

UNIVERSITÀ DEGLI STUDI DI TRIESTE

CORSO DI LAUREA IN OSTETRICIA

DISPENSA DIDATTICA

---

## 408ME – Informatica

[www.dmi.units.it/borelli](http://www.dmi.units.it/borelli)

---



<http://www.public-domain-photos.com>

*Autore*  
Massimo BORELLI, Ph.D.

Anno Accademico 2015 – 2016

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Stima del Peso Fetale</b>	<b>4</b>
2.1	Reperire una pubblicazione scientifica dalla rete . . . . .	4
2.2	Implementare una funzione nel Foglio Elettronico . . . . .	5
2.3	Implementare una funzione in Python . . . . .	6
2.4	Implementare una funzione in R . . . . .	7
2.5	Creare un grafico 'non banale' in R . . . . .	8
<b>3</b>	<b>Assegnazione Casuale al Trattamento</b>	<b>9</b>
3.1	Creare una lista di assegnazione casuale con il Foglio Elettronico . . . . .	9
3.2	generare numeri casuali con il Foglio elettronico . . . . .	11
3.3	Creare una lista di assegnazione casuale con R . . . . .	12
<b>4</b>	<b>Indici di rischio neonatale</b>	<b>14</b>

## 1 Introduzione

Questa dispensa raccoglie il contenuto di alcune lezioni del corso di Informatica rivolto alle studentesse del Corso di Laurea in Ostetricia dell'Università di Trieste, <http://dsm.units.it/it/didatt>. Lo scopo delle nostre lezioni è quello di prendere familiarità con un software di produttività personale che è usato comunemente per raccogliere ed organizzare i dati, il **foglio elettronico** (o **foglio di calcolo**), e con due linguaggi programmazione che vanno per la maggiore in ambito scientifico ed accademico, **Python** ed **R**. L'obiettivo didattico di questo corso è anche quello di mostrare che, oggi, la rete internet consente di svolgere moltissime azioni direttamente **online** ed in maniera gratuita, senza dover necessariamente 'scaricare' nel proprio computer i software; in omaggio alle filosofie del **free software** e dell'**open source**. Diciamo, dunque, qualche parola di più a proposito dei software che utilizzeremo.

**Il foglio di calcolo.** Tutti quelli che leggono questa dispensa, sicuramente, hanno già avuto qualche esperienza di uso con il **foglio elettronico**, che per metonimia tutti chiamano *Excel*, identificando l'applicativo con il nome del prodotto commerciale MS Excel®, <https://products.office.com/en-gb/buy/office>, venduto da Microsoft. Ricordiamo però che esistono validissime alternative gratuite, come **Apache Open Office Calc** si ottiene gratuitamente dal sito <https://www.openoffice.org/>.

E, tanto per dire, pensate che ci si potrebbe ancora procurare una copia funzionante di VisiCalc, il primo foglio elettronico della storia, ideato nel 1981 da Dan Bricklin, <http://www.danbricklin.com/history/vcexecutable.htm>. In queste dispense daremo preferenza a **Google Sheets**, <https://www.google.com/sheets/about/>, che vi permette di lavorare online direttamente con il vostro **browser** (Mozilla Firefox, Google Chrome, Apple Safari, Microsoft Internet Explorer, ...) e vi consente anche di condividere i fogli con i vostri colleghi di studio. Tutto quello che vi serve per accedere è un indirizzo mail, gratuito, con **GMail**, il servizio di posta elettronica di Google. Nell'immagine sottostante, Figura 1, vedete che abbiamo avuto accesso a Google Sheets per mezzo dell'account `informatica408me@gmail.com` :

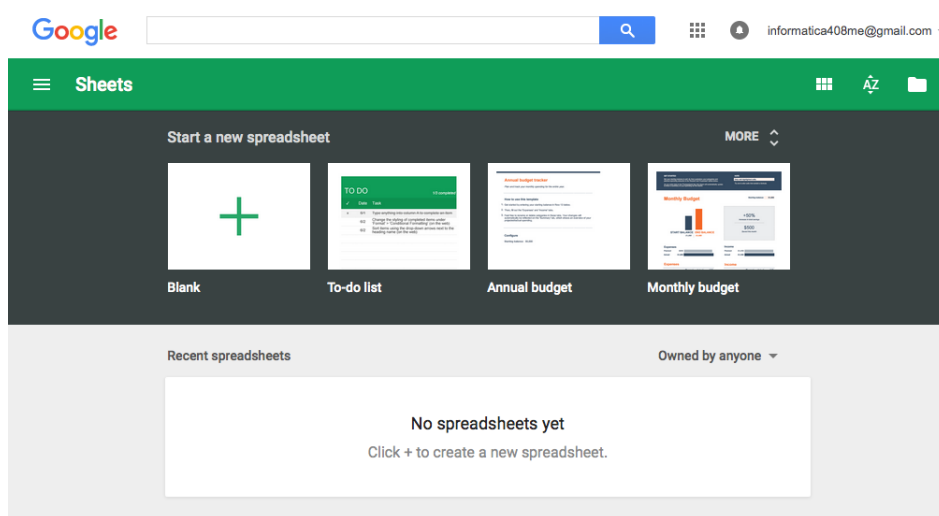


Figura 1: Google Sheets, il foglio di calcolo a cui si accede dalla vostra casella di posta GMail.

