da sovrapporsi alla normale standard  $Z$ :

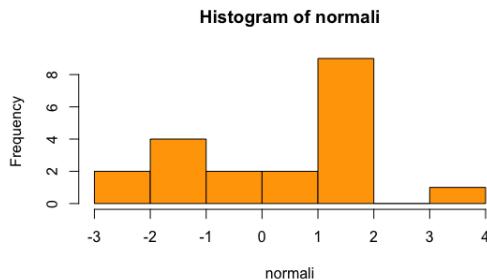
$$Z = \frac{X - \mu}{\sigma}$$

Pertanto, il peso  $x = 3800$  grammi di un neonato verrebbe standardizzato ottenendo quello che talvolta si chiama **z score** calcolando

$$z = \frac{3800 - 3500}{240} = 1.25$$

**Quesito 2.3.4** Vi è mai capitato di sentire la frase 'dobbiamo normalizzare i dati'? A vostro giudizio le parole 'normalizzare' o 'standardizzare' hanno lo stesso significato? Discutiamone.

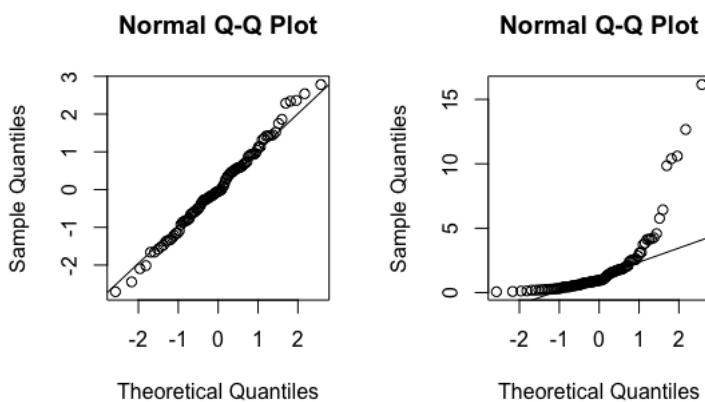
### Il grafico quantile-quantile



Quando generiamo casualmente con `rnorm` un numero esiguo di dati non dobbiamo stupirci se il loro istogramma non ha una forma di campana. Più in generale, l'istogramma non è lo strumento più adeguato per verificare se ciò su cui stiamo investigando si comporta o meno come previsto dalla distribuzione aleatoria di Gauss; a prima vista uno non ci crede, ma i dati che abbiamo raffigurato sono stati generati con questo codice:

```
set.seed(7155)
normali = rnorm(20)
hist(normali, col = "orange")
```

Esiste invece un grafico di immediata percezione che consente di valutare se i dati si adeguano alla normale: il **grafico quantile-quantile**, che tutti chiamano scherzosamente QQ plot. In R si realizza con il comando `qqnorm` e l'idea sottostante è tutto sommato semplice: si contano i dati (per esempio supponiamo che ce ne siano settanta) e si riordinano dal minore al maggiore; in questo modo abbiamo definito quelli che si chiamano i settanta *Sample Quantiles*. Si divide poi la campana di Gauss in settanta fettine di area uguale ad un settantesimo, e si considerano i punti sull'asse  $x$  che delimitano le fettine: abbiamo ora anche i *Theoretical Quantiles*. Rappresentandoli sul piano cartesiano, se si tratta di dati normali essi si adaggeranno sia lungo la diagonale del piano, che in particolare coinciderà con la linea che congiunge il primo con il terzo quartile della normale, che si individua con il comando `qqline`.



Osservate i due grafici; a sinistra i punti si adagiano alla diagonale, tranne qualche lieve imperfezione: possiamo assumere dunque che essi siano stati generati secondo una distribuzione

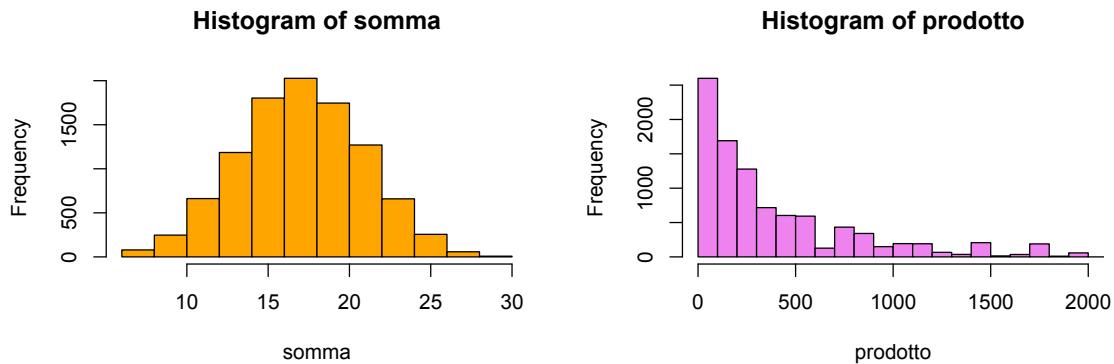
gaussiana. Quando invece nel QQplot si nota una forma curva, o una forma a serpente, si intuisce immediatamente che siamo in presenza di dati non modellabili con la distribuzione aleatoria normale (infatti, due si distribuisce come una variabile aleatoria log-normale di cui parliamo velocemente fra due minuti). Vedrete in seguito nel capitolo 4.1.6 come ci tornerà utile riuscire a distinguere la normalità dei dati per mezzo di questo strumento.

```
uno = rnorm(70)
due = exp(uno)
par(mfrow = c(1,2))
qqnorm(uno)
qqline(uno)
qqnorm(due)
qqline(due)
```

### La distribuzione log-normale

Riflettiamo su un esempio molto ingenuo ma illuminante [37]: supponiamo di avere una concentrazione media di batteri dell'ordine di  $10^6$ ; una divisione cellulare in più o in meno conduce ad una concentrazione di  $2 \cdot 10^6$  o di  $5 \cdot 10^5$ . Quindi in due situazioni ugualmente probabili otteniamo valori del tutto asimmetrici – precisamente moltiplicati o divisi per 2 attorno alla media. Situazioni di questo tipo sono frequentissime nelle scienze della vita, quando le leggi di natura si comportano in maniera moltiplicativa invece che additiva. Vediamolo con una simulazione, lanciando per millanta volte cinque dadi, e sommando (istogramma arancione) o moltiplicando (istogramma viola) gli esiti:

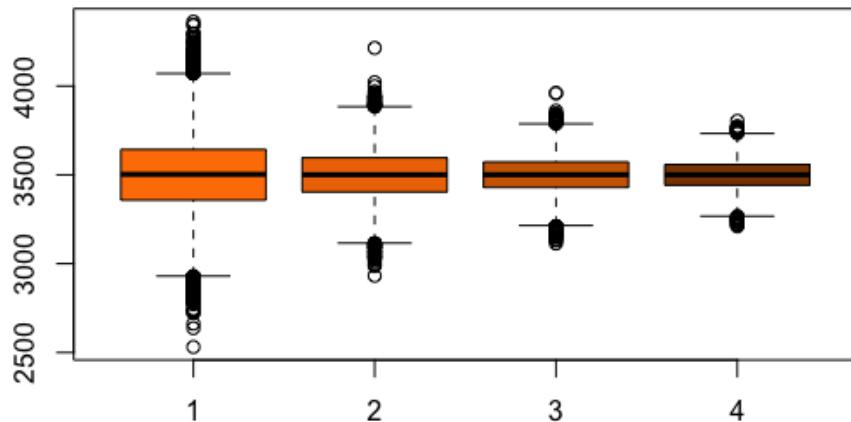
```
d1 = sample(1:6, size = 10000, replace = TRUE)
d2 = sample(1:6, size = 10000, replace = TRUE)
d3 = sample(1:6, size = 10000, replace = TRUE)
d4 = sample(1:6, size = 10000, replace = TRUE)
d5 = sample(1:6, size = 10000, replace = TRUE)
somma = d1 + d2 + d3 + d4 + d5
prodotto = d1 * d2 * d3 * d4 * d5
par(mfrow = c(1,2))
hist(somma, col = "orange")
hist(prodotto, col = "violet", xlim = c(0, 2000), breaks = seq(0,8000,100))
```



**Quesito 2.3.5** Vi ricordate del Problema 1.1? In presenza di una distribuzione dei dati come quella colore viola vi verrebbe ancora voglia di utilizzare la convenzione  $\text{Means} \pm \text{SD}$  per riassumere i dati? Ebbene, vi propongo di cercare su PubMed un centinaio di paper che descrivano il Body Mass Index dei loro pazienti, e di contare quanti lo hanno riassunto indicando media e deviazione standard. E poi vi invito a guardare la Figura 1, tipicamente log-normale, di Fonarow G. et al.: <https://www.sciencedirect.com/science/article/pii/S0002870306008271>

### 2.3.5 L'equazione più pericolosa, parte terza (ed ultima)

Mettiamo un tassello conclusivo ai concetti che riguardano l'errore standard  $\sigma/\sqrt{n}$  che abbiamo commentato e lasciato in sospeso nella sezione 2.3.2 e poi in particolare nell'Esercizio 2.7. Qui vogliamo cercare di illustrare (per l'ultima volta, poi prometto che smetto di parlarne) la relazione che intercorre tra i parametri  $\mu$  e  $\sigma$  che definiscono una variabile aleatoria normale (che nel seguito possiamo considerare come la popolazione di riferimento) e quelli della distribuzione della **media campionaria**, cioè la distribuzione dei dati empirici che si ottengono estraendo dei campioni casuali, di una certa dimensione, dalla popolazione normale.



Dimensione n campione	Media m della M.C.	Deviazione standard s della M.C.
4	3500.4	211.0
9	3500.4	141.5
16	3500.4	106.3
25	3500.4	85.4

L'esempio di questo paragrafo riguarda il peso dei neonati, che supponiamo provenire da una popolazione gaussiana della quale facciamo finta di non conoscere né  $\mu$  né  $\sigma$ . Osservando ripetutamente dei campioni, riusciamo a fornire una stima di  $\mu$  e di  $\sigma$ ? Ebbene, supponiamo che diligentemente le ostetriche Andrea, Bruna, Carla e Diana registrino il peso dei neonati che nascono nel loro ospedale, rispettivamente a gruppi di 4, 9, 16 e 25, e di ogni campione calcolano la media e se la trascrivono diligentemente. Dopo aver registrato millanta e millanta volte le loro medie, le

quattro ostetriche sono in grado di **inferire** (e non usiamo a caso questo verbo così aristotelico) quali possano essere i valori ignoti di  $\mu$  e di  $\sigma$  della popolazione gaussiana ignota? Ebbene sì, ma attenti all'apparente gioco di parole. Il valore medio  $m = 3500.4$  della media campionaria (cioè della distribuzione empirica dei millanta e millanta dati raffigurati nei quattro boxplot) è una stima della media  $\mu$  della popolazione. Invece le deviazioni standard  $s$  della media campionaria che ciascuna di esse ottiene non sono direttamente una stima di  $\sigma$ ; ma se ciascuna di esse considera la quantità  $s \cdot \sqrt{n}$  ottiene che  $\sigma \approx 211.0 \cdot \sqrt{4} \approx 141.5 \cdot \sqrt{9} \approx 106.3 \cdot \sqrt{16} \approx 85.4 \cdot \sqrt{25} \approx 420$ . Ebbene, adesso vi svelo il trucco: mi sono servito di `rnorm(n = 176400, mean = 3500, sd = 420)` per generare i millanta neonati di Andrea, Bruna, Carla e Diana (sul sito web del libro se siete interessati trovate il codice R per generare questa simulazione). Cosa ci dovrà rimanere in mente riguardo a questa simulazione finale? Nei piccoli campioni la variabilità degli esiti plausibili è più elevata di quella che si osserva nei grandi campioni: il boxplot di Andrea è molto più largo di quello di Diana; gli outlier di Andrea sono molto più sparsi di quelli di Diana. Ed aumentando la dimensione del campione l'incertezza si riduce: Andrea osservava campioni di dimensione  $n = 4$ , Diana di dimensione  $n = 25$ .

## 2.4 Riepilogone del capitolone

Oh, come è bello ritrovarsi tutti qui di nuovo assieme. Allora, questo secondo capitolo ci è servito a 'rimettere in ordine la cantina'. La trascrizione e la mutazione simulata sulla proteina Zinc Finger 521 ci ha fornito un espediente sia per presentare alcune tecniche molto basiche di programmazione con R, ma soprattutto per rivisitare il concetto di **casualità**, *randomness*, sfruttando la funzione `sample`, e per discutere di quali siano gli strumenti matematici più opportuni per lavorare con i fenomeni casuali: le **variabili aleatorie**. Ve ne sono tantissime, sono molto utili per far fare agli studenti degli eserciziotti di algebra ed interrogarli e dar loro un voto, alle superiori o nei corsi di laurea. Ma a noi qui interessa riconoscere il ruolo eminente nella biostatistica occupato da due variabili aleatorie ricorrenti. La prima è la variabile aleatoria **normale**, descritta dalla curva **gaussiana** con la sua tipica forma simmetrica cosiddetta a campana. Abbiamo osservato che le sue importanti proprietà in definitiva sono calcolabili a partire dalla sola conoscenza dei due **parametri** della distribuzione: la **media  $\mu$**  e la **deviazione standard  $\sigma$** . Ad esempio, la proprietà più nota è quella per la quale il 68 per cento oppure il 95 per cento dei dati sono racchiusi nell'area compresa tra più o meno una oppure due deviazioni standard attorno alla media:

[https://en.wikipedia.org/wiki/Standard\\_error#/media/File:Standard\\_deviation\\_diagram.svg](https://en.wikipedia.org/wiki/Standard_error#/media/File:Standard_deviation_diagram.svg)

Già nel primo capitolo si accennava al fatto che descrivere una variabile di un dataset per mezzo della media e della deviazione standard avesse a che fare con la **statistica parametrica**, senza però spiegare in dettaglio cosa avevo in mente. Ecco: qui abbiamo aggiunto un tassello per la comprensione di questo aggettivo, 'parametrico'; ed il discorso verrà definitivamente chiarito nella sezione 4.1.4.

Accanto poi alla distribuzione normale dei dati vi ho parlato delle distribuzioni di tipo **binomiale**, con le quali i tipici esiti biologici di negativo/positivo, sano/malato, wild-type/mutato vengono studiati. La conoscenza di queste due variabili aleatorie, e l'utilizzo della funzione `sample`, ci ha consentito di riprendere e completare il discorso 'dell'equazione più pericolosa' che avevamo lasciato in sospeso nel capitolo precedente, riguardo al concetto di **standard error**,  $\sigma/\sqrt{n}$ . Nel capitolo iniziale utilizzavamo l'errore standard per valutare l'affidabilità della media negli esperimenti a misure ripetute. Qui invece abbiamo visto, nell'esempio della distribuzione binomiale lanciando molte monetine, che in un certo senso esso ci libera dal problema di avere a che fare con un campione piccolo o grande di dati, e quindi siamo in presenza dell'oggetto matematico 'giusto'

per fare dell'**inferenza statistica**: passare dal particolare al generale; dedurre un comportamento universale esaminando il comportamento del nostro campione. E nell'esempio del peso dei neonati distribuito normalmente abbiamo ribadito come si possa provare a stimare i parametri di una popolazione ripetendo delle osservazioni su un certo numero di campioni. Questo approccio ovviamente è del tutto speculativo, perché nei nostri laboratori difficilmente avremo tempo, voglia e denaro sufficiente per ripetere millanta volte un esperimento: ne sapeva ben qualcosa il dottor William Sealy Gosset che tutto il mondo delle scienze della vita conosce con lo pseudonimo di Student e che poteva disporre solo di undici campi seminati ad orzo dei quali valutare la resa in termini di produzione di birra Guinness. Di tutto questo inizieremo ad occuparci nel prossimo capitolo. Ma prima, alcuni esercizi di comprensione e di approfondimento.

## 2.5 Esercizi ed attività di approfondimento

■ **Attività 2.1 — funzioni di R.** Riprendiamo i concetti relativi alle misure di tendenza centrale e di dispersione delle sezioni 1.1 e 1.2. Siccome la deviazione standard (e la varianza) vengono calcolate in relazione alle medie, le quali ovviamente dipendono dalla scala di misura adottata, se si vogliono confrontare due campioni che utilizzano scale di misura diverse (esempio: gradi Kelvin e gradi Farenheit) è opportuno servirsi del **coefficiente di variazione** (talvolta indicato con *RSD*, **relative standard deviation**):

$$CV = \frac{\sigma}{\mu}$$

Realizzate una funzione di R che ne calcoli il valore.

*Suggerimento: se non sapete proprio come fare nemmeno dopo aver riletto il paragrafo 2.1.2, guardate questo video.*



■

■ **Attività 2.2 — funzioni di R.** Ancora relativamente alle misure di tendenza centrale e di dispersione delle sezioni 1.1 e 1.2, una misura di posizione particolarmente utile ('robusta', cioè insensibile agli outlier) è la media del primo e del terzo quartile, che si chiama **midhinge** (il 'perno centrale'):

$$\frac{Q1 + Q3}{2}$$

Realizzate una funzione di R che ne calcoli il valore, utilizzando a vostro piacere le funzioni `quantile`, oppure `summary`.

*Suggerimento: osservate come cambia l'output in questi tre esempi, e sfruttate il concetto:*

```
summary(iris$Petal.Length)
summary(iris$Petal.Length) [4]
summary(iris$Petal.Length) [[4]]
```

■

■ **Attività 2.3 — variabili aleatorie finite.** ([49, pagina 84]) Nella tabella 4.3 Bernard Rosner riporta le frequenze relative di episodi di otite media riscontrati in una determinata popolazione di neonati durante i loro primi due anni di vita. Pensando alle variabili aleatorie, interpretiamo la prima riga come una rappresentazione dei possibili eventi, la seconda come la densità di probabilità:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0.129 & 0.264 & 0.271 & 0.185 & 0.095 & 0.039 & 0.017 \end{pmatrix}$$

Inserite in R questa variabile aleatoria finita con i comandi:

```
evento = 0:6
frequenza = c(0.129, 0.264, 0.271, 0.185, 0.095, 0.039, 0.017)
```

Calcolate il valore atteso (la speranza matematica) e la varianza di questa variabile aleatoria utilizzando le definizioni:

```
speranza = sum(evento * frequenza)
varianza = sum(((evento - speranza)^2) * frequenza)
```

ed osservate che la varianza può venir calcolata anche mediante la 'formula' di König (o di Huygens, o di Steiner):

```
sum(evento^2 * frequenza) - speranza^2
```

Generate casualmente 100000 casi di otite media e determinatene la media e la varianza. Cosa notate?

```
casuali = sample(evento, size = 100000, prob = frequenza, replace = TRUE)
mean(casuali)
var(casuali)
```

Da ultimo, confrontate le probabilità teoriche con le frequenze osservate:

```
par(mfrow = c(1,2))
barplot(frequenza)
barplot(table(casuali))
```

■ **Attività 2.4 — distribuzione binomiale.** Rileggiamo gli esempi della sezione 2.3.1. Come potremmo usare la funzione `pbinom` per stimare la probabilità che arrovolando 10 pazienti sintomatiche, non più di una di loro abbia un sarcoma maligno? ed invece, sempre con `pbinom`, la probabilità che almeno due delle dieci pazienti abbia un sarcoma maligno? ■

■ **Attività 2.5 — distribuzione binomiale.** I matematici possono dimostrarvi che, ripetendo per  $n$  volte un esperimento bernoulliano di probabilità  $p$ , la speranza matematica assume valore  $n \cdot p$  e la varianza  $n \cdot p \cdot (1 - p)$ . Sulla falsariga della Attività 2.3, riuscireste a verificare quanto valgono la speranza matematica e la varianza per il sarcoma uterino ( $p = 0.02$ ) in un campione di 450 pazienti sintomatiche? ■

■ **Attività 2.6 — distribuzione di Poisson.** Provate ad ideare una simulazione che vi convinca che in una variabile aleatoria di Poisson di parametro  $\lambda$  la media e la varianza sono coincidenti. ■

■ **Attività 2.7 — distribuzione normale.** [49, pagina 127] Nell'esempio 5.14 Bernard Rosner definisce la misura spirometrica denominata Capacità Vitale Forzata (FVC) come il volume di aria che può essere espirato da una persona con uno sforzo massimale di 6 secondi. Si considera che un bambino abbia una crescita polmonare normale (in senso queteletiano) se la sua FVC standardizzata  $X$  stia nel range di  $\pm 1.5\sigma$ . Supponendo che la FVC standardizzata si comporti come una normale (in senso gaussiano) standard, calcolare la percentuale  $Prob(-1.5 \leq X \leq 1.5)$  di bambini che stanno nel range di normalità. ■

■ **Attività 2.8 — distribuzione normale.** [49, pagine 120-121] Negli esempi 5.6 e 5.7 Bernard Rosner afferma che la pressione diastolica dei maschi della mia età (giovanissimi, cioè) si distribuisce in maniera gaussiana con media 80 mm Hg e deviazione standard 12 mm Hg. Quanto vale il 90-esimo percentile? E quanti potrebbero essere in percentuale i maschi che hanno una sistolica inferiore a 60 mm Hg? ■

■ **Attività 2.9 — distribuzione normale.** Giovanni Gallo [26] nelle sue tabelle di misurazioni ecografiche biometriche relative all'accrescimento fetale, indica che alla 42-esima settimana di gestazione un feto, 'normalmente', ha un diametro biparietale BPD compreso tra 93 mm (quinto percentile) e 101 mm (novantacinquesimo percentile). Sapreste calcolare la deviazione standard della ipotetica distribuzione gaussiana? ■

■ **Attività 2.10 — distribuzione uniforme.** La distribuzione aleatoria uniforme è molto semplice da descrivere: essa genera i numeri reali casuali che si dispongono in un intervallo di estremi  $[a, b]$  ( $a$  compreso,  $b$  escluso). Verificate che valori vale o zero o  $1/4$ , ed i valori  $1/4$  si susseguono sul rettangolo di base da 3 a 7.

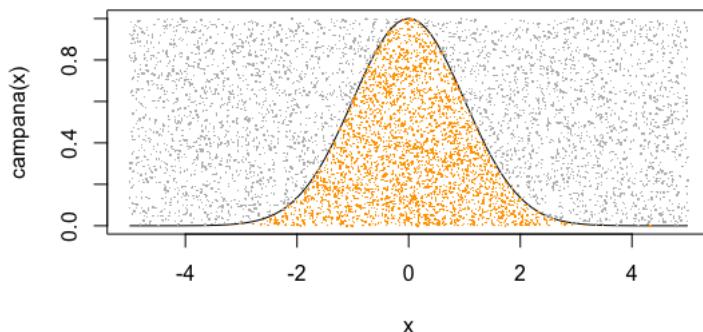
```
punti = seq(from = 0, to = 10, by = 0.1)
valori = dunif( punti, min = 3, max = 7)
valori
plot(punti, valori)
punif(5, min = 3, max = 7)
```

Un quesito: `punif(5, min = 3, max = 7)` individua la media, la mediana o la moda della distribuzione? ■

■ **Attività 2.11 — distribuzione uniforme.** Se andate ad esempio su Wikipedia [https://en.wikipedia.org/wiki/Normal\\_distribution](https://en.wikipedia.org/wiki/Normal_distribution), oppure se cercate un'immagine della banconota tedesca di dieci marchi dello scorso millennio, <https://www.google.com/search?q=zehn+mark+gauss>, osserverete che la funzione di densità di probabilità della variabile aleatoria normale dipende dalla funzione  $\exp(-(x^2/2))$  e presenta accanto a sé uno 'strano' coefficiente,  $1/\sqrt{2\pi}$ . Questo strano coefficiente ha a che fare con l'area della campana, e viene messo lì in modo tale che l'area valga uno. È sorprendente come il numero  $\pi$  salti fuori anche qui, e lo verifichiamo assieme, integrando la funzione  $\exp(-(x^2/2))$  con un **metodo Monte Carlo**. Definite la funzione campana e generate casualmente una caterva di punti casuali (ad esempio  $10^7$ ) che 'bombardino' il rettangolo di base  $[-L, L]$  ed altezza  $[0, 1]$ . Individuate quanti ne cadono aldisotto della campana con la condizione booleana `ycasuali < campana(xcasuali)` e contateli con `length(aldisotto)`. Adesso dividendo tale quantità per la caterva  $10^7$  avete una percentuale dell'area del rettangolo di base  $2 * L$  ed altezza 1 corrispondente proprio all'area arancione della campana di Gauss. A questo punto, indovinate quanto fa  $\sqrt{2\pi}$ .

```
campana = function(x){exp(-(x^2)/2)}
L = 5
xcasuali = runif(n = 10^7, min = -L, max = L)
ycasuali = runif(n = 10^7, min = 0, max = 1)
aldisotto = which(ycasuali < campana(xcasuali))
(length(aldisotto)/10^7) * 2 * L * 1
sqrt(2 * 3.1416)
```

Attenzione: non vi venga in mente di dare il comando `plot(xcasuali, ycasuali)`, se non volete impallare il vostro computer per qualche minutino. Se volete replicare il mio disegno, scegliete solo alcune migliaia di puntini, con un codice come questo:



```
curve(campana, -L, L)
points(xcasuali[1:5000], ycasuali[1:5000], pch = ".", col = "grey")
alcuni = aldisotto[1:3000]
points(xcasuali[alcuni], ycasuali[alcuni], pch = ".", col = "orange")
```

■ **Attività 2.12 — after hour.** Il professor Morrone è stato un esperto enologo e conoscitore di cocktail. E se anche a voi piace degustare di quando in quando un cocktail (attenzione: bevete con moderazione; l'alcol nuoce alla salute vostra, ed altri) provate il nostro *Raul*. Ecco gli ingredienti:

- mezzo lime tagliato in tre cunei
- 2.5 cucchiaini di zucchero di canna
- uno spruzzo di Angostura
- 20 ml di Triple Sec
- 20 ml di Curacao
- soda
- una fetta d'arancia per decorazione

In un bicchiere tumbler si pestino con il muddler i tre cunei di lime ricoperti con lo zucchero di canna. Aggiungere uno spruzzo di Angostura e ricoprire di ghiaccio tritato; aggiungere il Triple Sec ed il Curacao e completare con uno splash di soda. Guarnire con fetta d'arancia e servire con cannuccia corta biodegradabile. Mescolare prima di assaggiare.



# Seconda Parte

## 3 C'era una volta il p-value ..... 69

- 3.1 Il risultato è statisticamente significativo. E dunque?
- 3.2 Come nacque il t test
- 3.3 Ascesa e declino del p-value
- 3.4 La retta di regressione
- 3.5 Esercizi ed attività di approfondimento

## 4 Che cos'è un modello lineare ..... 83

- 4.1 I dettagli da conoscere
- 4.2 Ancova: unire i predittori numerici ai fattori
- 4.3 Facciamo il punto della situazione
- 4.4 Anova: la generalizzazione del t test
- 4.5 La meta finale: condurre un'analisi multivariabile
- 4.6 Riassuntone del capitolo
- 4.7 Perle di saggezza: tecniche di linearizzazione
- 4.8 Esercizi ed attività di approfondimento

## 5 I modelli lineari generalizzati ..... 121

- 5.1 I dettagli da conoscere
- 5.2 La meta finale: valutare l'accuratezza del modello logistico
- 5.3 Esercizi ed attività di approfondimento





### 3. C'era una volta il p-value

**Problema 3.1** ... bisognerebbe poter dare una qualche significatività ai nostri dati ...

**Problema 3.2** ... vorrei chiedere se il numero di pazienti coinvolti nel mio studio rappresenta una quantità che si possa definire statisticamente significativa ...

Nei giorni in cui stavamo per iniziare a comporre questo terzo capitolo, sui notiziari di tutto il mondo si era diffusa la notizia che per la prima volta si era riusciti a produrre un'immagine non di fantasia ma basata su reali osservazioni di natura interferometrica di quelle singolarità dello spazio tempo che chiamiamo buchi neri. Nel volume 875 del The Astrophysical Journal Letters erano apparsi ben sei articoli scientifici che illustravano i risultati della ricerca svolta dal consorzio Event Horizon Telescope, intitolati appunto *First M87 Event Horizon Telescope Results*. Come potrete immaginare si trattava di una ricerca che aveva comportato uno sforzo pluriennale di analisi dei dati, oltre che all'utilizzo di innovative tecniche computazionali e di ideazione di algoritmi di riconoscimento automatico, nelle quali le metodologie probabilistiche e statistiche rimanevano alla base di ogni discorso. Ebbene, l'invito che vi propongo è quello di provare a leggere i sei paper, con tutti i loro risultati:

<https://iopscience.iop.org/issue/2041-8205/875/1>

ed a coloro i quali riescano a trovarci scritta almeno una volta la parola **p value**, offro da bere una pinta di birra Guinness (oppure, a piacere, uno sciropo di tamarindo). E siccome – me lo sento – nei prossimi anni qualcuno di costoro si porterà a casa un Premio Nobel, evidentemente, non vi è dubbio che siamo in presenza di una 'significativa' scoperta scientifica. Come la mettiamo, allora? Dai, leggete qui avanti.

#### 3.1 Il risultato è statisticamente significativo. E dunque?

Sentite quest'altra storia. Il 17 aprile 2014 la chirurgia ginecologica laparoscopica subì una rilevante battuta d'arresto: la Food and Drug Administration emise una comunicazione di sicurezza che andava a limitare l'utilizzo ('*to discourage the use*') della tecnica laparoscopica denominata 'power

morcellation', una tecnica minimamente invasiva che permetteva di trattare i fibromi uterini in maniera locale (miomectomia) o globale (isterectomia) assicurando brevi tempi di degenza e di guarigione e bassa probabilità di infezione. Era successo purtroppo che pazienti colpite da sarcoma uterino, lamentando sintomi del tutto sovrapponibili con quelli dei fibromi, erano state trattate con la morcellazione laparoscopica: ma tale tecnica chirurgica poneva il rischio di disseminare porzioni di tessuto canceroso a livello addominale o pelvico, aggravando la probabilità di sopravvivenza delle pazienti oncologiche. Tutto ciò accadeva purtroppo, nella parola della FDA, '*because there is no reliable method for predicting whether a woman with fibroids may have a uterine sarcoma*'.

Era urgente mettersi alla ricerca di possibili metodi predittivi del sarcoma. Annalisa Di Cello [19] raccolse le cartelle di più di tre migliaia di pazienti affette da patologie benigne uterine (ad esempio i fibromi uterini, oppure le adenomiosi) e maligne (i sarcomi, appunto, molto aggressivi e dalla prognosi molto spesso infausta) ed iniziò a riflettere su come individuare un biomarcatore clinico efficace nella diagnosi del sarcoma uterino. Ne abbiamo già parlato nella sezione 2.3.1 dedicata alla variabile aleatoria binomiale: all'indirizzo

<http://www.biostatisticaumg.it/dataset/raul.csv>

dove abbiamo pubblicato un dataset estratto da quelle migliaia di pazienti, in cui sono riportati i valori di due importanti candidati al ruolo di biomarcatore:

- la glicoproteina CA125.
- gli isoenzimi LDH-1 ed LDH-3 dell'enzima L-lattato deidrogenasi.

Proviamo ora a fare un paio di grafici esplorativi con Orange: precisamente un boxplot del Ca125 versus l'Esito (benigno o maligno) ed uno scatterplot, il grafico sul piano cartesiano che la professoressa di matematica ci faceva fare alle superiori, di Ldhtre versus Ldhuno, mettendo in evidenza con due colori diversi l'Esito. Toccate il pulsante arancione per far partire il video tutorial per imparare come realizzarli facilmente:

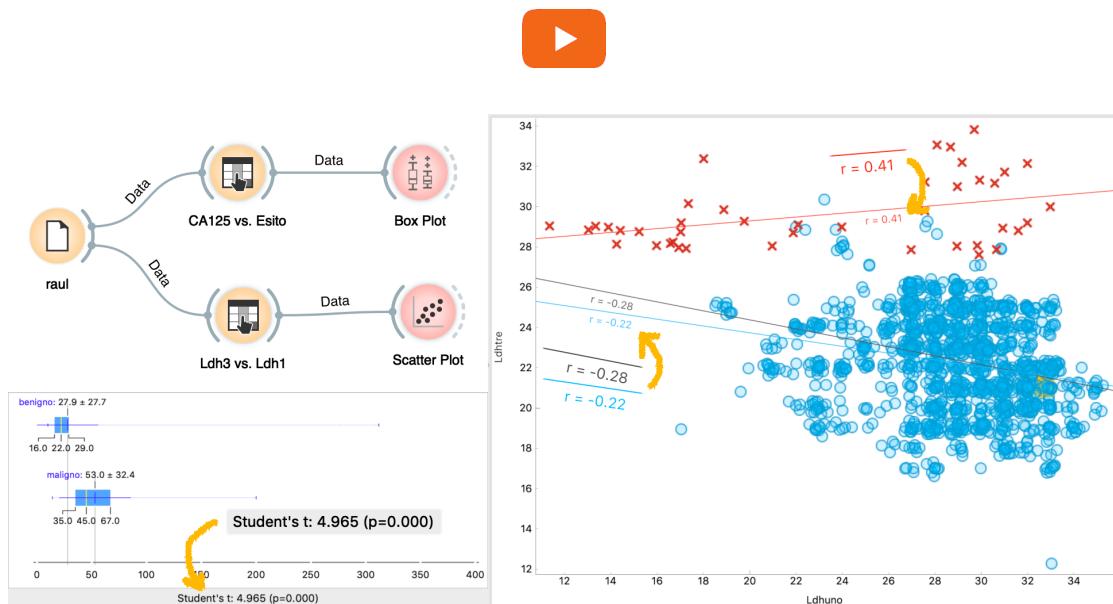


Figura 3.1: Il p-value ci garantisce di aver individuato un biomarcatore clinico efficace?

Ebbene, forse non ci crederete, ma la Figura 3.1 tocca esattamente la crisi in atto che coinvolge la questione del ' $p < .05$ ', ed indica anche quale sarà il modo per superare questa crisi. Così finalmente i biostatistici non dovranno più cercare modi eleganti ed educati per dare risposte a domande insensate come quelle dei problemi 3.1 e 3.2.

### 3.2 Come nacque il t test

**Problema 3.3** ... innanzitutto vorrei capire se, in media, i valori del CA125 delle pazienti con il sarcoma siano differenti da quelli delle signore con la patologia benigna.

Questa è una domanda del tutto analoga a quella che il matematico e chimico inglese di nascita e dublinese di adozione William Sealy Gosset, che tutti noi conosciamo con lo pseudonimo di 'Student', ebbe a porsi più di un centinaio di anni fa.



Ci sono mille e mille storie da leggere sulla vita e le opere di William Gosset. Per esempio, questa: [https://www.encyclopediaofmath.org/index.php/Gosset,\\_William\\_Sealy](https://www.encyclopediaofmath.org/index.php/Gosset,_William_Sealy)

Arriveremo ad una (parziale) risposta al Problema 3.3 fra alcune pagine, nel paragrafo 3.2.2 relativo al test t di Student effettuato su due diversi campioni. Ma dal punto di vista didattico vale la pena ripercorrere l'esperimento che William Gosset voleva affrontare e risolvere dal punto di vista statistico e che lo condusse a pubblicare il suo articolo fondamentale [53]. Si trattava di risolvere un problema di biotecnologie agrarie *ante-litteram*: se fosse utile e remunerativo, o meno, essiccare all'interno di speciali forni i semi di orzo prima di piantarli, in modo da assicurare una maggiore resa agricola e conseguentemente aumentare la produzione di birra Guinness a parità di costo. Ecco qui le parole di Gosset:

To test whether it is advantage to kiln-dry barley seed before sowing, seven varieties of barley were sown (both kiln-dried and not kiln-dried) in 1899 and four in 1900; the results are given in the table (...), expressed in Lbs. head corn per acre.

Not Kiln-Dried	Kiln-Dried	Difference
1903	2009	+106
1935	1915	-20
1910	2011	+101
2496	2463	-33
2108	2180	+72
1961	1925	-36
2060	2122	+62
1444	1482	+38
1612	1542	-70
1316	1443	+127
1511	1535	+24

La tabella qui sopra riporta i dati grezzi dell'esperimento; William Gosset in particolare cerca di semplificarsi i calcoli algebrici considerando la colonna a destra, che riporta la differenza di resa per ogni campo. Inseriamo 'a mano' proprio questi dati in R usando il comando c():

```
differenzaresa = c(106, -20, 101, -33, 72, -36, 62, 38, -70, 127, 24)
mean(differenzaresa)
```

```
> differenzaresa = c(106, -20, 101, -33, 72, -36, 62, 38, -70, 127, 24)
> mean(differenzaresa)
[1] 33.72727
```

Concretamente, William Gosset vuole decidere con un metodo statistico se la media delle rese dei semi 'Not Kiln-Dried' e quelle dei semi 'Kiln-Dried' siano diverse. Ovvero, se la media di differenzarea, che vale 33.7, sia un numero diverso dallo zero in senso probabilistico/statistico, non banalmente algebrico. È questo in definitiva lo scopo del comando `t.test` di R: discutiamone il significato passo per passo.

### 3.2.1 il comando `t.test` di R

Qui si vuole paragonare la media dei rendimenti nell'esperimento di Gosset,  $m = 33.7$ , con una situazione ipotetica nella quale effettuare o non effettuare il trattamento produca la medesima resa agricola, ossia una media di rendimenti nulla. Il risultato teorico di differenzarea quindi dovrebbe essere  $\mu = 0$ , se questa ipotesi teorica fosse vera. Ecco perché in gergo quest'ultima viene detta l'**ipotesi nulla** o, più raramente, **valore nullo del test**:

```
t.test(differenzarea)$estimate
t.test(differenzarea)$null.value

> t.test(differenzarea)$estimate
mean of x
33.72727
> t.test(differenzarea)$null.value
mean
0
```

In termini geometrici, William Gosset vuole capire se la distanza tra le due quantità (osservata  $m = 33.7$  e teorica  $\mu = 0$ ) si possa considerare nulla. Ma qui dobbiamo liberarci da un problema iniziale: se invece di usare la scala originale di misura in 'libbre per acro' avessimo usato le attuali misure del sistema internazionale, 'kilogrammo per metro quadrato', evidentemente la media osservata  $m$  sarebbe cambiata.

**Quesito 3.2.1** Vi ricordate dove abbiamo già incontrato il problema del cambiamento di media quando cambia la scala di misura adottata? No? Eh, per forza, se non fate i compiti per casa! :-)

L'idea vincente è quella di andare a considerare quello che in termini ingegneristici viene definito come il **rapporto segnale - rumore**; il 'segnale' è `mean(differenzarea)`, la distanza tra le medie  $m - \mu \equiv 33.7 - 0$ ; mentre il 'rumore' si quantifica stimando la deviazione standard della media campionaria, ossia la stima dell'errore standard  $s/\sqrt{n}$  di cui abbiamo discusso nel Capitolo 2. Questo particolare 'rapporto segnale rumore' storicamente si indica con  $t$ :

$$t = \frac{m - \mu}{s/\sqrt{n}}$$

Questa quantità  $t$  viene chiamata in inglese **test statistic** e per noi italiani questa parola rappresenta una fonte di grande confusione perché ci appare difficoltoso distinguere tra la 'scienza', *statistics*, e la 'quantità numerica' *statistic*, ossia *a quantity calculated from the data in a sample, which characterises an important aspect in the sample*. Il mio professore Andrea Sgarro mi raccomandava di tradurre quest'ultimo lemma in italiano con il termine di **consuntivo** (gli ispanici invece sono furbi: al femminile *la (ciencia) estadistica* ed al maschile *el estadístico (de test)*).

```
( mean(differenzarea) - 0 ) / ( sd(differenzarea) / sqrt(11) )
t.test(differenzarea)$statistic
```

```
> ( mean(differenzaresa) - 0 ) / ( sd(differenzaresa) / sqrt(11) )
[1] 1.690476
> t.test(differenzaresa)\$statistic
t
1.690476
```

Guardiamo la Figura 3.2 sottostante. Ora che ci siamo liberati dal problema della scala di misura, ci rimane da interpretare questo consuntivo  $t = 1.69$ : esso è vicino allo zero, oppure lontano? Dovremmo porre questa domanda in termini di probabilità, e la risposta possibile – prima della pubblicazione fondamentale di Student – sarebbe stata quella di andare a considerare la variabile aleatoria normale standard di media nulla e deviazione standard 1, e di interpretare il consuntivo  $t$  come il **quantile** della distribuzione. Se ricordate i discorsi del Capitolo 2, dovremmo valutare quest’area con il comando `pnorm(1.690476)`; o meglio, l’area della coda viola di destra equivarrebbe a  $1 - \text{pnorm}(1.690476)$ , e quindi per avere la probabilità che le distanze  $m - \mu$  oppure  $\mu - m$  siano più grandi di quella osservata, dovremmo raddoppiare la quantità:

```
2 * (1 - pnorm(1.690476))
> 2 * (1 - pnorm(1.690476))
[1] 0.09093693
```

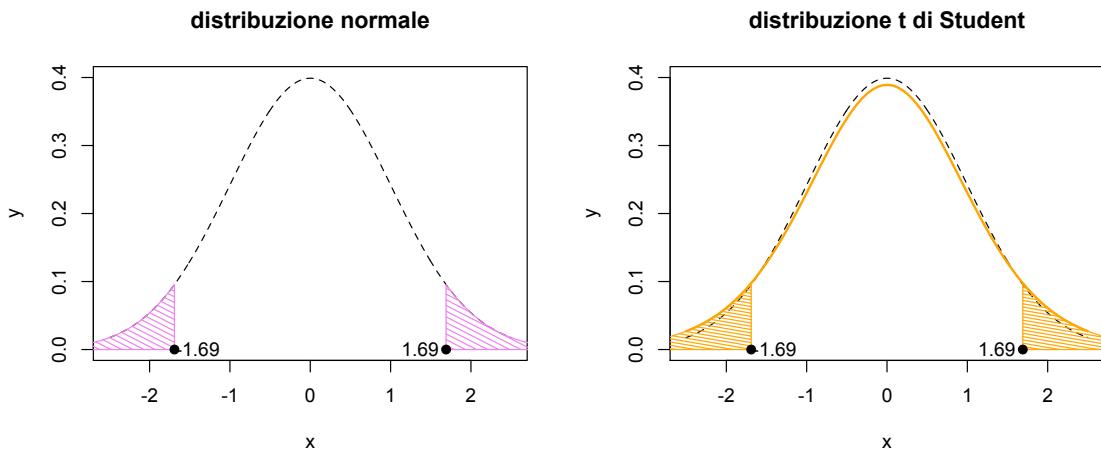


Figura 3.2: A sinistra, la probabilità di osservare per caso una distanza maggiore di quella dalla media del campione  $m = 33.7$  e quella teorica  $\mu = 0$  valutata erroneamente con la distribuzione gaussiana, `pnorm`: le aree delle code colore viola sono di circa il 9%, stima troppo ottimistica. A destra, le probabilità in colore arancio salgono al 12%, secondo la variabile aleatoria  $t$  di Student a 10 gradi di libertà: ‘le code sono più grosse quando la dimensione del campione è piccola’.

Nel suo articolo magistrale [53], William Gosset si accorge che tale valutazione del 9% sarebbe troppo ottimistica, e dopo una dozzina di pagine di calcoli di vera matematica egli ci fornisce la relazione integrale che descrive l’innovativa variabile aleatoria  $T = (X - M)/(S/\sqrt(n))$ , dove  $M$  ed  $S$  sono le variabili aleatorie media della media campionaria e deviazione standard della media campionaria. Tutto molto arzigogolato, vero? ed il tutto è posto in relazione alla dimensione del campione  $n$ , o meglio, al numero di parametri liberi che prendono il nome di **gradi di libertà** della distribuzione  $t$  di Student e nel nostro esempio sono 10, ossia la dimensione del campione 11 diminuita di 1 informazione (perché noi sappiamo già che  $m = 33.7$ ).

```

length(differenzaresa) - length(mean(differenzaresa))
t.test(differenzaresa)$parameter

> length(differenzaresa) - length(mean(differenzaresa))
[1] 10
> t.test(differenzaresa)\$parameter
df
10

```

Stiamo arrivando alla fine della nostra dettagliata spiegazione. In R la probabilità della distribuzione  $t$  di Student si può calcolare con il comando `pt`:

```

2 * (1 - pt(q = 1.690476, df = 10))
t.test(differenzaresa)$p.value

> 2 * (1 - pt(q = 1.690476, df = 10))
[1] 0.1218166
> t.test(differenzaresa)\$p.value
[1] 0.1218166

```

In conclusione, abbiamo effettuato il nostro primo test  $t$  di Student con R, sui dati originali dell'autore, il quale scopre che c'è una probabilità di circa il 12.2% che la differenzaresa non nulla sia semplicemente dovuta alla casualità di quell'esperimento, e non legata ad una reale efficacia del trattamento di essicazione 'Kiln-dry'. Vediamo se quindi siamo in grado di capire tutto quello che viene proposto dall'output del comando `t.test(differenzaresa)`:

```

> t.test(differenzaresa)

One Sample t-test

data: differenzaresa
t = 1.6905, df = 10, p-value = 0.1218
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-10.72710 78.18164
sample estimates:
mean of x
33.72727

```

Dunque, abbiamo spiegato bene quasi tutto, ci manca ancora il discorso del 95 percent confidence interval, e lo riprenderemo più avanti nel libro nella sezione dedicata all'approccio bayesiano all'analisi dei dati.

### 3.2.2 il test $t$ di Student tra due campioni

Adesso potete dare voi stessi una risposta al problema 3.3. Nel paragrafo precedente abbiamo visto come funziona il comando `t.test` applicato ad un unico campione, la `differenzaresa`. Si trattava di un trucco algebrico comodo da sfruttare per il fatto che i due gruppi che stavamo analizzando ('Not Kiln-Dried' e 'Kiln-Dried') avevano entrambi la medesima dimensione, 11. Questo però non è il caso del dataset `raul`, nel quale le pazienti benigne sono 1568, contro le 42 con patologia maligna:

```

www = "http://www.biostatisticaumg.it/dataset/raul.csv"
raul = read.csv(www, header = TRUE)
attach(raul)
names(raul)
table(Esito)

> www = "http://www.biostatisticaumg.it/dataset/raul.csv"
> raul = read.csv(www, header = TRUE)
> attach(raul)
> names(raul)
[1] "Anni"      "Ca125"     "Ldhuno"    "Ldhtre"    "Esito"     "Diagnosi"
> table(Esito)
Esito
benigno maligno
1568      42

```

Vi proponiamo un possibile modo di procedere. Si tratta però di un modo che adotteremo solo in questa singola occasione didattica, perché come vedremo esiste un modo di procedere molto più efficiente e, per così dire, universale.

Separiamo in due sottoinsiemi, che chiameremo verde e rosso, i valori della glicoproteina Ca125 in base all'Esito delle pazienti (che ricordiamo essere, in ordine alfabetico, benigno e maligno). Per farlo, useremo la funzione `split`. Dopo di che, eseguiremo il `t.test` sui due sottoinsiemi verde e rosso:

```

verde = split(Ca125, Esito)[[1]]
rosso = split(Ca125, Esito)[[2]]
t.test(verde, rosso)

> verde = split(Ca125, Esito)[[1]]
> rosso = split(Ca125, Esito)[[2]]
> t.test(verde, rosso)

```

*Welch Two Sample t-test*

```

data: verde and rosso
t = -4.9067, df = 42.584, p-value = 1.401e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-35.32373 -14.74089
sample estimates:
mean of x mean of y
27.94388 52.97619

```

Cosa vediamo? Vediamo che la media del Ca125 nelle pazienti del gruppo verde, ossia le benigne, vale approssimativamente 28, mentre in quelle del rosso, le maligne, vale circa 53. Molto probabilmente questa differenza non è dovuta al caso: il consuntivo  $t = -4.9$ , il famoso rapporto segnale-rumore si colloca quasi a 5 deviazioni standard dal centro, individuando nelle due code della distribuzione una probabilità praticamente trascurabile, dell'ordine di  $10^{-5}$ .

Quindi, abbiamo finalmente una risposta al problema 3.3? Beh, lo sappiamo che non è molto educato rispondere ad una domanda con un'altra domanda, ma io ve la faccio lo stesso:

**Quesito 3.2.2** Il test t di Student ci assicura che possiamo affermare con elevatissimo grado di fiducia che la glicoproteina Ca125 ha valori mediamente più bassi nelle donne con patologie uterine benigne rispetto a quelle affetta da sarcoma. La domanda importante è: il p-value = 0.000014 ci garantisce che il Ca125 sia un biomarcatore efficace nella predizione della malattia?

Nel frattempo che riflettete sul quesito 3.2.2, tenete in allenamento le dita sulla tastiera e risolvete il seguente esercizio.

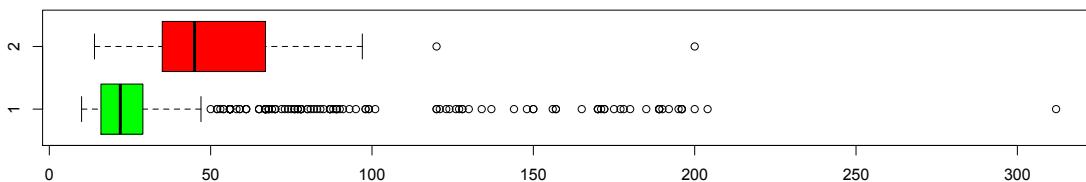
**Esercizio 3.1** A vostro giudizio, la media delle età (Anni) delle pazienti con il sarcoma è differente da quella delle pazienti con la patologia benigna? Adattate i comandi di verde e rosso che abbiamo appena visto. ■

### 3.3 Ascesa e declino del p-value

Il quesito 3.2.2 è cruciale. Sul mio libro di matematica al liceo, veniva riportata un'istanza del filosofo Ludovico Geymonat sul fatto che la Matematica possegga o meno un valore conoscitivo. Noi, molto più terra-terra, ci chiediamo se il p-value che abbiamo calcolato con il t test sia uno strumento che ci permette di affermare che il CA125 sia predittivo del tumore. Proviamo a riguardare la Figura 3.1 di poche pagine fa e riguardiamo in particolare il boxplot in basso:

```
boxplot(rosso, verde, col = c("red", "green"), horizontal = TRUE)
quantile(rosso, .2)
quantile(verde, .9)

> quantile(rosso, .2)
20%
32.4
> quantile(verde, .9)
90%
32
```



Supponiamo che una paziente con sintomi si presenti alla nostra osservazione con un valore di CA125 uguale a 32. Vi faccio alcune domande, e temo che voi abbiate già intuito dove andremo a parare:

- Sapendo che nella nostra casistica 1 donna su 5 tra i sarcomi aveva CA125 inferiore a 32, e che 1 donna su 10 tra le benigne aveva il CA125 superiore a 32, quale utilità nel porre la diagnosi avete nel sapere che avevamo un p-value = 0.000014?
- E cosa dire del fatto che 1 sarcoma su 7 della nostra casistica aveva valori di CA125 inferiori a 28, la media di verde?

- Ed ancora, una donna su quattro con patologia benigna non supera il valore di 16; eppure abbiamo avuto un caso non outlier di paziente maligna con CA125 uguale a 14. Che ne pensate?
- Infine, supponiamo che si presenti una paziente con CA125 maggiore ad esempio di 100. Molto raramente, le pazienti benigne hanno di questi valori; al contempo, nel sarcoma – che di sua natura è raro – questo è capitato 2 volte su 42, altrettanto raramente quindi. Ripetiamo, che abbiamo un p-value = 0.000014: dunque quale sarebbe la diagnosi plausibile in questo caso, a vostro giudizio?

Tutto, ahinoi, inizia a sedimentarsi attorno ad un'interpretazione assoluta di ciò che invece avrebbe dovuto avere un senso relativo: nel 1937 sir Ronald Aymer Fisher dava alle stampe il suo *The Design of Experiments* [25], ed a pagina 11 iniziava a raccontare della lady che beveva il the ed affermava di saper distinguere se il latte era stato aggiunto nella tazza prima del the oppure dopo. Ed a pagina 13, alla sezione intitolata *Significance*, scriveva:

Thus if he wishes to ignore results having probabilities as high as 1 in 20 – the probabilities being of course reckoned from the hypothesis that the phenomenon to be demonstrated is in fact absent –

A pagina 25 Ronald Fisher proseguiva indicando che ci potrebbero essere anche altri *familiar level of significance* oltre a quello del 5%, come quelli del 2% o dell'1%, e questa 'percezione di familiarità' proveniva dal fatto che i test statistici si calcolavano in base alle tavole di probabilità che si trovavano elencate nei testi allora circolanti (e tacciamo qui per decenza sulla diatriba tra Karl Pearson e Ronald Fisher). Non sappiamo dirvi poi con precisione perché nel giro di cinquanta anni queste affermazioni ragionevoli, in quanto legate ad un esperimento di tipo combinatorio in cui erano considerate solo otto tazze di the, quattro di un tipo e quattro di un altro, si sia arrivati ad un atteggiamento fideistico nei confronti del p-value. Non stiamo usando il termine fideistico a caso: abbiamo in mente il magistrale articolo di Stephen Ziliak e di Deirdre McCloskey, uscito nel 2008, dal titolo *The Cult of Statistical Significance*[64] e che vi invitiamo senza dubbio a leggere:



Stephen Ziliak and Deirdre McCloskey, *The cult of statistical significance*:

<https://www.deirdremccloskey.com/docs/jsm.pdf>

Abbiamo preso a prestito da loro la terminologia del **signal to noise ratio** per descrivere il consuntivo *t* del test di Student. Il loro articolo era apparso pochi anni dopo un altro macigno che scuoteva le fondamenta di quella che per scherzo mi viene da chiamare 'la bolla speculativa del p-value'. L'articolo era di John Ioannidis ed il suo titolo era ancora più esplosivo, *Why Most Published Research Findings Are False* [31]. Si trattava pur sempre però di articoli provenienti dall'accademia, ed avremmo dovuto attendere ancora un decennio prima che facessero ampio scalpore nella comunità medica. Tra tutti, ricordiamo il commento di Richard Horton, chief-editor di *The Lancet*, *Offline: what is medicine's 5 sigma?*[28]:



John Ioannidis, *Why Most Published Research Findings Are False*:

<https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124>

Richard Horton, *Offline: what is medicine's 5 sigma?*

<https://www.thelancet.com/pdfs/journals/lancet/PIIS0140-6736%2815%2960696-1.pdf>

La posizione più forte, al momento in cui sto redigendo queste righe, viene posta dalla prestigiosa rivista (dicono sempre così al telegiornale) Nature, in un manifesto sottoscritto da quasi un migliaio di scienziati che chiedono di mettere in soffitta il concetto di significatività statistica, *Scientists rise up against statistical significance*[2]:

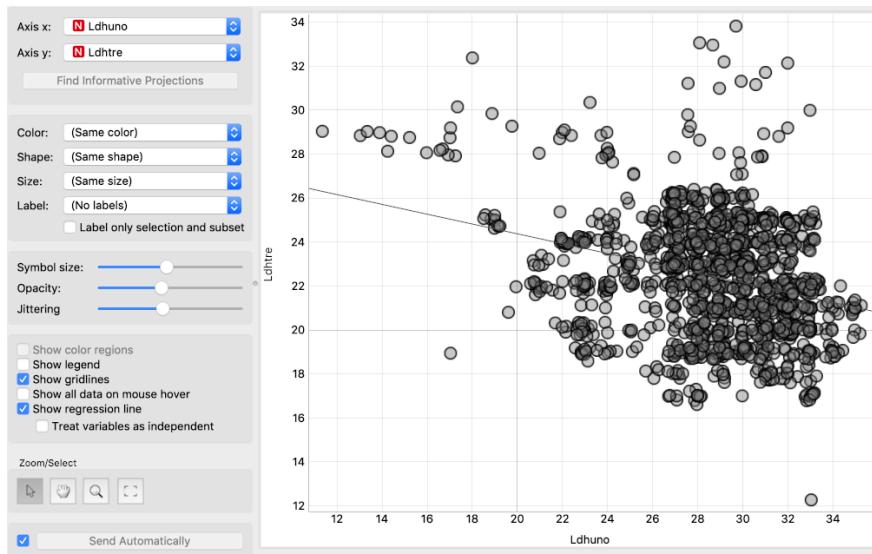


Valentin Amrhein et al., *Scientists rise up against statistical significance*:

<https://www.nature.com/articles/d41586-019-00857-9>

Ora, ammettiamolo, sono tutti dei bei discorsi, giusti, ineccepibili. Ma la domanda sorge immediata e la esprimiamo senza giri di parole nel linguaggio crudo di chi fa ricerca: se dobbiamo smetterla con questi p-value, con questa significatività, come facciamo a pubblicare i nostri lavori senza essere rigettati dai referee che non sono al corrente di questa profonda mutazione che sta sopraggiungendo? Vi avevo anticipato che nella Figura 3.1 erano raffigurati sia il problema attuale della statistica (in basso a sinistra, quel p-value significativo che però non garantisce di aver trovato un biomarcatore efficiente), che la sua soluzione. Essa è nascosta nel disegno a destra, in quelle rette colorate.

### 3.4 La retta di regressione



**Problema 3.4** ... noi saremmo interessati a capire come si correlano gli isoenzimi LDH1 ed LDH3 nelle nostre pazienti con la patologia uterina ...

In questo problema è del tutto appropriato usare il verbo 'correlare', nel senso latino di 'avere una relazione comune'. Vediamo in dettaglio il senso matematico di queste parole.

#### 3.4.1 Covarianza e correlazione

Riprendiamo i discorsi del primo Capitolo; quando consideriamo due variabili numeriche come Ldh1 ed Ldh3, certamente possiamo andare a misurare la loro dispersione attorno alla media servendoci della deviazione standard o, volendo, della varianza. Ma è altrettanto utile capire se esse possiedano una qualche caratteristica comune che le 'faccia variare assieme'. Per misurare questa 'variazione comune' consideriamo gli scarti che i dati hanno rispetto alla loro rispettiva media, e facciamo un po' di algebra, moltiplicando, sommando e dividendo:

$$\text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - m_x)(y_i - m_y)$$

Questo indice prende il nome di **covarianza**, ma avrete già intuito che qui siamo alle solite: la covarianza è una quantità soggetta alla scala di misura delle variabili che studiamo, e per liberarci da questa sgradevole limitazione si preferisce ripetere la medesima operazione che avevamo fatto nel 'rapporto segnale-rumore' di un paio di paragrafi fa, ossia dividere per le deviazioni standard:

$$\text{cor}(x, y) = \frac{\text{cov}(x, y)}{s_x \cdot s_y}$$

Siamo in presenza del cosiddetto **coefficiente di correlazione lineare** di Auguste Bravais e di Karl Pearson, il quale viene solitamente indicato con la lettera greca rho minuscola,  $\rho$ . Siccome i software preferiscono semplificare la faccenda, invece dell'alfabeto greco si preferisce usare l'alfabeto latino; infatti nella Figura 3.1 vedete riportata in colore nero l'indicazione  $r = -0.28$ . È immediato calcolare covarianza e correlazione con R:

```
cov(Ldhuno, Ldhtre)
cor(Ldhuno, Ldhtre)

> cov(Ldhuno, Ldhtre)
[1] -2.412192
> cor(Ldhuno, Ldhtre)
[1] -0.2813228
```

È interessante questo segno meno. Provate a vedere quello che succede in questo esercizio e provate a trarre qualche conclusione.

**Esercizio 3.2** Vi ricordate il dataset airquality di cento pagine fa? Provate a calcolare il coefficiente di correlazione della temperatura Temp rispetto ai livelli di Ozone:

```
attach(airquality)
cor(Temp, Ozone)

Oops! Cosa succede? Allora provate così:

cor(Temp, Ozone, use = "complete.obs")
```

Confrontate questo valore con quel  $r = -0.28$  degli isoenzimi Ldh, ed in particolare confrontate quella nuvola di punti con questa:

```
plot(Temp, Ozone)
```



### 3.4.2 L'idea di Francis Galton

Nel 1886 Francis Galton, cugino di Charles Darwin, pubblicava un interessante studio di antropologia intitolato *Regression Towards Mediocrity in Hereditary Stature* [27]. Galton voleva convincere i lettori 'in modo statistico' del fatto che i figli nati da genitori più alti della media tendono a *regredire* verso la statura media della popolazione. Ed escogitò un intuitivo metodo grafico, disegnando la retta che descriveva 'nel miglior modo possibile' la nuvola di punti che aveva disegnato, che rappresentava sull'asse delle ascisse la statura di 930 figli da adulti, e sull'asse delle ordinate la statura di un loro genitore. Se da genitori alti fossero nati sempre figli alti, e da genitori bassi fossero

nati sempre figli bassi, la retta sarebbe stata una diagonale inclinata di 45 gradi. Invece la retta che emergeva era inclinata di meno, aveva una pendenza più lieve: era nata la **retta di regressione** (e Sergio Invernizzi [30] giustamente si chiede perché a Galton non interessasse parlare dei figli nati da genitori più bassi, i quali tendono a progredire verso la statura media, e noi oggi non parliamo della 'retta di progressione').



In rete ci sono due siti mooooo carini che si occupano di queste faccende in maniera visuale:

- <https://setosa.io/>
- <https://seeing-theory.brown.edu>

Torniamo a riguardare ancora una volta la Figura 3.1; in essa compaiono tre rette di regressione: la retta rossa, con una pendenza crescente, relativa alle pazienti con neoplasia maligna; la retta azzurra relativa alle pazienti benigne, che condivide una pendenza decrescente con la retta di regressione nera, la quale è la 'migliore retta' che descrive la nuvola di punti prescindendo dall'Esito (benigno o maligno), come se i punti fossero tutti di colore nero. Nell'Esercizio 3.2 avevamo intuito che il segno positivo o negativo della correlazione avesse a che fare con l'andamento della nuvola di punti, crescente o decrescente. Ebbene, non vi stupirete se vi dico che il coefficiente angolare  $b$  di ciascuna delle tre rette, essendo una  $b$  una pendenza del tipo  $\Delta y / \Delta x$ , se viene moltiplicata per una frazione della forma  $\Delta x / \Delta y$  (e l'idea più naturale è quella di considerare il 'disordine' della nuvola, misurato dal rapporto delle deviazioni standard  $\sigma_x / \sigma_y$ ) si trasforma in un numero puro adimensionale, che è proprio il coefficiente di correlazione lineare:

$$\rho = b \cdot \frac{\sigma_x}{\sigma_y}$$

Come vedremo in dettaglio nel prossimo capitolo, noi possiamo facilmente calcolare la pendenza della retta nera della Figura 3.1, che risulta essere  $b = -0.223$ . Usiamo questa informazione per verificare la relazione che lega tra loro la correlazione  $\rho$  e pendenza  $b$ :

```
-0.223 * sd(Ldhuno) / sd(Ldhtre)
cor(Ldhuno, Ldhtre)

> -0.223 * sd(Ldhuno) / sd(Ldhtre)
[1] -0.2812561
> cor(Ldhuno, Ldhtre)
[1] -0.2813228
```

Alcune pagine fa avevamo posto il Quesito 3.2.2, chiedendoci se il  $p$ -value = 0.000014 ci potesse garantire che il Ca125 sia un biomarcatore efficace nella predizione della malattia. Qui saremmo nella medesima condizione, perché se interrogassimo il software R scopriremmo che anche le rette rossa, azzurra e nera della Figura 3.1 hanno tutte e tre un  $p$ -value < 0.001 (qualunque cosa significhi questo  $p$ -value). E non sarebbe questa informazione a farci capire quello che invece l'intuizione ci suggerisce: che è meglio descrivere il comportamento dei dati delle Figura 3.1 per mezzo delle due rette rossa ed azzurra invece che con la sola retta nera.

Bene: siamo pronti per fare il grande salto. Nel Capitolo 4 impareremo che la retta di regressione è un prototipo perfetto di quello che gli statistici chiamano il **modello lineare**; scopriremo con stupore che anche il t test (di un campione, sezione 3.2, o di due campioni, sezione 3.2.2) ricada in seno al concetto di modello lineare, e vedremo come il professor Hirotugu Akaike [1] abbia saputo interpretare il concetto di disordine, o meglio di entropia, traslandolo dai settori della termodinamica, della probabilità e dell'informatica teorica a quello della statistica, creando lo strumento che va sotto il nome di **criterio di informazione di Akaike** il quale consente di scegliere un modello lineare piuttosto che un altro.

### 3.5 Esercizi ed attività di approfondimento

■ **Attività 3.1 — covarianza e correlazione.** Calcolate il coefficiente di correlazione tra la lunghezza dei petali e la lunghezza dei sepali nel dataset più petaloso che c’è, *iris*. ■

■ **Attività 3.2 — correlazione non vuol dire rapporto di causa-effetto.** Si sono sparsi fiumi di inchiostro (e forse se ne dovranno spargere ancora) per spiegare che la correlazione non misura quanto un fenomeno influisca su un altro. Andate a sorridere sul sito <http://www.tylervigen.com/spurious-correlations> nel quale ad esempio si pone in relazione il tasso di divorzi nel Maine con il locale consumo di margarina; o la spesa per la ricerca scientifica e spaziale degli Stati Uniti ed i suicidi per impiccagione, soffocamento e strangolamento. ■





## 4. Che cos'è un modello lineare

**Problema 4.1** ... come faccio ad affermare che gli isoenzimi LDH3 ed LDH1 hanno un comportamento diverso nelle pazienti con una patologia benigna rispetto ai casi maligni?

In questo capitolo riprenderemo in esame in maniera approfondita il dataset `raul` tratto dalla ricerca di Annalisa Di Cello [19], nel quale vi sono 1568 pazienti con una patologia uterina di natura benigna e 42 con un sarcoma uterino di natura maligna. Questo dataset ci fornirà un esempio-guida completo per trattare due punti di particolare rilevanza:

1. introdurre il concetto statistico di **modello lineare**, spiegando quali siano gli aspetti tecnico - matematici da tenere presente;
2. spiegare come si possa scegliere tra vari modelli statistici quello che in qualche modo risulta essere il 'migliore', operando quella che si chiama **selezione del modello** statistico.

Il punto qualificante di questo Capitolo consiste nel fatto che riusciremo a dare una risposta al Problema 4.1 **senza parlare di 'significatività'**, senza cioè ricorrere alle farraginose questioni del p-value, così ben esposte sul recente editoriale apparso su Nature [2]:

We agree, and call for the entire concept of statistical significance to be abandoned.  
(...) Rather, and in line with many others over the decades, we are calling for a stop to the use of P values in the conventional, dichotomous way - to decide whether a result refutes or supports a scientific hypothesis. (...) Whatever the statistics show, it is fine to suggest reasons for your results, but discuss a range of potential explanations, not just favoured ones. Inferences should be scientific, and that goes far beyond the merely statistical. Factors such as background evidence, study design, data quality and understanding of underlying mechanisms are often more important than statistical measures such as P values or intervals.

In particolare andiamo a riprendere la Figura 3.1 che avevamo creato con Orange nel capitolo precedente e la realizziamo con R: il nostro obiettivo è quello di convincere un lettore del perché sia 'giusto' utilizzare il modello statistico con due rette di regressione, quella azzurra e quella

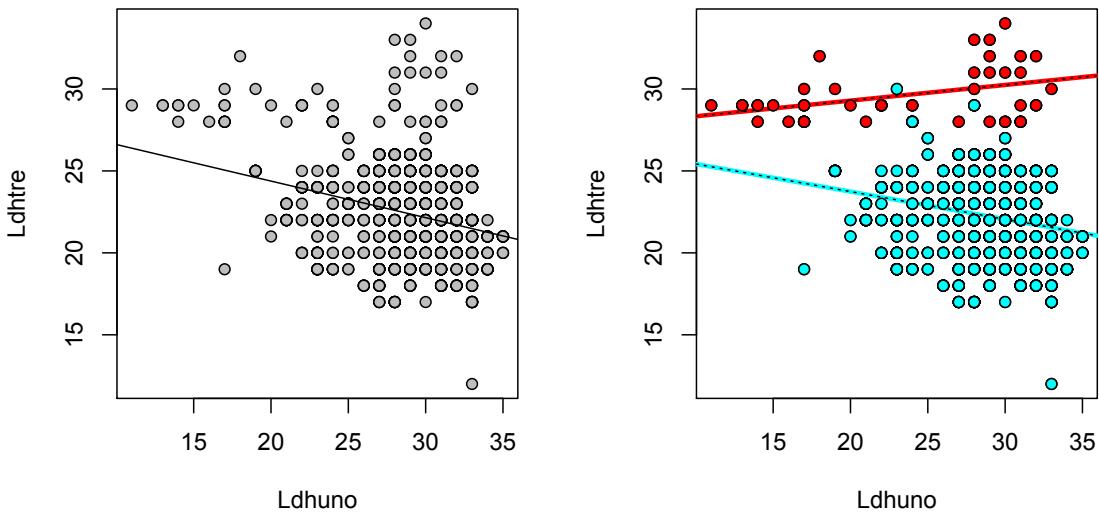


Figura 4.1: Due possibili modelli lineari che 'spiegano' il comportamento di Ldhtre in funzione di Ldhuno. perché dovremmo preferire quello di destra a quello di sinistra? La risposta ci arriverà alla fine di questo Capitolo, nella Sezione 4.5.

rossa, rispetto al modello semplice della singola retta di regressione nera. Per prima cosa, ci conviene imparare la simbologia che Wilkinson e Rogers [62] hanno proposto di usare negli ormai lontani anni '70. Per descrivere il modello lineare di sinistra in cui Ldhtre viene posto in relazione semplicemente con Ldhuno scriveremo:

```
formula1 = Ldhtre ~ Ldhuno
```

Invece nella seconda ipotesi, in cui l'informazione sull'Esito concorre a modificare la pendenza delle rette di regressione azzurra e rossa, useremo la notazione:

```
formula2 = Ldhtre ~ Ldhuno * Esito
```

Occorrono un paio di precisazioni. Vedete che in questa seconda ipotesi ci sono due rette, quella azzurra e quella rossa, ma qui non si parla di due modelli lineari; si parla di **un modello** lineare **con interazione** tra i due predittori, Ldhuno ed Esito. Il simbolo **\*** è stato scelto da Wilkinson e Rogers proprio per indicare il fatto che le covariate Esito e Ldhuno interagiscono tra su Ldhtre in una maniera non indipendente. Avrò modo anche di spiegare l'utilità dei simboli **+** e **:**, che hanno un loro preciso significato nella notazione statistica. E, da ultimo, spiegheremo come – sorprendentemente – la notazione:

```
formula3 = Ldhtre ~ Esito
```

individui un modello lineare di cui conosciamo già gli scopi ed il significato: il t test di Student.

## 4.1 I dettagli da conoscere

Per comodità didattica lasciamo in sospeso la questione delle patologie uterine (alla quale comunque continueremo a fare riferimento) ed andiamo a considerare un dataset molto più ameno, fresher:

si tratta di un gruppo di 65 ragazze e ragazzi iscritti (tanti anni fa) al primo anno di medicina dell'università di Trieste:

```
www = "http://www.biostatisticaumg.it/dataset/fresher.csv"
fresher = read.csv( www, header = TRUE )
attach(fresher)
str(fresher)

...
> str(fresher)
'data.frame':      65 obs. of  10 variables:
 \$ gender : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
 \$ height : int  155 157 158 158 158 160 163 163 164 165 ...
 \$ weight : int  53 50 48 49 58 45 51 52 68 57 ...
 \$ shoesize: int  36 37 36 36 38 37 39 37 40 37 ...
 \$ smoke   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 2 1 1 1 ...
 \$ gym     : Factor w/ 3 levels "not","occasional",...: 1 2 2 2 1 2 3 3 1 2 ...
 \$ heartrate: int  62 64 74 66 65 72 80 74 78 76 ...
 \$ systolic : int  90 120 95 95 140 120 125 120 100 105 ...
 \$ diastolic: int  60 70 75 70 70 80 80 70 65 80 ...
 \$ day     : int  10 17 16 9 27 30 19 2 14 1 ...
```

Come vedete, ho raccolto 10 colonne di dati relativi a questi miei studenti: in questo capitolo ci interesseranno soprattutto il loro genere `gender` – una variabile fattore a due livelli, `f` and `m`; la loro statura `height` in centimetri ed il loro peso `weight` in chilogrammi. Iniziamo ad occuparci della relazione più semplice (che sarebbe analoga alla situazione della `formula1`, raffigurata nel pannello di sinistra nella Figura 4.1):

```
> relation1 = weight ~ height
```

Siamo dunque interessati a scoprire una possibile relazione che predica il peso `weight` degli studenti `fresher` in funzione della loro statura `height`, ed in tal caso diremo che la statura risulterà essere un **predittore** del peso `weight`. Il fatto che `weight` preceda il segno della tilde significa che la pensiamo come un output, come una variabile dipendente posizionata sull'asse delle ordinate, `y`; al contrario `height` rappresenta l'input, la variabile indipendente situata sull'ascissa `x`.

Abbiamo detto che Galton cercava l'equazione della retta di regressione del tipo  $y = a + b \cdot x$  in modo tale che l'intercetta  $a$  e la pendenza  $b$  riuscissero a far attraversare alla retta la nuvola di punti grigi 'nel migliore modo possibile'. Questo concetto di 'migliore modo' viene espresso nei libri in una forma matematica nota con il nome di **Teorema di Gauss e Markov** ([22, pagina 15]): in questo teorema si dimostra che tale retta è la 'Best Linear Unbiased Estimate' ('BLUE'), secondo un metodo statistico esplorato già nel 1755 sulle sponde del Mare Adriatico dal nobile dalmata Ruggero Boscovich / Ruđer Bošković [52]: il **metodo dei minimi quadrati** (Ordinary Least Square, OLS). Spiegheremo fra poco come funziona questo metodo; ma prima chiediamo ad R di farci scoprire quanto valgono questi due coefficienti  $a$  e  $b$  con il comando `lm`, le iniziali di **linear model**:

```
lm(relation1)
```

```
...
```

**Coefficients:**

(Intercept)	height
-83.8906	0.8539

Abbiamo scoperto che  $a = -83.9$  e  $b = 0.854$ , ossia che la retta tratteggiata ha equazione

$$y = -83.9 + 0.854 \cdot x$$

In realtà, il comando `lm` chiede al software R di valutare una ben più vasta quantità di informazioni, che si possono esplicitare a video con il versatile comando `summary`, a suo tempo introdotto nel paragrafo 1.2.1. Come vedete vi sono (almeno) due tipi di informazioni che dovremo discutere: quelle dedicate ai `Coefficients` e quelle dedicate ai `Residuals`.

```
model1 = lm(relation1)
summary(model1)

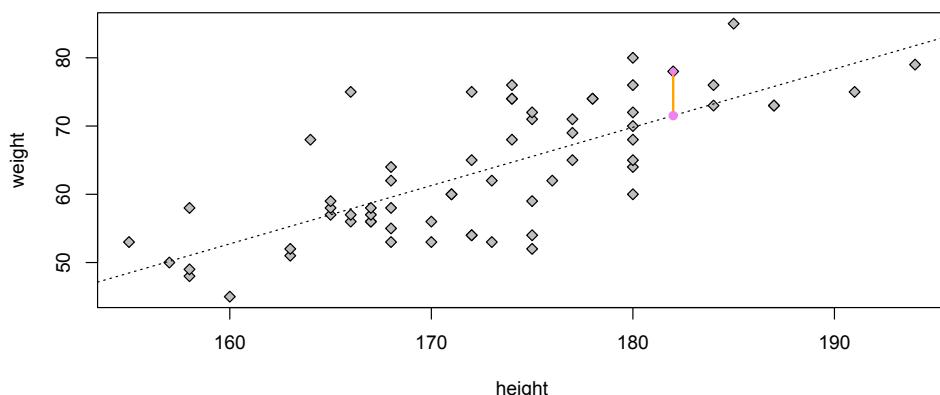
> model1 = lm(relation1)
> summary(model1)

...
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 ***
height       0.85392    0.09649   8.85 1.18e-12 ***

...
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
```

#### 4.1.1 I residui di un modello lineare



Cominciamo ad analizzare questo disegno per capire in dettaglio gli elementi del `summary`. Vedete quel diamante di colore violetto bordato di grigio? Si tratta di uno/a studente/ssa di statura 182 centimetri e di peso 78 chili. Il segmento (teorico) di colore arancione rappresenta uno dei 65 **residui**, tanti quanti sono i punti grigi della nuvola. Visto che conosciamo i **coefficienti** (ovvero i **parametri del modello lineare**)  $a$  e  $b$  possiamo calcolarne la lunghezza; infatti l'ipotetico pallino viola collocato sulla retta tratteggiata ha la medesima ascissa  $x = 182$  ma l'ordinata vale  $a + b \cdot 182 = -83.9 + 0.854 \cdot 182 \approx 71.5$ . Dunque il residuo (segmento arancione) ha lunghezza  $78 - 71.5 = 6.5$ . Volendo, potremmo anche far calcolare al software la lunghezza di tutti questi 65 bastoncini con il comando `resid()`. Riportiamo per brevità solo alcune delle righe di output, osservando che alcuni residui hanno un segno negativo, a significare che il punto della nuvola sta al di sotto della retta di regressione e non al di sopra:

```
resid(model1)

> resid(model1)
      1          2          3          4          5
 4.532268586 -0.175580381 -3.029504865 -2.029504865  6.970495135
...
      61          62          63          64          65
10.914534081 -2.793314886 -2.793314886 -4.209012820 -2.770786270
```

Volendo, potremmo cercare di individuare tra questi residui chi sia quello più grande tra quelli positivi, e chi tra quelli negativi, o chi sia il mediano, o quanto valga la media di questi residui. Fare insomma la statistica descrittiva dei residui:

```
summary(resid(model1))

> summary(resid(model1))
   Min. 1st Qu. Median Mean 3rd Qu. Max.
-13.546 -4.209 -1.569  0.000  4.431 17.139
```

Molto interessante: la media è zero; gli altri quantili elencati sono esattamente quelli che venivano riportati alla voce `Residuals` della pagina precedente.

### 4.1.2 La devianza di un modello lineare

Il metodo dei minimi quadrati di Ruggero Boscovich / Ruđer Bošković [52] (che però di solito si attribuisce a Carl Frederick Gauss) parte generalizzando l'idea di base del teorema di Pitagora: si considerano i residui, si elevano tutti al quadrato e si sommano tra loro tutte queste quantità ottenendo un risultato numerico che si chiama **devianza**:

```
sum(resid(model1)^2)
deviance(model1)

> sum(resid(model1)^2)
[1] 2628.627
> deviance(model1)
[1] 2628.627
```

È chiaro che spostando la pendenza  $b$  della retta, o l'intercetta  $a$ , le lunghezze dei residui cambieranno e cambierà di conseguenza la devianza ottenuta. Ebbene, è possibile dimostrare con una certa facilità che esistono un unico valore di  $a$  ed un unico valore di  $b$  che rende più piccola possibile la devianza; ecco perché parliamo di metodo dei 'minimi' quadrati e parliamo di **devianza del modello lineare tout court**.

### 4.1.3 La componente aleatoria di un modello lineare

.. wie kann Etwas aus seinem Gegensatzentstehen, zum Beispiel [...] Wahrheit aus Irrthuemern? (Friedrich Nietzsche, *Menschliches, Allzumenschliches*)

Su come possa nascere qualcosa dal suo opposto e come possa scaturire la verità dall'errore, lo vediamo immediatamente. Se indichiamo con  $\varepsilon$  il residuo 6.5, la situazione dello/a studente/ssa di statura  $x = 182$  centimetri e di peso  $y = 78$  chili viene perfettamente descritta nella relazione algebrica:

$$78 = -83.9 + 0.854 \cdot 182 + 6.5$$

Nella formulazione  $y = a + b \cdot x + \varepsilon$  la parte che riguarda i coefficienti  $a + b \cdot x$  viene detta **predittore lineare** del modello, o ancor meglio, **effetti fissi**; mentre l'insieme dei 65 residui  $\varepsilon$  si chiama **componente aleatoria** (se siete latinisti) o **componente stocastica** (se siete grecisti) del modello lineare, o ancor meglio, **effetti casuali**. Nel modello lineare, oltre alla richiesta algebrica di soddisfare il principio dei minimi quadrati, abbiamo un'ulteriore richiesta di natura statistica: si chiede di poter considerare questi 65 residui come dei numeri casuali che provengano da una variabile aleatoria gaussiana, come quelli che abbiamo imparato a simulare nella Sezione 2.3.4: una sequenza di 65 numeri casuali distribuiti normalmente con media zero e con deviazione standard pari a quella riportata nel summary del modello lineare alla voce **Residual standard error**,  $\approx 6.46$ . Quest'ultima richiesta in effetti è estremamente utile, e per capire bene il perché facciamo un paio di esercizi.