**Il linguaggio di programmazione Python.** Guido van Rossum è un informatico olandese che durante le vacanze di Natale del 1989 ha inventato un linguaggio di programmazione, dandogli il nome da una serie televisiva comica degli anni '70. Da allora, **Python** è diventato uno dei linguaggi di programmazione più diffusi al mondo, preceduto solo da Java e dal linguaggio C (con le varianti C++ e C#). I sistemi operativi iOS ('Mac') e Linux installano direttamente 'in maniera nativa' tale linguaggio. Al contrario, nei sistemi operativi Windows il linguaggio di programmazione non è pre-installato, ma lo si può scaricare gratuitamente da <https://www.python.org/>. Esistono inoltre degli ambienti computazionali interattivi, i cosiddetti **IPython Notebook**, che consentono di sviluppare gli *script* online. In queste dispense, ci serviremo di



Figura 2: La targa dell'automobile di Guido van Rossum, il 'benevolo dittatore a vita' di Python.

<https://www.pythonanywhere.com/try-ipython/>, il terminale dei comandi offerto da Python Anywhere.

**Il linguaggio di programmazione R.** R è anche un linguaggio di programmazione, ma la comunità scientifica lo adoperava soprattutto per le sue eccellenti capacità grafiche e statistiche. All'Università di Trieste molti sono gli studiosi che utilizzano ed insegnano ad usare R, il quale si ottiene gratuitamente dall'indirizzo <https://www.r-project.org/>. Da alcuni anni sta anche avendo rapida diffusione l'ambiente di sviluppo integrato (IDE) **R Studio**, <https://www.rstudio.com/>. In queste nostre esercitazioni, ci serviremo invece dell'ambiente online R Fiddle, <http://www.r-fiddle.org/>:



Figura 3: R Fiddle, un ambiente online di programmazione in linguaggio R.

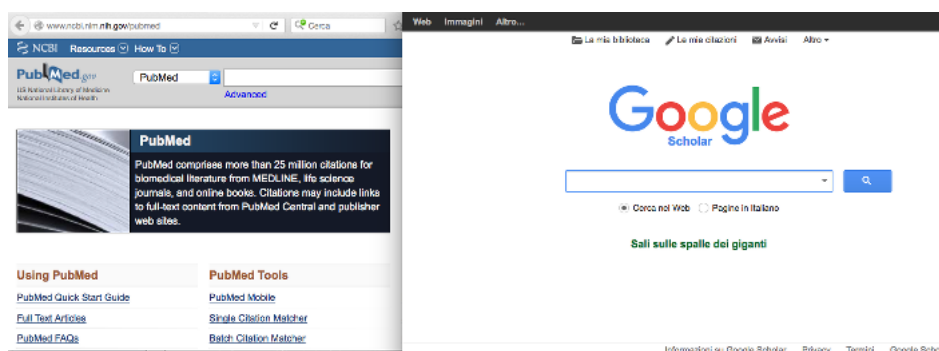
**Dulcis in fundo.** Per concludere questa parte introduttiva vi diciamo che anche questa dispensa è stata composta **online** ed utilizzando un software gratuito. Infatti, ci siamo serviti del 'linguaggio di markup'  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , <https://www.latex-project.org/>, modificando un 'template' offerto da Latex Templates, <http://www.latextemplates.com> e componendolo via web con Overleaf, <https://www.overleaf.com/>.

**Prerequisiti teorici.** Queste nostre dispense 'coprono' esclusivamente la parte del corso dedicata alla conoscenza di alcuni ambienti applicativi. Per quanto riguarda la parte di conoscenza teorica, abbiamo fatto riferimento al testo di Dennis P. Curtin, Kim Foley, Kunal Sen, Cathleen Morin intitolato *Informatica di base*, quinta edizione, 2012, McGraw-Hill.

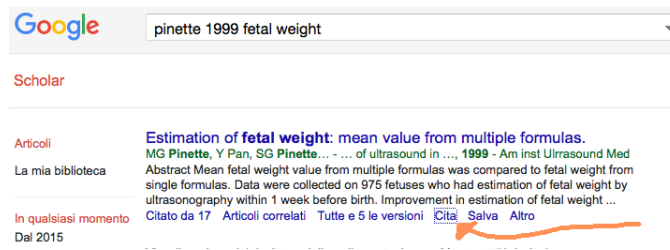
## 2 Stima del Peso Fetale

### 2.1 Reperire una pubblicazione scientifica dalla rete

Spesso ci si avvale di **Google Scholar** (<https://scholar.google.it/>) o di **Pubmed** (<http://www.ncbi.nlm.nih.gov/pubmed>) per reperire le pubblicazioni scientifiche utili alle nostre ricerche o al nostro aggiornamento professionale.