**Esercizio 4.1** Generate 65 numeri casuali gaussiani con media nulla e deviazione standard 6.46:

```
numericasuali = rnorm(65, mean = 0, sd = 6.46)
```

Mediane questi numericasuali generate 65 pesifittizi, utilizzando gli effetti fissi  $a$  e  $b$ :

```
pesifittizi = -83.9 + 0.854 * height + numericasuali
```

Ora disegnate due grafici: a sinistra la nube di punti reale; a destra, la nube di punti fittizi. Non sembra anche a voi che i due grafici si assomiglino?

```
par(mfrow = c(1,2))
plot(height, weight)
plot(height, pesifittizi)
```

■

Confrontate la situazione che emerge replicando la medesima idea sull'esempio del dataset raul. Riprendiamo la formula1 che esplicitava la singola retta di regressione di colore nero tratteggiato modello1 e determiniamone le informazioni essenziali,  $a$ ,  $b$  e  $\sigma$ :

```
modello1 = lm(formula1)
modello1
summary(modello1)$sigma

> modello1 = lm(formula1)
> modello1
```

*Call:*  
*lm(formula = formula1)*

*Coefficients:*

(Intercept)	Ldhuno
28.8408	-0.2231

```
> summary(modello1)\$sigma
[1] 2.502858
```

Molto bene, abbiamo scoperto che nella Figura 4.1 la retta nera tratteggiata ha equazione  $y = 28.8 - 0.223 \cdot x$ , e che i residui  $\varepsilon$  si distribuiscono con media nulla e deviazione standard  $\sigma = 2.5$  (si tratta del Residual standard error del modello1 che si può leggere nel summary). Ripetiamo l'esercizio precedente, generando 1610 numeri casuali che perturbano i valori stimati dell'LDH3:

```
numericasuali = rnorm(1610, mean = 0, sd = 2.5)
LDH3fittizio = 28.8 - 0.223 * Ldhuno + numericasuali
par(mfrow = c(1,2))
plot(Ldhuno, Ldhtre)
plot(Ldhuno, trunc(LDH3fittizio))
```

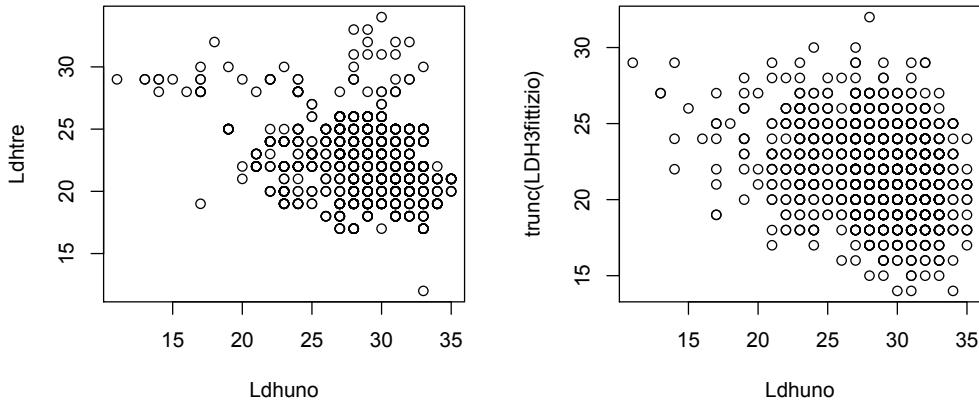


Figura 4.2: A sinistra, i dati reali di Ldhtre in funzione di Ldhuno. A destra, i dati di un LDH3fittizio simulato in base al modello1: non vanno affatto bene.

Vedete che la nube di destra della Figura 4.2 ora non assomiglia più a quella di sinistra, come invece accadeva nell'esercizio 4.1? Ebbene, in questa semplice simulazione abbiamo una chiave per capire che in un modello lineare è necessario occuparsi non solo della struttura degli effetti fissi, ma anche di quella degli effetti casuali.

#### 4.1.4 Il modello nullo è importante

Ora, una questione che potrebbe apparire a prima vista irrilevante: che senso avrebbe una retta di regressione 'orizzontale'? Lo stesso senso che avrebbe interpellare al telefono una fattucchiera, dirle il nostro giorno di nascita e farsi predire il peso. Una sciocchezza? Appunto: proviamo a predire il peso *weight* dei nostri *fresher* conoscendo il giorno *day* del loro compleanno:

```

formulascioccia = weight ~ day
modellosciocco = lm(formulascioccia)
modellosciocco
mean(weight)

> formulascioccia = weight ~ day
> modellosciocco = lm(formulascioccia)
> modellosciocco

Call:
lm(formula = formulascioccia)

Coefficients:
(Intercept)      day
6.352e+01    1.618e-05

> mean(weight)
[1] 63.52308

```

Otteniamo una retta che ha equazione  $y = 63.52 + 0.000016 \cdot x$ , praticamente una retta orizzontale (infatti la pendenza  $b$  è praticamente zero) che passa per la media del weight, 63.5. Dunque, siamo in questa situazione: abbiamo delle informazioni sul weight, vorremmo interpretarle in senso statistico, ma (facciamo finta che) non abbiamo predittori validi da associare loro. Gli statistici Wilkinson e Rogers [62] hanno anche ideato un modo di indicare il **modello lineare nullo**, utilizzando il simbolo 1. Confrontiamo a tale proposito il **summary** con la media e la deviazione standard del peso weight:

```

relation0 = weight ~ 1
modellonullo = lm(relation0)
summary(modellonullo)
mean(weight)
sd(weight)

> relation0 = weight ~ 1
> modellonullo = lm(relation0)
> summary(modellonullo)

Call:
lm(formula = relation0)

Residuals:
    Min     1Q   Median     3Q    Max 
-18.523 -7.523 -1.523  9.477 21.477 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 63.523     1.191   53.36 <2e-16 ***
...
Residual standard error: 9.598 on 64 degrees of freedom

```

```
> mean(weight)
[1] 63.52308
> sd(weight)
[1] 9.598352
```

Proviamo ad interpretare il `summary` del `modelonullo` alla luce dei primissimi discorsi che abbiamo fatto nel Capitolo 1 sulla statistica descrittiva. L'intercetta del `modelonullo` vale 63.5, e rappresenta il 'baricentro' dei dati; si tratta di un indice di posizione dei dati ed è infatti la media, `mean(weight)`. Comunicare la media dei dati significa trasmettere una ('1') sola informazione; ma siccome gli studenti erano 65, per descrivere completamente il loro peso ci mancherebbero da conoscere ancora 64 ulteriori informazioni: 64 degrees of freedom. I dati non sono ovviamente concentrati tutti nella media ma vi è una naturale variabilità dei pesi; infatti la deviazione standard `sd(weight)` vale circa 9.6, e la componente aleatoria del modello lineare è descritta dal `Residual standard error`. Adesso finalmente ci risulta chiaro perché nella Sezione 1.3 avevamo detto che media e deviazione standard sono un modo 'parametrico' di descrivere i dati: perché essi sono i parametri (1 effetto fisso + 1 effetto casuale) del modello lineare nullo.

#### 4.1.5 Hirotugu Akaike, un nome da ricordare per sempre

Ora, un grande passo finale che arricchisce il concetto di devianza del modello lineare. Riprendiamo i dati dei nostri `fresher`, ricordando che la pendenza del `model1` era  $b = 0.854$  (se racchiudiamo un comando di R tra parentesi tonde otteniamo immediatamente a video il contenuto dell'output):

```
( b = summary(model1)$coefficient[2] )
> (b = summary(model1)\$coefficient[2])
[1] 0.8539245
```

Nella Sezione 3.4.2 avevamo vagheggiato la possibilità di moltiplicare la pendenza  $b$  per un qualcosa che fosse il suo inverso algebrico e che tenesse conto del disordine dei dati, del tipo  $b \cdot \frac{\sigma_x}{\sigma_y}$ . Sarebbe stato questo il coefficiente di correlazione lineare  $\rho$  di Bravais e Pearson:

```
b * sd(height) / sd(weight)
cor(height, weight)

> b * sd(height) / sd(weight)
[1] 0.7444353
> cor(height, weight)
[1] 0.7444353
```

Siccome il coefficiente  $\rho$  ha lo stesso segno positivo o negativo della pendenza  $b$  della retta, per liberarsi da questo impiccio gli statistici hanno da sempre preferito utilizzare il suo valore elevato al quadrato  $\rho^2$ , definendolo con il nome di **coefficiente di determinazione** e denotandolo con il simbolo  $R^2$ . Questo coefficiente viene proposto nel `summary` del modello lineare alla voce `Multiple R-squared`:

```
cor(height, weight)^2
summary(model1)$r.squared

> cor(height, weight)^2
[1] 0.554184
> summary(model1)\$r.squared
[1] 0.554184
```

Ora, in un paese ed in un tempo molto molto lontano, viveva un fisico di nome Ludwig Eduard Boltzmann che studiava la termodinamica e parlava di **entropia**; ed in un altro paese ed in un altro tempo molto ma non molto lontano c'erano due matematici, Richard Leibler e Solomon Kullback detto Kully [10], che si occupavano di decifrare i codici segreti nazisti e che parlavano della **misura di informazione di Kullback e Leibler**. E questi ultimi due matematici si erano accorti che la loro misura di informazione altro non era che l'entropia di Boltzmann cambiata di segno (la *negentropy*). Colpo di scena: installiamo in R il pacchetto **rsq**, acronimo di **r squared**, e calcoliamo il coefficiente R quadrato secondo Kullback e Leibler con la funzione **rsq.kl**:

```
install.packages("rsq")
library(rsq)
rsq.kl(model1)

...
> library(rsq)
> rsq.kl(model1)
[1] 0.554184
```

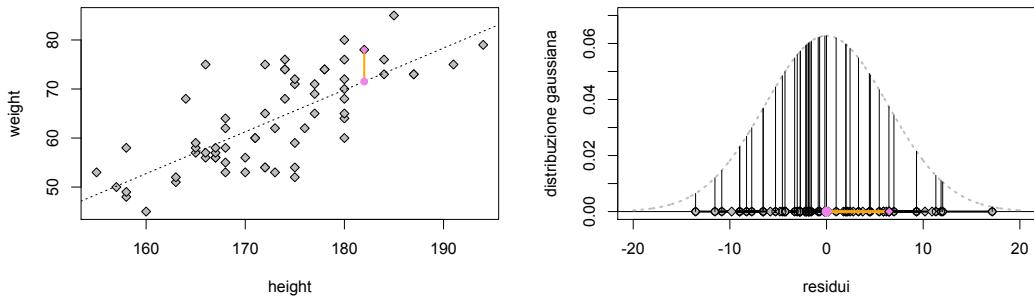
Cosa c'entra tutto questo con la devianza? Ecco l'illuminazione. Consideriamo il modello in questione e raffrontiamolo con il modello nullo. Anzi, consideriamo il rapporto tra le devianze dei due modelli e calcoliamo 'quanto manca' alla perfezione del 100 per cento:

```
deviance(model1)
model0 = lm(weight ~ 1)
deviance(model0)
1 - deviance(model1)/deviance(model0)

> deviance(model1)
[1] 2628.627
> model0 = lm(weight ~ 1)
> deviance(model0)
[1] 5896.215
> 1 - deviance(model1)/deviance(model0)
[1] 0.554184
```

Meraviglioso: si tratta proprio del coefficiente  $R^2$ , ossia la divergenza di Kully e Leibler. Adesso abbiamo quest'ultima domanda, riferendoci al disegno sottostante: immaginando quindi che tutti i residui siano delle realizzazioni di una variabile aleatoria gaussiana, abbiamo un modo per calcolare quale sia la probabilità che si manifestino proprio quei 65 residui? Ossia proprio quei 65 diamanti grigi rispetto alla retta di regressione grigia che prendiamo per buona?

Procediamo (con la fantasia) in questo modo. Ribaltiamo la retta di regressione in verticale, in modo tale che 'tutti i residui cadano al suolo'; in particolare, nella figura a destra, vediamo rappresentato nei colori arancioni e violetto il residuo  $\varepsilon \approx 6.5$  dello/a studente/ssa di statura 182 centimetri e di peso 78 chili. Per conoscere con quale 'probabilità istantanea' questo si possa verificare dovremmo usare il comando **dnorm** introdotto nella Sezione 2.3.4; non c'è dubbio sulla media zero di questa gaussiana, ma per conoscere la deviazione standard dobbiamo fare un cambio di scala, ricorrendo alla **stima di massima verosimiglianza** (in inglese '**ML, Maximum Likelihood**') del residual standard error, in cui teniamo conto dei 2 effetti fissi ( $a$  e  $b$ ). Compare dunque una radice quadrata che aggiusta di poco la stima di  $\sigma$  passando da 6.46 a 6.36:



```

sigmamodel1 = summary(model1)$sigma
sigmamodel1
sigmaMLmodel1 = sigmamodel1 * sqrt( (65 - 2) / 65 )
sigmaMLmodel1

> sigmamodel1 = summary(model1)\$sigma
> sigmamodel1
[1] 6.459431
> sigmaMLmodel1 = sigmamodel1 * sqrt( (65 - 2) / 65 )
> sigmaMLmodel1
[1] 6.359278

```

Allora, la 'probabilità istantanea' di osservare quel particolare residuo  $\varepsilon \approx 6.5$  vale circa il 3.7%:

```

dnorm(6.5, mean = 0, sd = sigmaMLmodel1)

> dnorm(6.5, mean = 0, sd = sigmaMLmodel1)
[1] 0.03720817

```

Ora, se volessimo conoscere la probabilità di osservare esattamente quei 65 residui, cioè se volessimo conoscere la **verosimiglianza del modello lineare**, bisognerebbe calcolare tutte quelle 65 probabilità e moltiplicarle tra di loro. Ma il risultato sarebbe un numero piccolissimo, vicinissimo allo zero; ecco perché da sempre si preferisce considerare il logaritmo della verosimiglianza, in gergo la **logverosimiglianza**, che trasforma i numeri 'positivi piccoli' in numeri 'negativi grandi', e le moltiplicazioni in somme:

```

sum(log(dnorm(resid(model1), mean = 0, sd = sigmaMLmodel1)))
logLik(model1)

> sum(log(dnorm(resid(model1), mean = 0, sd = sigmaMLmodel1)))
[1] -212.4755
> logLik(model1)
'log Lik.' -212.4755 (df=3)

```

Ebbene, non vi stupirà più di tanto sapere che la logverosimiglianza del modello si possa mettere in relazione con tutti i discorsi precedenti relativi alla devianza del modello lineare. Ma veniamo al punto finale del discorso: l'ingegner Hirotugu Akaike nel 1974 [1] si accorse che penalizzando (ossia, 'amplificando la misura di disordine') la logverosimiglianza (-212.5) con il

numero di parametri del modello (2 effetti fissi ed 1 effetto casuale,  $df=3$ ), si poteva ottenere un criterio di scelta dei modelli statistici che tenesse conto sia di quanto 'costa' un modello (appunto, il numero di parametri) sia di quanto esso 'riduce il disordine' della nube di punti. Nasceva allora il **criterio di informazione di Akaike**, AIC:

```
2 * ( 3 - logLik(model1) )
AIC(model1)

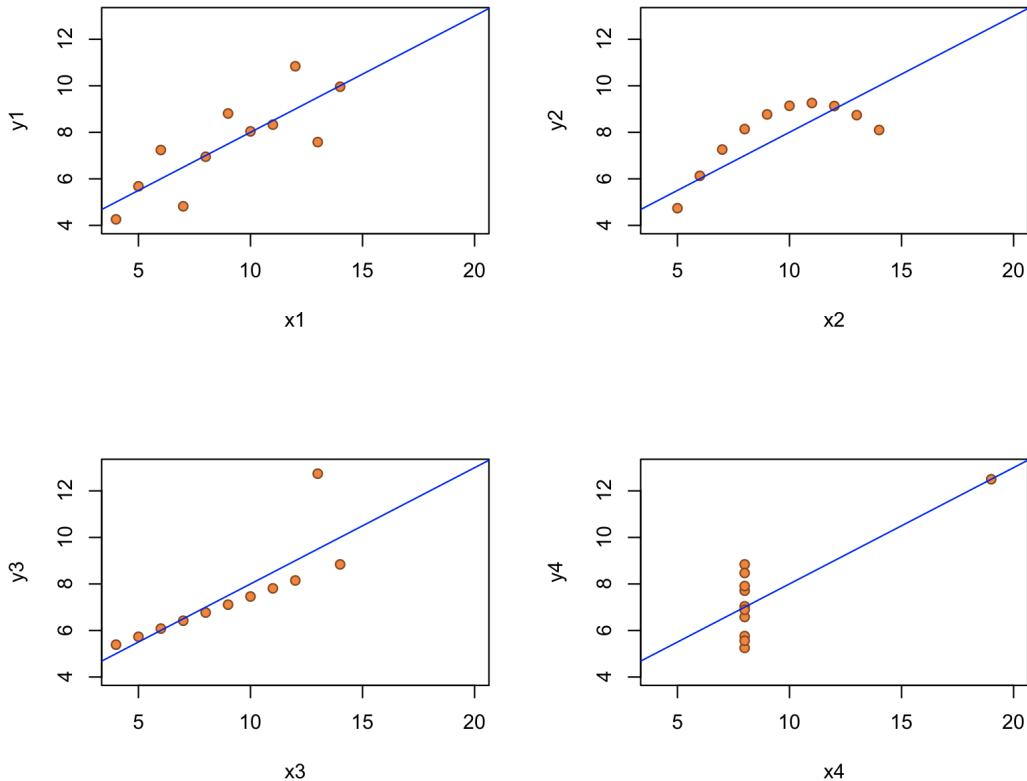
> 2 * ( 3 - logLik(model1) )
'log Lik.' 430.9509 (df=3)
> AIC(model1)
[1] 430.9509
```

Vedremo già nella sezione 4.3 come questo criterio AIC sarà lo strumento che ci permetterà di scegliere il modello statistico più opportuno, in barba a tutti i famigerati p-value.

#### 4.1.6 La diagnostica del modello lineare

Correva l'anno 1973, un anno prima che Hirotugu Akaike ci insegnasse a calcolare il criterio AIC, ed in un paper didattico [4] Francis Anscombe illustrava quattro dataset artificiali di notevole interesse. Si trattava di quattro semplici nuvole di punti che potete vedere raffigurate anche su Wikipedia:

[https://en.wikipedia.org/wiki/Anscombe%27s\\_quartet](https://en.wikipedia.org/wiki/Anscombe%27s_quartet)



La particolarità di questi dataset consisteva nel fatto che tutti avevano come modello lineare la medesima retta  $y = 3 + 0.5 \cdot x$  ed il medesimo coefficiente di determinazione  $R^2 = 2/3 = 0.66$ . Tuttavia, chiunque di noi vedendo i disegni si riesce a rendere conto che se nel primo disegno la

retta di regressione appariva essere un buon modello statistico, nel secondo l'evidente curvatura della nube rende irragionevole il modello lineare, mentre nel terzo e nel quarto disegno in maniera addirittura drammatica è presente un punto isolato dagli altri che possiede una 'forza di leva', un *leverage*, tale da spostare il modello lineare da quello che apparirebbe verosimilmente essere il modello 'giusto' (nel terzo disegno una retta deterministica ma con pendenza minore, nel quarto una retta deterministica verticale).

**Esercizio 4.2** Provate a riproporre da voi il disegno che trovate su Wikipedia, e che è del tutto analogo a quello raffigurato qui sopra. Avrete bisogno del comando `attach(anscombe)` per importare i dati, e poi con `str(anscombe)` oppure direttamente con `anscombe` vi potrete fare un'idea di quanti siano i dati e di come si chiamino le variabili. Ricordatevi che con lo stranissimo comando `par(mfrow = c(2,2))` si riescono ad organizzare quattro plot in maniera affiancata, due per due.

Abbiamo dunque bisogno di qualche strumento che ci consenta di giudicare la 'qualità' del nostro modello lineare, perché se tutto risulta immediato nel caso bidimensionale, non è altrettanto immediato quando si deve compiere un'analisi multivariata. Spieghiamoci con un esempio concreto: ricorderete che nel Capitolo 1 avevamo introdotto il dataset `airquality`, nel quale la nociva presenza di Ozone nell'aria poteva dipendere dalla quantità di radiazione solare `Solar.R` rilevata quel giorno, dall'intensità del vento `Wind` e dalla Temperatura dell'aria espressa in gradi Fahrenheit. Avevamo già detto che non si trattasse di un dataset di tipo cross-section, ma Julian Faraway [22, pagina 62] ci rassicura e, almeno dal punto di vista didattico, possiamo procedere nell'analisi ipotizzando una prima relazione. Ecco i comandi, ed uno stralcio dell'output (così risparmiamo un po' di inchiostro, o di elettroni come preferite):

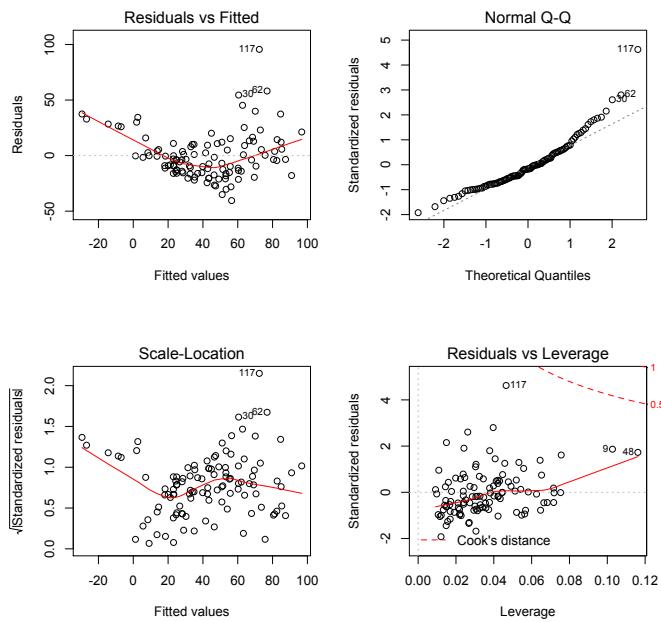
```
attach(airquality)
ipotesi1 = Ozone ~ Solar.R + Wind + Temp
modellodiscutibile = lm(ipotesi1)
summary(modellodiscutibile)

...
Residual standard error: 21.18 on 107 degrees of freedom
(42 observations deleted due to missingness)
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.5948
F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16
```

Tutto sommato, si vede un ottimo p-value ed il coefficiente  $R^2$  supera il 60 per cento: potrebbe andare bene, no? No! Ed ora spieghiamo perché. Digitiamo questo comando:

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

I quattro grafici che otteniamo in gergo si chiamano i **plot diagnostici** e servono a fare chiarezza se le ipotesi matematiche richieste dal modello lineare siano o meno verificate. Il primo dei quattro è forse il più importante e rappresenta il grafico dei residui del modello lineare rispetto ai valori predetti, 'fitted'. Siccome i residui dovrebbero poter essere equiparati a dei numeri generati casualmente, quel grafico dovrebbe assomigliare ad un caos di punti, una specie di via lattea nelle romantiche notti di agosto. Ed inoltre, siccome i residui dovrebbero avere media nulla, quella linea rossa dovrebbe apparire grossomodo come una retta orizzontale passante per lo zero. E ciò non accade, ci sono due aspetti che ci preoccupano: notiamo innanzitutto una pronunciata curvatura della riga rossa; inoltre, i punti appaiono addensarsi attorno alla riga, mentre invece dovrebbero



essere sparpagliati in maniera informe in tutto il riquadro. Dunque nutriamo forti dubbi sulla adeguatezza del modello discutibile.

Nel secondo pannello ritroviamo il grafico di cui abbiamo parlato nella Sezione 2.3.4, il QQ plot. Siccome i residui dovrebbero essere generati come una variabile aleatoria gaussiana, in questo grafico dovremmo vedere una 'diagonale' di punti e non una forma del tipo di una gondola o di un serpente.

Il terzo grafico è una variante del primo; in esso i residui negativi vengono 'ribaltati' considerando il loro valore assoluto positivo. Questo grafico è efficace per individuare se il 'residual standard error', ossia la dispersione  $\sigma$  che genera i residui  $\varepsilon$  nella variabile aleatoria gaussiana  $N(0, \sigma)$  è un numero costante, indipendente dalla posizione o dalla dimensione dei punti; ovvero la sua non-costanza si manifesti in un disegno con la forma di cuneo, o di fetta di torta. Qui è evidente la forma triangolare della nube, fatta salva la presenza di uno sparuto gruppetto di residui nel settore di sinistra: è questo un problema di **eteroschedasticità** dei residui.

Infine il quarto grafico mette in luce il problema che Anscombe esibiva nel suo terzo e quarto esempio, laddove un punto isolato riusciva ad avere una forza di leva sufficiente a spostare la retta di regressione. Il quarto grafico diagnostico valuta una quantità algebrica detta la **distanza di Cook**, ossia una misura congiunta di 'leverage' e di ampiezza del residuo  $\varepsilon$  in un singolo numero, che riesce ad illustrare la presenza di punti isolati particolarmente influenti i quali – se presenti – appaiono come punti isolati al di fuori di alcune iperboli rosse. In questo caso, fortunatamente non vi sono evidenti punti con forza di leva.



Per approfondire ulteriori aspetti dei plot diagnostici date un'occhiata al documento intitolato *Data Quality Assessment Statistical Methods for Practitioners* pubblicato dall'EPA, la Agenzia di Protezione Ambientale degli Stati Uniti (sul sito <https://nepis.epa.gov>, o semplicemente cercandolo con Google).

**Esercizio 4.3** Controllate che nel dataset `anscombe`, il terzo esempio esibisce un punto isolato:

```
> attach(anscombe)
```

```
> m3 = lm(y3 ~ x3)
> par(mfrow = c(2,2))
> plot(m3)
```



#### 4.1.7 Lineare non è sinonimo di rettilineo

Nella sezione precedente i plot diagnostici hanno messo in luce che la ipotesi1 nella quale l'Ozone viene posto in relazione in maniera additiva alle tre variabili Solar.R + Wind + Temp presenta problemi di eteroschedasticità nella componente aleatoria del modello e di curvatura negli effetti fissi. Proviamo a migliorare il modello introducendo un termine speciale di cui spiegheremo immediatamente il significato:

```
ipotesi2 = Ozone ~ Solar.R + Wind + Temp + I(Temp^2)
modellomigliore = lm(ipotesi2)
summary(modellomigliore)

...
Residual standard error: 19.93 on 106 degrees of freedom
(42 observations deleted due to missingness)
Multiple R-squared:  0.6544,      Adjusted R-squared:  0.6414
F-statistic: 50.18 on 4 and 106 DF,  p-value: < 2.2e-16
```

Osserviamo che abbiamo avuto un miglioramento del coefficiente R quadrato, passando dal 60% al 65%, ma non è questo il dato realmente importante. Ciò che importa, ed il terzo dei plot diagnostici lo mette in luce, è che abbiamo eliminato l'eteroschedasticità dei residui, ossia la forma triangolare se ne è andata via. Ed anche nel primo plot diagnostico la curvatura della riga rossa è molto meno evidente, fatto salvo il comportamento di una decina di punti strani, che vediamo anche deformare il Q-Q plot nella sua coda di destra.

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

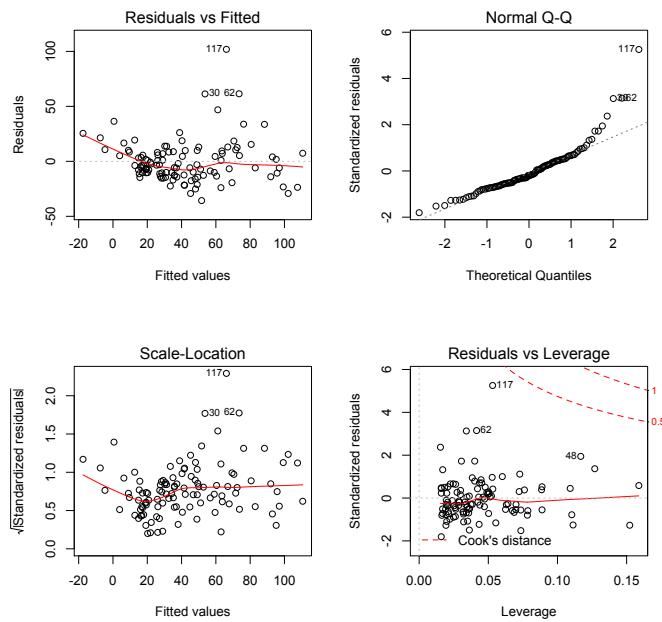
A questo punto vogliamo spiegare cosa significa quel termine che eleva Temp al quadrato:

```
ipotesi2 = Ozone ~ Solar.R + Wind + Temp + I(Temp^2)
```

Potremmo pensarla come un termine di curvatura che trasforma la 'retta di regressione' in una 'parabola di regressione', confermando ciò che avevo anticipato nel titolo, ossia che la parola 'lineare' in matematica non significa esattamente che ciò che raffiguriamo sia 'rettilineo'. In realtà le cose sono più banali, abbiamo semplicemente creato una nuova variabile che prende i valori di Temp e li eleva al quadrato (ecco per esempio i primi tre valori incolonnati per confronto con quelli di Temp):

```
cbind(Temp, I(Temp^2))[1:3,]

> cbind(Temp, I(Temp^2))[1:3,]
      Temp
[1,] 67 4489
[2,] 72 5184
[3,] 74 5476
```



e poi con `lm` si va a cercare quali siano quei cinque coefficienti  $a, b, c, d, e$  che in base al metodo dei minimi quadrati ottimizza la devianza della funzione

$$y = a + b \cdot \text{Solar.R} + c \cdot \text{Wind} + d \cdot \text{Temp} + e \cdot (\text{Temp})^2$$

#### 4.1.8 Anche il t test è un modello lineare

Vi andrebbe ora una buona pinta di birra Guinness con qualche stuzzichino per aperitivo? Potremmo così festeggiare la 'scoperta' del fatto che il modello lineare `lm` può risultare del tutto equivalente al test t di Student. Vi ricordo brevemente cosa avevamo imparato nella storia del Capitolo 3.2:

```
differenzaresa = c(106, -20, 101, -33, 72, -36, 62, 38, -70, 127, 24)
t.test(differenzaresa)
```

```
> differenzaresa = c(106, -20, 101, -33, 72, -36, 62, 38, -70, 127, 24)
> t.test(differenzaresa)
```

*One Sample t-test*

```
data: differenzaresa
t = 1.6905, df = 10, p-value = 0.1218
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-10.72710 78.18164
sample estimates:
mean of x
33.72727
```

Raffrontate voi stessi le quantità che abbiamo calcolato qui sopra (la media 33.7, il quantile t = 1.69, il p-value 12.2%) con il modello nullo seguente:

```

ipotesinulla = differenzaresa ~ 1
modelloGosset = lm(ipotesinulla)
summary(modelloGosset)

> ipotesinulla = differenzaresa ~ 1
> modelloGosset = lm(ipotesinulla)
> summary(modelloGosset)

...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.73     19.95   1.69   0.122
Residual standard error: 66.17 on 10 degrees of freedom

```

Avrete certamente riconosciuto che l'intercetta 33.7 del modelloGosset è proprio la media della differenzaresa. Ora controllate il valore 66.17 dell'errore standard dei residui ed il valore 19.95 dell'errore standard della media, ed intanto calcoliamo:

```

sd(differenzaresa)
sd(differenzaresa)/sqrt(11)

> sd(differenzaresa)
[1] 66.17113
> sd(differenzaresa)/sqrt(11)
[1] 19.95135

```

ed infine convincetevi che la cosa funziona perfettamente, ricordando il discorso del 'rapporto segnale rumore':

$$t = \frac{x_m - \mu}{s/\sqrt{n}} = \frac{33.7 - 0}{19.95} = 1.69$$

Conclusione? Il t test ad un solo campione è del tutto equivalente ad un modello lineare nullo.

**Esercizio 4.4** Riprendete la Sezione 3.2.2 nella quale avevamo considerato le 1568 pazienti benigne del dataset raul e le 42 maligne 'splittandole' in due gruppi (quello verde e quello rosso) in modo tale da capire se il biomarcatore Ca125 si esprimesse in maniera differente tra i due gruppi, in senso statistico. Ecco uno stralcio dell'output:

```

t = -4.9067, df = 42.584, p-value = 1.401e-05
...
mean of x mean of y
27.94388 52.97619

```

Controllate che il seguente modellobiomarker offre delle informazioni apparentemente diverse:

```

ipotesibiomarker = Ca125 ~ Esito
modellobiomarker = lm(ipotesibiomarker)
summary(modellobiomarker)

```

In questo esercizio c'è un passaggio che merita attenzione. Vi sarete accorti che il summary offre due coefficienti: (Intercept), che vale circa 27.94, ed Esitomaligno che vale circa 25.03. Osservate che il numero 27.94 è la media del gruppo verde, mentre per ottenere la media del gruppo rosso, circa 52.97, dovremmo combinare tra loro quei due numeri:  $27.94 + 25.03 = 52.97$ . La spiegazione di questa 'addizione' sta nel fatto che il modello lineare è sempre rappresentato nella forma  $y = a + b \cdot x$  e per convenzione si rispetta l'ordine alfabetico: al livello benigno si attribuisce  $x = 0$ , e quindi  $y = 27.94 + 25.03 \cdot 0 = 27.94$ , mentre al livello maligno si attribuisce  $x = 1$ , e quindi  $y = 27.94 + 25.03 \cdot 1 = 27.94 + 25.03 = 52.97$ .

**Esercizio 4.5** Riprendete l'esercizio precedente, e convincetevi immediatamente sul fatto che il Ca125 non è affatto un valido predittore dell'Esito, esaminando i plot diagnostici:

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

Concludiamo questa sezione riprendendo il dataset `fresher` degli studenti di medicina, e ci ricordiamo di quanto abbiamo appena visto, ossia che la relazione:

```
relation1 = weight ~ height
```

fornisce un valido modello lineare, ossia abbiamo stabilito che la statura `height` è un predittore dal peso `weight`. Adesso vogliamo osservare che anche il genere `gender` è un predittore dal peso `weight`:

```
relation2 = weight ~ gender
model2 = lm(relation2)
summary(model2)

...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.625     1.199   47.237 < 2e-16 ***
gender      13.587     1.682    8.076 2.63e-11 ***
...
Residual standard error: 6.781 on 63 degrees of freedom
Multiple R-squared:  0.5087,          Adjusted R-squared:  0.5009
F-statistic: 65.22 on 1 and 63 DF,  p-value: 2.627e-11
```

Vogliamo discutere quindi nella prossima sezione di come sia possibile riunire tra loro i predittori `height` e `gender` per 'migliorare la predittività' del modello statistico.

## 4.2 Ancova: unire i predittori numerici ai fattori

Abbiamo iniziato questo Capitolo 4 mostrando nella ormai famosa Figura 4.1 due possibili situazioni che descrivevano il comportamento di `Ldhtre` in funzione di `Ldhuno`, e ci chiedevamo perché dovremmo preferire la situazione di destra a quella di sinistra o viceversa. Della situazione di sinistra nella Figura 4.1, quella della retta di regressione di colore nero, ci siamo occupati in dettaglio nella sezione 4.1 utilizzando la relazione:

```
formula1 = Ldhtre ~ Ldhuno
```

nella quale la variabile numerica Ldhtre vuole essere predetta dalla variabile numerica Ldhuno. Nella sezione 4.1.8 abbiamo invece interpretato il t test alla luce di un modello lineare, in cui la variabile numerica Ca125 vuole essere predetta dalla variabile fattore Esito:

```
ipotesibiomarker = Ca125 ~ Esito
```

Ora ritornando al pannello destro della Figura 4.1, siamo in una situazione in cui la variabile numerica Ldhtre vuole essere predetta da una variabile numerica Ldhuno congiuntamente ad una variabile fattore, Esito:

```
formula2 = Ldhtre ~ Ldhuno * Esito
```

Vediamolo in maggiore dettaglio sul dataset `fresher`, ampliando il modello introdotto nella Sezione 4.1; vogliamo capire se è opportuno modificare la `relation1` nella quale volevamo predire il peso `weight` solamente per mezzo della statura `height` introducendo l'informazione relativa al `gender`: in magenta le f, in azzurro / blu i m.

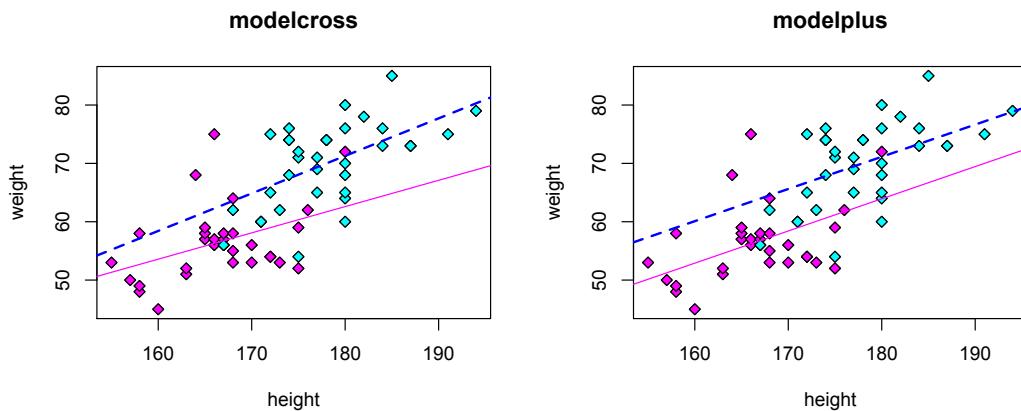


Figura 4.3: Due modelli lineari che 'spiegano' il comportamento del `weight` in funzione di `height`. A sinistra nel `modelcross` maschi e femmine hanno due rette con pendenze diverse; a destra nel `modelplus` la pendenza è la stessa, le rette sono parallele. Quale sarà il migliore modello?

Spieghiamo dunque quale differenza intercorra tra le due simbologie:

```
relationcross = weight ~ height * gender
relationplus = weight ~ height + gender
```

Nel modello descritto dalla formula `relationcross` le due rette di regressione non sono parallele tra di loro, come invece accade nel modello relativo alla relazione `relationplus`. Esaminiamo il `summary` del modello lineare corrispondente:

```
modelcross = lm(relationcross)
summary(modelcross)

...
Coefficients:
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -18.5041   30.1517  -0.614   0.5417
height       0.4505    0.1807   2.493   0.0154 *
genderm     -25.9617  42.9882  -0.604   0.5481
height:genderm 0.1925    0.2493   0.772   0.4430
...
```

I primi due coefficienti, (Intercept) ed height, si riferiscono ai punti il cui gender compare per primo in ordine alfabetico, ossia le f: la retta rosa ha pertanto equazione  $y = -18.50 + 0.45 \cdot x$ . Invece, i termini (genderm) ed height:genderm esprimono le quantità che perturbano rispettivamente l'intercetta e la pendenza della retta rosa, trasformandola in quella blu:  $y = (-18.50 - 25.96) + (0.45 + 0.19) \cdot x$ , ossia  $y = -44.46 + 0.64 \cdot x$ . Notate qui che la notazione di Wilkinson e Rogers utilizza anche il simbolo : per indicare il termine di interazione. Per capirci, invece di scrivere:

```
relationcross = weight ~ height * gender
```

avremmo potuto specificare il medesimo modello con la formula:

```
relationcross = weight ~ height + gender + height:gender
```

ma nessuno in pratica utilizza questa scrittura e tutti preferiscono usare il simbolo \*. Vediamo invece cosa accade nella situazione raffigurata a destra, modelplus:

```
modelplus = lm(relationplus)
summary(modelplus)

...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -35.3725   20.7162  -1.707 0.092736 .
height       0.5517    0.1241   4.447 3.68e-05 ***
genderm     7.1965    2.0606   3.492 0.000889 ***
...
```

Le due rette nella figura sono parallele, possiedono quindi la medesima pendenza; leggiamo questa informazione in corrispondenza di height, 0.55. La retta rosa parte da una quota più bassa (Intercept -35.4) rispetto a quella blu (Intercept + genderm,  $-35.4 + 7.2 \approx -28.2$ ). Pertanto in questo modello lineare la retta rosa ha equazione  $y = -35.4 + 0.55 \cdot x$ , mentre la retta blu è  $y = -28.2 + 0.55 \cdot x$ .

### 4.3 Facciamo il punto della situazione

Riassumiamo quanto abbiamo veduto sinora relativamente al dataset fresher, nel quale abbiamo preso in considerazione cinque possibili relazioni che 'spiegano' il comportamento del peso degli studenti:

```
relation0 = weight ~ 1
relation1 = weight ~ height
relation2 = weight ~ gender
relationplus = weight ~ height + gender
relationcross = weight ~ height * gender
```

Da queste relazioni abbiamo stimato i parametri dei cinque modelli lineari associati ad esse: `model0`, `model1`, `model2`, `modelplus` e `modelcross`. Ripetiamo il loro significato algebrico, statistico e geometrico:

1. `model0` riassume il peso `weight` degli studenti semplicemente per mezzo di due parametri: la media 63.5 Kg e la deviazione standard 9.6 Kg. Questi due numeri sono rispettivamente l'effetto fisso e l'effetto casuale ('residual standard error') del modello lineare nullo, che geometricamente si può rappresentare come la retta orizzontale  $y = 63.5$
2. `model1` descrive il peso `weight` degli studenti in funzione della loro statura `height`, per mezzo di tre parametri: due effetti fissi che rappresentano l'intercetta e la pendenza della retta di regressione  $y = -83.9 + 0.85 \cdot x$ , ed un effetto casuale 6.5 (come vedete, minore di quello del precedente `model0`) che descrive la variabilità della nuvola di punti che circonda la retta di regressione; o più propriamente, dei residui pensati come perturbazioni casuali del modello distribuite secondo una variabile aleatoria gaussiana di media nulla e deviazione standard 6.5.
3. `model2` è il celeberrimo t test di Student: si vuole descrivere il peso `weight` degli studenti in funzione del loro genere `gender`, ancora per mezzo di tre parametri: due effetti fissi che rappresentano rispettivamente il peso medio 56.6 Kg ('intercetta') del primo gruppo secondo l'ordine alfabetico –le femmine– e l'incremento di 13.6 Kg (il differenziale, la 'pendenza' `genderm`) del secondo gruppo per ordine alfabetico –i maschi–; ed ancora un effetto casuale 6.8 che descrive la variabilità (sempre in termini gaussiani) della nuvola di punti attorno alle loro medie corrispondenti. Geometricamente possiamo visualizzare due rette orizzontali, la retta 'rosa'  $y = 56.6$  e la retta 'azzurra'  $y = 70.2$ .
4. `modelplus` descrive il peso `weight` degli studenti sia in funzione della loro statura `height` che del loro genere `gender`. Stavolta occorrono quattro parametri: due effetti fissi che rappresentano l'intercetta e la pendenza della retta di regressione  $y = -28.2 + 0.55 \cdot x$  per le femmine, un effetto fisso 7.2 che rappresenta l'incremento di intercetta per la retta di regressione  $y = -35.4 + 0.55 \cdot x$  dei maschi; ed infine un effetto casuale 6.0 che descrive la variabilità dei residui gaussiani. Un tempo veniva chiamato modello *ancova without interaction*.
5. `modelcross` infine descrive il peso `weight` degli studenti sia in funzione della loro statura `height` che del loro genere `gender` ma utilizzando cinque parametri: due effetti fissi per la retta di regressione 'rosa' delle femmine  $y = -18.5 + 0.45 \cdot x$ ; due effetti fissi per la retta di regressione 'azzurra' dei maschi  $y = -44.5 + 0.64 \cdot x$ ; un effetto casuale 6.0 per i residui gaussiani. Tempo fa lo si chiamava *ancova with interaction*.

Ciascuno di questi modelli lineari descrivere in senso statistico il peso `weight` degli studenti, in un modo più o meno affidabile: è logico intuire che più parametri ci sono nel modello e maggiore sarà la capacità predittiva di quest'ultimo. Tuttavia, aggiungere parametri rappresenta un 'costo' che si cerca di evitare, valendo come paradigma il principio di parsimonia di Ockham: *frustra fit per plura quod fieri potest per pauciora*.



Si intuisce immediatamente il fatto che, aumentando il numero di parametri di un modello e calibrandoli sui dati raccolti nel nostro dataset di interesse andiamo incontro al rischio di **overfitting**: <https://en.wikipedia.org/wiki/Overfitting>

Abbiamo però già discusso il fatto che possiamo validamente misurare quanto sia 'disordinata' una nuvola di punti in raffronto al modello statistico che la descrive: si tratta del concetto espresso dal criterio di informazione di Akaike della sezione 4.1.5, che in R si può calcolare immediatamente con la funzione `AIC()`:

```
AIC(model0, model2, model1, modelplus, modelcross)
```

```
> AIC(model0, model1, model2, modelplus, modelcross)
      df      AIC
model0    2 481.4611
model2    3 437.2702
model1    3 430.9509
modelplus 4 421.2780
modelcross 5 422.6457
```

Siccome il libro l'ho scritto conoscendo già la soluzione del problema, mi sono permesso di riarrangiare l'ordine dei modelli in modo da evidenziare il comportamento 'convesso', 'parabolico', dei criteri AIC: vedete che aumentando il numero di parametri il disordine inizia a diminuire, ma poi ad un certo punto il 'costo' dei parametri (df, degrees of freedom) inizia a prevalere ed il criterio AIC inizia a risalire. Abbiamo quindi individuato un possibile candidato a quello che potrebbe essere il **modello minimale adeguato**, ossia il modello statistico che meglio descrive la nube di punti ma nel modo più parsimonioso possibile: modelplus.

#### 4.4 Anova: la generalizzazione del t test

Nel dataset `fresher` la variabile `gym` che classifica in modo sommario gli studenti in base all'attività fisica che dichiarano di svolgere durante la loro settimana-tipo durante il periodo delle lezioni: praticamente nulla, sporadica, o considerevole:

```
levels(gym)

> levels(gym)
[1] "not" "occasional" "sporty"
```

Sarebbe del tutto ragionevole ipotizzare che l'attività fisica possa influenzare il peso, e dunque vorremmo indagare innanzitutto sulla relazione:

```
relation3 = weight ~ gym
```

Osservate però che `gym` è un fattore a tre livelli, non a due; quindi non ha senso chiedere di eseguire un t test:

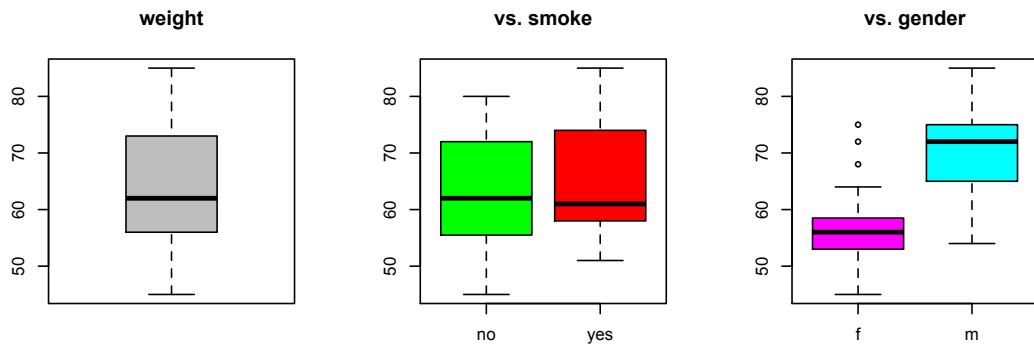
```
t.test(relation3)

> t.test(relation3)
Error in t.test.formula(relation3) :
  grouping factor must have exactly 2 levels
```

Infatti, non sarebbe proprio possibile da un punto di vista algebrico provare a modificare il calcolo del consuntivo che abbiamo introdotto nella sezione 3.2.1:

$$t = \frac{x_m - \mu}{s/\sqrt{n}}$$

A dire il vero non ci sarebbe problema per modificare la misura del 'rumore',  $s/\sqrt{n}$ , in cui compaiono solo radici e frazioni; ma modificare la misura del 'segnale',  $x_m - \mu$ , sarebbe un guaio. Tanto per fare un esempio sciocchissimo, pensando appunto di aver a che fare con tre medie, ad esempio 80 70 e 60, chi saprebbe dire quanto fa  $80 - 70 - 60$ ? Fa  $80 - 10 = 70$ , oppure fa  $10 - 60 = -50$ ? Ebbene, sarà forse una sciocchezza, ma questo è proprio il punto per il quale non possiamo fare un t test quando abbiamo più di due gruppi.



La soluzione invece risiede nell'osservare che quando le medie tra i gruppi differiscono, gli effetti si fanno sentire anche sulla dispersione dei dati (ed ecco il perché del nome *An.o.va.*, Analysis of variance). Per fare due esempi concreti, raffrontiamo il boxplot grigio di sinistra relativo ai dati del `weight`, suddividendolo nei due gruppi `smoke` (no oppure yes) e `gender` (f oppure m, la `relation2` che ben conosciamo). Ad occhio, vedete che nel caso `smoke` i box hanno pressappoco la medesima estensione di quello grigio? Ed invece il box rosa appare considerevolmente ridotto, e le mediane sono molto discoste – cosa che invece non accade nei boxplot verde e rosso, le mediane sono praticamente allineate a quella del box grigio. Ora, ricordiamo che i 'parametri' di un modello lineare coinvolgono le medie, e non le mediane; ma comunque la sostanza non cambia, come si evince dalla tabella seguente:

gruppo	media	dev. standard
tutti i 65 fresher	63.5	9.6
51 non fumatori	63.1	9.5
14 fumatori	65.1	10.3
32 femmine	56.6	6.4
33 maschi	70.2	7.1

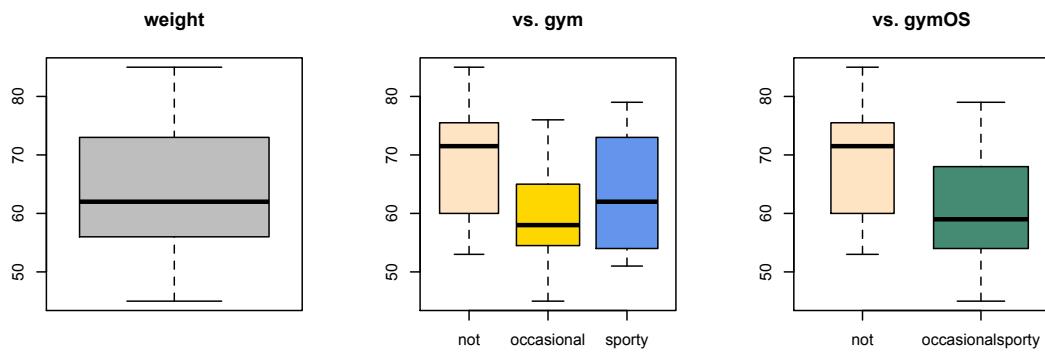
Proseguiamo dunque il ragionamento alla luce di quanto abbiamo imparato sinora: in un modello lineare vogliamo (usando possibilmente pochi parametri) ridurre il disordine, la dispersione dei dati. Stiamo investigando sulle due relazioni:

```
relation0 = weight ~ 1
relation3 = weight ~ gym
```

e vogliamo capire se la `relation3` sia 'migliore' della `relation0` in termini del criterio di informazione di Akaike, ossia che riduca il disordine dei dati senza richiedere troppi parametri. Graficamente, `relation0` riguarda i dati raffigurati nel boxplot grigio di sinistra, mentre `relation3` illustra il peso suddiviso nei gruppi `not` (colore biscotto), `occasional` (colore oro) e `sporty` (colore fiordaliso).

Notate che le larghezze dei box differiscono; si tratta di un modo molto comodo per rappresentare all'interno del boxplot anche la frequenza relativa degli elementi che compongono i vari gruppi. In R si utilizza l'opzione `varwidth = TRUE`; ad esempio, il pannello centrale è stato realizzato con la sintassi:

```
boxplot(weight ~ gym, main = "vs. gym", col = c("bisque", "gold",
```



```
"cornflowerblue"), varwidth = TRUE)
```

Vediamo dunque se si debba preferire la situazione di sinistra o quella centrale:

```
model3 = lm(relation3)
AIC(model0, model3)

> model3 = lm(relation3)
> AIC(model0, model3)
      df      AIC
model0 2 481.4611
model3 4 473.1096
```

Dunque il model3 ha un criterio di informazione minore del modello nullo, e quindi è opportuno tenere conto del fattore *gym*. 'Quando c'era il p-value', avremmo detto che 'la Anova è significativa'; e dunque avremmo tratto la conclusione che il peso dei nostri studenti è differente a seconda dell'attività fisica. Tuttavia questa appare essere solo una risposta parziale alla questione: se riguardiamo il pannello centrale, vediamo che sia il gruppo *occasional* che quello *sporty* hanno una mediana più bassa del gruppo *not*. Potremmo provare a raggrupparli assieme, come vediamo nel pannello di destra della Figura precedente, nel boxplot di colore acquamarina che abbiamo denominato *occasionalsporty*. Per fare questo dobbiamo creare una nuova variabile, che chiamiamo *gymOS*, con soli due livelli. In R di solito si utilizzano questi comandi:

```
gymOS = gym
levels(gymOS)
levels(gymOS) [2] = "occasionalsporty"
levels(gymOS) [3] = "occasionalsporty"
levels(gymOS)

> gymOS = gym
> levels(gymOS)
[1] "not"   "occasional" "sporty"
> levels(gymOS) [2] = "occasionalsporty"
> levels(gymOS) [3] = "occasionalsporty"
> levels(gymOS)
[1] "not"   "occasionalsporty"
```

Spieghiamo un istante i comandi. Inizialmente, 'fotocopiamo' la variabile `gym` in `gymOS`, e con il comando `levels(gymOS)` vediamo che non è cambiato nulla, i gruppi sono tre. Poi, attribuiamo al secondo ed al terzo livello del fattore il medesimo nome, `occasionalsporty`. Quindi di fatto il secondo ed il terzo livello divengono indistinguibili ed il comando `levels(gymOS)` mostra che ora abbiamo solo due gruppi. Ripetiamo la analisi dei criteri di informazione di Akaike:

```
relation3bis = weight ~ gymOS
model3bis = lm(relation3bis)
AIC(model0, model3bis, model3)

> relation3bis = weight ~ gymOS
> model3bis = lm(relation3bis)
> AIC(model0, model3bis, model3)
      df      AIC
model0    2 481.4611
model3bis 3 472.7146
model3    4 473.1096
```

Tutto chiaro? A quanto ci risulta, i `fresher` che fanno saltuaria o costante attività sportiva hanno un `weight` inferiore a quelli sedentari, ma non possiamo affermare invece che tra gli `occasional` ed i `sporty` il peso sia diverso in senso statistico. Siamo giunti ad una conclusione importante, che classicamente si otteneva mediante un processo di analisi statistica noto con gli appellativi di **multiple comparison**, o procedura dei **confronti multipli**, o anche **post-hoc analysis**. Si tratta di un argomento che ha comportato fiumi di inchiostro, e carteggi infiniti tra i referee dei paper alla ricerca della migliore 'correzione di Bonferroni', 'residui studentizzati in maniera onesta secondo Tukey', del 'false discovery rate secondo Benjamini' e chi più ne ha più ne metta. Intendiamoci, la questione sicuramente meriterebbe di essere conosciuta ed approfondita, e senza dubbio vi raccomando un approccio moderno e pragmatico come proposto nel testo di Bretz, Hothorn, e Westfall [8].

C'è ancora forse un dubbio da dipanare: abbiamo provato a riunire i livelli `occasional` e `sporty`, e ci è andata bene. Ma perché invece non abbiamo provato a riunire, ad esempio, `not` e `sporty`, lasciando da soli quelli del gruppo `occasional`? Ebbene, provatelo a fare per esercizio e vedrete che la cosa non funziona bene.

**Esercizio 4.6** Verificate che adottare il modello `model3ter` è veramente una pessima idea:

```
gymNS = gym
levels(gymNS)[1] = "notsporty"
levels(gymNS)[3] = "notsporty"
relation3ter = weight ~ gymNS
model3ter = lm(relation3ter)
AIC(model0, model3bis, model3, model3ter)
```

Prima di passare alla sezione più importante di tutto il libro, voglio ricordarvi che con la `relation3` ci siamo occupati di quella che tutti conoscono con il nome di **one-way anova**. Per capire come eseguire la **two-way anova** o la **two-way anova with interaction** in cui compaiono due predittori di tipo fattore:

```
relationtwowayplus = weight ~ gym + sport
relationtwowaycross = weight ~ gym * sport
```

dobbiamo oltrepassare il traguardo de 'La meta finale' fra cinque righe. Se invece eravate alla ricerca di capire come funzionasse la **nested factor anova**, allora dovete pazientare sino alla Parte III del libro; qui finora ci stiamo occupando esclusivamente di **crossed factors**: per capirci, `gym` e `gender` sono due fattori 'crossed' perché nel dataset `fresher` possono sussistere senza alcuna restrizione sia `f` che `m` che siano `not`, oppure `sporty`, oppure `occasional`.

#### 4.5 La meta finale: condurre un'analisi multivariabile

Abbiamo iniziato questo Capitolo 4 ponendoci una domanda centrale, relativamente al Problema 4.1, nel quale si voleva capire come poter affermare il fatto che gli isoenzimi LDH3 ed LDH1 abbiano una concreta capacità predittiva, supportata da una 'solida' argomentazione statistica, nel discriminare la presenza di una patologia benigna rispetto ad una patologia maligna nelle pazienti che presentano una massa uterina con una ben definita sintomatologia. Abbiamo qui volutamente usato l'aggettivo 'solido', che non possiede un significato particolare, invece del termine 'significativo': infatti vi avevamo annunciato sin dal Capitolo precedente che nella Figura 3.1 veniva rappresentata in modo paradigmatico sia la crisi in cui la letteratura scientifica oggi versa, essendosi fanaticamente appoggiata su un'interpretazione fideistica del simbolo ' $p < .05$ ' e di quel livello di fiducia arbitrariamente scelto come assoluto; ed al contempo veniva lì rappresentata la soluzione che avrebbe potuto superare quella crisi – il criterio di informazione di Akaike applicato al modello lineare.

Ci eravamo chiesti in particolare nella Figura 4.1 perché dovremmo preferire la spartana retta di regressione di colore nero in cui i pazienti non vengono distinti in base all'`Esito`, rispetto al modello Ancova con interazione (Sezione 4.2) di sinistra in cui due rette di regressione distinte 'spiegano' il comportamento di `Ldhtre` in funzione di `Ldhuno`. Adesso che conosciamo il modo di utilizzare la funzione `AIC()` non abbiamo alcun dubbio – `Ldhtre` ed `Ldhuno` si comportano in maniera diversa nelle pazienti con patologia maligna rispetto a quelle benigne:

```
uno = lm(Ldhtre ~ Ldhuno)
due = lm(Ldhtre ~ Ldhuno * Esito)
AIC(uno, due)

> AIC(uno, due)
      df      AIC
uno  3 7527.116
due  5 7220.747
```

Ora vediamo di annodare tutti i fili per dare un esempio completo di come condurre un'analisi multivariabile (tempo fa si diceva analisi multivariata) nell'individuare il modello minimale adeguato, sulla falsariga del dataset `fresher`. Lo faremo in cinque passi.

##### Step 1: ricercare il modello minimale adeguato di tipo additivo.

L'obiettivo ormai è chiaro: individuare quali siano i migliori predittori del `weight` degli studenti. Partiamo dal modello massimale, quello nel quale compaiono tutte le covariate del dataset (beh, non proprio tutte: manca la variabile `day`, ma ci rifiutiamo di credere che il giorno del compleanno di una persona influisca sul peso di quella persona):

```
maximalrel = weight ~ gender + height + shoesize + smoke + gym +
             + systolic + diastolic
maximalmodel = lm(maximalrel)
```

Ora ci possiamo affidare alla funzione di R denominata `step()`, la quale esegue una ricerca passo per passo ('stepwise search') di quale sia il modello minimale adeguato in termini del criterio di informazione di Akaike. L'output è decisamente corposo, ma noi andiamo immediatamente a leggere le righe finali:

```
step(maximalmodel)
...
Call:
lm(formula = weight ~ height + shoesize + gym )
...

```

Abbiamo individuato un primo candidato che soddisfa la nostra analisi multivariabile. Conserviamolo temporaneamente facendo un po' di copia-incolla dalla Console di R all'editor dei comandi:

```
formulaTemporanea01 = weight ~ height + shoesize + gym
modelloTemporaneo01 = lm(formulaTemporanea01)
```

### **Step 2: si controllano i fattori a più livelli.**

Questa fase corrisponde a quella che era la procedura dei *multiple comparison* della Anova (Sezione 4.4). In pratica confrontiamo il criterio di Akaike di `modelloTemporaneo01` con quello che si ottiene sostituendo `gym` con `gym2` (Sezione 4.4):

```
formulaTemporanea02 = weight ~ height + shoesize + gym0S + heartrate
modelloTemporaneo02 = lm(formulaTemporanea02)
AIC(modelloTemporaneo01, modelloTemporaneo02)

> formulaTemporanea02 = weight ~ height + shoesize + gym0S
> modelloTemporaneo02 = lm(formulaTemporanea02)
> AIC(modelloTemporaneo01, modelloTemporaneo02)
      df      AIC
modelloTemporaneo01 7 385.3987
modelloTemporaneo02 5 387.4298
```

Abbiamo un peggioramento del criterio di informazione nel `modelloTemporaneo02`, quindi tratteniamo per il momento il `modelloTemporaneo01`.

### **Step 3: ricercare se vi siano termini di interazione.**

Si inizia, con pazienza, a controllare se qualche coppia di predittori interagisca con la risposta:

```
formulaTemporanea03 = weight ~ height * shoesize + gym
formulaTemporanea04 = weight ~ height + shoesize * gym
formulaTemporanea05 = weight ~ height * gym + shoesize
modelloTemporaneo03 = lm(formulaTemporanea03)
modelloTemporaneo04 = lm(formulaTemporanea04)
modelloTemporaneo05 = lm(formulaTemporanea05)
AIC(modelloTemporaneo01, modelloTemporaneo03, modelloTemporaneo04,
     modelloTemporaneo05)
```

```

...
> AIC(modelloTemporaneo01, modelloTemporaneo03, modelloTemporaneo04,
modelloTemporaneo05)
      df      AIC
modelloTemporaneo01 7 385.3987
modelloTemporaneo03 7 388.2541
modelloTemporaneo04 8 390.9618
modelloTemporaneo05 8 389.8529

```

I termini di interazione evidentemente hanno un 'costo' sproporzionato al vantaggio che offrono nella riduzione del disordine. Quindi, continuiamo a ritenere valido il modelloTemporaneo01.

#### **Step 4: ricercare se vi siano termini di curvatura.**

Come abbiamo visto nella Sezione 4.1.7, 'lineare' non significa 'rettilineo'; proviamo con pazienza se vi sia qualche termine di curvatura per i predittori numerici:

```

formulaTemporanea06 = weight ~ height + shoesize + gym + I(height^2)
formulaTemporanea07 = weight ~ height + shoesize + gym + I(shoesize^2)
modelloTemporaneo06 = lm(formulaTemporanea06)
modelloTemporaneo07 = lm(formulaTemporanea07)
AIC(modelloTemporaneo01, modelloTemporaneo06, modelloTemporaneo07)

...
> AIC(modelloTemporaneo01, modelloTemporaneo06, modelloTemporaneo07)
      df      AIC
modelloTemporaneo01 7 385.3987
modelloTemporaneo06 7 388.2136
modelloTemporaneo07 7 388.3710

```

Nulla di fatto, continuiamo a trattenere il modelloTemporaneo01. Se per caso qualcuno di questi termini di curvatura fossero stati rilevanti, avrebbe avuto senso ritornare al passo 3 e verificare se anche essi interagissero con gli altri predittori, utilizzando nuovamente il simbolo \*.

#### **Step 5: stabilire se l'intercetta abbia rilevanza nel modello**

Talvolta, da un punto di vista biologico / medico / fisico ha senso ipotizzare che in presenza di valori nulli dei predittori anche la risposta valga zero. La notazione di Wilkinson e Rogers [62] consente di utilizzare equivalentemente la simbologia - 1 oppure + 0. Vediamo:

```

formulaTemporanea08 = weight ~ height + shoesize + gym - 1
modelloTemporaneo08 = lm(formulaTemporanea08)
AIC(modelloTemporaneo01, modelloTemporaneo08)

...
> AIC(modelloTemporaneo01, modelloTemporaneo08)
      df      AIC
modelloTemporaneo01 7 385.3987
modelloTemporaneo08 6 387.0734

```

No, nemmeno il modelloTemporaneo08 è migliore; quindi avevamo imboccato la soluzione più adatta sin dall'inizio al primo step. *Senza fallo!*, direbbe ridacchiando Gianni Morrone.

### Step 6: eseguire la diagnostica del modello

Vogliamo dunque verificare se le ipotesi matematiche del modello lineare sono verificate nel modelloTemporaneo01 per poterlo dichiarare il nostro modello minimale adeguato. Fatelo voi per esercizio, e discutiamo assieme il risultato.

**Esercizio 4.7** Esegui la diagnostica del modello minimale adeguato:

```
modminadeg = lm(weight ~ height + shoesize + gym)
par(mfrow = c(2,2))
plot(modminadeg)
```

■

Come vedete, nel primo disegno (Residuals vs. Fitted) la nube di punti è caotica ('residui identicamente distribuiti') e la linea rossa (lo 'smoother') è praticamente orizzontale ('assenza di drift') ed adagiata lungo lo zero ('media nulla dei residui'). Nel grafico quantile quantile, tutto sommato i residui appaiono plasmarsi lungo la diagonale tratteggiata: buon segno di normalità. Nel terzo plot, quantunque la presenza dei punti numero 15 e numero 45 e 50 effettivamente sembri influenzare leggermente l'omoschedasticità dei residui, non si notano comunque delle forme particolari nella nube dei punti: bene. Nel quarto ed ultimo plot (Residuals vs. Leverage) vedete che lo smoother rosso è praticamente orizzontale e nella nube di punti non appare alcuna iperbole rossa che segnali la presenza di punti dotati di forza di leva. Concludiamo perciò che questo modello lineare ci appare essere proprio un modello minimale adeguato.

#### 4.5.1 Interpretare il modello minimale adeguato

Ora che abbiamo individuato un modello minimale adeguato a descrivere il weight, commentiamolo ed interpretiamolo.

```
modminadeg = lm(weight ~ height + shoesize + gym)
summary(modminadeg)

...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -71.8185   12.6131  -5.694 3.96e-07 ***
height        0.3560    0.1378   2.583  0.01224 *  
shoesize      1.9135    0.4203   4.552 2.65e-05 ***
gymoccasional -5.0198   1.4679  -3.420  0.00113 ** 
gymsporthy    -7.1266   1.4977  -4.758 1.27e-05 *** 
...
Residual standard error: 4.51 on 60 degrees of freedom
...
```

Il primo commento da fare è che height, shoesize e gym si sono rivelati essere tre predittori sufficienti a modellare il comportamento di weight, in una maniera non correlata tra di essi (o meglio: i tre predittori non interagiscono tra loro nel predire il peso). Il secondo commento non riguarda i predittori presenti nel modello bensì le covariate che ne sono assenti: ad esempio, siccome gender non è presente nel modello, significa che da un punto di vista statistico non è importante distinguere esplicitamente tra femmine e maschi – in realtà la questione è più sottile: si potrebbe osservare che shoesize in effetti è una **variabile proxy** del fattore gender, essendo che le ragazze di solito hanno un piedino, mentre i ragazzi hanno di solito un piedone:

```
table(gender, shoesize > 40)

gender FALSE TRUE
  f    29     3
  m     0    33
```

Per interpretare il modello possiamo ragionare in questo modo: supponiamo di avere uno studente di statura 182 centimetri, che porta le scarpe con il numero di taglia 43 e che non è particolarmente sportivo. La stima teorica del suo peso diverrebbe dunque:

$$-71.8 + 0.356 \cdot 182 + 1.91 \cdot 43 = 75.3$$

Il fatto che l'errore standard residuale valga 4.5 significa per noi che sarebbe plausibile che, circa in una settantina di studenti su cento simili a questo, il peso osservato vari nel range  $75.3 \pm 4.5$ , ossia da 70.8 Kg a 79.8 Kg. Oppure, che circa in novantacinque soggetti su cento il peso osservato vari nel range  $75.3 \pm 2 \cdot 4.5$ , ossia da 66.3.8 Kg a 84.3 Kg (e se questo ragionamento vi sfugge dovete riprendere il discorso della Sezione 2.3.4 e dell'Esercizio 2.9).

## 4.6 Riassuntone del capitolo

Giunti a questo punto del libro, il professore professore Morrone si sarebbe affacciato allo stipite dell'uscio, appoggiandovi l'avambraccio ed al suo invito "*professore Borelli, purifichiamo?*" ci saremmo alzati dalle scrivanie ed usciti furtivamente sul retro dell'Ateneo per stargli accanto a respirare passivamente il fumo della sua sigaretta, che avrebbe aspirato avidamente e rapidamente. Ed io avrei fatto un rapido riepilogo, gesticolando, su cosa abbiamo visto sin qui.

Sicuramente, la statistica da sempre ha aiutato gli studiosi a gestire l'incertezza delle misure effettuate nei loro esperimenti o raccolte sui loro pazienti. Ed abbiamo capito che se il design dello studio è semplice e coinvolge poche variabili (il che non è un limite dello studio, anzi!), allora i **test statistici** possono essere lo strumento principe per una attendibile inferenza. Tuttavia, la risposta di un test è una misura di probabilità (il famigerato p-value) che non è collegabile a ciò che un ricercatore vorrebbe, ossia alla 'probabilità di avere ragione', o alla 'probabilità di avere scoperto qualcosa che sia maggiormente efficace rispetto a quello che sappiamo già'; men che meno ad una misura della 'forza', o ad una classifica della 'qualità' dei predittori su cui stiamo indagando.

Adottando invece l'approccio più moderno (moderno per modo di dire, perché comunque utilizziamo strumenti matematici tardo ottocenteschi, ma che si sono arricchiti certamente nella seconda metà del novecento) dei **modelli statistici**, nei quali una **risposta** viene posta in relazione con un esiguo numero di **predittori**, possiamo operare una cernita della qualità dei modelli comparando il disordine dei dati che essi riescono a modellare, tenendo conto del costo che questa operazione comporta, ossia, la quantità di 'termini' coinvolti nel modello. Lo strumento che permette di operare questa scelta non è una misura di probabilità ma è un numero, che non possiede una scala assoluta di valori, e che va sotto il nome di **Criterio di Informazione di Akaike**. C'è da aggiungere che il criterio AIC è solamente uno dei possibili criteri che possono venire utilizzati, ma di certo esso è il più noto nella comunità dei ricercatori.

Sin qui ci siamo riferiti ad un particolare tipo di modello statistico che va sotto il nome di **modello lineare**: esso è particolarmente adatto a modellare risposte di tipo numerico nelle quali la variabilità dei risultati possa venir interpretata come un processo casuale di tipo gaussiano. Nel Capitolo 5 ci occuperemo di risposte diverse da questa, come ad esempio il caso degli indici di rischio o dei biomarcatori clinici, nei quali si vuole predire una risposta binomiale (esempio tipico: in base a questi biomarcatori la nostra paziente ha una lesione benigna oppure maligna?)

Ma prima di concludere questo Capitolo vogliamo ribadire ciò che abbiamo citato nella Sezione 4.1.7, ossia che la parola 'lineare' non significa affatto 'dati dritti come una retta'. Ed in questo

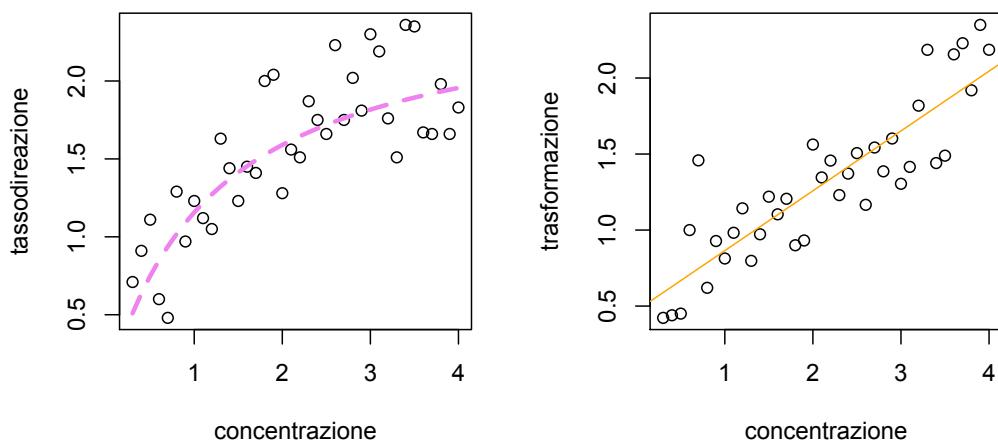
approfondimento di fine Capitolo discutiamo in dettaglio di questo fatto: si tratta di una mezza dozzina di paginette che in prima lettura potete evitare di leggere, e ritornarci sopra quando ne avrete bisogno in futuro.

## 4.7 Perle di saggezza: tecniche di linearizzazione

Vi sono dei fenomeni della Natura che possono venir modellati, *senza fallo*, avrebbe detto il nostro professore, per mezzo di equazioni matematiche di tipo non lineare. Questo significa che noi conosciamo una funzione matematica  $f$  con la quale la risposta  $y$  viene approssimativamente predetta da un dato  $x$ ; ma accade anche che, se consideriamo la somma di due dati  $x_1 + x_2$ , la risposta che otteniamo  $f(x_1 + x_2)$  non equivale affatto alla somma delle risposte  $f(x_1) + f(x_2)$ : è questo il concetto base della nonlinearità.

Tuttavia vi sono parecchi casi nei quali possiamo operare delle opportune trasformazioni algebriche che trasformano i dati e le relazioni che ne escono riformulate si comportano in modo lineare. Vediamo alcuni casi particolari piuttosto diffusi.

### 4.7.1 Il modello iperbolico.



Un esempio biochimico di legge iperbolica è dato dalla *cinetica di Michaelis-Menten*[29], la quale descrive l'andamento della velocità di una reazione catalizzata da enzimi. In tal caso la risposta  $y$  (il tasso di reazione enzimatica) dipende dalla concentrazione  $x$ , tramite la legge:

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

Il nostro problema è riuscire a scoprire quali siano i valori dei coefficienti  $A$  e  $B$  che caratterizzano la curva viola tratteggiata nel pannello di sinistra.

 Date un'occhiata a Wikipedia, all'indirizzo [https://en.wikipedia.org/wiki/Michaelis-Menten\\_kinetics](https://en.wikipedia.org/wiki/Michaelis-Menten_kinetics) e vedrete che la figura ha un tipico andamento iperbolico.