Nelle pagine finali di questa dispensa troverete la citazione bibliografica di un articolo scientifico [2] pubblicato dal medico Michael Pinette e dai suoi collaboratori, relativamente alla stima del peso fetale in relazione a certi parametri biometrici misurati con tecniche ecografiche. Per citare correttamente l'articolo anche noi ci siamo avvalsi di Google Scholar, digitando le **parole chiave** pinette 1999 fetal weight, come potete vedere nella figura seguente, e cliccando sull'opzione Cita, qui segnalata dalla freccia arancione:



**azione 01** cercate l'articolo scientifico in questione servendovi di **Pubmed**; dopo averlo reperito, ed aver osservato che si tratta di un *Free full text*, scaricate una copia in formato .pdf

Adesso che abbiamo sottomano questo articolo scientifico, leggiamolo per cercare di capire cosa significano le sigle BPD, AC, FL ed EFW. Cerchiamo inoltre di interpretare il significato delle 'formule' che vengono riportate nella Tabella 1 di pagina 814 (Shepard 1, Hadlock 2 e 3, Combs 4).

## 2.2 Implementare una funzione nel Foglio Elettronico

Ci proponiamo di realizzare un foglio di calcolo nel quale, ponendo come input dei valori fetali esemplificativi (espressi in *cm*) del diametro biparietale BPD, della circonferenza addominale AC, della lunghezza del femore FL e della circonferenza cranica HC, si ottenga in output il calcolo dei pesi stimati secondo le quattro formule di Shepard (espressi in *Kg*), di Hadlock e di Combs (espressi in *g*) riportate nell'articolo.

	A	B	C	D
1	BPD	9,7	Shepard (1)	3,601
2	AC	34,2	Hadlock (2)	3451
3	FL	7,5	Hadlock (3)	3335
4	HC	32,8	Combs (4)	3249

**azione 02** creare un foglio di calcolo nel quale, inserendo in input i valori delle celle B1:B4, si ottenga in output nelle celle D1:D4 i pesi stimati secondo le formule riportate nella Tabella 1.

In conclusione di questo paragrafo, troverete la soluzione a questo quesito. Può essere utile forse dare un suggerimento 'matematico'. Se ad esempio nella formula di Shepard vediamo scritto che:

$$\log_{10}(\textit{weight}) = 1.7492 + 0.166 \times \textit{BPD} + 0.046 \times \textit{AC} - 2.646 \times \frac{\textit{AC} \times \textit{BPD}}{1000}$$

questo significa che:

$$\textit{weight} = 10^{1.7492+0.166 \times \textit{BPD}+0.046 \times \textit{AC}-2.646 \times \frac{\textit{AC} \times \textit{BPD}}{1000}}$$

### 2.3 Implementare una funzione in Python

---

**azione 03** digitare in Google le parole chiave *ipython terminal online* e scegliere una risorsa web che emuli il terminale dei comandi Python. Definire di conseguenza una funzione che implementi la formula di Shepard (1) e ne visualizzi il valore per  $\textit{BPD} = 9.7$  e  $\textit{AC} = 34.2$ , usando questi comandi:

```
def shepard(bpd, ac):  
    weight = 10**(-1.7492+0.166*bpd+0.046*ac-0.002646*bpd*ac)  
    return weight  
  
print(shepard(9.7, 34.2))
```

---

Commentiamo le istruzioni che abbiamo inserito. La parola `def` significa che stiamo per definire una funzione, che chiamiamo `shepard`, e che calcolerà in output un valore, `weight`, sulla base di due variabili di input, `bpd` e `ac`. Il valore di `weight` si otterrà con un calcolo algebrico che coinvolge somme (+), moltiplicazioni (\*) ed elevamento a potenza (\*\*). È molto importante osservare che alla fine della riga

```
def shepard(bpd, ac):
```

abbiamo digitato i due punti; se questo non viene fatto, l'interprete dei comandi non è capace di definire la funzione. Inoltre, avete notato che le sue righe sottostanti,

```
weight = 10**(-1.7492+0.166*bpd+0.046*ac-0.002646*bpd*ac)
return weight
```

non sono 'allineate' a sinistra a caporiga, come la parola `def`, ma hanno qualche spazio bianco innanzi: si tratta della **indentazione** assolutamente necessaria a Python per individuare quanti e quali siano i comandi da attribuire alle definizioni della funzione (che tra l'altro contribuisce ad una migliore leggibilità in caso di programmi e funzioni complicate).

## 2.4 Implementare una funzione in R

Il proposito innanzitutto è quello di replicare quanto abbiamo appena fatto con Python, per notare quali siano le (lievi, ma sostanziali) differenze tra i due linguaggi di programmazione. Ecco lo *script* che ci occorre:

```
shepard = function(bpd, ac)
{10^(-1.7492 + 0.166*bpd + 0.046*ac - 0.002646*bpd*ac)}
```

```
shepard(9.7, 34.2)
```

Notate che è scomparsa la parola `def` che definiva la funzione, ma che è stata sostituita dalla parola `function`. Noterete che non vi sono più i due punti alla fine degli argomenti della funzione, e nemmeno l'indentazione a caporiga; tuttavia, l'espressione algebrica ora è racchiusa tra due parentesi graffe, obbligatorie se vi sono più comandi (ad onor del vero, in questo particolare esempio le parentesi graffe non sarebbero state necessarie, in quanto vi è una sola istruzione all'interno della funzione, e quindi non vi è ambiguità nell'interpretazione dei comandi). Infine, il simbolo di elevamento a potenza è il consueto 'cappuccio', invece del doppio asterisco.