Supponiamo dunque che siano dati la concentrazione  $x$  del substrato ed il tassodireazione  $y$ :

```
concentrazione = seq(from = 0.3, to = 4.0, by = 0.1)
```

```
tassodireazione = 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 del predittore con la risposta è affidata al rapporto delle variabili,  $x/y$ :

```
trasformazione = concentrazione / tassodireazione
formulalineare = trasformazione ~ concentrazione
modello = lm (formulalineare)
modello

...
(Intercept)  concentrazione
0.4708          0.3936
```

La retta di regressione arancione del pannello di destra ha equazione  $y = 0.47 + 0.39 \cdot x$ . Come si collegano ora questi coefficienti  $a = 0.47$  e  $b = 0.39$  con i parametri incogniti  $A$  e  $B$  delle relazione iperbolica? Con un paio di passaggi algebrici si può vedere che:

```
A = 1 / 0.3936
B = 0.4708 / 0.3936
```

ossia  $A = 2.54$  e  $B = 1.20$ . Dunque l'iperbole di regressione viola tratteggiata ha equazione  $y = \frac{2.54 \cdot x}{1.20+x}$  ed i due grafici della pagina precedente si realizzano con i seguenti comandi:

```
iperbole = (A * concentrazione) / (B + concentrazione)
par(mfrow = c(1,2))
plot(concentrazione , tassodireazione)
lines(concentrazione, iperbole, col = "violet", lty = 2, lwd = 3)
plot(concentrazione, trasformazione)
abline(modello, col = "orange")
```

#### 4.7.2 Il modello esponenziale.

Il Tecnezio 99m è un tracciante 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 conoscere questa serie di dati, raffigurata nel pannello di sinistra della prossima pagina:

```
x = seq(from = 1, to = 24, by = 1)
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)
```

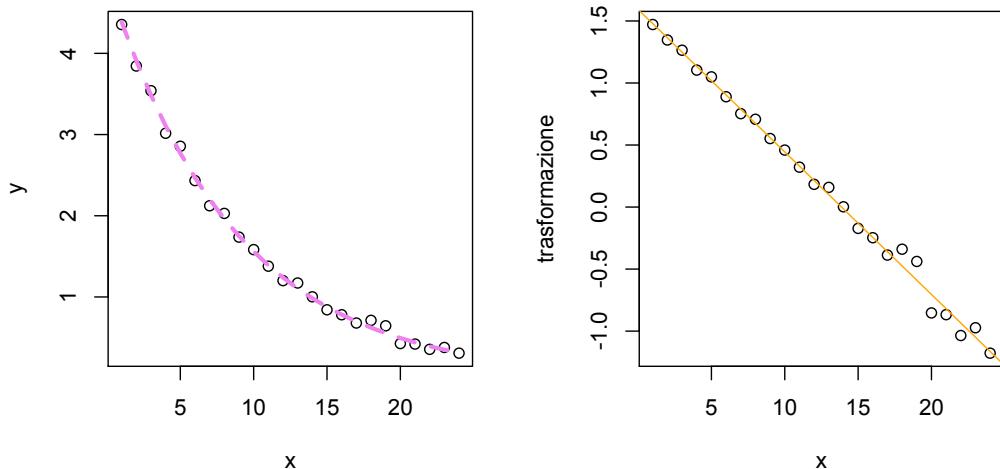
Considerando il logaritmo della risposta  $y$  ci riconduciamo ad un problema lineare:

```
trasformazione = log(y)
modello = lm (trasformazione ~ x)
modello
( A = exp(modello$coefficients[[1]]) )
```

```
( B = modello$coefficients[[2]] )
esponenziale = A * exp( B * x)
par(mfrow = c(1,2))
plot(x,y)
lines(x, esponenziale, col = "violet", lty = 2, lwd = 3)
plot(x, trasformazione)
abline(modello, col = "orange")

...
(Intercept)           x
1.5932        -0.1149
...
```

Dai parametri  $a$  e  $b$  della retta di regressione del modello arancione abbiamo potuto ricalcolare i parametri incogniti  $A = \exp(a) = \exp(1.59) = 4.92$  e  $B = b = -0.11$  dell'esponenziale viola.



### 4.7.3 Il modello maxima function.

Il dataset `analgesia` che avete incontrato negli Esercizi del primo Capitolo in realtà non è di tipo cross-section, ma è stato tratto da uno studio longitudinale, in cui il dolore dei pazienti veniva valutato in 7 occasioni nell'arco di 72 ore dalla fine dell'intervento odontoiatrico, secondo una scala soggettiva da 1 a 10 (con tutti i limiti metodologici che questo comporta). Supponiamo di considerare la *curva del dolore* di una certa paziente trattata con il farmaco di tipo ketorolac:

```
ore = c(1, 6, 12, 24, 36, 48, 72)
dolore = c(1, 7, 8, 4, 2, 1, 1)
```

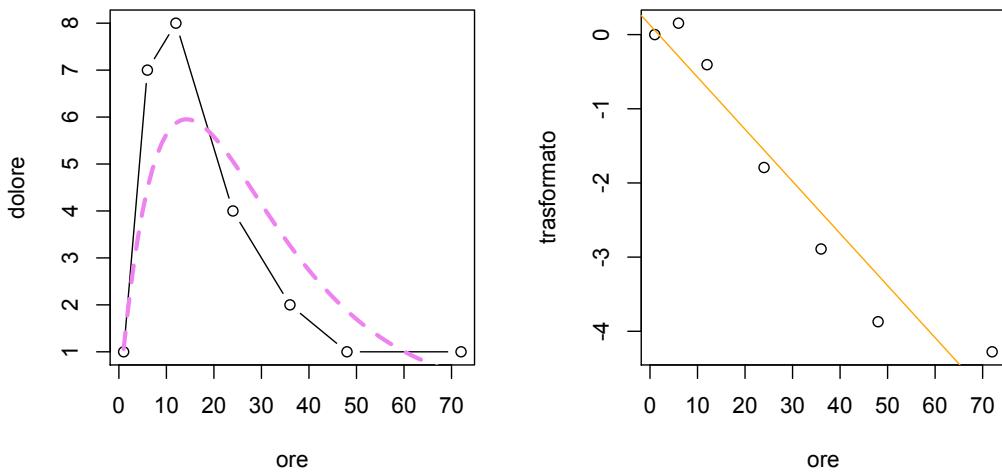
Questa curva può venir modellata con una legge nonlineare del tipo  $y = A \cdot x \cdot \exp(B \cdot x)$ . Il cambio di variabile da operare sulla  $y$  per rendere il problema lineare è del tipo  $\log(y/x)$ :

```
trasformato = log(dolore / ore)
modello = lm (trasformato ~ ore)
modello
```

```
( A = exp(modello$coefficients[[1]]) )
( B = modello$coefficients[[2]] )
tempo = 1:72
maxima = A * tempo * exp( B * tempo)
par(mfrow = c(1,2))
plot(ore, dolore, "b")
lines(tempo, maxima, col = "violet", lty = 2, lwd = 3)
plot(ore, trasformato)
abline(modello, col = "orange")

...
(Intercept)          ore
 0.12849      -0.07026
...
```

I parametri  $A = 1.14$  e  $B = -0.07$  che rappresentano la curva viola di sinistra stavolta dipendono da  $a = 0.13$  e  $b = -0.07$  per mezzo delle relazioni  $A = \exp(a)$  e  $B = b$ . Non sembra molto adeguato, vero? Purtroppo in questo tipo di andamenti due parametri non sono sufficienti a fare le cose per bene, ce ne potrebbero occorrere tre, o addirittura quattro: vediamo come si procede in tal caso nei prossimi esempi.



#### 4.7.4 Il modello potenza.

Nei primi tre casi che abbiamo trattato le cose tutto sommato sono risultate abbastanza routinarie: si tratta di stimare due parametri ignoti  $A$  e  $B$  di una certa legge matematica (viola) e lo si fa trasformandola in una retta (arancione) di cui è banale calcolare l'intercetta  $a$  e la pendenza  $b$  mediante il comando `lm` del modello lineare. Nei prossimi esempi le cose si complicano, perché cercheremo di stimare tre parametri ignoti. Cominciamo con la cosiddetta *legge di von Bertalanffy* del tipo  $y = K + A \cdot x^B$ , che potrebbe essere impiegata per modellare le fasi iniziali della crescita cellulare.



Per maggiori ragguagli: [http://en.wikipedia.org/wiki/Ludwig\\_von\\_Bertalanffy](http://en.wikipedia.org/wiki/Ludwig_von_Bertalanffy)

Prendiamo una serie di dati  $y$ , raccolti ad unità di tempo  $x$  arbitrarie:

```
x = 1:7
y = c(1.2, 1.3, 1.7, 2.8, 5.5, 10.9, 20.2)
```

Non possiamo appunto sperare di ottenere le stime dei tre parametri incogniti ( $K, A$  e  $B$ ) che definiscono la curva  $y = K + A \cdot x^B$  da una retta di regressione che possiede solo due parametri, la pendenza  $a$  e l'intercetta  $b$ . Possiamo fare qualche tentativo, iniziando a stimare il valore di  $K$ : siccome per  $x = 1$  l'equazione diventa  $y = K + A \cdot 1^B = K + A$ , questo significa che si dovrebbe avere  $K + A = 1.2$ . 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 logaritmici, su  $x$  e su  $y$ :

```
stimaK = 1.1
ytrasforma = log(y - stimaK)
xtrasforma = log(x)
modello = lm (ytrasforma ~ xtrasforma)
modello

...
(Intercept)  xtrasforma
-2.989        2.785
...
```

Per ritrovare i coefficienti della relazione, operiamo le trasformazioni inverse;  $A = \exp(-2.989) = 0.050$  e  $B = b = 2.785$ :

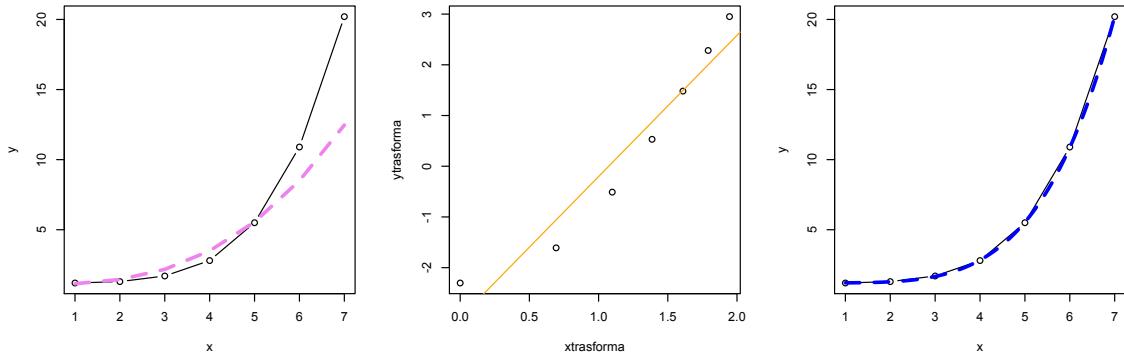
```
(A = exp(modello$coefficients[[1]]))
(B = modello$coefficients[[2]])
```

Ma il grafico viola che otteniamo a sinistra nella prossima pagina non è affatto dei migliori: esso risente della scelta arbitraria che abbiamo fatto stimando  $K$ . Come uscirne? Potremmo ritentare l'analisi con altri valori di  $K$  leggermente più grandi o leggermente più piccoli e vedere se vi sono dei miglioramenti. Ma una strategia certamente molto più efficace è quella di affidare ad R la ricerca dei migliori valori di  $K, A$  e  $B$  alla funzione `nls`, ben illustrata nel libro di Ritz e Streibig [45]. Non ci addentriamo qui nei dettagli ben esposti nel testo; diciamo semplicemente si tratta di un metodo approssimato, o come si dice anche **iterativo** o **ricorsivo**, in cui si sfruttano le stime ai minimi quadrati di Boscovich / Gauss-Markov in seno alla relazione nonlineare (`nls` è l'acronimo di nonlinear least squares). Per far partire l'algoritmo iterativo possiamo utilizzare la nostra stima di  $K = 1.1$  ed i valori corrispondenti di  $A = 0.50$  e  $B = 2.785$  che abbiamo utilmente calcolato poco fa:

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

```
...
Parameters:
Estimate Std. Error t value Pr(>|t|)
kappa 1.2090780 0.0230875 52.37 7.96e-07 ***
A 0.0036473 0.0001665 21.91 2.57e-05 ***
```

```
B      4.3981333  0.0234541  187.52 4.85e-09 ***
...
Residual standard error: 0.03797 on 4 degrees of freedom
Number of iterations to convergence: 12
Achieved convergence tolerance: 7.464e-07
```



L'algoritmo dopo 12 iterazioni converge ad una soluzione approssimata che soddisfa l'errore che di default viene prefissato nell'ordine di  $10^{-6}$  e le stime desiderate dei parametri sono:  $K = 1.2091$ ,  $A = 0.00365$  e  $B = 4.398$ . Si noti che disponendo anche del loro errore standard potremmo essere in grado di valutarne anche la loro affidabilità. Ma ad occhio, guardando il grafico blu di destra, siamo giunti ad una situazione più che soddisfacente:

```
par(mfrow = c(1,3))
plot(x, y, "b")
lines(x, potenza, col = "violet", lty = 2, lwd = 3)
plot(xtrasforma, ytrasforma)
abline(modello, col = "orange")
plot(x, y, "b")
sequenza = seq(from = 1, to = 7, by = .1)
stimanls = 1.209 + 0.00365 * sequenza ^ 4.398
lines(sequenza, stimanls, col = "blue", lty = 2, lwd = 3)
```

#### 4.7.5 Il modello logistico.

Il modello di crescita delle popolazioni proposto nel 1838 da Pierre-Francois Verhulst si può rappresentare nella forma  $y = \frac{K}{1+A\exp(B\cdot x)}$ . Anche in questo caso i parametri sono tre, e la nonlinearità obbliga a fornire una stima di  $K$  da cui partire; siccome di solito il coefficiente  $B$  è negativo, per  $x$  'molto grande' i valori di  $y$  tendono a  $K$  (i matematici sono espertissimi nel calcolare i limiti per  $x$  che tende ad infinito). Prendiamo in esame i dati di una mia carissima amica, anche lei allieva del professor Morrone, la dottoressa Teresa Calimeri, relativamente ad uno studio concernente certi *scaffold* adatti a studiare lo sviluppo del mieloma multiplo [11]. Importiamo i dati che sono presenti nel dataset `mieloma.csv`:

```
www = "http://www.biostatisticaumg.it/dataset/mieloma.csv"
mieloma = read.csv(www, header = TRUE)
attach(mieloma)
```

Con il comando `str(mieloma)` potete controllare che vi sono 110 osservazioni di 4 variabili (`patient`, `time`, `chain` e `paraprotein`); in effetti vi sono 9 patient contrassegnati con le sigle p1, p2, ..., p9 (lo si vede dalla `table(patient)`), mentre la variabile risposta di interesse di Teresa è `paraprotein`. Le osservazioni sono state condotte talvolta in duplicato e talvolta in triplicato (infatti con `table(time)` compaiono cinque occasioni temporali, rispettivamente a 0, 25, 45, 60 ed 80 minuti). Per fare un esempio illustrativo concentriamoci sulla prima curva del primo paziente:

```
x = time[1:5]
y = paraprotein[1:5]
y[1] = 1
```

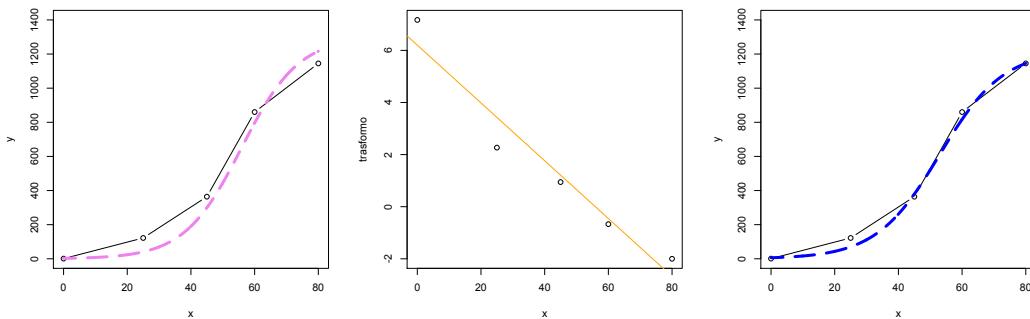
Spieghiamo il senso di quell'istruzione `y[1] = 1`; siccome anche qui avremo a che fare con la funzione logaritmo per trasformare la `y`, e non possiamo calcolare il logaritmo di zero, facciamo un lieve cambiamento ponendo uguale ad 1 il valore iniziale di ciascuna paraproteina (che sarebbe stata zero). Il grafico di sinistra ci mostra che, grossomodo, al tempo 55 la curva presenta un punto di flesso che vale approssimativamente 650. Dunque, per simmetria, è abbastanza ragionevole ipotizzare che il *plateau* della curva, ossia l'asintoto orizzontale, si aggiri attorno al valore di 1300:

```
stimaK = 1300
```

Adesso operiamo le consuete manipolazioni algebriche:

```
trasformo = log(- 1 + stimaK/y)
modello = lm (trasformo ~ x)
modello

...
(Intercept)           x
6.2016            -0.1109
```



Come vediamo nel pannello centrale della figura, la retta di regressione arancione descrive ragionevolmente bene la nube di punti. Come in precedenza, calcoliamo le stime dei parametri  $A$  e  $B$  (come al solito,  $\exp(6.202) = 494$  e  $-0.111$ ), invertendo le relazioni algebriche:

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

Tuttavia le stime ottenute per  $A$ ,  $B$  e  $K$  danno alla curva logistica, detta anche **sigmoide**, di colore viola tratteggiato: ad occhio, non è un granché. Ma allora possiamo mettere in moto il metodo `nls`:

```

curva = nls(y ~ kappa / ( 1 + A * exp(B * x) ) ,
            start = list(kappa = 500, A = 2.52, B = -0.43), trace = TRUE)
summary(curva)

...
      Estimate Std. Error t value Pr(>|t|)
kappa 1222.17708   75.12507 16.269  0.00376 ***
A       199.76981  153.84950   1.298  0.32368
B      -0.10133    0.01668  -6.075  0.02604 *
...

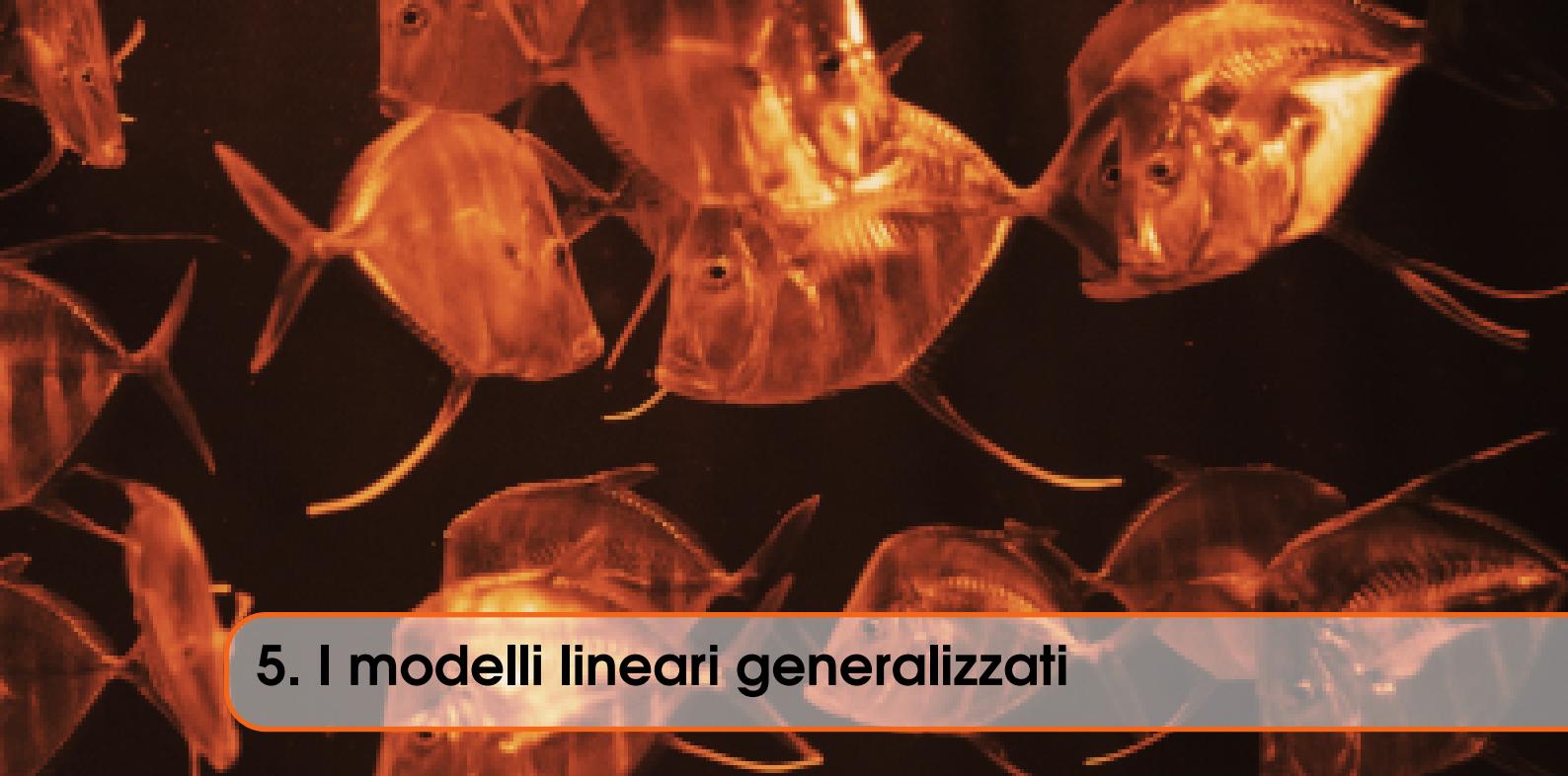
```

Dopo poche iterazioni (nel nostro esempio, 6) otteniamo delle stime più soddisfacenti, come vedete nella curva blu tratteggiata di sinistra. C'è però ancora un piccolo dettaglio di cui tenere conto: osservate che la stima di  $A = 199.8$  reca con sé un'incertezza conspicua, misurata dallo standard error 153.8. Questo capita molto spesso se si ha a che fare con dati provenienti da scale di misura molto diverse: un accorgimento da tenere sempre a mente è quello di pre-processare i dati per mezzo della funzione `scale()` di R, la quale opera una standardizzazione dei dati numerici, come avevamo anticipato nella sezione 2.3.4 (ed in particolare nel Quesito 2.3.4).

## 4.8 Esercizi ed attività di approfondimento

■ **Attività 4.1 — retta di regressione.** Una delle prime cose che si impara a proposito delle retta di regressione è che essa "passa" per il baricentro della nuvola di punti. Per baricentro, o centro di massa, in statistica si intende proprio il punto che ha per coordinate la media dei dati delle ascisse, e la media dei dati delle ordinate. Per esercizio, provate a realizzare con R il grafico di sinistra della Figura 4.1, facendo apparire un bel punto blu a forma di diamante che indichi il baricentro, e verificate che la retta di regressione lo attraversa. ■

■ **Attività 4.2 — Ancova.** Leggete il paper di Roberta Venturella [57] dall'indirizzo <http://www.biostatisticaumg.it/biostatistica/jmig.pdf> e verificate quanto viene affermato usando i dati originali del dataset `salpinge` pubblicati all'indirizzo: <http://www.biostatisticaumg.it/dataset/salpinge.csv> ■



## 5. I modelli lineari generalizzati

Sono passate ormai molte pagine da quando nella Sezione 2.3.1 abbiamo adottato come esempio guida il dataset `raul` (risk assessment in uterine lesions) di Annalisa Di Cello [19]:

```
www = "http://www.biostatisticaumg.it/dataset/raul.csv"
raul = read.csv(www, header = TRUE)
attach(raul)
names(raul)

...
[1] "Anni"      "Ca125"     "Ldhuno"    "Ldhtre"    "Esito"
[6] "Diagnosi"
```

Ed abbiamo anche detto nella Sezione 3.1 che lo scopo della ricerca era quello di escogitare possibili metodi predittivi del sarcoma uterino, stante il fatto che nel 2014 la Food and Drug Administration aveva emesso un *warning* che scoraggiava l'utilizzo della morcellazione laparoscopica nel trattamento chirurgico dei fibromi, giacché non erano noti metodi affidabili per predire la malignità delle masse uterine in fase preoperatoria. Vi ricorderete anche che nella Sezione 3.1 avevamo realizzato qualche grafico esplorativo per capire se la glicoproteina Ca125 potesse considerarsi un accurato predittore dell'*Esito* (benigno o maligno) della lesione uterina. E da lì ci eravamo accorti che sebbene il Ca125 "fosse significativo" non ci sarebbe stato di alcuna utilità nella pratica clinica.

Ora noi vogliamo estendere il discorso che abbiamo iniziato nel Capitolo precedente, riguardo ai modelli lineari; in quel frangente la risposta era di tipo numerico, ed un modello che ammettesse una distribuzione gaussiana per 'spiegare' la varibilità dei residui era ottimale. Ma in questo caso la risposta desiderata riguarda l'*Esito*, benigno o maligno: dobbiamo quindi cercare dei modelli che si adattino ad una distribuzione aleatoria binomiale, come quella di cui abbiamo discusso nella Sezione 2.3.1, e che ci consentano di discutere la validità di una relazione come questa:

```
indicedirischio1 = Esito ~ Anni + Ca125 + Ldhuno + Ldhtre
```

## 5.1 I dettagli da conoscere

Non sto qui a raccontarvi dell'emozione di quando nell'ottobre del 2007 potei assistere ad un ciclo di lezioni tenuto proprio a Trieste dal professor Peter McCullagh. L'irlandese Peter McCullagh è, appunto, coautore con il britannico John Ashworth Nelder del testo *Generalized Linear Models*, pubblicato nel 1983; e John Nelder è stato colui il quale assieme allo scozzese Robert William MacLagan Wedderburn ideò e pubblicò nel 1972 sul prestigiosissimo Journal of the Royal Statistical Society il concetto di modello lineare generalizzato [41]. Di cosa stiamo parlando? Parliamo di un concetto matematico non banale, anzi.

**Vocabolario 5.1 — modello lineare generalizzato.** Un modello lineare generalizzato è un insieme di tre strumenti matematici:

1. una **relazione**, che usualmente viene denominata il **predittore lineare**, la quale connette la risposta del dataset con una o più covariate. Ad esempio, il nostro `indicedirischio1`.
2. una famiglia di **variabili aleatorie** adatte a modellare la risposta (o, per essere giustamente più precisi, i residui del predittore lineare). Ad esempio, la distribuzione binomiale per l'Esito.
3. una **funzione di collegamento** che trasforma ('inietta') il valore atteso della variabile aleatoria che modella la risposta nel valore medio del predittore lineare. E qui devo fare qualche parola di approfondimento perché nel Capitolo precedente relativo ai modelli lineari questo argomento era stato sottaciuto, in quanto 'invisibile'.

### 5.1.1 La funzione di collegamento

Nel quesito 3.2.2 ci chiedevamo se il Ca125 fosse un biomarcatore efficace nella predizione dell'Esito. Utilizzando il linguaggio dei modelli statistici, vogliamo occuparci del seguente predittore lineare:

```
indicedirischio2 = Esito ~ Ca125
```

Ma il ragionamento immediato è: Esito è una variabile aleatoria bernoulliana, con probabilità stimata 1568/1610 di insuccesso e  $p = 42/1610$  di successo. Mentre invece Ca125 non rappresenta una probabilità, perché i suoi valori numerici non sono compresi tra zero ed uno. Ma ci viene in soccorso la nozione di *unità logistica*, o **logit**:  $\text{logit}(p) = \log(p/(1-p))$ . Avendo una lavagna a disposizione, potremmo mostrarvi che avendo una probabilità di successo  $0 < p < 1$ , e considerando il logaritmo del rapporto delle quote di probabilità  $\log(p/(1-p))$  (in inglese, il logaritmo dell'**odds ratio**), potremmo collegarci al predittore lineare  $y = a + b \cdot x$ :

$\text{logit}(p) = a + b x$	$p = (1-p) \cdot \exp(a + b x)$	$p \cdot (1 + \exp(a + b x))^{-1} = \exp(a + b x)$
$\log\left(\frac{p}{1-p}\right) = a + b x$	$p = \exp(a + b x) - p \cdot \exp(a + b x)$	$p = \frac{\exp(a + b x)}{1 + \exp(a + b x)}$
$\frac{p}{1-p} = \exp(a + b x)$	$p + p \cdot \exp(a + b x) = \exp(a + b x)$	

Il succo di questa lavagna è il seguente: conoscendo il valore  $x$  (il Ca125 della nostra paziente) valutiamo il predittore lineare  $a + b \cdot x$  e lo sostituiamo nella funzione inversa del logit, che chiamiamo **funzione logistica**, o anche **sigmoide** (ed è proprio un caso particolare di quella funzione che Teresa Calimeri ci ha fatto incontrare nella sezione 4.7.5):

$$p = \frac{\exp(a + b \cdot x)}{1 + \exp(a + b \cdot x)}$$

**Esercizio 5.1** Disegnate una sigmoide. Vedete che la ascissa  $y$  (e sottolineo ascissa, ossia, l'asse orizzontale) è una variabile numerica continua, mentre **sigmoide** diventa un valore di probabilità, sempre compreso tra 0 ed 1 (sull'asse verticale delle ordinate):

```
y = seq(-6, 6, 0.1)
p = exp(y) / (1 + exp(y))
plot(y, p)
```

■

### 5.1.2 Ad ogni variabile aleatoria, la sua funzione di link

Perché mai allora quando abbiamo introdotto il modello lineare ed il comando `lm` questa funzione di collegamento non era apparsa? Ebbene, perché per definizione i residui del modello lineare avevano una media nulla e la loro deviazione standard era proprio la dispersione della variabile aleatoria gaussiana che si andava a cercare. Quindi, di fatto, nel caso `lm` la funzione di collegamento sarebbe stata la funzione identità, che ad ogni valore  $y$  associa il valore  $y$  stesso: quindi una specie di fantasma invisibile.

Potrebbero esserci anche altri casi. Per esempio, quando conduciamo esperimenti in cui dobbiamo contare cellule, la distribuzione di Poisson potrebbe fare al caso nostro. In tal caso, la funzione di collegamento adatta è la funzione logaritmo, la cui inversa (che 'inietta' il predittore lineare nella risposta di tipo *count*) è ovviamente la funzione esponenziale. Ecco un riassuntino della situazione:

modello	variabile aleatoria	(inversa della) funzione di link
lineare	<code>family = gaussian</code>	identità, $v = u$
regressione logistica	<code>family = binomial</code>	sigmoide, $v = \exp(u)/(1 + \exp(u))$
regressione di Poisson	<code>family = poisson</code>	esponenziale, $v = \exp(u)$

### 5.1.3 Interpretare una regressione logistica

#### il caso di un fattore

Facciamo un esempio didattico per capirci meglio. Abbiamo capito che se una paziente ha una massa uterina maligna, i valori dei biomarcatori saranno particolarmente elevati. Dividiamo perciò le nostre pazienti in due gruppi; quelle che non superano il valore mediano del Ca125 e quelle che invece lo superano. Lo ribadisco: scegliere la mediana come cut-off non ha alcuna giustificazione clinica, lo facciamo solamente per creare una tabella per esercizio.

```
Ca125High = Ca125 > median(Ca125)
table(Esito, Ca125High)
```

...

<i>Ca125High</i>		
Esito	FALSE	TRUE
<i>benigno</i>	818	750
<i>maligno</i>	4	38

Per definizione, l'**odds ratio** di una tavola di contingenza si calcola 'in diagonale':  $(818 \cdot 38) / (750 \cdot 4) \approx 10.4$ . Sappiamo che l'odds ratio è una misura di associazione tra due eventi dicotomici: quanto più esso è 'lontano' da 1 (cioè vicino allo zero, o grande verso l'infinito) tanto più gli eventi sono associati tra loro.



Altri dettagli utili da conoscere: [https://en.wikipedia.org/wiki/Odds\\_ratio](https://en.wikipedia.org/wiki/Odds_ratio)

Calcoliamo infine il log - odds ratio:

```
log((818 * 38) / ( 750 * 4))
```

...

```
[1] 2.338081
```

Adesso cerchiamo di capire cosa rappresentano i coefficienti che otteniamo nel `summary` di un modello lineare generalizzato binomiale e di metterli in relazione con le informazioni riportate nella tavola di contingenza che abbiamo appena calcolato.

Esito vs.			
Ca125High	FALSE	TRUE	
Benigno	818	750	
Maligno	4	38	

```
indicedirischio3 = Esito ~ Ca125High
logistico3 = glm(indicedirischio3, family = binomial)
summary(logistico3)
```

...

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-5.3206	0.5012	-10.615	< 2e-16 ***
Ca125HighTRUE	2.3381	0.5281	4.427	9.53e-06 ***

Iniziamo ora ad osservare che il coefficiente  $b = 2.3381$  del predittore lineare del modello lineare generalizzato è esattamente uguale al valore che abbiamo calcolato con il log - odds ratio. Fissiamo ora la nostra attenzione sul gruppo delle donne della colonna FALSE, ossia quelle che hanno il Ca125 al di sotto della mediana. Ci sono 4 pazienti con patologia maligna, dunque la probabilità di successo  $p$  in quel gruppo vale:

```
> 4 / (818 + 4)
[1] 0.00486618
```

Se adesso calcoliamo il *logit* di questa probabilità otteniamo il coefficiente  $a$  del predittore lineare, l'intercetta del modello:

```
> log(0.00486618/(1-0.00486618))
[1] -5.320568
```

Equivalentemente, se consideriamo la funzione sigmoide, ritroviamo il valore di probabilità del gruppo di pazienti della colonna FALSE:

```
> exp(-5.3206) / (1 + exp(-5.3206))
[1] 0.004866025
```

Abbiamo quindi mostrato che i coefficienti  $a$  e  $b$  del predittore lineare nella regressione logistica sono immediatamente riconducibili ai valori di probabilità che emergono dalla tavola di contingenza; in particolare  $a$  riguarda la probabilità del gruppo a basso rischio, mentre  $\exp(b)$  è l'odds ratio della tabella. Se poi ricordate quanto abbiamo annunciato nella sezione 4.1.8, in cui spiegavamo come si interpretano i coefficienti di un modello lineare quando si esegue il t test tra due gruppi, allora vi sarà immediato capire perché  $a + b \cdot 1 = -5.3206 + 2.3381 = -2.9825$  rappresenti il logit della probabilità di avere una patologia maligna nel gruppo di pazienti della colonna TRUE, ossia quelle che hanno il Ca125 al di sopra della mediana:

```
> exp(-2.9825) / (1 + exp(-2.9825))
[1] 0.04822276
> 38/(750+38)
[1] 0.04822335
```

### il caso di una variabile numerica

Ora riprendiamo il caso dell'indicedirischio2 in cui ci illudiamo di predire l'Esito della massa uterina in funzione del Ca125:

```
indicedirischio2 = Esito ~ Ca125
modello = glm(indicedirischio2, family = binomial)
summary(modello)

...
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.131762   0.209732 -19.700 < 2e-16 ***
Ca125        0.013832   0.002837   4.876 1.08e-06 ***
...
```

Siccome il biomarcatore Ca125 assume tutta una possibile gamma di valori numerici, non possiamo creare come prima una tabella di contingenza con 'infinite' colonne; allora ovviamente interpretiamo i coefficienti  $a$  e  $b$  alla luce di quanto abbiamo detto a proposito della sigmoide. Per esempio, ipotizziamo che una nuova paziente si presenti alla nostra attenzione con un valore di Ca125 pari a 132. Calcoliamo l'indice di rischio (il predittore lineare):

```
> -4.132 + 0.014 * 132
[1] -2.284
```

Trasformiamo questo valore numerico -2.284 in una stima della probabilità  $p$  di avere una neoplasia maligna, utilizzando la sigmoide, ossia l'inversa della funzione di collegamento:

```
> exp(-2.284) / (1+exp(-2.284))
[1] 0.09245677
```

In base a questo modello logistico, vorremmo poter affermare che la probabilità  $p$  di una massa uterina maligna sia attorno al 10 per cento. In realtà non siamo affatto sicuri di essere in presenza di un modello predittivo che sia adeguato. Ritorneremo fra poco su questo argomento, ma appare piuttosto strano che una paziente con un valore piuttosto elevato di Ca125 abbia una probabilità piuttosto bassa di malignità. Per esercizio, provate a disegnare la curva logistica che emerge dal modello e convincetevi che le cose non vanno affatto bene (nonostante l'acclarata 'significatività').

**Esercizio 5.2** Commentate l'andamento di questa sigmoide.

```
plot(Ca125, (as.numeric(Esito)-1))
xx = seq(0,300)
pl = -4.132 + 0.014 * xx
yy = exp(pl)/(1+exp(pl))
lines(xx,yy, col = "orange")
```

**Esercizio 5.3** Andate a leggere l'articolo di Richard Moore in cui si definisce l'indice R.O.M.A. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3594101/> ed in particolare cercate la sezione 'Predictive Probability Calculations'. Guardate la formula relativa alla 'Predicted Probability (PP)' e scoprirete l'errore di stampa.

#### 5.1.4 Problemi con lo standard error

Allora, vi sono piaciute le fotografie in apertura di capitolo? Debora De Bartolo oltre ad essere un'eminente artista della fotografia, che pubblica le sue foto su Vogue, è un'anatomopatologa e durante gli studi di dottorato alla UMG si è occupata del dataset tossicologia; si tratta di un dataset di 185 osservazioni di 8 variabili, in formato .csv (il carattere separatore è infatti una virgola): ID è un contatore di identificazione delle persone decedute, di cui conosciamo l'anno del decesso, il loro genere (ossia f e m, una variabile fattore a due livelli), la loro eta al momento del decesso ed una generica descrizione della causa di morte. Inoltre si conoscono gli esiti degli esami tossicologico ed alcolemia (entrambe variabili fattore a due livelli, negativo e positivo; di quest'ultima, il livello positivo si ha quando la variabile numerica alcol è maggiore di zero). Come vedete, praticamente tutte le persone sono decedute in un incidente stradale; solamente tre persone sono decedute rispettivamente per un incidente domestico, per un incidente in volo con elicottero e per suicidio.

```
www = "http://www.biostatisticaumg.it/dataset/tossicologia.csv"
tossicologia = read.csv(www, header = TRUE)
attach(tossicologia)
table(causa)

...
causa
incidentedomestico incidenteelicottero
    1                  1
incidentestradale      suicidio
    182                 1
```

Volendo associare l'alcolemia (negativo o positivo) alle covariate del dataset, proviamo ad eseguire una regressione logistica, riportando parte dell'output in una tabella per comodità di lettura:

```
prova = glm(alcolemia ~ genere + eta + causa + tossicologico,
            family = binomial)
summary(prova)
```

Osservate il fatto che nella colonna degli standard error appaiono alcuni valori molto elevati, del tipo 1455 o 2058. Ricordate quanto dicevamo in apertura di questo libro, a proposito della cosiddetta

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-16.7264	1455.3977	-0.01	0.9908
generem	1.0717	0.5783	1.85	0.0638
eta	0.0016	0.0082	0.19	0.8500
causaincidenteelicottero	0.0187	2058.2429	0.00	1.0000
causaincidentestradale	15.1462	1455.3976	0.01	0.9917
causasuicidio	0.0078	2058.2429	0.00	1.0000
tossicologicopositivo	0.0460	0.4819	0.10	0.9240

equazione più pericolosa, nella sezione 1.4.2: l'errore standard dipende dalla dimensione  $n$  del gruppo che stiamo esaminando, e dunque se abbiamo poca informazione (1 incidente in elicottero) conseguentemente abbiamo molta incertezza (effetto 0.02 con errore standard 2058.2).

Altri problemi con l'errore standard si manifestano anche nei casi di **collinearità**. Ci spieghiamo con un esempio: nella sua tesi di laurea in Medicina e Chirurgia presso la UniTs, intitolata *Convulsioni ed epilessia nei primi due anni di vita: caratterizzazione e prognosi*, la dottoressa Lucrezia Ietri si è occupata del dataset epilessie. Il suggerimento tecnico è quello di 'staccare' il dataset tossicologia con il comando `detach` prima di importare epilessie, per non creare conflitto con la voce causa presente in entrambi i dataset:

```
detach(tossicologia)
www = "http://www.biostatisticaumg.it/dataset/epilessie.csv"
epilessie = read.csv(www , header = TRUE)
attach(epilessie)
```

Quello che qui si vuole accettare è se la **prognosi** (buona o cattiva) sia legata alle covariate del dataset (l'esito del primo elettroencefalogramma, la causa, l'esame obiettivo neurologico, lo sviluppo psicomotorio e l'età all'esordio). Osserviamo anche in questo caso la tabella del **summary**:

```
prova = glm(prognosi ~ primoEEG + causa + eoneuro + svilpsicom
            + etaallesordio, family = binomial)
summary(prova)
```

	Estimate	Std. Error	z value	Pr(> z )
causasintomatico	-15.5730	1696.2449	-0.01	0.9927
eoneurop	16.1697	1696.2452	0.01	0.9924
...	...	...	...	...

Qui gli standard error di causasintomatico e di eoneurop hanno due ordini di grandezza superiori rispetto alle stime dei loro coefficienti (e le stime, in pratica, si cancellano tra loro: sono due numeri quasi opposti, ovvero a somma quasi nulla). L'arcano si spiega osservando che le variabili esame obiettivo e causa sono praticamente 'la stessa variabile', sono cioè fortemente associate. Ecco dunque il significato della parola collinearità:

```
table(eoneuro, causa)
```

...

causa

<i>eoneuro nonsintomatico sintomatico</i>		
<i>n</i>	35	1
<i>p</i>	1	14

Ma ripetendo l'analisi dopo aver eliminato una delle due covariate (ad esempio, l'esame obiettivo neurologico) la situazione cambia drasticamente:

	Estimate	Std. Error	z value	Pr(> z )
causasintomatico	0.0436	0.9656	0.05	0.9640
	..	..	..	..

### 5.1.5 La sovradispersione

Andando a riguardare le sezioni 2.3.2 e 2.3.4 in cui si spiega come si generano in maniera casuale dei numeri distribuiti, rispettivamente, secondo una binomiale con `rbinom` e secondo una normale con `rnorm`, si osserva chiaramente che nella normale siamo liberi di scegliere sia la media che la deviazione standard della distribuzione. Al contrario, nella binomiale, fissata la dimensione `size` e la probabilità `prob` di successo, tutto avviene di conseguenza. In dettaglio, si potrebbe andare a verificare che `size * prob` rappresenta il valore atteso (la media) della distribuzione aleatoria, mentre `size * prob * (1-prob)` ne rappresenta la varianza, ossia il quadrato della deviazione standard.

Tutto questo per dire che mentre in un modello lineare dovevamo preoccuparci di verificare con i plot diagnostici se la media dei residui fosse ragionevolmente nulla, e che non ci fosse eteroschedasticità in questi ultimi, nel modelli lineari generalizzati la faccenda è più sottile. Per discutere i dettagli a fondo, vi rimandiamo per esempio al testo di Julian Faraway [23]. Per farla invece più breve, andiamo a considerare il dataset `gdm` dell'amico endocrinologo Eusebio Chiefari [12] in cui ci si occupa del diabete gestazionale. Nelle analisi preliminari, si cercava di capire se il numero di gravidanze `Ngrav` e l'indice di massa corporea materno `BMIpreGrav` potessero essere dei validi predittori dell'insorgenza del diabete mellito gestazionale, GDM:

```

www = "http://www.biostatisticaumg.it/dataset/gdm.csv"
gdm = read.csv(www, header = TRUE)
attach(gdm)
prova = glm( factor(GDM) ~ Ngrav + BMIpreGrav , family = binomial)
summary(prova)

...
Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.11413    0.29777 -13.816 < 2e-16 ***
Ngrav        0.46692    0.06688   6.981 2.93e-12 ***
BMIPreGrav   0.09382    0.01124   8.344 < 2e-16 ***
...
(Dispersion parameter for binomial family taken to be 1)
Null deviance: 2562.1 on 2283 degrees of freedom
Residual deviance: 2430.2 on 2281 degrees of freedom
AIC: 2436.2
...

```

Eccoci qui al succo della storia: il fatto che nella distribuzione binomiale la media e la deviazione standard sono legate tra loro in maniera algebrica comporta [23] che la devianza residua

non debba superare il numero di gradi di libertà del modello. Se questo accade, come in questo caso in cui la devianza residua vale 2430 mentre i gradi di libertà sono 2281, siamo in presenza del fenomeno denominato **sovradispersione**. Come si corregge la faccenda? Agendo su un ulteriore parametro presente nei modelli lineari generalizzati che si chiama **parametro di dispersione**, che per default viene sempre fissato al valore 1. La stima di tale parametro avviene in maniera euristica; è sufficiente utilizzare la sintassi `family = quasibinomial`:

```
prova2 = glm( factor(GDM) ~ Ngrav + BMIpreGrav , family = quasibinomial)
summary(prova2)

...
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.11413   0.29961 -13.732 < 2e-16 ***
Ngrav        0.46692   0.06730   6.938 5.15e-12 ***
BMIPreGrav   0.09382   0.01131   8.292 < 2e-16 ***
...
(Dispersion parameter for quasibinomial family taken to be 1.012403)
Null deviance: 2562.1 on 2283 degrees of freedom
Residual deviance: 2430.2 on 2281 degrees of freedom
AIC: NA
...
```

L'algoritmo euristicamente stima il miglior valore del parametro di dispersione (1.012) risolvendoci il problema della sovradispersione. Il prezzo da pagare è che non abbiamo più modo di valutare il criterio di informazione di Akaike del modello. Tuttavia potete osservare che, seppur di poco, si è ridotta l'affidabilità delle stime dei coefficienti del modello: infatti gli standard error sono leggermente aumentati, quindi i t value si sono leggermente avvicinati allo zero, peggiorando lievemente i p-value dei coefficienti. Ma ormai noi abbiamo accettato che questi non sono più dei discorsi che hanno un'interesse assoluto.

## 5.2 La meta finale: valutare l'accuratezza del modello logistico

Ritorniamo all'articolo dell'Esercizio 5.3. Una decina di anni fa infatti Richard Moore et al. pubblicarono [40] il 'R.O.M.A., Risk of Ovarian Malignancy Algorithm', un metodo per stimare la probabilità di malignità in una massa ovarica. La dottessa Shadi Najaf, ora ginecologa al Kantonsspital Baden di Zurigo, esplorava la possibilità di aumentare l'accuratezza dell'algoritmo. Carichiamo il suo dataset `roma`:

```
www = "http://www.biostatisticaumg.it/dataset/roma.csv"
roma = read.csv(www, header = TRUE)
attach(roma)
tail(roma)

...
logHE4 logCA125 logCA19.9 logCEA AgePatient Menopause Histology
205  4.26    3.46    3.15   0.98       43     ante    benign
206  3.63    4.14    3.34   1.10       51     ante    benign
207  3.49    2.94   -0.04   0.94       42     ante    benign
208  3.72    4.12    2.13   0.96       36     ante    benign
209  4.06    2.20   -0.02   0.38       55     post    benign
210  3.96    4.03    1.67   0.71       63     post malignant
```

Come si vede dal comando `tail`, Shadi Najaf aveva raccolto dati retrospettivi su 210 pazienti distinguendone la `Histology` la quale sarebbe potuta essere associata alla `AgePatient`, alla `Menopause` ed a quattro possibili biomarcatori (trasformati in scala logaritmica): il `logHE4`, il `logCA125`, il `logCA19.9` ed infine il `logCEA`.

Nell'articolo di Moore et al. l'attenzione si focalizzava su due predittori lineari, l'uno per le donne in pre-menopausa, l'altro per le pazienti in post-menopausa, rispettivamente:

$$\begin{aligned} P.I. &= -12.0 + 2.38 \cdot \log(HE4) + 0.0626 \cdot \log(CA125) \\ P.I. &= -8.09 + 1.04 \cdot \log(HE4) + 0.732 \cdot \log(CA125) \end{aligned}$$

Osserviamo che i coefficienti di `logHE4` e di `logCA125` cambiano nelle due formule; questo implica che il predittore `Menopause` interagisce con i due biomarcatori – notate la analogia con la 'Ancova' della Sezione 4.2 nella quale si cercava di capire attraverso la `relationcross` se il `weight` dei `fresher` dipendessero dalla `height` in maniera specifica nei due diversi `gender`? Quindi, il modello proposto da Moore et al. [40] si esplicita nel modo seguente:

```
moorerelation = Histology ~ Menopause * logHE4 + Menopause * logCA125
```

Il primo passo che Shadi doveva compiere era verificare se anche nel suo dataset si giungeva al modello proposto dagli autori. Possiamo replicare la sua analisi: si parte esplicitando il **modello massimale additivo**, nel quale come abbiamo detto nella sezione 4.5 tutte le covariate del dataset sono presenti collegate da un segno più:

```
maximalrelation = Histology ~ logHE4 + logCA125 + logCA19.9 + logCEA
+ AgePatient + Menopause
```

Definito il predittore lineare massimale, calcoliamo il modello lineare generalizzato e lo semplifichiamo per mezzo della funzione `step`, utilizzando il criterio AIC per individuare un possibile candidato:

```
maximalmodel = glm(maximalrelation, family = binomial)
step(maximalmodel)

...
Coefficients:
(Intercept)      logHE4      logCA125   Menopausepost
-14.3770        2.3382       0.6845        0.9378

...
Residual Deviance: 107.3          AIC: 115.3
```

Le tre covariate che Moore et al. propongono sono presenti anche in questa nostra analisi. Ma adesso si tratta di andare anche a verificare se sia opportuno considerare le interazioni della menopausa con i livelli dell'`HE4` o del `CA125`. Facciamo dei tentativi, facendo interagire di volta in volta `Menopause` con uno dei due biomarcatori (`attempt1` e `attempt2`), e per sicurezza proviamo a controllare anche se i due biomarcatori interagiscono tra loro (`attempt3`). Da ultimo, raffrontiamo tutti questi modelli per mezzo del criterio AIC:

```
attempt0 = Histology ~ Menopause + logHE4 + logCA125
attempt1 = Histology ~ Menopause * logHE4 + logCA125
attempt2 = Histology ~ logHE4 + Menopause * logCA125
attempt3 = Histology ~ Menopause + logHE4 * logCA125
```

```

moorerelation = Histology ~ Menopause * logHE4 + Menopause * logCA125

modelattempt0 = glm(attempt0, family = binomial)
modelattempt1 = glm(attempt1, family = binomial)
modelattempt2 = glm(attempt2, family = binomial)
modelattempt3 = glm(attempt3, family = binomial)
mooremodel = glm(moorerelation, family = binomial)

AIC(modelattempt0, modelattempt1, modelattempt2, modelattempt3, mooremodel)

...
      df      AIC
modelattempt0 4 115.2610
modelattempt1 5 115.8854
modelattempt2 5 116.7179
modelattempt3 5 116.4479
mooremodel     6 117.4887

```

Diversamente da quanto affermato da Moore et al. in [40], i dati di Shadi Najaf non danno evidenza che sia necessario andare a considerare l'interazione con la variabile Menopause. Per completezza, andiamo a verificare se sia opportuno inserire un termine di curvatura sui biormarcatori:

```

attempt4 = Histology ~ Menopause + logHE4 + logCA125 + I(logCA125^2)
attempt5 = Histology ~ Menopause + logHE4 + I(logHE4^2) + logCA125

modelattempt4 = glm(attempt4, family = binomial)
modelattempt5 = glm(attempt5, family = binomial)

AIC(modelattempt0, modelattempt4, modelattempt5, mooremodel)

...
      df      AIC
modelattempt0 4 115.2610
modelattempt4 5 116.9350
modelattempt5 5 117.1341
mooremodel     6 117.4887

```

Nulla di fatto: il modello minimale adeguato che Shadi Najaf individua differisce da quello proposto da Moore et al., ed è molto più semplice. Commentiamo il `summary` in maniera approfondita:

```

summary(modelattempt0)

...
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.3770    2.6739  -5.377 7.58e-08 ***
Menopausepost  0.9378    0.5700   1.645 0.099912 .
logHE4        2.3382    0.6524   3.584 0.000338 ***
logCA125      0.6845    0.2029   3.374 0.000741 ***
...
```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 201.58 on 209 degrees of freedom  
 Residual deviance: 107.26 on 206 degrees of freedom  
 AIC: 115.26

Number of Fisher Scoring iterations: 6

Partiamo dal basso. Per calcolare la devianza dei residui, diversamente da quanto accadeva nel caso `lm`, non si dispone di un metodo algebrico esatto, ma ci si deve affidare a delle procedure ricorsive: si deve infatti massimizzare la **verosimiglianza** del modello (ne avevamo parlato nella sezione 4.1.5) di solito utilizzando il metodo numerico della derivata di Newton e Raphson. Questa procedura va sotto al nome di **Fisher Scoring**, e se volete degli approfondimenti vi rimandiamo ad esempio a [23]. Siccome la devianza dei residui vale circa 107.3 rispetto a 206 gradi di libertà del modello non abbiamo da preoccuparci del problema della sovradispersione: bene, non occorre invocare la `family = quasibinomial` e teniamo fermo ad 1 il parametro di dispersione.

Possiamo ora confrontare i predittori lineari proposti da Moore et al. con quelli di Shadi Najaf, sempre distinguendo tra pre e post menopausa:

$$P.I. = -14.38 + 2.34 \cdot \log(HE4) + 0.68 \cdot \log(CA125)$$

$$P.I. = -13.44 + 2.34 \cdot \log(HE4) + 0.68 \cdot \log(CA125)$$

noteremo che ora l'unica variazione si manifesta nel valore leggermente più basso dell'intercetta, mentre i coefficienti dei biomarcatori non variano.

Anche nei modelli lineari generalizzati `glm` è possibile eseguire una parziale diagnostica del modello, come avevamo discusso nella sezione 4.1.6, disponendo dei comandi `residuals` ed `influence`; ci permettiamo di rimandarvi al libro di Julian Faraway [23] per un approfondimento.

### 5.2.1 La curva ROC

Si tratta ora di valutare la qualità dell'indice predittivo proposto da Shadi Najaf. Per farlo visivamente, disponiamo di un grafico nato in seguito alla battaglia di Pearl Harbor, la **curva ROC**:



Vi interessa la storia della receiver operating characteristic curve? Eccola: [https://en.wikipedia.org/wiki/Receiver\\_operating\\_characteristic#History](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#History)

Per disegnarlo, abbiamo innanzitutto bisogno di trasformare, almeno in maniera temporanea, la variabile fattore `Menopause` con i suoi livelli `ante / post` ad una variabile numerica di tipo 0 / 1; per farlo usiamo questa sintassi:

**Esercizio 5.4** Verificate che questo comando fa al caso nostro, comprendendo passo per passo i cambiamenti che accadono:

```
(-1 + as.numeric(Menopause))
```

Usiamo questo pezzettino di istruzione per creare `iSN`, l'indice predittivo di Shadi Najaf:

```
Menopause01 = -1 + as.numeric(Menopause)
iSN = -14.38 + 0.94 * Menopause01 + 2.34 * logHE4 + 0.68 * logCA125
```

A questo punto, dobbiamo installare un pacchetto aggiuntivo, per esempio pROC [46] (ma anche ROCR [50] andrebbe benissimo):

```
install.packages("pROC")
library(pROC)
```

Con il comando `roc` calcoliamo tutta una serie di parametri, il più noto dei quali è l'area al di sotto della curva, `auc`, la quale fornisce un modo (molto) approssimativo di paragonare diversi predittori tra di loro. Ci vuole però una certa cautela, perché bisogna avere molto chiaro in mente se del preditore in esame vogliamo evidenziare le qualità della sensibilità, o della specificità, o di entrambe accettando qualche necessario compromesso.

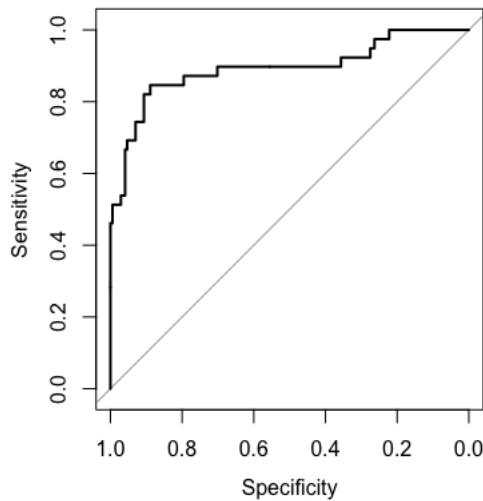
```
iSNroc = roc(Histology ~ iSN, auc=TRUE)
iSNroc
```

*Call:*

```
roc.formula(formula = Histology ~ iSN, auc = TRUE)
```

*Data: iSN in 171 controls (Histology benign) < 39 cases (Histology malignant).*  
*Area under the curve: 0.892*

Possiamo ovviamente anche visualizzare la curva ROC con il comando `plot(iSNroc)`:



### 5.3 Esercizi ed attività di approfondimento

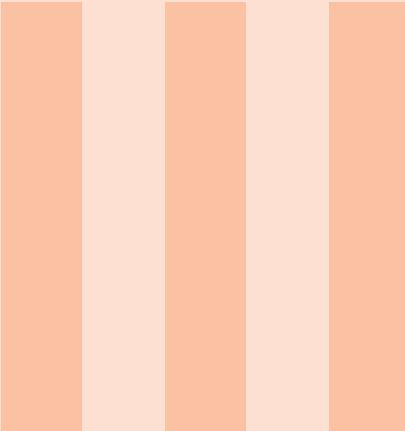
■ **Attività 5.1 — la curva ROC.** Riprendete in esame il dataset `raul` e definite l'indice predittivo come descritto in [19]:

```
umgrisk = Ldhtre + 24 / Ldhuno
```

A questo punto, valutate l'area sotto la curva ROC e disegnatela, e capirete l'emozione di Annalisa Di Cello quando si è accorta di questo risultato:

```
annalisa = roc(Esito ~ umgrisk, auc=TRUE)
annalisa
plot(annalisa)
```

■



# Terza Parte

<b>6</b>	<b>Misure ripetute .....</b>	<b>137</b>
6.1	Lo strano caso delle gemelle Alice ed Ellen	
6.2	Il modello lineare ad effetti misti	
6.3	Intermezzo ectopico, inattuale ed (abbastanza) opzionale	
6.4	La selezione di un modello ad effetti misti	
6.5	La Anova repeated measures, i modelli misti e la qRT-PCR	
6.6	Esercizi ed attività di approfondimento	
<b>7</b>	<b>Sopravvivenza .....</b>	<b>155</b>
7.1	I dati di tipo Time-to-Event	
7.2	La cornice matematica	
7.3	Le curve di sopravvivenza con R	
7.4	I modelli semiparametrici	
7.5	Esercizi ed attività di approfondimento	
	<b>Bibliografia .....</b>	<b>171</b>
	Articoli	
	Libri	
	Risorse Web	
	<b>Index .....</b>	<b>175</b>





## 6. Misure ripetute

Sono trascorse più di cento pagine da quando nel primo Capitolo abbiamo introdotto l'esempio delle qRT-PCR di Stefania Scalise e Maria Teresa De Angelis [17], nelle quali gli esperimenti venivano effettuati in triplicato tecnico e duplicato biologico. Nello stesso capitolo avevamo anche accennato al fatto che Emanuela Chiarella [14] occupandosi della proteina Zinc Finger nel paper *ZNF521 Represses Osteoblastic Differentiation in Human Adipose-Derived Stem Cells* aveva effettuato delle misure di espressione del mRNA all'inizio dell'esperimento e successivamente ripetuto tali misure al quinto, al decimo ed al ventesimo giorno di incubazione. Come dicevamo, il professore Gianni Morrone aveva di questa proteina una rilevante conoscenza e l'articolo in questione apparve proprio pochi giorni che egli ci lasciasse. Inoltre, una trentina di pagine fa vi dicevamo che Teresa Calimeri aveva studiato il comportamento di alcune paraproteine presenti nello sviluppo del mieloma multiplo [11] in cinque occasioni temporali, rispettivamente a 0, 25, 45, 60 ed 80 minuti dall'inizio dell'esperimento.

Potremmo continuare questa introduzione con una lista smisurata di articoli e di tesi in cui gli esperimenti od i clinical trial vengano condotti in modo analogo a questo; nella ricerca biomedica è consuetudine effettuare **misure ripetute** sui medesimi soggetti all'interno di un campione prescelto, generando dataset popolati da **dati correlati** (ossia, caratterizzati dal fenomeno della '**pseudoreplication**'). Queste misure possono avvenire in diverse occasioni temporali (quando seguiamo il decorso di una patologia nei pazienti – in tal caso parliamo di **esperimenti longitudinali**) – oppure, potrebbe accadere che il tempo non giochi un ruolo di interesse, ma le misure vengano ripetute semplicemente per aumentare l'affidabilità (sottointeso: della stima di un parametro non noto a priori quale ad esempio la media della popolazione delle unità statistiche prese a riferimento), nei termini in cui abbiamo parlato nella sezione 1.4.2. In quest'ultimo frangente, meno utilizzata in ambito biomedico (tipica invece della ricerca sociale) è la terminologia di **dati in cluster**: dati nei quali le covariate del dataset possiedono una naturale gerarchia (per fare un esempio sciocco: la media dei voti di matematica nelle varie classi e sezioni di un medesimo istituto scolastico, nei quali però insegnano un giovane ed aitante prof. di manica larga ed un anziano e ripugnante prof. severissimo). Questo concetto invece si sovrappone in maniera perfetta ad esempio negli esperimenti di qRT-PCR, in cui ciascun replicato tecnico 'comprende', 'racchiude' in sé un certo

numero di replicati biologici, e in questo caso comunemente si parla di fattori nidificati, **nested factors**.

## 6.1 Lo strano caso delle gemelle Alice ed Ellen

Divertiamoci con questa storiella inventata. Le gemelle Alice ed Ellen sono due anziane signore che, dopo aver condotto una vita artistica di grande successo, decidono di riprendere gli studi di biostatistica che avevano interrotto alcuni decenni fa. Alice ed Ellen decidono di fare uno **studio osservazionale**: alzarsi dal letto assieme ogni mattina ed immediatamente pesarsi, per rispondere alla seguente domanda: *le gemelle Alice ed Ellen hanno lo stesso peso?* All'indomani, eseguito il primo esperimento e preso nota del responso della bilancia (accuratissima, digitale, che non si lascia perturbare dalle onde gravitazionali, ecc. ecc.) la situazione è la seguente:

Alice	Ellen
73.60	73.80

A questo punto, Alice ed Ellen sarebbero propense a decidere che *non hanno* lo stesso peso, giacché, ragionando da un punto di vista puramente matematico, i due numeri non coincidono. Ma le gemelle sanno che, nella Natura, la variabilità la fa da padrona e così scelgono di fare un secondo esperimento, ossia di pesarsi per cinque mattine consecutive (**studio osservazionale longitudinale**):

	Alice	Ellen
1	73.60	73.80
2	73.40	73.50
3	74.10	74.60
4	73.50	73.80
5	73.20	73.60

Per dirimere la questione esse ricorrono al celebre t test di Student, quello a due campioni della sezione 3.2.2, che era poi l'unico test di cui hanno un vago ricordo dei loro studi in gioventù.

```
alice = c(73.6, 73.4, 74.1, 73.5, 73.2)
ellen = c(73.8, 73.5, 74.6, 73.8, 73.6)
t.test(alice, ellen, var.equal = TRUE)

...
Two Sample t-test

data: alice and ellen
t = -1.2227, df = 8, p-value = 0.2562
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.865794 0.265794
sample estimates:
mean of x mean of y
73.56      73.86
```

Alice ed Ellen si ricordano che a quel tempo se il p-value era superiore al 5 per cento si diceva che la differenza non era significativa, il che in pratica - loro, almeno, l'avevano capita in questo modo

- le rendeva propense a decidere che esse *hanno* lo stesso peso, in base al p-value = 0.2562. Le gemelle tuttavia ricordano che l'affidabilità delle misure aumenta con il numero di repliche e scelgono perciò di continuare a pesarsi complessivamente per tre settimane:

	Alice	Ellen		Alice	Ellen
1	73.60	73.80	12	74.10	74.60
2	73.40	73.50	13	73.60	73.80
3	74.10	74.60	14	73.40	73.60
4	73.50	73.80	15	74.10	74.40
5	73.20	73.60	16	73.50	73.70
6	74.00	74.40	17	73.20	73.50
7	73.60	73.80	18	74.00	74.40
8	73.30	73.50	19	73.60	73.90
9	74.20	74.30	20	73.30	73.60
10	73.60	73.90	21	74.20	74.50
11	73.40	73.60	-	-	-

Inserendo i dati in R e ripetendo il comando t.test, ecco la sorpresa:

```
...
Two Sample t-test

data: peso by gemella
t = -2.4594, df = 40, p-value = 0.01834
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.51183215 -0.05007261
sample estimates:
mean in group alice mean in group ellen
73.66190          73.94286
```

Colpo di scena! Alice ed Ellen ora si sentono confuse più che mai, perché ora dovrebbero decidere che *non hanno* lo stesso peso, in base al p-value = 0.01834, significativo; contrariamente a quello che era accaduto nel secondo esperimento. Tutto ciò è molto strano. Di chi è la colpa?

### 6.1.1 Tutta colpa di Student?

Abbiamo iniziato il Capitolo 1 dicendo che dovremmo sempre distinguere il caso in cui si voglia effettuare l'analisi dei dati condotti in un design sperimentale di tipo cross-section oppure nel caso delle misure ripetute. Importiamo il dataset gemelle:

```
www = "http://www.biostatisticaumg.it/dataset/gemelle.csv"
gemelle = read.csv(www, header = TRUE)
attach(gemelle)
str(gemelle)

...
'data.frame': 42 obs. of 2 variables:
$ gemella: Factor w/ 2 levels "alice","ellen": 1 1 1 1 1 1 1 1 1 ...
$ peso   : num 73.6 73.4 74.1 73.5 73.2 74 73.6 73.3 74.2 73.6 ...
```

Si tratta di un dataset 'tecnicamente sbagliato', perché viene preso per scontato un elemento non inutile: il *timestamp*, il tempo che passa, il quale usualmente è rappresentato da un contatore a numeri interi (sono pochi i frangenti nei quali è necessario utilizzare il *calendar time*); il dataset quindi non si presenta come una serie temporale, analoga all'esempio airquality di cui abbiamo discusso nella sezione 1.1, ed il peso di Alice e di Ellen non viene raccolto su due colonne diverse. Proviamo allora ad interpretare il *summary* del modello, alla luce di quanto dicevamo nella sezione 4.1.3 riguardo alla componente aleatoria di un modello lineare.

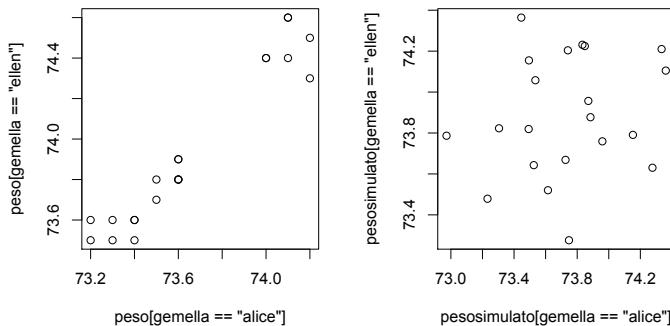
```
ttestsbagliato = lm(peso ~ gemella)
ttestsbagliato
summary(ttestsbagliato)$sigma

...
(Intercept)  gemellaellen
    73.662      0.281
...
[1] 0.3701673
```

Ci ricordiamo due fatti: il primo, che il predittore lineare (gli 'effetti fissi') è descritto dalla 'retta'  $y = 73.66 + 0.28 \cdot x$ , dove  $x$  vale 0 per `alice` ed 1 per `ellen` (quindi, il peso medio di Alice è circa 73.7 mentre quello di Ellen è circa 73.9); il secondo, che il predittore lineare  $y$ , ossia i pesi medi delle due artiste, viene perturbato in maniera aleatoria da 42 residui, distribuiti normalmente con media nulla e deviazione standard  $\epsilon = 0.37$  ('effetti casuali').

Utilizziamo dunque questi tre parametri per una simulazione che confronteremo con la situazione sperimentale. Simulando infatti con `rnorm` 42 numeri casuali che provengano da una variabile aleatoria gaussiana con media zero e con quella deviazione standard, dovremmo ottenere un grafico paragonabile a quello del peso delle nostre due anziane artiste.

```
set.seed(123) # per la riproducibilità
pesosimulato = c(rep(73.7, 21), rep(73.9, 21)) + rnorm(42, 0, 0.37)
par(mfrow = c(1,2))
plot(peso[gemella == "alice"], peso[gemella == "ellen"])
plot(pesosimulato[gemella == "alice"], pesosimulato[gemella == "ellen"])
```



La situazione di sinistra è quella del dataset `gemelle`: i pallini vanno a disporsi lungo la diagonale del grafico perché le gemelle conducono una vita speculare, giorno dopo giorno e quindi anche il peso dell'una varia come il peso dell'altra.

Viceversa, a destra, la simulazione creata a partire dal modello lineare `ttestsbagliato` non rispecchia affatto la situazione originale. Detto in altri termini, la simulazione ottenuta a destra mediante il modello statistico non rappresenta il fenomeno sperimentale di sinistra: pertanto il modello statistico prescelto non è adeguato alla realtà descritta dai dati sperimentali. A sinistra, ci troviamo in una situazione di elevata **informazione**; a destra, in una situazione di assenza di informazione, ovvero di elevata **entropia** [10]. E questo contravviene alla richiesta di adeguatezza di un modello statistico [15]. E dunque, il **modello lineare ad effetti fissi** `lm` si conferma essere appropriato solamente quando siamo in presenza di dati indipendenti e non, come in questo caso, di **dati correlati** [58].

La proposta risolutiva è quella di introdurre nel modello lineare un'ulteriore variabile aleatoria, o meglio, un'ulteriore componente di effetti casuali, che tenga conto del fatto che sono state effettuate delle repliche di misura sul medesimo soggetto, all'interno del modello prescelto. Questa tecnica che si perfezionò una trentina di anni fa [36] va sotto il nome di **modelli lineari ad effetti misti**. R dispone di un pacchetto 'storico' (nel senso di – absit iniuria verbis – obsoleto) per effettuare i calcoli, `n1me` [44], ma lo standard de facto è divenuto ormai il pacchetto `lme4` [5].

Per capire come funzionano le cose cominceremo con un dataset che in qualche modo assomiglia agli esempi delle rette di regressione relative al peso degli studenti `fresher` che abbiamo discusso nella sezione 4.2. Passeremo poi a vedere le questioni tipiche che accadono quando analizziamo i dati delle qRT-PCR.

## 6.2 Il modello lineare ad effetti misti

### 6.2.1 Un esempio introduttivo

Il professor Umberto Lucangelo di Trieste è un esperto delle tecniche di ventilazione ad alta frequenza (`hfvp`, *high frequency percussive ventilation*) nei pazienti affetti da gravi sindromi respiratorie. Il dataset `percussive` riporta i dati di uno studio [39] in cui il gruppo `hfvp` di 35 pazienti caratterizzati da grave compromissione respiratoria vengono trattati con un ventilatore ad alta frequenza per 12 ore consecutive, e gli effetti di tale ventilazione si devono comparare con un gruppo `control` di 35 pazienti ventilati in maniera convenzionale. La risposta che ci interessa modellare è una indice di ossigenazione alveolare ( $PaO_2/FiO_2$ ).

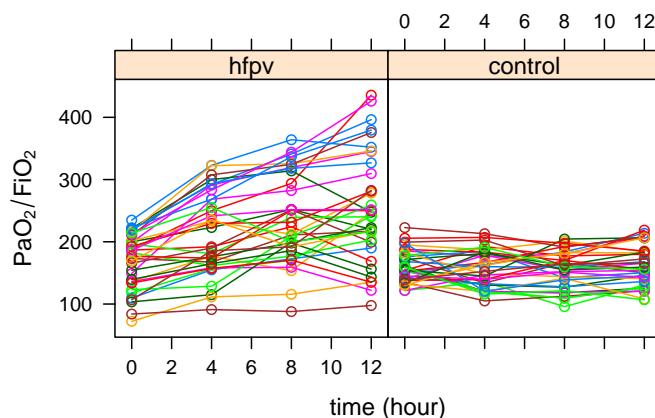


Figura 6.1: L'indice di respirazione alveolare  $PaO_2/FiO_2$  mostra un comportamento generalmente crescente nel gruppo `hfvp`, mentre sembra rimanere stazionario nel gruppo `control`. Come possiamo certificare questa differenza in maniera statistica?

Osservate le analogie e le differenze ad esempio con il grafico relativo al model cross che avevamo fatto nella sezione 4.2. In quel disegno ogni diamante raffigurato rappresentava una delle 65 **unità statistiche**, ossia ciascuno degli studenti fresher. Qui invece ci sono 280 pallini, 140 nel pannello a sinistra e 140 nel pannello a destra. Infatti su ciascuno delle 70 **unità statistiche** (i pazienti, 35 trattati e 35 controlli) abbiamo raccolto 4 **misure ripetute** nelle diverse occasioni temporali. In altre parole, in uno studio di tipo cross-section l'unità statistica veniva rappresentata da un punto; nel design a misure ripetute l'unità statistica viene qui rappresentata con una spezzata, una traiettoria.

### 6.2.2 Come impostare il dataset

Il primo scoglio da superare è apparentemente banale, ma esiziale: bisogna che i dati che abbiamo raccolto siano stati sistemati in quello che viene definito come **long format** (cosa che succede praticamente sempre se i dati li raccogliete ad esempio in un database, come quelli di tipo .php che si raccolgono via web), invece che nello **short format** (cosa che succede praticamente sempre se i dati li raccogliete a mano in un foglio elettronico nel vostro personal computer). Capirete immediatamente la faccenda dando un'occhiata al disegno qui sotto, in cui un foglio dati di tipo short viene trasformato in tipo long: i valori numerici raccolti (in questo esempio la  $PaO_2/FiO_2$  dei pazienti) vengono ri-arrangiati in un'unica colonna; il che comporta che bisogna ripetere in colonna per quattro volte il 'tempo' in cui sono stati raccolti ed il 'gruppo' cui appartengono. Inoltre anche l'identificativo dell'unità statistica deve venir ripetuto – e non ci stancheremo mai di ripetere ai nostri studenti di **N O N** utilizzare mai nome e cognome per individuare i pazienti: qui lo abbiamo fatto solamente a titolo didattico. Vi suggeriamo invece di raccogliere in un foglio separato le informazioni anagrafiche delle persone (esempio un contatore progressivo ed il nome, il cognome, la data di nascita, il numero di telefono) e poi creare in maniera automatica un identificatore che estragga qualche lettera e qualche cifra di queste informazioni e le concatensi tra loro creando un codice univoco.

	A	B	C	D	E	F
1	paziente	gruppo	tempo00	tempo04	tempo08	tempo12
2	Mario Rossi	trattato	220	292	341	396
3	Maria Verdi	controllo	186	176	180	209
4	Mario Bianchi	trattato	100	201	319	345



	A	B	C	D
1	paziente	gruppo	tempo	paf
2	Mario Rossi	trattato	0	220
3	Mario Rossi	trattato	4	292
4	Mario Rossi	trattato	8	341
5	Mario Rossi	trattato	12	396
6	Maria Verdi	controllo	0	186
7	Maria Verdi	controllo	4	176
8	Maria Verdi	controllo	8	180
9	Maria Verdi	controllo	12	209
10	Mario Bianchi	trattato	n	100

Vi suggeriamo inoltre di evitare di trasformare 'a mano' i dati da short a long, a meno che non si tratti di un dataset molto molto contenuto. Fatelo fare ad R sfruttando in maniera opportuna le istruzioni `rep` e `seq`:

**Esercizio 6.1** Provate ad importare in R un dataset realizzato in formato short e sfruttando l'istruzione `data.frame` trasformatelo in uno in formato long. Fate attenzione a creare dei `names` che non siano in conflitto con quelli esistenti. ■

Il dataset `percussive` è già predisposto in 'formato long'. Facciamo solo delle piccole trasformazioni, traslando la variabile originale `time` di dodici ore (il tempo 0 dello studio era stato fissato dopo le prime dodici ore trascorse dal ricovero nel reparto di terapia intensiva) ed individuiamo i due gruppi (`treatment`) di pazienti con le sigle `hfpv` e `control` invece che con la codifica originale 1 e 0 (si noti che così facendo invertiamo il consueto ordine che R utilizza – alfabetico e numerico – nei livelli di un fattore). Da ultimo, notiamo che `percussive` è un file di formato `.txt` (nel quale di default il separatore di campo è il segno di tabulazione, e non la virgola

decimale) e quindi preferiamo usare il comando `read.table` invece che `read.csv` (ma si sarebbe potuto continuare ad usare `read.csv` specificando all'interno l'opzione `sep` del separatore).

```
www = "http://www.biostatisticaumg.it/dataset/percussive.txt"
percussive = read.table(www, header = TRUE)
attach(percussive)
time = time - 12
treatment = factor(treatment)
treatment1 = treatment[1:70]
levels(treatment)[1] = "hfpv"
levels(treatment)[2] = "control"
str(percussive)
head(percussive)
tail(percussive)

...
subject treatment time pafi
1      1          0   12 220.7
2      2          0   12 198.8
3      3          0   12 200.0
...
278    68         1   24 206.4
279    69         1   24 156.0
280    70         1   24 165.4
```

Già che ci siamo, prima di procedere, vi dico anche che si può realizzare la Figura 6.1 sfruttando il pacchetto `lattice` in questo modo:

```
library(lattice)
xyplot(pafi ~ time | treatment, type = "b", groups = subject,
       xlab = "time (hour)", ylab = expression(Pa0[2] / Fi0[2]))
```

### 6.2.3 L'analisi non appropriata

Allora, tenendo a mente il conto esempio della gemelle Alice ed Ellen 6.1, proviamo a condurre un'analisi non appropriata comparando tra loro due modelli ad effetti fissi, quello in cui il `time` interagisce con il `treatment` e quello invece in cui non vi è interazione:

```
ancovasbagliatocross = lm(pafi ~ time * treatment)
summary(ancovasbagliatocross)
ancovasbagliatoplus = lm(pafi ~ time + treatment)
summary(ancovasbagliatoplus)
AIC(ancovasbagliatocross, ancovasbagliatoplus)

...
df      AIC
ancovasbagliatocross 5 3021.629
ancovasbagliatoplus  4 3042.522
```

Come vediamo, il criterio di informazione di Akaike più piccolo ci indica di preferire il modello con interazione, facendoci propendere per la scelta che il comportamento temporale della pafi

all'interno dei due gruppi sia differente in senso statistico. Intendiamoci, questo è esattamente il titolo del paper [39] del professor Lucangelo ed è esattamente quello che vogliamo dimostrare. Ma se avessimo condotto questa analisi 'il referee ci avrebbe segato il paper', per avere utilizzato un modello ad effetti fissi, nel quale si ipotizza che le osservazioni sono indipendenti; al contrario, se un certo Mario Bianchi del gruppo hfpv ha riportato i valori di pafi 199, 291, 319, nelle prime tre osservazioni, sarebbe ben strano che la quarta osservazione si discostasse di molto da 345 (sono i dati presi ad esempio del paziente numero 3): come ho già detto all'inizio del Capitolo, qui siamo in presenza di **dati correlati**.

### 6.2.4 Interpretare il modello ad effetti misti

Il modello ancovasbagliatocross sarebbe stato caratterizzato dai seguenti coefficienti:

```
> ancovasbagliatocross
```

*Call:*

```
lm(formula = pafi ~ time * treatment)
```

*Coefficients:*

(Intercept)	time	treatmentcontrol	time:treatmentcontrol
177.039	6.923	-19.395	-6.841

Come sappiamo, questo significa che per il gruppo dei trattati (hfpv) la retta di regressione che meglio descrive le traiettorie della Figura 6.1 avrebbe avuto equazione  $y = 177.0 + 6.9 \cdot x$ , mentre per il gruppo dei controlli avremmo avuto una retta praticamente orizzontale, di equazione  $y = 157.6 + 0.1 \cdot x$ .

Per comodità, riportiamo anche il summary completo di ancovasbagliatocross che adesso andremo a confrontare con il modello ad effetti misti.