---

**azione 04** digitare in Google le parole chiave *R fiddle*. Digitare i comandi della formula di Shepard (1) e cliccare sul pulsante color verde **Run Code** per ottenere il risultato.

---



## 2.5 Creare un grafico 'non banale' in R

Ci proponiamo ora di realizzare un grafico del tipo 'a curve di livello' per riuscire a valutare, visivamente ma in maniera sufficientemente approssimativa, la funzione `shepard` su una 'griglia' di punti relativi al diametro biparietale BPD e alla circonferenza addominale AC. Otterremo un grafico (che alcuni chiamano 'a mappa di calore', 'heat map') utilizzando il comando `image` e sovrapporremo ad esso le curve di livello mediante il comando `contour`. L'istruzione `seq` genera delle sequenze di numeri, mentre l'istruzione `outer` nel nostro caso consente di creare il 'pavimento bidimensionale AC-BPD' sul quale la funzione `shepard` verrà calcolata.

Siccome i comandi iniziano ad essere in numero non trascurabile, è buona norma digitarli preventivamente in un semplice editor di testo (il blocco note/notepad, ad esempio; ma in rete si possono scaricare gratuitamente eccellenti editor di testo, come Tinn-R e Textwrangler).

---

**azione 05** digitare sul vostro editor di testo preferito (attenzione: vietato usare MS Word o Open Office Writer, per evitare che il software usi dei caratteri non-Ascii per rappresentare trattini, apostrofi, virgolette inglesi e altri orpelli consimili) i comandi:

```
shepard = function(bpd, ac)
{10^(-1.7492 + 0.166*bpd + 0.046*ac - 0.002646*bpd*ac)}
ac = seq(from = 30, to = 35, by = .05)
bpd = seq(from = 6, to = 10, by = .05)
z = outer(ac, bpd, shepard)
image(ac, bpd, z)
contour(ac, bpd, z, add = TRUE)
```

Copiare tutta la sequenza di comandi e incollarla in R Fiddle. Cliccare sui pulsanti color verde **Run Code** e color grigio **Graphs** per ottenere il risultato.

---

**Soluzioni** Ecco alcune possibili soluzioni dei quesiti del paragrafo 2.2.

	A	B	C	D	E	F	G
1	BPD	9,7	Shepard (1)	3.601			
2	AC	34,2	Hadlock (2)	3451			
3	FL	7,5	Hadlock (3)	3335			
4	HC	32,8	Combs (4)	3249			
5							

	A	B	C	D	E
1	BPD	9,7	Shepard (1)	3.601	
2	AC	34,2	Hadlock (2)	3451	
3	FL	7,5	Hadlock (3)	3335	
4	HC	32,8	Combs (4)	3249	
5					

### 3 Assegnazione Casuale al Trattamento

#### 3.1 Creare una lista di assegnazione casuale con il Foglio Elettronico

Supponiamo che in uno studio sperimentale clinico controllato e casualizzato (RCT, randomized controlled trial) si debbano arruolare 84 partecipanti, dei quali 42 da inviare al braccio del trattamento e 42 al braccio del placebo. Vogliamo creare una lista casuale ed anonima come nella Figura sottostante:

	A	B
1	partecipanti	braccio
2	Paziente01	placebo
3	Paziente02	placebo
4	Paziente03	trattamento
5	Paziente04	placebo
6	Paziente05	trattamento
83	Paziente82	trattamento
84	Paziente83	trattamento
85	Paziente84	placebo
86		

---

**azione 01** digitare in Google le parole chiave *Open Office* per scaricare la *suite* di applicativi di produttività personale. Avviate un nuovo foglio di calcolo con *Open Office Calc*. Eseguite ora le seguenti azioni.

---

**azione 02** Iniziamo nella colonna C a creare un'intestazione **contatore** e una sequenza intera da 101 a 184, digitando nella cella C3 la formula = C2 + 1, copiandola ed incollandola:

	A	B	C
1			contatore
2			101
3			102
4			103
5			104
6			105
7			106
83			182
84			183
85			184

**azione 03** Nella colonna D con l'intestazione duecifre usiamo la **funzione di testo DESTRA** per estrarre gli ultimi due caratteri che appaiono nella colonna adiacente. Si noti che mentre nella colonna C i numeri sono automaticamente allineati a destra della cella, nella colonna D essi sono allineati a sinistra come i caratteri alfabetici (i.e. un testo).

	A	B	C	D
1			contatore	duecifre
2			101	=DESTRA(C2;2)
3			102	02
4			103	03
5			104	04
6			105	05
7			106	06
83			182	82
84			183	83
85			184	84

**azione 04** Nella colonna E con l'intestazione assieme usiamo la funzione **CONCATENA** per congiungere la parola Paziente racchiusa tra i doppi apici, e la numerazione progressiva a due cifre della colonna D:

C	D	E	F
contatore	duecifre	assieme	
101	01	=CONCATENA("Paziente";D2)	
102	02	Paziente02	
103	03	Paziente03	
104	04	Paziente04	
105	05	Paziente05	
106	06	Paziente06	
182	82	Paziente82	
183	83	Paziente83	
184	84	Paziente84	

### 3.2 generare numeri casuali con il Foglio elettronico

Adesso abbiamo bisogno di 'simulare' il lancio di una moneta all'interno del Foglio di calcolo: questo si può fare per mezzo di (raffinati) algoritmi che generano sequenze di numeri pseudo-casuali. Essendo che i numeri sono, appunto, generati da algoritmi essi sono in definitiva tutto meno che 'casuali'; ma agli occhi di un qualsiasi sperimentatore umano essi ci appaiono del tutto imprevedibili. Per il nostro scopo, questo è più che sufficiente.

**azione 05** Nella colonna G con l'intestazione **gruppi**, con un copia-incolla creiamo consecutivamente 42 celle **trattamento** e 42 celle **placebo**

E	F	G
assieme		gruppi
Paziente01		trattamento
Paziente02		trattamento
Paziente03		trattamento
Paziente41		trattamento
Paziente42		trattamento
Paziente43		placebo
Paziente44		placebo
Paziente45		placebo
Paziente46		placebo

**azione 06** Nella colonna I con l'intestazione **acaso** usiamo la funzione **CASUALE** per generare 84 numeri casuali distribuiti uniformemente nell'intervallo da 0 ad 1 (0 compreso, 1 escluso):

G	H	I
gruppi		acaso
trattamento		0,397486641
trattamento		0,557971777
trattamento		0,831659208
placebo		0,273798108
placebo		0,724804994
placebo		0,797542932
placebo		0,304056164

**azione 07** Copiamo le celle della colonna I ed incolliamole nella colonna H in **modalità speciale**, incollando i contenuti delle celle come valori numerici e non come funzioni. Dobbiamo fare questo passaggio poiché i Fogli di calcolo, usualmente, hanno attivata la funzione di ricalcolo automatico dei valori; ed infatti, noteremo proprio che la colonna I verrà ricalcolata, perdendo i valori originali che ora sono memorizzati in H.

G	H	I
gruppi	acaso	acaso
trattamento	0,397486641	0,642737483
trattamento	0,557971777	0,488884165
trattamento	0,831659208	0,676171996
placebo	0,273798108	0,555609326
placebo	0,724804994	0,125944385
placebo	0,797542932	0,747284807
placebo	0,304056164	0,615761548

**azione 08** Eliminiamo la colonna I che è diventata inutile; selezioniamo entrambe le colonne G ed H ed **ordiniamole** per **acaso** (e non per **gruppi**). Vedremo che adesso la colonna H ospita una sequenza di numeri perfettamente ordinati, ed il 'disordine' si è trasferito nella colonna G, creando una sequenza casuale come volevamo.

G	H
gruppi	acaso
placebo	0,022233164
placebo	0,024941457
trattamento	0,046798519
placebo	0,969422975
trattamento	0,975099717
trattamento	0,986986665
placebo	0,99476557

Per concludere, è sufficiente copiare el colonne E e G nelle colonne A e B, cambiando le intestazioni come volevamo all'inizio.

### 3.3 Creare una lista di assegnazione casuale con R

Vogliamo ora vedere come possiamo svolgere con R il medesimo compito. Si tratterà di far eseguire i seguenti comandi, che vogliamo però discutere passo per passo:

```
gruppi = rep( c("placebo", "trattamento"), 42 )
braccio = sample(gruppi)
numeri = 101:184
adestra = substr(numeri, 2, 3)
paziente = rep("Paziente", 84)
partecipanti = paste0(paziente, adestra)
data.frame(partecipanti, braccio)
```

L'istruzione `rep` consente di ripetere per 42 volte consecutivamente il vettore `c(-placebo, trattamento)`, ossia un vettore che ha in sè soli due elementi, la parola 'placebo' e la parola 'trattamento', appunto. Conseguentemente, il vettore `gruppi` contiene 84 elementi, 42 volte la parola 'placebo' e 42 volte la parola 'trattamento', in analogia con quello che accadeva nella colonna G del foglio elettronico.

Con il comando `sample`, che in inglese significa 'campione', noi estraiamo in maniera (pseudo-)casuale gli 84 elementi del vettore `gruppi` – in analogia a quanto succede quando mischiamo un mazzo di carte; l'estrazione genera dunque un vettore che chiamiamo `braccio` e corrisponde a quanto si voleva nella colonna B del Foglio elettronico.