```
...
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 177.0394 7.4645 23.718 < 2e-16 ***
time         6.9227 0.9975  6.940 2.79e-11 ***
treatmentcontrol -19.3949 10.5563 -1.837 0.0672 .
time:treatmentcontrol -6.8405 1.4106 -4.849 2.07e-06 ***
...
Residual standard error: 52.78 on 276 degrees of freedom
...
```

Vediamo invece in cosa consiste il modello ad effetti misti. Carichiamo la library aggiuntiva:

```
library(lme4)
```

Vogliamo introdurre degli effetti casuali che modifichino queste due rette per ciascun paziente, modellando la variabilità biologica che osserviamo in Natura: alcuni pazienti partono da condizioni migliori, altri da peggiori (differenti intercetta  $a$ ); alcuni riacquistano più velocemente i livelli di ossigenazione di riferimento, altri purtroppo non migliorano (differenti pendenza  $b$ ).

Guardiamo innanzitutto come si specifica la formula:

```
effettimisti1 = pafi ~ time * treatment + (time | subject)
```

Quel termine aggiunto tra parentesi indica ad R che l'unità statistica `subject` comparirà nelle righe del dataset ripetutamente. Infatti, il numero 1 apparirà quattro volte, e così il 2, e via via sino al paziente 70. Questo non accade nei design sperimentali di tipo cross-section. La presenza invece del simbolo `time` indica al software che vogliamo perturbare – nel senso appena ricordato – entrambi i coefficienti  $a$  e  $b$  della retta di regressione  $y = a + b \cdot x$ . In certi frangenti potrebbe essere sensato perturbare casualmente solo il coefficiente  $a$ , in altri solo il coefficiente  $b$ ; in tal caso si dovrebbe usare, rispettivamente, la simbologia:

```
... + (1 | subject)
... + (time - 1 | subject)
```

Ma non ci dilunghiamo su questi aspetti e proseguiamo. Utilizziamo il comando `lmer` (linear mixed effect regression) e vediamo di interpretare il `summary` del modello:

```
modellomisto1 = lmer(effettimisti1)
summary(modellomisto1)

...
Random effects:
Groups      Name        Variance Std.Dev. Corr
subject    (Intercept) 1232.12   35.102
           time         14.24    3.774   0.22
Residual               433.32   20.816

...
Fixed effects:
              Estimate Std. Error t value
(Intercept) 177.0394   6.6234  26.729
time          6.9227   0.7495   9.237
treatmentcontrol -19.3949   9.3670  -2.071
time:treatmentcontrol -6.8405   1.0599  -6.454
```

### Gli effetti fissi.

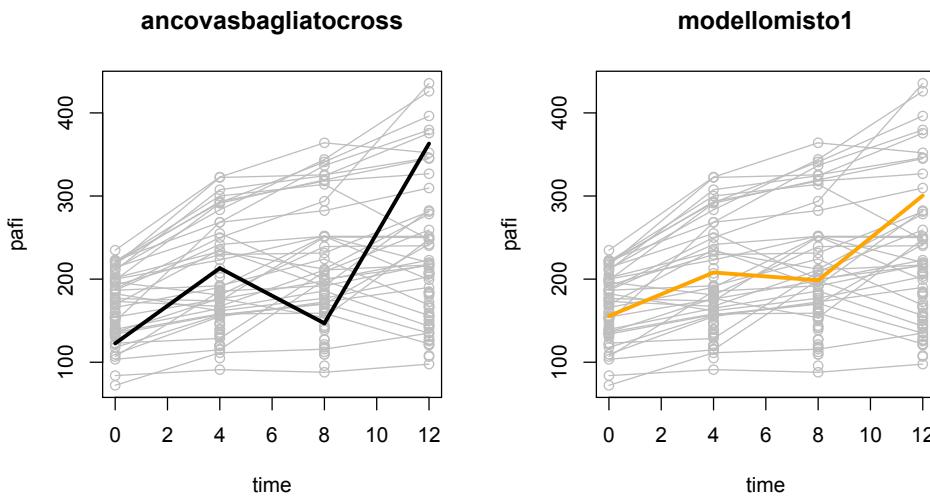
Concentriamo innanzitutto l'attenzione sulle `Estimate` dei `Fixed effects`; siamo confortati nel ritrovare esattamente i coefficienti  $a$  e  $b$  che avevamo ottenuto nel modello inappropriato `ancovasbagliatocross`. Questo non solo significa che il paper di Umberto Lucangelo [39] diceva la verità. Ora possediamo un'interpretazione metodologicamente corretta del fenomeno in base ai termini di **popolazione** e **campione**: nella popolazione dei pazienti `hfpv`, i ventilati ad alta frequenza da cui abbiamo tratto un campione di trentacinque soggetti, l'ossigenazione `pafi` ha un comportamento modellabile dalla retta crescente  $y = 177.0 + 6.9 \cdot x$ . Il comportamento nella popolazione dei controlli invece è dato dalla retta quasi orizzontale  $y = 157.6 + 0.1 \cdot x$ .

### Gli effetti casuali.

Per spiegare la variabilità che si osserva '**between**', cioè tra le singole trentacinque traiettorie grigie ossia i singoli pazienti appartenenti ad una determinata popolazione, oppure '**within**', cioè all'interno di una singola traiettoria (che ovviamente non sarà mai una retta perfetta), abbiamo bisogno delle informazioni relative alle **componenti della varianza**, ossia degli effetti casuali.

Iniziamo con quella dei `Residual`, andando a rileggere quella che era la componente aleatoria di `ancovasbagliatocross`: lì avevamo un `Residual standard error` di circa 52.8. Dunque, per un ipotetico trentaseiesimo paziente `hfpv`, i quattro valori della `pafi` al tempo 0, al tempo 4, al tempo 8 ed al tempo 12 si sarebbero dovuti simulare con le quattro formule:

$$1. \quad y = 177.0 + 6.9 \cdot 0 + \epsilon_1$$



2.  $y = 177.0 + 6.9 \cdot 4 + \varepsilon_2$
3.  $y = 177.0 + 6.9 \cdot 8 + \varepsilon_3$
4.  $y = 177.0 + 6.9 \cdot 12 + \varepsilon_4$

dove  $\varepsilon_1, \varepsilon_2, \varepsilon_3$  e  $\varepsilon_4$  sono quattro numeri casuali generati con `rnorm` di media nulla e deviazione standard 52.8. Ebbene, questa affermazione non è plausibile:

```
set.seed(1234)
residuiacaso = rnorm(4, 0, 52.8)
residuiacaso

...
[1] -54.372650  8.574832 -85.360546 103.313520
```

Utilizziamo questi `residuiacaso` e disegniamo nel pannello di sinistra in colore nero la traiettoria risultante. Vi sembra una traiettoria plausibile, che 'si comporta' come le altre grigie? Al contrario, consideriamo nel modello ad effetti misti `modellomisto1` la Std.Dev.dei Residual 20.8 e generiamo, per paragone usando lo stesso seme, altri quattro `residuimixed`:

```
set.seed(1234)
residuimixed = rnorm(4, 0, 20.8)
residuimixed

...
[1] -21.419529  3.377964 -33.626882  40.699266
```

Osservate ora la traiettoria arancione nel pannello di destra: i 'cambi di direzione' della traiettoria sono ancora imprevedibili (ed anzi, essendo stati generati dal medesimo seed sono proporzionali ai precedenti) ma sono più contenuti, proprio perché la deviazione standard che li ha generati è molto più contenuta, dando vita ad una simulazione che 'si confonde' con i dati realmente osservati – e questo è ciò che si desidera per modellare la *variabilità 'within'*.

Passiamo ora alla *variabilità 'between'*, cioè al modo di poter simulare la pendenza e la quota della retta di regressione pertinente ad uno degli ipotetici pazienti del gruppo `hfpv`. Vedete che l'intercetta  $a$  è caratterizzata da una deviazione standard 35.1, mentre la pendenza  $b$  ha una deviazione standard pari circa a 3.8. C'è di mezzo però anche un termine di correlazione, `Corr`,

0.22. Allora, faremo così. Chi di voi vorrà capire bene la questione, dovrà sciropparsi la lettura del successivo Intermezzo ectopico, inattuale ed abbastanza opzionale 6.3 (*abbastanza* opzionale, ma non *troppo*: in quanto lì si spiega perché la somma di due dromedari non fa un cammello, come molti autori di statistica medica invece sembrano suggerire). Chi invece si sente già sufficientemente soddisfatto, potrà saltare l'Intermezzo.

Comunque sia, tutti quanti adesso devono sottoporsi a questa attività obbligatoria. Troppo esigente? Su, dai, forza, dopo vi dò un bel voto – il professore Morrone mi raccomandava sempre di dare dei voti graduati, in modo da differenziare il rendimento di ciascuna/o dottoranda/o; ma io scantonavo cercando di cambiar discorso e magnificando al contempo la generale, uniformemente diffusa ed eccellente qualità degli iscritti ai corsi di dottorato della UMG. Una buona comprensione dell'esercizio che segue è a mio modo di vedere propedeutica per la comprensione del paragrafo dedicato alla rt-qPCR.

■ **Attività 6.1 — obbligatoria.** Durante il suo dottorato in Ingegneria Biomedica, Salvatore Scaramuzzino si è occupato nell'ambito dell'*imaging* diagnostico di una possibile classificazione automatica della severità delle patologie a livello di grossi vasi. Importate dall'indirizzo <http://www.biostatisticaumg.it/dataset/frattale.csv> il dataset *frattale* ed osservate che la risposta BC misurata rispetto al predittore DBC compare in trenta righe, ma i valori di StudyID sono in numero minore. Quanti? Si tratta dunque di un design di ricerca con misure ripetute. Tenete conto inoltre che per definizione se  $DBC = 0$  allora  $BC = 0$ , e viceversa; dunque, avrete bisogno di usare un modello senza intercetta:

```
ipotesi1 = BC ~ DBC - 1
ipotesi2 = BC ~ DBC - 1 + (DBC - 1 | StudyID)
```

Ora spiegate ed interpretate 'per filo e per segno' cosa ottenete dai comandi:

```
modello1 = lm(ipotesi1)
summary(modello1)
```

e cosa invece dai comandi:

```
library(lme4)
modello2 = lmer(ipotesi2)
summary(modello2)
```

In particolare spiegate perché gli errori standard dei due modelli sono diversi, spiegate cosa rappresentano le deviazioni standard degli effetti casuali, perché non appare la parola Corr nel summary, ed infine con rnorm fate qualche simulazione di dati ulteriori. ■

### 6.3 Intermezzo ectopico, inattuale ed (abbastanza) opzionale

*Persona che agisce e si comporta con eccessiva pedanteria, che nello svolgimento di un'attività mette, e soprattutto pretende dagli altri, una cura meticolosa, eccessiva, spesso inutile, anche nei particolari più insignificanti, che è rigidamente attaccata ai principi e ai regolamenti (...) che non sa liberarsi da schemi mentali rigidi e minuziosi.*

Pignoli. Noi matematici siamo persone pignole, in perfetto accordo con la definizione data dal Vocabolario Treccani. Ed è per questo che sento l'urgenza di raccontarvi cosa significano quelle due righe di output che abbiamo appena visto tra gli effetti casuali del summary(modello1).

```
...
Groups      Name      Variance Std.Dev. Corr
subject    (Intercept) 1232.12   35.102
            time        14.24    3.774   0.22
...

```

Solamente che, per farlo bene, avrò bisogno di parlare della distribuzione **normale bivariata**. Avrei potuto raccontarvelo nel secondo Capitolo, ma mi sembrava di appesantirlo più di quanto già esso non lo fosse. Poi, avrei potuto parlarvene dalle parti della Figura 4.2, ma se lo facevo avrei dovuto svelare un 'qualcosina di segreto' che Annalisa Di Cello deve tenere per sé fino al 2020, quando lei potrà rivelarvi i contenuti della domanda di Brevetto Industriale che è emersa in seguito ai suoi studi di dottorato. È andata a finire che ve la racconto qui, e lo faccio ricordando che:

1. Sergio Invernizzi [30] ha ripescato fuori una bella immagine del *living histogram*, <http://i.imgur.com/Fet8VWi.jpg>, in cui si intuisce che la somma di due distribuzioni normali non garantisce di ottenere una distribuzione normale.
2. Martin Bland [6] e Bernard Rosner [49] invece sembrano suggerire di sì, almeno ad una prima lettura dei loro testi.
3. Abbiamo dei controsensi matematici che ci fanno capire che no, non è così ma che 'proiettando' una normale bivariata lungo due assi cartesiani si ottiene ancora due normali univariate, ma non vale il viceversa.
4. Per generare una normale bivariata a partire da due normali univariate disponiamo di un codice R piuttosto semplice di tre righe, che coinvolgono la deviazione standard delle due normali univariate, e la correlazione che intercorre tra queste.

Ecco, vorrei dirvi tutto questo per spiegarvi quelle due righe degli effetti casuali del `summary` del `modellomisto1`, ma mi manca il tempo per scriverlo bene. Pazienza, lo farò prima che arrivi la stagione delle piogge. Lasciamo in sospeso questa sezione ed andiamo alla prossima.

## 6.4 La selezione di un modello ad effetti misti

Adesso salta fuori la questione delle Sezioni 4.5 e 5.2: se abbiamo due modelli ad effetti misti, come facciamo a scegliere il migliore? Possiamo affidarci anche in questo caso al Criterio di Informazione di Akaike? Dunque, la risposta è *ni*. Cioè, *sì*: potremo in certi casi utilizzare il criterio AIC, ma *no*: non senza dover fare qualche aggiustamento.

### 6.4.1 Selezione con il Criterio di Informazione di Akaike

Riprendiamo il dataset `percussive` e vediamo di comparare la relazione precedente `effettimisti1` con la nuova:

```
effettimisti1 = pafi ~ time * treatment + (time | subject)
effettimisti2 = pafi ~ time + treatment + (time | subject)
```

Se ricordate, si ripropone qui quanto già dicevamo a proposito della Figura 4.3: come facciamo a capire se nei due gruppi `hfpv` e `control` le pendenze siano effettivamente diverse, e quindi l'ossigenazione migliori con la ventilazione ad altra frequenza; oppure se i due gruppi abbiano la medesima pendenza (il che significherebbe che ventilando convenzionalmente o ad alta frequenza si ottengono i medesimi outcome; e questa quindi sarebbe la 'ipotesi nulla' che il lavoro di Umberto Lucangelo desidera confutare). Ora si starebbe poco a confrontare il `modellomisto1` precedente con il nuovo

```
modellomisto2 = lmer(effettimisti2)
summary(modellomisto2)
```

ma sarebbe *sbagliato* andare a comparare  $AIC(\text{modellomisto1})$  con  $AIC(\text{modellomisto2})$ . Per ragioni su cui non ci dilunghiamo (e vi rimandiamo ad esempio a [58] oppure a [23]) negli effetti fissi del modello gli errori standard vengono stimati con il metodo della massima verosimiglianza ristretta, 'REML'; mentre, per calcolare correttamente il criterio di informazione di Akaike occorrono le stime di massima verosimiglianza. Sono certamente parole molto tecniche, ma sappiate che non sono null'altro che una generalizzazione del concetto molto semplice che abbiamo introdotto nella sezione 1.2 a proposito della deviazione standard di una popolazione (nella quale si divideva per  $n$ ) o di un campione (in cui si divide per  $n - 1$ ). E che abbiamo re-incontrato nella sezione 4.1.5 quando la stima di massima verosimiglianza di  $\sigma$  aggiustava da 6.46 a 6.36 il valore della deviazione standard: tutto questo accadeva proprio perché si abbandonava il criterio REML di stima per adottare il criterio ML. Quindi ora con l'opzione `REML = FALSE` stimiamo due nuovi modelli:

```
modellomisto1ml = lmer(effettimisti1, REML = FALSE)
modellomisto2ml = lmer(effettimisti2, REML = FALSE)
```

e ne calcoliamo i criteri di informazione:

```
...
> AIC(modellomisto1ml, modellomisto2ml)
      df      AIC
modellomisto1ml  8 2782.380
modellomisto2ml  7 2813.826
...

```

Perfecto, decisione presa: i gruppi `hfpv` e `control` ossigenano in maniera differente con il passare del tempo. Attenzione però che ora i due modelli `modellomisto1ml` e `modellomisto2ml` non ci servono più: il risultato finale da considerare è e rimane il modello stimato con il criterio REML, ossia `modellomisto1`.

#### 6.4.2 Selezione con il 'parametric bootstrap'

Potrebbe invece accadere che si debba paragonare tra loro modelli che non differiscono negli effetti fissi, come `effettimisti1` e `effettimisti2`, ma invece possiedono una struttura diversa nella componente aleatoria. Ad esempio, ritorniamo al fatto che nella relazione `effettimisti1` gli effetti casuali che perturbano l'intercetta  $a$  e la pendenza  $b$  della retta della popolazione dando luogo alla retta di regressione del singolo paziente (variabilità 'between') sono legati da un termine di correlazione, 0.22:

```
...
Random effects:
Groups      Name        Variance Std.Dev. Corr
subject    (Intercept) 1232.12   35.102
            time         14.24    3.774   0.22
...

```

Visto che questo termine di correlazione potrebbe apparirci 'piccolo', ci chiediamo se non sia il caso di eliminarlo dalla struttura del nostro modello e risparmiare un parametro ridondante. Per fare ciò, abbiamo a disposizione due sintassi per specificare il modello: `time - 1`, oppure `time + 0`. Una vale l'altra, potete scegliere quella che vi piace di più. Io preferisco la prima, mi sembra più comprensibile. Dunque vogliamo decidere se adottare `effettimisti1` oppure `effettimisti3`:

```
effettimisti1 = pafi ~ time * treatment + (time | subject)
effettimisti3 = pafi ~ time * treatment + (time - 1|subject) + (1|subject)
```

Ora, come consuetudine, il summary:

```
modellomisto3 = lmer(effettimisti3)
summary(modellomisto3)

...
Random effects:
Groups      Name        Variance Std.Dev.
subject     (Intercept) 1325.60   36.409
subject.1   time        15.48    3.934
Residual                422.45   20.554
...
Fixed effects:
                     Estimate Std. Error t value
(Intercept)       177.0394   6.8061  26.012
time              6.9227   0.7701   8.989
treatmentcontrol -19.3949   9.6253  -2.015
time:treatmentcontrol -6.8405   1.0891  -6.281
...
```

Ovviamente le stime degli effetti fissi non mutano. Ma osservate adesso che non compare più la colonna che correlava gli effetti casuali della 'varianza between': da 35.1 e 3.8 siamo passati a 36.4 e 3.9.

Ma ecco le note dolenti. Per stabilire se sia opportuno scegliere modellomisto1 oppure modellomisto3 come modello minimale adeguato non possiamo più ricorrere all'utilizzo del Criterio di Informazione di Akaike, né a qualunque altro criterio di informazione. Ci sono delle ragioni ben precise, e mi ricordo che avevo provato a spiegarle durante una *Summer School* organizzata a Bologna dalla Associazione Nazionale dei Biotecnologi Italiani una decina di anni fa. Ora, ricordo che il mio corso iniziava verso le nove del mattino e finiva verso le diciotto, e dovevo trattare questo argomento verso le 15.30, proprio prima della coffee break. Siccome ricordo ancora le facce sgomentate e gli sbadigli dei poveri corsisti, 'pongo la questione di fiducia' e vi dico che effettivamente sarebbe stato proprio il modellomisto3 quello ad essere quello minimale adeguato. Se proprio volete sapere il come ed il perché, vi rimando alle Lecture Notes del corso ([https://dmi.units.it/pubblicazioni/Quaderni\\_Didattici/56\\_2011.pdf](https://dmi.units.it/pubblicazioni/Quaderni_Didattici/56_2011.pdf)) o meglio ancora, cercate sul testo di Julian Faraway [23] i dettagli del *parametric bootstrap*.

## 6.5 La Anova repeated measures, i modelli misti e la qRT-PCR

*in preparazione*

### Problema 6.1 ... ma come sarebbe a dire anova repeated measures

- esempio guida: densitometrie su 3 sedi del s.n.c. (brainstem, cerebellum e cortex)

La domanda di ricerca: visto che le densitometrie costano, devo per forza farle in tre sedi o potrei risparmiare facendole su due o su una? Ed in tal caso quale?

- importiamo densitometry (attenzione formato txt non csv)

```
www = "http://www.biostatisticaumg.it/dataset/densitometry.txt"
densitometry = read.table(www, header = TRUE)
attach(densitometry)
tail(densitometry)
# subject measure      region
```

- osservate che il dataset è sbilanciato. Questo sarebbe stato un grande problema per la old-fashioned 'one-way anova repeated measures'

```
table(subject)
```

Per comodità di scala passiamo alla radice quadrata:

```
intensity = sqrt(measure)
```

- cerchiamo di vedere la situazione

```
situazione = intensity ~ region
boxplot(situazione)
tapply(intensity, region, mean)
```

- discutiamo la analisi **sbagliata** ad effetti fissi: vedete che gli standard error sono uguali? non è logico.

```
sbagliato = lm(intensity ~ region)
summary(sbagliato)
```

- vedete che gli standard error sono uguali? non è logico.
- impostiamo il modello iniziale

```
rel1 = intensity ~ region + (region | subject)
```

- adesso facciamo il mixed model e discutiamo i coefficients

```
mixed1 = lmer(rel1)
summary(mixed1)
```

- vediamo che regioncortex -0.06 (s.e. 0.28) si confonde con lo zero, e quindi possiamo ipotizzare che le intensità della corteccia e del tronco encefalico brainstem si possano accomunare

```
regionStemCo = region
levels(regionStemCo)
levels(regionStemCo)[1]
levels(regionStemCo)[1] = "brainstemcortex"
levels(regionStemCo)
levels(regionStemCo)[3] = "brainstemcortex"
levels(regionStemCo)
```

- proviamo con mixed2

```
rel2 = intensity ~ regionStemCo + (regionStemCo | subject )
```

```
mixed2 = lmer(rel2)
summary(mixed2)
```

- Adesso si che cerebellum appare diverso da brainstemcortex 0.67 (s.e. 0.21) fa circa 3.2 deviate dall'origine, buon indizio.
- Per sicurezza, verifichiamo che mixed2 differisce dal modello nullo, nel quale basterebbe raccogliere dati solo su una delle tre sedi del s.n.c.

```
rel0 = intensity ~ 1 + (1 | subject )
mixed0ML = lmer(rel0, REML = FALSE)
mixed2ML = lmer(rel2, REML = FALSE)
AIC(mixed0ML, mixed2ML)
```

**Problema 6.2** ... ma insomma, come si fa a fare l'analisi dei dati della PCR?

E qui si apre un altro di quei mondi smisurati in cui si sono affastellate mille credenze e mille congetture negli ultimi venti anni, da quando Livak e Schmittgen [38] hanno proposto nel 2001 l'idea del 'Delta Delta  $C_T$ ' – e ne abbiamo parlato già nella sezione 1.4.1. A fronte di questa diffusione, non ha avuto altrettanta fortuna invece l'articolo di Juan Pedro Steibel e Rosangela Poletto, *A powerful and flexible linear mixed model framework for the analysis of relative quantification RT-PCR data*, seppure apparso su Genomics [51] nel 2009.

*in preparazione*

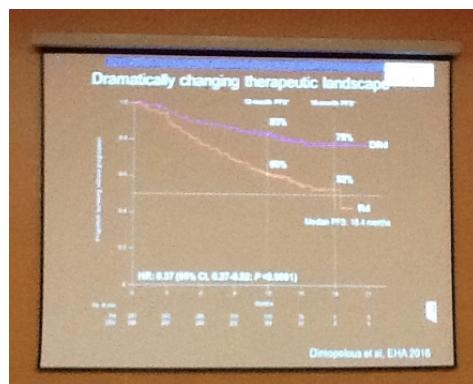
```
www = "http://www.biostatisticaumg.it/dataset/pcr.csv"
pcr = read.csv(www, header = TRUE)
str(pcr)
```

**6.6 Esercizi ed attività di approfondimento**



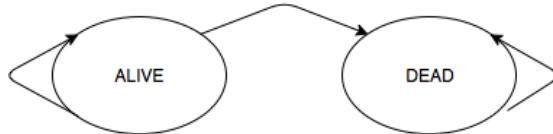
## 7. Sopravvivenza

Come avrete già visto dalla primissima pagina di questo libro, il professor Morrone si aggirava spessissimo per l'ateneo portandosi al collo una delle sue macchine fotografiche; mi ricordo anche che durante l'estate del 2016 in concomitanza con il corso di biostatistica per i dottorandi si tenne anche alla UMG il congresso internazionale *Multiple Myeloma 2016*, e mentre Gianni in piedi in prima fila in aula magna fotografava – giustamente – il Dottor Leif Bergsagel della Mayo Clinic, io in piccionaia fotografavo con il telefonino la prima slide che quest'ultimo presentava ai numerosi presenti; ed ero contentissimo perché all'indomani avrei dovuto trattare il discorso della analisi di sopravvivenza ed avrei potuto usare proprio questa immagine come motivazione:



### 7.1 I dati di tipo Time-to-Event

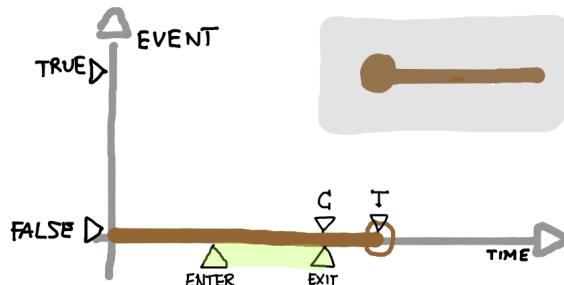
Abbiamo introdotto nella sezione 2.3.1 gli eventi dicotomici, che vengono modellati da un punto di vista probabilistico con le variabili aleatorie benoulliana o binomiale. Adattando questi concetti potremmo forse provare ad estenderli ad un **modello di sopravvivenza** molto semplice, a due stati, come schematizzato barbaramente in questa figura:



Dal punto di vista matematico ci interesserebbe poter valutare le **probabilità di transizione** tra gli stati di questo grafico – che prende il nome di **catena di Markov** – e che indichiamo con i  $P_{AA}, P_{AD}, P_{DD}, P_{DA}$ : ad esempio,  $P_{AD}$  rappresenta la probabilità di passare, diciamo domani, nello stato  $D$ , DEAD, sapendo che, diciamo oggi, siamo ancora nello stato  $A$ , ALIVE. Tecnicamente si dice che lo stato  $D$  è **assorbente** perché avvenuta la transizione in quello stato ahinoi non possiamo più ritornare nello stato antecedente:  $P_{DD} = 1, P_{DA} = 0$ . Ora, voi capite immediatamente che i modelli di questo genere saranno anche molto interessanti da studiare dal punto di vista probabilistico (e anche molto utili: la MCMC Monte Carlo Markov Chain è una tecnica usatissima nell'approccio bayesiano all'inferenza statistica), ma in questo frangente non ci sono di grande aiuto in quanto non tengono conto esplicitamente del 'tempo che passa'.

Nella **analisi di sopravvivenza** invece la variabile risposta che 'caratterizza' ciascun paziente è del tipo (`enter`, `exit`, `event`), laddove `enter` rappresenta il tempo (misurato secondo il *calendar time*) in cui inizia la nostra osservazione; `exit` quando finisce la nostra osservazione; infine la variabile logica `event` assume il valore `TRUE` qualora al tempo `exit` si sia manifestato l'evento di interesse (`DEAD`); altrimenti, essa assume il valore `FALSE`. Se decidiamo che per i nostri pazienti si stabilisce come 'tempo zero' il tempo di insorgenza di una determinata patologia, allora si dà per scontato che `enter` valga zero e la tripla si riduce ad una coppia (`time`, `event`).

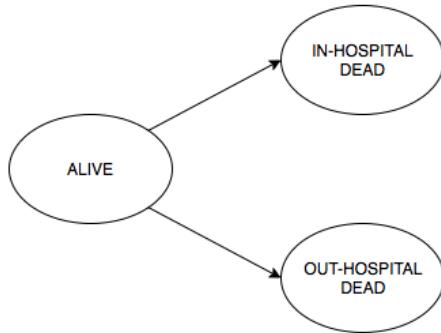
Abbiamo però due perplessità. Il primo problema è che con questa codifica, morire a 17 anni o morire a 71 anni rappresenterebbero due eventi equivalenti in questo contesto: non abbiamo un'evidenza specifica del 'tempo biologico'. Vedremo come risolvere questo problema. Il secondo problema: abbiamo detto che nella tripla (`enter`, `exit`, `event`) possiamo avere `event = FALSE` qualora il tempo conclusivo della nostra osservazione (`exit`) anticipi quello dell'evento morte. Parliamo dunque di **eventi censurati a destra**, come vediamo nell'immagine qui sotto: il tempo  $T$  in cui accade `event = TRUE` segue l'evento  $C$  (abbiamo  $T > C$ ) in cui registriamo `exit`; nell'intervallo di osservazione (`enter`, `exit`) la **funzione indicatrice** di color marrone mostra che `event = FALSE`. Vedremo come gestire, in maniera abbastanza soddisfacente, questa difficoltà.



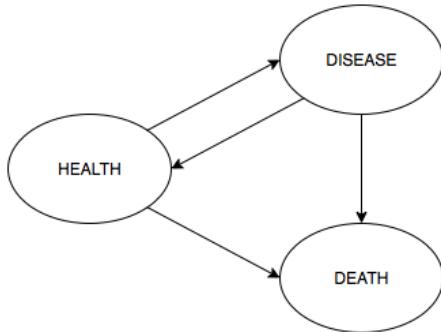
### 7.1.1 Rischio competitivo ed i modelli multistato

Modelli di sopravvivenza più avanzati sono ad esempio i modelli di **rischio competitivo**: ad esempio, il paziente può muoversi in due diversi stati assorbenti, rappresentati dalla mortalità intraed extra-ospedaliera. Nel secondo stato assorbente vi possono essere dati censurati, mentre nel

primo no – non è ragionevole supporre che la direzione ospedaliera ignori la data di decesso **exit** di un suo paziente.



Invece un **modello multistato** ad esempio è il cosiddetto *illness-death model with recovery*: il paziente può entrare dello stato assorbente DEATH da uno qualsiasi dei due stati precedenti, nei quali è possibile ritranciare per un numero qualsiasi di volte.



Che si tratti di un argomento corposo e difficilmente liquidabile in una decina di paginette lo si intuisce immediatamente dalle possibilità di analisi offerte da R:



Provate a dare un'occhiata a <https://cran.r-project.org/web/views/Survival.html> e capite cosa intendo per 'corposo'.

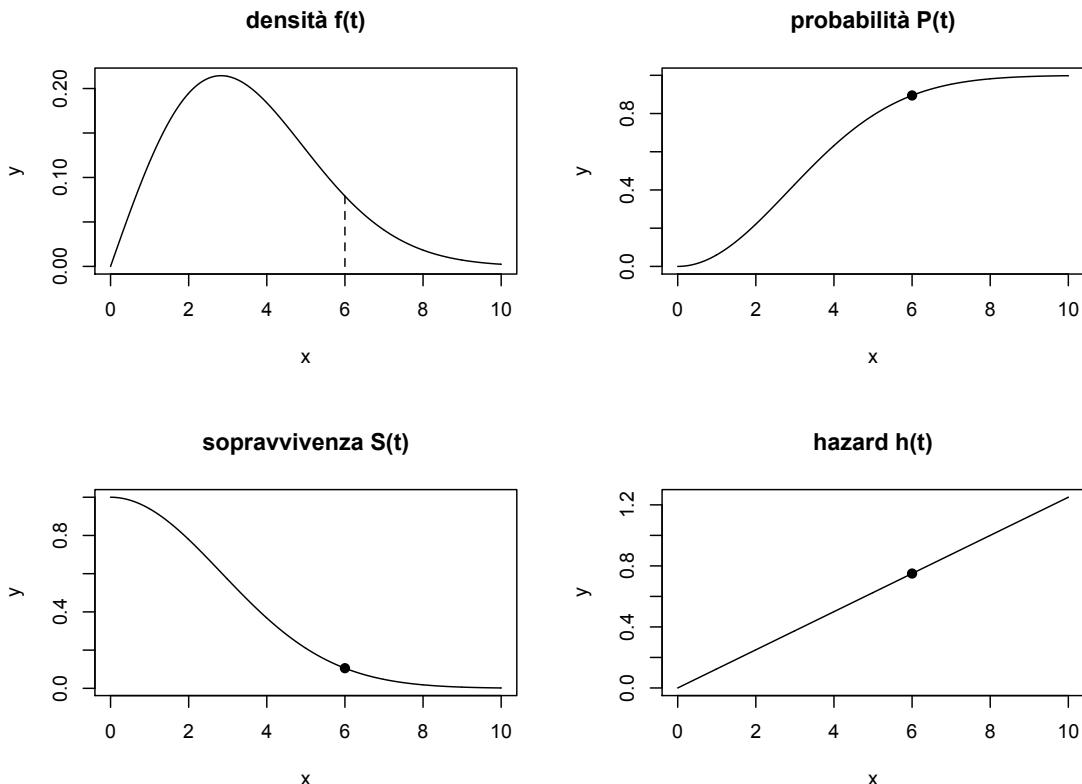
## 7.2 La cornice matematica

Ahi! Avete già capito che qui si tratta di rimboccarsi le maniche .. E già avete intuito che io non vi sgriderò più di tanto se ci rivediamo allegramente alla sezione 7.3, by-passando le prossime quattro pagine. L'analisi di sopravvivenza è 'matematicamente' più difficile da affrontare perché non è immediato introdurre, come invece facevamo nel modello lineare, il concetto di 'residuo', ossia quegli errori casuali che perturbano il modello e sulla base dei quali stabiliamo le misure di devianza o di verosimiglianza che consentono di selezionare due modelli tra di loro privilegiando il modello minimale adeguato: il solo fatto che coesistano tempi in cui abbiamo osservato l'evento e tempi (dati censurati) in cui non lo abbiamo osservato complica la faccenda in maniera soverchia. Ecco perché i `glm` non ci saranno più di grande aiuto.

### 7.2.1 Le definizioni

Matematicamente, ci sono di aiuto cinque funzioni, legate tra di loro, che descrivono in maniera equivalente il rischio di morte. Facciamo un esempio: mettiamo il caso che abbiamo seguito un certo

numero di pazienti ed abbiamo visto che l'outcome sfavorevole sopraggiunge molto frequentemente circa dopo tre anni l'insorgenza della patologia, e dopo circa in dieci anni quasi nessuno sopravvive (diciamo anni ma possiamo pensare a mesi, settimane o unità di tempo arbitrarie). Nella figura che segue, il pannello in alto a sinistra rappresenta la curva di **densità**  $f(t)$  (nel senso già introdotto nella sezione 2.3.4 a proposito della gaussiana) non è altro che una rappresentazione matematica dell'**istogramma** degli eventi. In quel disegno, il fatto che  $f(6) = 0.09$  dice che circa il 9 per cento del nostro gruppo di pazienti è deceduto 'in concomitanza' del sesto anno computato dall'istante dell'insorgenza della patologia.



Se ci chiediamo invece quale possa essere la percentuale di persone che hanno avuto outcome sfavorevole nei primi 6 anni, lo possiamo leggere in alto a destra nel grafico della **probabilità**  $P(t)$ : il pallino nero ci indica uno  $0.90 = 90\%$  di persone. Questo significa, riguardando il pannello in alto a sinistra, che nell'intervallo da 0 a 6 della funzione di densità è racchiusa un'area pari allo  $0.90 = 90\%$  dell'area totale di quella densità. In termini di integrale:

$$P(6) = \int_0^6 f(t)dt = 0.90$$

Se invece ci chiediamo quale possa essere la percentuale di persone che sopravviveranno ai primi 6 anni, lo leggiamo dal grafico della **sopravvivenza**  $S(t)$ , che è esattamente il grafico della probabilità  $P(t)$  'capovolto'; infatti, il pallino nero vale qui circa  $0.10 = 1 - 0.90$ . In simboli matematici:

$$S(6) = 1 - P(6) = 1 - \int_0^6 f(t)dt = 1 - 0.90 = 0.10$$

Per mezzo di queste tre funzioni, possiamo valutare una nuova quantità che chiamiamo 'rischio', **hazard**  $h(t)$ ; esso si può ottenere calcolando il rapporto  $f(t)/S(t)$ , quantificando la **probabilità**

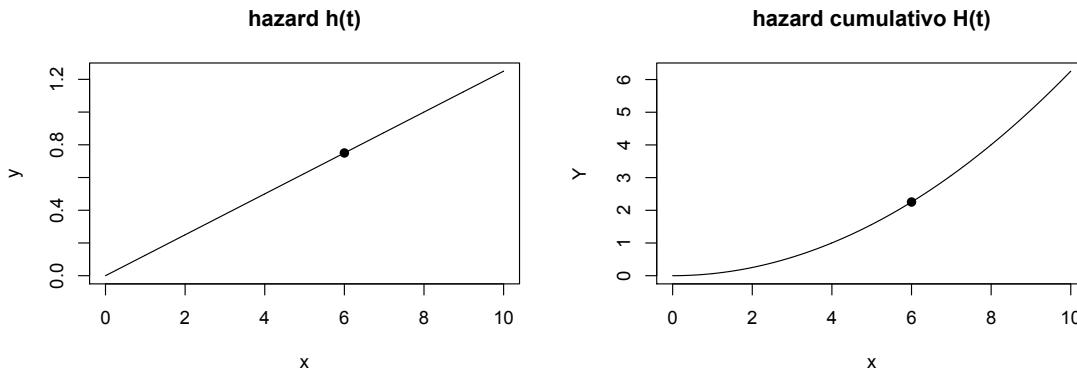
*istantanea* di morte al tempo  $t$ . Nel pannello in basso a destra, infatti,  $h(6)$  vale circa 0.8; ed abbiamo già visto che  $f(6)/S(6) = 0.08/0.10 = 0.8$ . In questo particolare esempio l'hazard assomiglia ad una retta poiché abbiamo scelto come densità una particolare distribuzione di dati  $f(t)$  (molto usata) che si chiama di Weibull, e che è una vera e propria variabile aleatoria che sarebbe dovuta apparire nel Capitolo 2.

Da ultimo potremmo anche considerare l'**hazard cumulativo**  $H(t)$ , che per l'hazard  $h(t)$  gioca il medesimo ruolo della probabilità  $P$  rispetto alla densità  $f$ :

$$H(6) = \int_0^6 h(t)dt$$

Nel disegno sottostante, il pannello a sinistra mostra che  $h(6) = 0.8$ ; pertanto, se calcoliamo l'area (= integrale) del triangolo di base 6 ed altezza 0.8 abbiamo:

$$\frac{6 \cdot h(6)}{2} = \frac{6 \cdot 0.8}{2} = 2.4 = H(6)$$



### 7.2.2 L'approccio non parametrico

Storicamente, il modo più agevole per affrontare l'analisi di sopravvivenza nei design di ricerca più semplici avveniva attraverso l'approccio non parametrico, evitando così di preoccuparsi se ad esempio la distribuzione esponenziale o la distribuzione di Weibull fossero adeguate a modellare i dati in nostro possesso. In altre parole, gli **stimatori non parametrici** della sopravvivenza  $S(t)$  venivano in aiuto allo statistico che non conosceva o non riusciva a fornire un'espressione analitica della 'vera legge matematica' che descrivesse la densità  $f(t)$  degli eventi:

- lo **stimatore di Nelson - Ålen**:  $\hat{H} = \sum_{s \leq t} \hat{h}(s)$
- lo **stimatore di Kaplan - Meier**:  $\hat{S} = \prod_{s < t} (1 - \hat{h}(s))$

Nella prossima sezione vedremo come fare per calcolare esplicitamente e visualizzare graficamente tutte queste quantità.

### 7.2.3 L'approccio parametrico

In certi casi fortunati abbiamo informazioni 'a priori' sui fenomeni su cui stiamo investigando che ci possono suggerire la legge matematica della densità  $f(t)$  degli eventi (come ad esempio capitava quando nella sezione 2.3.4 discutevamo dei fenomeni distribuiti in maniera gaussiana, o in maniera log-normale). Vediamo due possibilità favorevoli dal punto di vista matematico.

### La distribuzione esponenziale

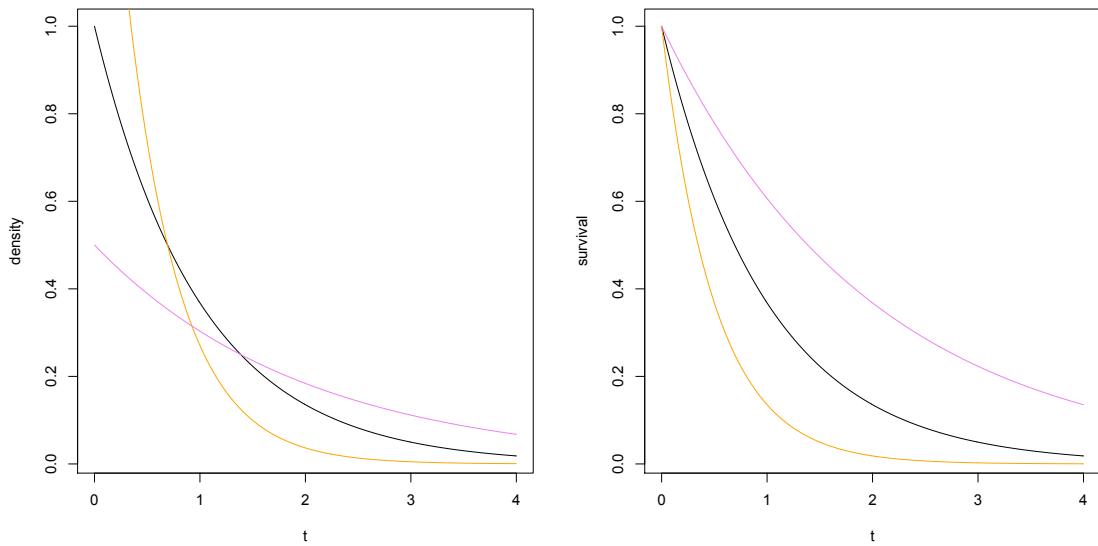
La distribuzione esponenziale è la più semplice di tutte le distribuzioni aleatorie utilizzate nell'analisi di sopravvivenza, ed è caratterizzata dal fatto che il tasso di rischio, hazard rate,  $\lambda > 0$  è costante; il che è un vantaggio matematico, ma è uno svantaggio biomedico. Il vantaggio matematico è che i calcoli per determinare  $P$ ,  $S$ ,  $h$  ed  $H$  diventano particolarmente facili da fare 'con carta e matita':

$$f(t; \lambda) = \lambda \exp(-\lambda t)$$

$$S(t; \lambda) = \exp(-\lambda t)$$

$$h(t; \lambda) = \lambda$$

Lo svantaggio biomedico sta nel fatto che, essendo il rate  $\lambda$  costante, un individuo giovane ed un individuo anziano avrebbero sempre la medesima probabilità di soccombere (ma c'è un 'trucco standard' per bypassare questa limitazione: suddividere 'piece-wise' le età in classi di rischio; l'hazard rimane una funzione discontinua, ma  $f$  ed  $S$  diventano continue - anche se non derivabili). Per disegnare la distribuzione esponenziale con R disponiamo dei comandi `dexp` e `pexp`. Ricordiamoci che essa è una legge che dipende da un solo parametro, come accade nella binomiale o nella Poisson del Capitolo 2.



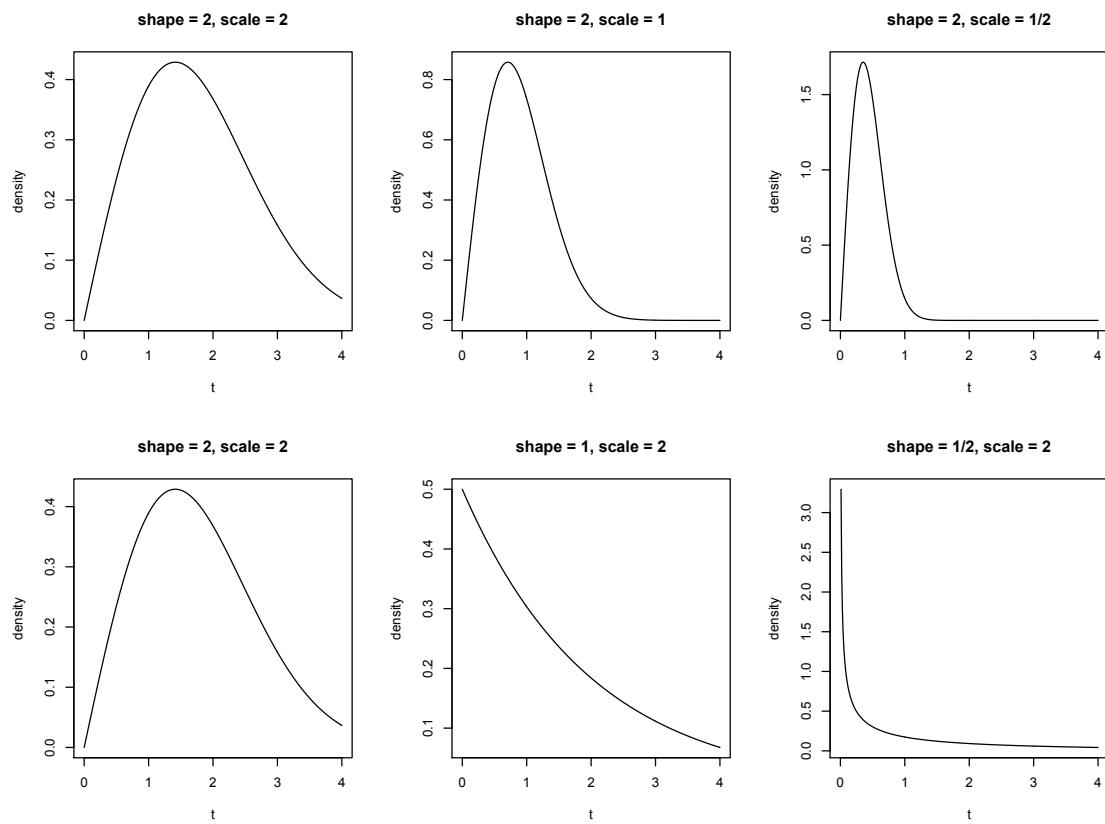
```
t = seq(from = 0, to = 4, by = 0.01)
density1 = dexp(t, rate = 1)
density2 = dexp(t, rate = 2)
density05 = dexp(t, rate = 0.5)
survival1 = 1 - pexp(t, rate = 1)
survival2 = 1 - pexp(t, rate = 2)
survival05 = 1 - pexp(t, rate = 0.5)
par(mfrow = c(1,2))
plot(t, density1, "l", ylab = "density")
lines(t, density2, col = "orange")
```