Si trattava ora di creare la colonna A, quella dei `partecipanti`: lo facciamo nelle quattro righe successive, considerando innanzitutto la sequenza di numeri interi che va

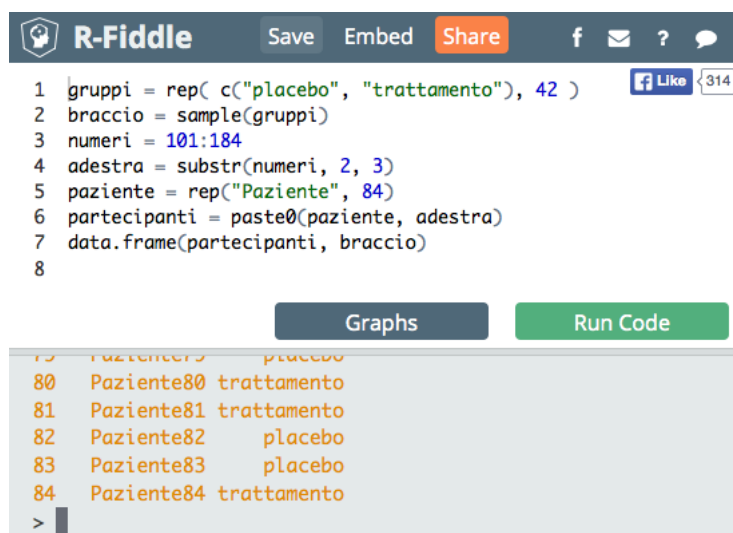
da 101 a 184, e con il comando `substr` estraiamo la sottostringa che inizia nella seconda posizione e finisce nella terza posizione (ossia 01, 02, 03, .. , 84). Chiamiamo questa sottostringa `adestra`. Ripetiamo la parola 'Paziente' per 84 volte creando il vettore `paziente` e incolliamo questi due vettori assieme, senza spazi bianchi in mezzo, con il comando `paste0`.

L'ultima riga crea il 'tipico oggetto' di interesse di R, che è il `data.frame` e che possiamo considerare come una rappresentazione del dataset di cui ci occupiamo nelle nostre analisi statistiche: in pratica, una tabella con due colonne, come volevamo nella Figura iniziale di questo Capitolo.

---

**azione 09** digitare in *R Fiddle* le sette righe di comandi precedenti, per ottenere l'output desiderato con il pulsante verde **Run Code**.

---



```
1 gruppi = rep( c("placebo", "trattamento"), 42 )
2 braccio = sample(gruppi)
3 numeri = 101:184
4 adestra = substr(numeri, 2, 3)
5 paziente = rep("Paziente", 84)
6 partecipanti = paste0(paziente, adestra)
7 data.frame(partecipanti, braccio)
8
```

Graphs Run Code

```
79 Paziente80 placebo
80 Paziente80 trattamento
81 Paziente81 trattamento
82 Paziente82 placebo
83 Paziente83 placebo
84 Paziente84 trattamento
>
```

## 4 Indici di rischio neonatale

Nei reparti di terapia intensiva neonatale si utilizzano comunemente degli indici di rischio che valutano lo stato di salute dei loro sfortunati pazienti. Uno di questi indici si chiama CRIB II [1] e tiene in considerazione cinque elementi:

- il genere del neonato, femminile o maschile
- il peso alla nascita, espresso in grammi
- la settimana di gestazione alla nascita, dalla 22-esima alla 32-esima
- la temperatura corporea rilevata all'ammissione al reparto
- l'eccesso di basi misurato tramite l'emogasanalisi

Il punteggio di rischio si ottiene in maniera additiva, sommando gli *scores* che sono tabulati all'interno dell'articolo [1]. Tanto per fare un esempio, supponendo di avere un neonato di sesso maschile, con un peso di 1900 grammi, nato alla 29-esima settimana di gestazione, con una temperatura di 33,5 gradi e un eccesso di basi pari a -14 mmol/L, procediamo in questo modo, traendo gli scores dalle seguenti tabelle:

**Birthweight (g) and gestation (weeks):**  
 The maximum (worst) score for birthweight and gestation is 15, which is obtained for a 22 week male infant of less than 501 g birthweight

Male infants											Female infants																
Birthweight (g)	22	23	24	25	26	27	28	29	30	31	32	Birthweight (g)	22	23	24	25	26	27	28	29	30	31	32				
2751 to 3000											0											0					
2501 to 2750											1	0										1	0				
2251 to 2500										3	0	0									2	0	0				
2001 to 2250										2	0	0									1	0	0				
1751 to 2000										1	0	0									3	1	0	0			
1501 to 1750						6	5	3	2	1	0					6	4	3	1	0	0	0					
1251 to 1500					8	6	5	3	3	2	1					7	5	4	3	2	1	1					
1001 to 1250			12	10	9	8	7	6	5	4	3	3			11	10	8	7	6	5	4	3	3	3			
751 to 1000			12	11	10	8	7	7	6	6	6	6			11	10	9	8	7	6	5	5	5	5			
501 to 750			14	13	12	11	10	9	8	8	8	8			13	12	11	10	9	8	8	7	7	7			
251 to 500			15	14	13	12	11	10	10			14	13	12	11	10	10	10			14	13	12	11	10	10	10

Temperature at admission (°C)	
<29-6	5
29-7 to 31-2	4
31-3 to 32-8	3
32-9 to 34-4	2
34-5 to 36	1
36-1 to 37-5	0
37-6 to 39-1	1
39-2 to 40-7	2
>40-8	3

Base excess (mmol/L):	
<-26	7
-26 to -23	6
-22 to -18	5
-17 to -13	4
-12 to -8	3
-7 to -3	2
-2 to 2	1
>3	0

Sesso, peso alla nascita e	3
Temperatura all'ammissione	2
Eccesso di basi	4
Punteggio totale CRIB II	9

Il punteggio totale CRIB II viene successivamente trasformato in maniera lineare determinando il cosiddetto log-odds della mortalità  $G$ , per mezzo della formula

$$G = -6.476 + 0.450 \times CRIB II$$

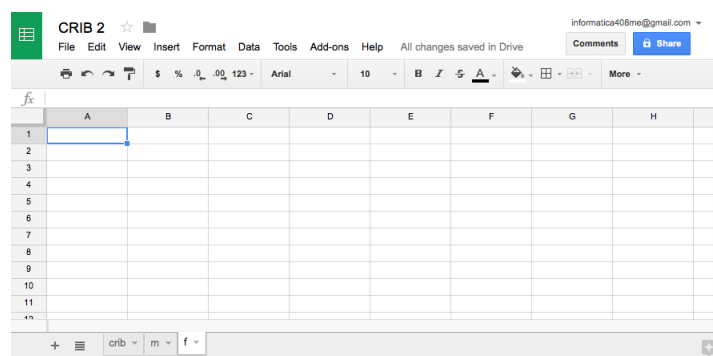
$$= -6.476 + 0.450 \times 9 = -2.426$$

ed infine per mezzo di  $G$  calcoliamo la probabilità  $P$  di un outcome sfavorevole per mezzo della funzione logistica (come vi insegneranno nei corsi di statistica):

$$P = \frac{\exp(G)}{1 + \exp(G)}$$

$$= \frac{\exp(-2.426)}{1 + \exp(-2.426)} = \frac{0.0884}{1 + 0.0884} = 0.08 = 8\%$$

**azione 01** digitare in Google le parole chiave *Google Sheets* ed entrare nell'ambiente con il vostro account GMail. Rinominate il foglio da *Untitled spreadsheet* a **CRIB 2**. Duplicate il foglio corrente per due volte, e rinominate i tre fogli rispettivamente **crib**, **m** e **f**. Eseguite ora, passo per passo le azioni che vi proponiamo qui di seguito.





**azione 02** Nelle colonne A e B inseriamo i dati del paziente in questione:

	A	B	C
1	genere	m	
2	peso alla nascita	1900	
3	settimana gestazione	29	
4	temperatura	33,5	
5	base excess	-14	

**azione 03** Nel foglio **m** ricopiamo gli *scores* pubblicati in [1] per i pazienti maschi, a partire dalla cella A1. Facciamo lo stesso nel foglio **f** per le femmine.

	A	B	C	D	E	F	G	H	I	J
1	15	14	13	12	11	10	10			
2	14	13	12	11	10	9	8	8	8	8
3		12	11	10	8	7	7	6	6	6
4		12	10	9	8	7	6	5	4	3
5				8	6	5	3	3	1	
6					6	5	3	2	0	
7								3	1	0
8										2
9										3
10										0
11										0

Si tratta ora di escogitare un modo per individuare la 'giusta' riga degli scores a partire dal peso del paziente. Osserviamo questo schema:

7	1750	Birthweight (g)	2001to 2250
:	:		1751to 2000
4	1000		1501to 1750
3	750		1251to 1500
2	500		1001to 1250
1	250		751 to 1000
			501 to 750
			251 to 500

Potete intuire che se i numeri naturali 1, 2, 3, ..., vengono moltiplicati per 250, e al risultato si aggiunge 1, si ottiene esattamente l'estremo sinistro delle classi utilizzate da CRIB II (251, 501, 751, ...). Inversamente, se io parto ad esempio del peso alla nascita di 1900 grammi, sottraggo 1 e divido per 250 ottengo 7.596; la parte intera di questo numero, cioè 7, rappresenta proprio la riga di nostro interesse.

**azione 04** Nella cella D2 inseriamo la funzione:

$$=INT((B2-1)/250)$$

fx   =INT((B2-1)/250)				
	A	B	C	D
1	genere	m		
2	peso alla nascita	1900	riga	7
3	settimana gestazione	29	colonna	H

È la volta ora dell'indice di colonna; noi vogliamo che ad esempio alla 29-esima settimana di gestazione corrisponda la colonna H:

64	65	66					71				
A	B	C	.	.	.	H					
22	23	24	25	26	27	28	29	30	31	32	
<b>Gestation (weeks)</b>											

Ci vengono in ausilio il codice ASCII ([www.asciitable.com](http://www.asciitable.com)) e il codice UNICODE ([unicode-table.com/en/](http://unicode-table.com/en/)), che ci fanno sapere sapendo che alla lettera H corrisponde il numero 71; quindi aggiungendo a 29 il numero costante 43, ed utilizzando la funzione CHAR, il gioco è fatto<sup>1</sup>:

=CHAR((B3 +43))				
A	B	C	D	
genere	m			
peso alla nascita	1900	riga	7	
settimana gestazione	29	colonna	H	

Ora si tratta di 'riunire' le informazioni di colonna e riga, ed andare a recuperare il contenuto della cella H7 situata però nel foglio di lavoro individuato dalla lettera m, inerente il genere del paziente, conservata nella cella B2. Per questo proposito, dobbiamo