```
lines(t, density05, col = "violet")
plot(t, survival1, "l", ylab = "survival")
lines(t, survival2, col = "orange")
lines(t, survival05, col = "violet")
```

### La distribuzione di Weibull

Negli anni '50 il matematico svedese Waloddi Weibull descrisse le interessanti proprietà della distribuzione che porta il suo nome e che possiamo rappresentare usando i comandi di R `dweibull` e `pweibull`. Si tratta di una distribuzione caratterizzata da due parametri (analogamente alla gaussiana, che ha un *parametro di centralità*, la media  $\mu$ , e un *parametro di forma*, la deviazione standard  $\sigma$ ):

- il *parametro di forma* (`shape`),  $p > 0$
- il *parametro di scala* (`scale`),  $\lambda > 0$



```
t = seq(from = 0, to = 4, by = 0.01)
par(mfrow = c(2,3))
density22 = dweibull(t, shape = 2, scale = 2)
plot(t, density22, "l", ylab = "density", main = "shape = 2, scale = 2")
density21 = dweibull(t, shape = 2, scale = 1)
plot(t, density21, "l", ylab = "density", main = "shape = 2, scale = 1")
density205 = dweibull(t, shape = 2, scale = 1/2)
plot(t, density205, "l", ylab = "density", main = "shape = 2, scale = 1/2")
density22 = dweibull(t, shape = 2, scale = 2)
plot(t, density22, "l", ylab = "density", main = "shape = 2, scale = 2")
```

```

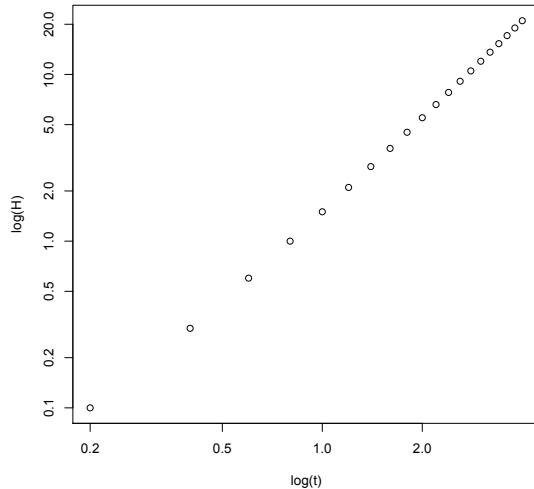
density12 = dweibull(t, shape = 1, scale = 2)
plot(t, density12, "l", ylab = "density", main = "shape = 1, scale = 2")
density052 = dweibull(t, shape = 1/2, scale = 2)
plot(t, density052, "l", ylab = "density", main = "shape = 1/2, scale = 2")

```

Purtroppo, c'è una po' di confusione in giro tra i libri ed i software. Anche perché il parametro di scala  $\lambda > 0$  della Weibull usa la stessa lettera greca  $\lambda$  in uso nella esponenziale (ma in quel caso  $\lambda$  non è la scala, è il **rate**; la scala, in quel caso, sarebbe  $1/\lambda$  (che pasticcio!) In ogni caso, la funzione di rischio di una Weibull è:

$$h(t; p, \lambda) = \frac{p}{\lambda} \left( \frac{t}{\lambda} \right)^{(p-1)}$$

La distribuzione di Weibull, tra le altre belle proprietà matematiche, è caratterizzata dal fatto che - su scale logaritmiche - il rischio cumulativo  $H(t)$  ed il tempo si comportano in modo lineare; ossia il grafico di  $\log(H(t))$  e  $\log(t)$  è una retta. Ecco quindi che se ci affidiamo ad un approccio non parametrico, e consideriamo lo stimatore  $\hat{H}$  di Nelson - Ålen, possiamo visualmente apprezzare se il fenomeno che stiamo studiando è ben modellato da una distribuzione di Weibull (o esponenziale, che è una Weibull con  $p = 1$ ).



```

t = seq(from = 0, to = 4, by = 0.2)
dens = dweibull(t, shape = 2, scale = 2)
Prob = pweibull(t, shape = 2, scale = 2)
Surv = 1 - Prob
haz = dens/Surv
Haz = cumsum(haz)
plot(t, Haz, log = "xy", xlab = "log(t)", ylab = "log(H)")

```

### 7.3 Le curve di sopravvivenza con R

Vi sono molti pacchetti di R che consentono di ottenere stime di base di sopravvivenza e delle loro relative rappresentazioni, ed il più noto è sicuramente **survival**[54] – ma devo dire che ho trovato anche di interesse il pacchetto **ehaed** il libro relativo, [9], da cui queste note attingono a man

bassa. Come esempio-guida sceglieremo il dataset addome relativo a 75 pazienti oncologici di cui conosciamo l'Eta all'intervento in cui furono trattati con una determinata Tecnica chirurgica (open oppure laparo). Ci interessa stimare la sopravvivenza espressa in Mesi, essendo la variabile logica Statoattuale l'indicatore dei dati censurati. Importiamo in R il dataset, e diamo un'occhiata a cinque pazienti scelti casualmente:

```
library(survival)
url = "http://www.biostatisticaumg.it/dataset/addome.csv"
addome = read.csv(url, header = TRUE)
attach(addome)
names(addome)
head(addome)
str(addome)
addome[sample(1:75, 5),]

...
> addome[sample(1:75, 5),]
  Mesi Statoattuale Tecnica Eta
62    15            1   open  88
31     2            0   open  59
75    58            1   open  86
32    37            0   open  50
11   168            0  laparo 65
```

### 7.3.1 L'approccio non parametrico

Scopriamo innanzitutto a cosa serve la funzione Surv:

```
Mesi
Surv(Mesi, Statoattuale)

> Mesi
[1] 108  39  94  23  78 114  14  77    1  83 168  90  87 184
...
[71]  43  25  34  17   58
> Surv(Mesi, Statoattuale)
[1] 108+ 39+ 94+ 23    78+ 114+ 14+ 77+    1+ 83+ 168+ 90+ 87+ 184+
...
[71]  43   25   34   17   58
```

Come vedete Surv giustappone un segno positivo ai dati censurati a destra; per esempio il primo paziente dopo 108 mesi di osservazione risultava ancora in vita, mentre al 58-esimo mese di osservazione constatavamo il decesso del 75-esimo paziente.

Iniziamo ora ad investigare sul 'modello nullo' di sopravvivenza, mediante il comando survfit ed interpretiamo cosa succede riordinando dal minore al maggiore i tempi di sopravvivenza:

```
relazionenulla = Surv(Mesi, Statoattuale) ~ 1
km = survfit(relazionenulla)
sort(Surv(Mesi, Statoattuale))
summary(km)
```

```

...
[1]   1+   2+   3+   4+   4+   5+ 10+ 10+ 14+ 15+ 15  15  15  16+
[15] 17   20+  20+  21   23+  23   23   24   25   28+  29+  30+  30+  30+
[29] 33   34+  34   35+  37+  38   38   39+  40+  41   43   48   48   49
[43] 52+  58   58   63   64+  65+  69   74+  77+  78+  82+  83+  87+  90+
[57] 93+  94+  97+ 102+ 107+ 108+ 109+ 109  113+ 114+ 119+ 132+ 133+ 133+
[71] 147+ 155+ 157+ 168+ 184+
...
time n.risk n.event survival std.err lower 95%\% CI upper 95%\% CI
 15     66      3    0.955  0.0256      0.906    1.000
 17     61      1    0.939  0.0296      0.883    0.999
 21     58      1    0.923  0.0332      0.860    0.990
...
 63     30      1    0.615  0.0674      0.496    0.763
 69     27      1    0.592  0.0687      0.472    0.744
109     13      1    0.547  0.0770      0.415    0.721

```

Come vedete, nei 14 mesi iniziale tutti i pazienti sopravvivono, ma a partire dal 15-esimo mese si registrano `n.risk = 3` decessi. Quindi dei 75 pazienti iniziali i primi 9 sono sopravvissuti ed `n.risk = 75 - 9 = 66` pazienti risultano essere a rischio di morte all'inizio del quindicesimo mese. Mentre, all'inizio del 109-esimo mese avremo 13 persone viventi a rischio ed un decesso.

La colonna `survival` che stima con il metodo di Kaplan e Meier la funzione  $S(t)$  si calcola moltiplicando di volta in volta fra di essi:

$$\frac{n.risk - n.event}{n.risk}$$

sicché  $0.955 = (66 - 3)/66$ , mentre  $0.939 = 0.955 \cdot (61 - 1)/61$  e così via. Osservate anche un fenomeno tipico delle curve di sopravvivenza: siccome con il trascorrere del tempo il numero dei sopravviventi decresce, l'inaffidabilità delle stime aumenta; ecco perché la colonna `std.err` via via va aumentando.

### 7.3.2 L'approccio parametrico

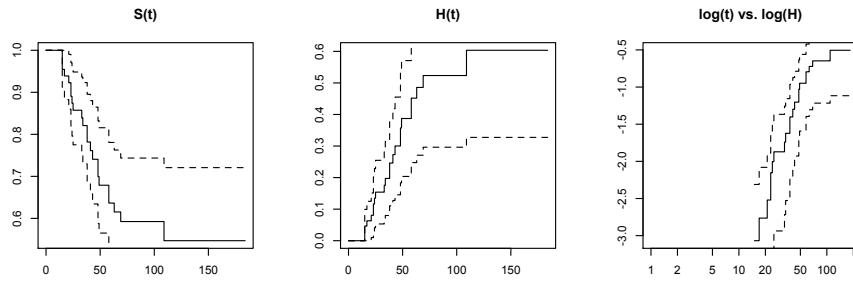
Come annunciato nel precedente paragrafo, possiamo valutare 'ad occhio' se i nostri dati si conformano ad una distribuzione di Weibull se si dispongono approssimativamente in maniera rettilinea su scala logaritmica nel piano  $(t, \hat{H}(t))$ . Nel grafico seguente abbiamo rappresentato da sinistra a destra rispettivamente le stime della funzione di sopravvivenza  $S(t)$  (secondo Kaplan e Meier), del rischio cumulato  $H(t)$  (secondo Nelson ed Ålen) e della trasformazione logaritmica di quest'ultimo rispetto alla scala logaritmica dei tempi. Abbiamo detto alla fine della sezione 7.2.3 che proprio quest'ultimo grafico ci consente di valutare visivamente se, tramite un'andamento rettilineo del grafico, i dati si conformino alla distribuzione aleatoria di Weibull.

```

par(mfrow = c(1,3))
plot(km, fun = "surv", main = "S(t)")
plot(km, fun = "cumhaz", main = "H(t)")
plot(km, fun = "cloglog", main = "log(t) vs. log(H)")

```

Con qualche riserva, possiamo provare ad illuderci che quel grafico a destra sia rettilineo, e quindi ipotizzare che la distribuzione di Weibull possa fare al caso nostro e sia adeguata a rappresentare i dati. Per stimare i parametri di forma  $p$  e di scala  $\lambda$  della distribuzione di Weibull, ci conviene far riferimento al comando `phreg` del pacchetto `eha`[9]:

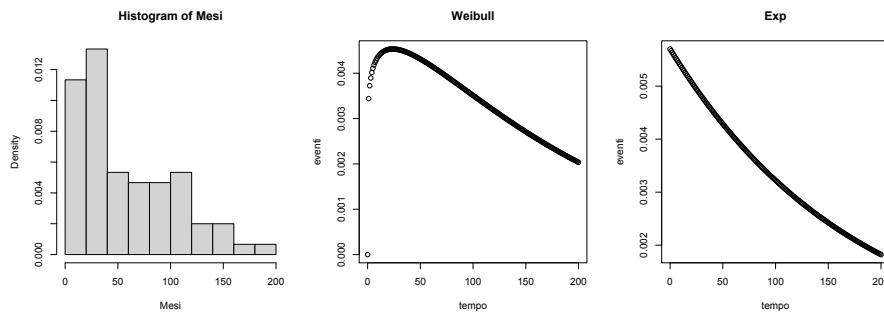


```
install.packages(eha)
library(eha)
fitWeibull = phreg(relazionenulla, data = addome)
fitWeibull

...
Covariate      W.mean      Coef Exp(Coef)  se(Coef)  Wald p
log(scale)      5.163          0.224    0.000
log(shape)      0.114          0.170    0.503
```

Senza dubbio prendiamo in debita considerazione il parametro  $\lambda = \exp(5.163) = 175$ . Al contrario, il fatto che  $p = \exp(0.114) = 1.12$  sia sospettosamente vicino ad uno, e che l'errore standard 0.17 sopravanza la stima del coefficiente 0.11, ci fa pensare di rinunciare a considerare la distribuzione di Weibull ma di considerare la semplice distribuzione esponenziale di tasso  $1/\lambda = 1/175 = 0.0057$ . Ritireremo su questo argomento in seguito.

```
par(mfrow = c(1,3))
hist(Mesi, freq = FALSE, col = "lightgrey")
tempo = 0:200
eventi = dweibull(tempo, shape = 1.12, scale = 174.68)
plot(tempo, eventi, main = "Weibull")
1/exp(fitWeibull[1]$coefficients[[1]]) # 0.005726756
eventi = dexp(tempo, rate = 0.0057)
plot(tempo, eventi, main = "Exp")
```



**Esercizio 7.1** Provate a verificare graficamente con una, o più, simulazioni aleatorie se i Mesi si distribuiscono con una densità esponenziale di tasso 0.0057. Rimarrete forse delusi della non somiglianza, e probabilmente rinuncerete all'idea di affidarvi ad un modello parametrico per studiare il dataset addome.

```
Mesisimulati = trunc(rexp(75, rate = 0.005727))

relazionenulla = Surv(Mesi, Statoattuale) ~ 1
relacionesimula = Surv(Mesisimulati, Statoattuale) ~ 1

km = survfit(relazionenulla)
kmsimula = survfit(relacionesimula)

par(mfrow = c(1,2))
plot(km)
plot(kmsimula, xlim = c(0, max(Mesi)))
```



## 7.4 I modelli semiparametrici

Il 15 luglio scorso, il leggendario statistico Sir David Cox ha compiuto 95 anni. Ad egli dobbiamo l'intuizione di un modo, semplice nel suo concetto ma molto efficace, per stabilire se vi sia una 'differenza di rischio' tra diversi gruppi di una medesima popolazione – senza dover conoscere necessariamente il rischio di ciascuno dei gruppi. Si parla appunto di modelli semiparametrici perché non saremo in grado di stimare 'chi sia' il **baseline hazard**, la funzione di rischio del gruppo di riferimento (i.e. i controlli, od il primo livello in ordine alfabetico), ma diremo di quanto si aggrava proporzionalmente questo rischio nel secondo gruppo.

### 7.4.1 L'aspetto matematico

Un po' di matematica. Indichiamo con  $h_L(t)$  la funzione di rischio del gruppo laparo, con  $h_O(t)$  quella del gruppo open. Diremo che i due gruppi hanno un rischio proporzionale se riusciamo a determinare un numero costante  $\psi > 0$ , indipendente dal tempo  $t$ , per cui accada che:

$$h_O(t) = \psi \cdot h_L(t)$$

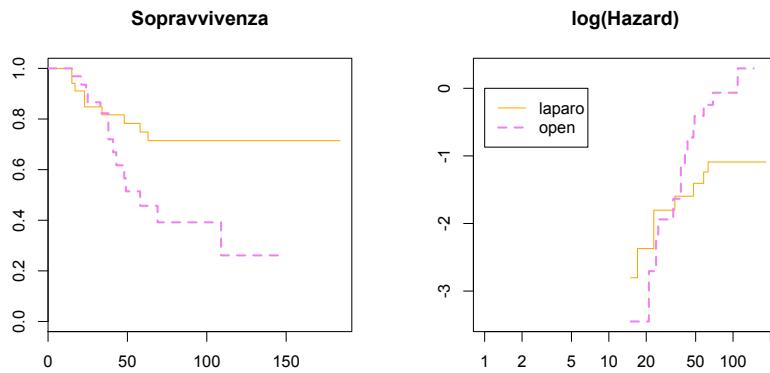
Il primo trucco ingegnoso è quello di considerare il logaritmo di  $\psi$ ,  $\beta = \log(\psi)$  e scrivere:

$$h_O(t) = \exp(\beta) \cdot h_L(t)$$

Ora, non è affatto detto che in ogni caso i rischi di due gruppi siano proporzionali; è un'ipotesi che va verificata controllando – almeno 'ad occhio' – il parallelismo degli stimatori di Nelson Ålen disegnati su scala log-log. E qui, diremmo proprio che le cose non funzionano come vogliamo: non siamo in presenza di rischi proporzionali tra i due gruppi perché nel pannello di destra l'andamento è forse rettilineo (i.e. la distribuzione di Weibull potrebbe andar bene) ma le rette non sono parallele, e non possiamo dire che tra loro due ci sia una distanza costante  $\beta = \log(\psi)$ .

Per conoscere i valori di  $\beta$  e di  $\psi = \exp(\beta)$  avremmo potuto fare riferimento ai valori `coef` ed `exp(coef)` proposti dalla funzione 'Cox proportional hazards' `coxph` del pacchetto `survival`:

```
relazioneT = Surv(Mesi, Statoattuale) ~ Tecnica
modelloco1 = coxph(relazioneT, data = addome)
```



modellocox1

```
...
      coef  exp(coef)  se(coef)      z      p
Tecnicaopen 0.9108     2.4862   0.4354  2.092 0.0365
...

```

L'idea ingegnosa di Cox ci avrebbe innanzitutto permesso di quantificare il rischio del gruppo *laparo* rispetto a quello *open* in questo modo:

$$h_L(t) = \exp(\beta) \cdot h_O(t) = 2.48 \cdot h_O(t)$$

ma ancor meglio, definendo in maniera astratta la funzione di rischio  $h(t; x)$ :

$$h(t; x) = \exp(\beta x) \cdot h_O(t)$$

che per  $x = 0$  ed  $x = 1$  restituisce rispettivamente gli hazard dei due gruppi:

$$h(t; 0) = \exp(\beta \cdot 0) \cdot h_O(t) = 1 \cdot h_O(t) = h_O(t)$$

$$h(t; 1) = \exp(\beta \cdot 1) \cdot h_O(t) = \exp(\beta) \cdot h_O(t) = h_L(t)$$

Pensando perciò alla  $\mathbf{x}$  come ad un vettore di covariate, il termine esponenziale  $\exp(\beta \mathbf{x})$  consente nella *regressione di Cox* di descrivere, in maniera moltiplicativa, la differenza di rischio tra due (o più) gruppi in un campione introducendo una (o più) covariate  $\mathbf{x}$ . In parole semplici, riusciremmo a determinare – con metodi che ci ricordano i modelli lineari generalizzati e le loro funzioni di collegamento – ad esempio i coefficienti  $a$  e  $b$  di una regressione del tipo:

$$\frac{h_L(t)}{h_O(t)} = \exp(a + b \cdot x)$$

valutando quindi amplificazione o la riduzione del rapporto tra i rischi di due gruppi.

Tuttavia dubitiamo della bontà della stima del  $\text{coef} = \beta = 0.911$ , che condurrebbe ad una costante di proporzionalità  $\exp(\text{coef}) = \psi = 2.486$  stante il fatto che i rischi non ci sono apparsi proporzionali nel grafico della pagina precedente.

### 7.4.2 Interpretare un modello di Cox

Proponiamoci di interpretare in un modello di Cox quale effetto potrebbe avere valutare l'Eta del paziente oltre che sapere con quale Tecnica sia stato operato per stimare il rischio di decesso.

```
relazioneTE = Surv(Mesi, Statoattuale) ~ Tecnica + Eta
modellocox2 = coxph(relazioneTE, data = addome)
modellocox2

...
            coef  exp(coef)  se(coef)      z      p
Tecnicaopen 0.59613    1.81508  0.43818  1.360  0.173681
Eta          0.06617    1.06841  0.01734  3.815  0.000136
...
```

Innanzitutto seguiamo sempre l'**ordine alfabetico** e quindi nella covariata Tecnica si assume che laparo è il riferimento ed open è il livello successivo. Pertanto  $h_O(t) = 1.815 \cdot h_L(t)$ . Come dicevamo, non siamo in grado di dire precisamente quale sia il rischio ad un certo tempo  $t$  per un paziente del gruppo laparo,  $h_L(t)$ ; ma sappiamo che un paziente con le medesime caratteristiche ma del gruppo open ha un rischio 1.815 volte superiore (ossia di circa l'82 per cento in più). Questo fatto è caratteristico appunto dei modelli di **tipo semiparametrico**. Analogamente, non siamo in grado di dire quale sia il rischio  $h(t;N)$  per un paziente di  $N$  anni di età, ma sappiamo che per un paziente con le medesime caratteristiche ma più anziano di un anno avremo che:

$$h(t;N+1) = 1.068 \cdot h(t;N)$$

ossia un aumento di rischio di circa il 7% per ogni anno di età in più.

### 7.4.3 Selezione del modello

Proviamo per esercizio a stabilire quale possa essere il migliore modello, scegliendo tra i seguenti predittori:

```
relazionenulla = Surv(Mesi, Statoattuale) ~ 1
relazioneTE = Surv(Mesi, Statoattuale) ~ Tecnica + Eta
relazioneT = Surv(Mesi, Statoattuale) ~ Tecnica
relazioneE = Surv(Mesi, Statoattuale) ~ Eta
relazioneTEcross = Surv(Mesi, Statoattuale) ~ Tecnica * Eta
```

Stimiamo i parametri dei modelli:

```
modellocox0 = coxph(relazionenulla, data = addome)
modellocox1 = coxph(relazioneT, data = addome)
modellocox2 = coxph(relazioneTE, data = addome)
modellocox3 = coxph(relazioneE, data = addome)
modellocox4 = coxph(relazioneTEcross, data = addome)
```

e mediante la funzione `extractAIC`, nella quale tra l'altro vengono determinati i gradi di libertà equivalenti (edf, equivalent degrees of freedom for the fitted model) del modello, scegliamo il modello minimale adeguato, che risulta essere il `modellocox3` nel quale la sopravvivenza è legata esclusivamente alla Eta

```
> extractAIC(modellocox0)
[1] 0.0000 171.8548
> extractAIC(modellocox1)
[1] 1.000 169.338
> extractAIC(modellocox3)
[1] 1.0000 150.9064
> extractAIC(modellocox2)
[1] 2.0000 151.0026
> extractAIC(modellocox4)
[1] 3.0000 152.2792
```

**Esercizio 7.2** Riprendete il discorso dei 'termini di curvatura' della sezione 4.5, quarto step, ed introducetelo nella relazioneE. Ne vale la pena? ■

## 7.5 Esercizi ed attività di approfondimento

■ **Attività 7.1 — melanoma.** Prendete in esame il dataset melanoma di Giusy Schipani, pubblicato all'indirizzo <http://www.biostatisticaumg.it/dataset/melanoma.csv>. Osservate che nascita, exeresimelanoma ed ultimocontrollo sono date 'giuliane' (o più precisamente, misurate in giorni a partire dal primo gennaio del 1900).

- Calcolate il tempo di sopravvivenza dopo l'intervento chirurgico.
- Il Sesso e lo SpessoreBreslow della lesione sono dei predittori della sopravvivenza? Individuate il modello minimale adeguato ed interpretatelo, fornendo anche un grafico opportuno. ■





## Bibliografia

### Articoli

- [1] Hirotugu Akaike. "A new look at the statistical model identification". In: *Automatic Control, IEEE Transactions on* 19.6 (1974), pagine 716–723 (citato alle pagine 80, 93).
- [3] Edgar Anderson. "The species problem in Iris". In: *Annals of the Missouri Botanical Garden* 23.3 (1936), pagine 457–509 (citato a pagina 17).
- [4] Francis J Anscombe. "Graphs in statistical analysis". In: *The American Statistician* 27.1 (1973), pagine 17–21 (citato a pagina 94).
- [5] Douglas Bates et al. "Fitting Linear Mixed-Effects Models Using lme4". In: *Journal of Statistical Software* 67.1 (2015), pagine 1–48. DOI: 10.18637/jss.v067.i01 (citato a pagina 141).
- [7] Heather M Bond et al. "Early hematopoietic zinc finger protein (EHZF), the human homolog to mouse Evi3, is highly expressed in primitive human hematopoietic cells". In: *Blood* 103.6 (2004), pagine 2062–2070 (citato alle pagine 7, 35).
- [11] T Calimeri et al. "A unique three-dimensional SCID-polymeric scaffold (SCID-synth-hu) model for in vivo expansion of human primary multiple myeloma cells". In: *Leukemia* 25.4 (2011), pagine 707–712 (citato alle pagine 118, 137).
- [12] Carmelo Capula et al. "A new predictive tool for the early risk assessment of gestational diabetes mellitus". In: *Primary Care Diabetes* 10.5 (2016). PMID: 27268754, pagine 315–323 (citato a pagina 128).
- [13] Maria Vittoria Caruso, Attilio Renzulli e Gionata Fragomeni. "Influence of IABP-Induced abdominal occlusions on aortic hemodynamics: a patient-specific computational evaluation". In: *ASAIO Journal* 63.2 (2017), pagine 161–167 (citato alle pagine 15, 20).
- [14] Emanuela Chiarella et al. "ZNF521 Represses Osteoblastic Differentiation in Human Adipose-Derived Stem Cells". In: *International journal of molecular sciences* 19.12 (2018), pagina 4095 (citato alle pagine 15, 31, 137).

- [17] Maria Teresa De Angelis et al. “Short-term retinoic acid treatment sustains pluripotency and suppresses differentiation of human induced pluripotent stem cells”. In: *Cell death & disease* 9.1 (2018), pagina 6 (citato alle pagine 15, 29, 137).
- [18] Janez Demšar e Blaž Zupan. “Orange: Data mining fruitful and fun-a historical perspective”. In: *Informatica* 37.1 (2013) (citato a pagina 7).
- [19] Annalisa Di Cello et al. “A more accurate method to interpret lactate dehydrogenase (LDH) isoenzymes’ results in patients with uterine masses”. In: *European Journal of Obstetrics and Gynecology and Reproductive Biology* in press (2019). DOI: 10.1016/j.ejogrb.2019.03.017. URL: 10.1016/j.ejogrb.2019.03.017 (citato alle pagine 15, 48, 70, 83, 121, 133).
- [24] Ronald A Fisher. “The use of multiple measurements in taxonomic problems”. In: *Annals of eugenics* 7.2 (1936), pagine 179–188 (citato a pagina 17).
- [27] Francis Galton. “Regression towards mediocrity in hereditary stature.” In: *The Journal of the Anthropological Institute of Great Britain and Ireland* 15 (1886), pagine 246–263 (citato a pagina 79).
- [28] Richard Horton. “Offline: what is medicine’s 5 sigma?” In: *The Lancet* 385.9976 (2015), pagina 1380 (citato a pagina 77).
- [31] John PA Ioannidis. “Why most published research findings are false”. In: *PLoS medicine* 2.8 (2005), e124 (citato a pagina 77).
- [32] John PA Ioannidis. “The reproducibility wars: successful, unsuccessful, uninterpretable, exact, conceptual, triangulated, contested replication”. In: *Clinical chemistry* 63.5 (2017), pagine 943–945 (citato a pagina 6).
- [36] Nan M Laird, James H Ware et al. “Random-effects models for longitudinal data”. In: *Biometrics* 38.4 (1982), pagine 963–974 (citato a pagina 141).
- [37] Eckhard Limpert, Werner A. Stahel e Markus Abbt. “Log-normal Distributions across the Sciences: Keys and Clues”. In: *BioScience* 51.5 (2001), pagine 341–352 (citato a pagina 59).
- [38] Kenneth J Livak e Thomas D Schmittgen. “Analysis of relative gene expression data using real-time quantitative PCR and the 2-  $\Delta\Delta CT$  method”. In: *methods* 25.4 (2001), pagine 402–408 (citato alle pagine 30, 153).
- [39] Umberto Lucangelo et al. “Early short-term application of high-frequency percussive ventilation improves gas exchange in hypoxicemic patients”. In: *Respiration* 84.5 (2012), pagine 369–376 (citato alle pagine 141, 144, 145).
- [40] Richard G Moore et al. “Comparison of a novel multiple marker assay vs the Risk of Malignancy Index for the prediction of epithelial ovarian cancer in patients with a pelvic mass”. In: *American journal of obstetrics and gynecology* 203.3 (2010), 228–e1 (citato alle pagine 129–131).
- [41] John Ashworth Nelder e Robert WM Wedderburn. “Generalized linear models”. In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pagine 370–384 (citato a pagina 122).
- [42] Kersti Pärna et al. “Alcohol consumption in Estonia and Finland: Finbalt survey 1994-2006”. In: *BMC Public Health* 10.1 (2010), pagina 261 (citato a pagina 56).
- [44] Jose Pinheiro et al. “nlme: Linear and Nonlinear Mixed Effects Models”. In: (2019) (citato a pagina 141).
- [46] Xavier Robin et al. “pROC: an open-source package for R and S+ to analyze and compare ROC curves”. In: *BMC Bioinformatics* 12 (2011), pagina 77 (citato a pagina 133).

- [50] T. Sing et al. “ROCR: visualizing classifier performance in R”. In: *Bioinformatics* 21.20 (2005), pagina 7881. URL: <http://rocr.bioinf.mpi-sb.mpg.de> (citato a pagina 133).
- [51] Juan Pedro Steibel et al. “A powerful and flexible linear mixed model framework for the analysis of relative quantification RT-PCR data”. In: *Genomics* 94.2 (2009), pagine 146–152 (citato a pagina 153).
- [53] Student. “The probable error of a mean”. In: *Biometrika* (1908), pagine 1–25 (citato alle pagine 71, 73).
- [55] Nick Thieme. “R generation”. In: *Significance* 15.4 (2018), pagine 14–19 (citato a pagina 6).
- [57] Roberta Venturella et al. “Three to five years later: long-term effects on ovarian function of prophylactic bilateral salpingectomy”. In: *Journal of Minimally Invasive Gynecology* 24.1 (2017). PMID: 27621194, pagine 145–150 (citato alle pagine 15, 120).
- [59] Howard Wainer. “The most dangerous equation”. In: *American Scientist* 95.3 (2007), pagina 249 (citato a pagina 31).
- [62] GN Wilkinson e CE Rogers. “Symbolic description of factorial models for analysis of variance”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 22.3 (1973), pagine 392–399 (citato alle pagine 84, 90, 110).
- [63] P Zaffino et al. “Radiotherapy of Hodgkin and non-Hodgkin lymphoma: A nonrigid image-based registration method for automatic localization of prechemotherapy gross tumor volume”. In: *Technology in cancer research & treatment* 15.2 (2016), pagine 355–364 (citato alle pagine 16, 20).
- [64] Stephen T Ziliak e Deirdre N McCloskey. “The cult of statistical significance”. In: *Ann Arbor: University of Michigan Press* 27 (2008) (citato a pagina 77).

## Libri

- [6] Martin Bland. *An introduction to medical statistics*. Ed. 3. Oxford University Press, 2000 (citato alle pagine 16, 56, 148).
- [8] Frank Bretz, Torsten Hothorn e Peter Westfall. *Multiple comparisons using R*. CRC Press, 2010 (citato a pagina 107).
- [9] Göran Broström. *Event History Analysis with R*. CRC Press, 2012 (citato alle pagine 162, 164).
- [10] Kenneth P Burnham e David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003 (citato alle pagine 92, 141).
- [15] Michael J Crawley. *Statistics: an introduction using R*. John Wiley & Sons, 2005 (citato a pagina 141).
- [16] Michael J Crawley. *The R book*. John Wiley & Sons, 2012 (citato a pagina 51).
- [20] Sorin Drăghici. *Statistics and data analysis for microarrays using R and bioconductor*. Chapman e Hall/CRC, 2016 (citato a pagina 16).
- [22] Julian J Faraway. *Linear models with R*. CRC Press, 2004 (citato alle pagine 85, 95).
- [23] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2005 (citato alle pagine 128, 132, 149, 150).
- [25] Ronald Aylmer Fisher. *The design of experiments*. Oliver e Boyd; Edinburgh; London, 1937 (citato a pagina 77).

- [29] Sergio Invernizzi. *Matematica nelle Scienze Naturali*. Trieste: Edizioni Goliardiche, 1996. ISBN: 8886573170 (citato alle pagine 16, 113).
- [30] Sergio Invernizzi, Maurizio Rinaldi e Federico Comoglio. *Moduli di matematica e statistica – Con l’uso di R*. Zanichelli, 2018 (citato alle pagine 80, 148).
- [33] Michael C Joiner e Albert Van der Kogel. *Basic clinical radiobiology*. CRC press, 2016 (citato a pagina 53).
- [34] Owen Jones, Andrew P. Robinson e Robert Maillardet. *Introduction to scientific programming and simulation using R*. Chapman e Hall/CRC, 2009 (citato a pagina 35).
- [43] Steven Piantadosi. *Clinical trials: a methodologic perspective*. John Wiley & Sons, 2017 (citato a pagina 43).
- [45] Christian Ritz e Jens Carl Streibig. *Nonlinear regression with R*. Springer Science & Business Media, 2008 (citato a pagina 117).
- [48] Vijay K Rohatgi. *Statistical inference*. Jonh Wiley & Sons, 1984 (citato alle pagine 22, 47, 53).
- [49] Bernard A Rosner. *Fundamentals of biostatics*. Duxbury Press, 1995 (citato alle pagine 32, 62–64, 148).
- [52] Stephen M Stigler. *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1986 (citato alle pagine 85, 87).
- [54] Terry M Therneau e Patricia M Grambsch. *Modeling survival data: extending the Cox model*. Springer Science & Business Media, 2000 (citato a pagina 162).
- [56] William N. Venables e Brian D. Ripley. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002. URL: <http://www.stats.ox.ac.uk/pub/MASS4> (citato a pagina 34).
- [58] Geert Verbeke e Geert Molenberghs. *Linear mixed models for longitudinal data*. Springer Science & Business Media, 2000 (citato alle pagine 141, 149).
- [60] Hadley Wickham e Garrett Grolemund. *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc., 2016 (citato alle pagine 6, 28).

## Risorse Web

- [2] Valentin Amrhein, Sander Greenland e Blake McShane. *Scientists rise up against statistical significance*. <https://www.nature.com/articles/d41586-019-00857-9>. Accessed: 2019-03-20. 2019 (citato alle pagine 78, 83).
- [21] EMBOpress. *Author Guidelines*. <http://emboj.embopress.org/authorguide>. Accessed: 2019-03-20 (citato a pagina 31).
- [26] Giovanni Gallo. *Biometria Fetale*. <http://web.tiscali.it/giovannigallo/tabelle/tabelle.htm>. Accessed: 2019-04-08 (citato a pagina 64).
- [35] Tatsuki Koyama. *Beware of Dynamite*. <http://biostat.mc.vanderbilt.edu/wiki/pub/Main/TatsukiRcode/Poster3.pdf>. Accessed: 2019-03-20 (citato a pagina 31).
- [47] Mark D Robinson, Davis J McCarthy e Gordon K Smyth. *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data*. <https://www.bioconductor.org/packages/devel/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>. Accessed: 2019-04-06 (citato a pagina 50).
- [61] Claus O. Wilke. *Fundamentals of Data Visualization*. <https://serialmentor.com/dataviz/>. Accessed: 2019-03-27 (citato a pagina 28).



## Indice analitico

AIC, 94  
ancova, 100  
anova, 104  
    one-way, 107  
    two-way, 107  
    two-way with interactions, 107  
  
bar plot, 34  
boxplot, 26, 27

calendar time, 140  
campione, 22  
chiave primaria, 24  
coefficiente di variazione, 62  
collinearità, 127  
confronti multipli, 107  
consuntivo, 72  
correlazione, 79  
covarianza, 79  
cut off, 34

dataset  
    airquality, 20  
    analgesia, 33  
    anscombe, 94  
    cholesterol, 24  
    epilessie, 127  
    iris, 17  
    raul, 48, 70, 74  
    tossicologia, 126  
    fresher, 85

design  
    cluster, 138  
    cross section, 26  
    longitudinale, 138  
    misure ripetute, 138  
    trasversale, 26  
  
devianza, 87  
deviazione standard, 21  
distanza di Cook, 96  
distribuzione  
    bernoulliana, 48  
    binomiale, 43, 49  
    binomiale negativa, 50  
    di Poisson, 53  
    gaussiana, 54  
    log-normale, 59  
    normale standard, 55  
  
dot plot, 31  
dynamite plot, 31

eteroschedasticità, 96  
  
fattore, 17  
    factor, 17  
    levels, 17  
Fisher Scoring, 132  
formato  
    .csv, 24  
    .txt, 36  
funzione di collegamento, 122

- funzione logistica, 123  
gradi di libertà, 73  
intervallo interquartile, 23  
istogramma, 34  
  
leverage, 96  
logit, 122  
  
media, 16, 25  
mediana, 16, 23, 25  
melanoma, 169  
midhinge, 62  
minimi quadrati, 85  
moda, 16  
    frequenze, 18  
modello minimale adeguato, 104  
multiple comparison, 107  
  
notazione  
    di Wilkinson e Rogers, 84  
  
odds ratio, 122, 124  
outlier, 27  
  
parametro di dispersione, 129  
percentile, 25  
popolazione, 22  
post-hoc analysis, 107  
predittore, 85  
predittore lineare, 122  
proxy, 111  
  
QQ plot, 58  
quantile, 25, 73  
quartili, 23  
  
range, 23  
regressione, 80  
residui, 86  
ROC, 132, 133  
  
sigmoide, 123  
sovradispersione, 128  
standard error, 30  
standardizzazione, 57  
  
t test, 72  
    a due campioni, 75  
test statistic, 72  
timestamp, 140  
  
variabile aleatoria  
    finita, 62  
verosimiglianza, 92, 132