<sup>1</sup>Precisiamo che la funzione CHAR ha una sua corrispondente in MS Excel e in Open Office: si tratta della funzione CODICE.CARATT()

servirci della funzione CONCAT()<sup>2</sup>. Inseriamo quindi, nelle celle D4, D5 e D6 le seguenti funzioni:

=CONCAT(B1,"!")  
 =CONCAT(D4,D3)  
 =CONCAT(D5,D2)

per creare la stringa m!H7

=CONCAT(D5,D2)

A	B	C	D
genere	m		
peso alla nascita	1900	riga	7
settimana gestazione	29	colonna	H
temperatura	33,5	referimento1	m!
base excess	-14	referimento2	m!H
		referimento2	m!H7

**azione 05** Nella cella F1 inseriamo la funzione:

=INDIRECT(D6)

per andare a leggere il valore che è contenuto non nella cella D6, bensì nella cella il cui indirizzo viene specificato dalla cella D6, ossia nella cella H7 del foglio m.

=INDIRECT(D6)

A	B	C	D	E	F
genere	m			score1	3
peso alla nascita	1900	riga	7		
settimana gestazione	29	colonna	H		
temperatura	33,5	referimento1	m!		
base excess	-14	referimento2	m!H		
		referimento2	m!H7		

**azione 06** Nella cella F2 inseriamo la funzione:

=ABS(INT((B4-36)/1.6))

<sup>2</sup>A quanto sembra, la funzione CONCAT implementata in Google Sheets non è altrettanto flessibile di quanto lo sia la funzione CONCATENA() di MS Excel e di Open Office Calc. Infatti, mentre negli ultimi due software si possono concatenare molte celle tra loro, in Google Sheets apparentemente si possono concatenare solo due celle, e quindi nel nostro caso dobbiamo agire ricorsivamente, per tre volte.

controllando di avere digitato il punto decimale (e non la virgola decimale, come siamo soliti in Europa) anche nella cella B3.

`=ABS(INT((B4-36)/1.6))`

A	B	C	D	E	F
genere	m			score1	3
peso alla nascita	1900	riga	7	score2	2
settimana gestazione	29	colonna	H		
temperatura	33.5	riferimento1	m!		
base excess	-14	riferimento2	m!H		
		riferimento2	m!H7		

Per dare una spiegazione della genesi di questa formula (approssimata), possiamo osservare questi passaggi:

			Temperature at admission (°C)	
			≤29.6	5
			29.7 to 31.2	4
			31.3 to 32.8	3
			32.9 to 34.4	2
			34.5 to 36	1
			36.1 to 37.5	0
			37.6 to 39.1	1
			39.2 to 40.7	2
			≥40.8	3

-3,9	-6,3	29,7
-2,9	-4,7	31,3
-1,9	-3,1	32,9
-0,9	-1,5	34,5
0,1	0,1	36,1
1,0	1,6	37,6
2,0	3,2	39,2
3,0	4,8	40,8

*Note: Purple arrows in the second table point from -36 to -0,9 and from 1.6 to 1,6.*

**azione 07** Un ragionamento analogo a quello appena visto permette di giustificare la funzione che andiamo ad inserire nella cella F3:

$$=INT(1.5 - B5/5)$$

`=int(1.5-B5/5)`

A	B	C	D	E	F
genere	m			score1	3
peso alla nascita	1900	riga	7	score2	2
settimana gestazione	29	colonna	H	score3	4
temperatura	33.5	riferimento1	m!		
base excess	-14	riferimento2	m!H		
		riferimento2	m!H7		

**azione 08** Calcoliamo automaticamente la somma degli scores nella cella F4 usando la funzione di somma, SUM, ottenendo il punteggio CRIB II in maniera completamente automatica a partire dai dati di input:

**=SUM(F1:F3)**

	A	B	C	D	E	F
1	genere	m			score1	3
2	peso alla nascita	1900	riga	7	score2	2
3	settimana gestazione	29	colonna	H	score3	4
4	temperatura	33.5	riferimento1	m!	<b>CRIB II</b>	<b>9</b>
5	base excess	-14	riferimento2	m!H		
6			riferimento2	m!H7		

**azione 09** Trasformiamo il punteggio CRIB II nel cosiddetto log-odds della mortalità,  $G$ :

**$f_x$  = -6.476 + 0.45 \* F4**

	A	B	C	D	E	F
1	genere	m			score1	3
2	peso alla nascita	1900	riga	7	score2	2
3	settimana gestazione	29	colonna	H	score3	4
4	temperatura	33.5	riferimento1	m!	<b>CRIB II</b>	<b>9</b>
5	base excess	-14	riferimento2	m!H	G	-2.426
6			riferimento2	m!H7		

**azione 10** Inseriamo la relazione logistica nella cella F6, nascondiamo le celle C e D (senza eliminare i contenuti!) e diamo un tocco di colore finale:

**$f_x$  = EXP(F5) / (1 + EXP(F5))**

	A	B	E	F
1	genere	m	score1	3
2	peso alla nascita	1900	score2	2
3	settimana gestazione	29	score3	4
4	temperatura	33.5	<b>CRIB II</b>	<b>9</b>
5	base excess	-14	G	-2.426
6			<b>probability</b>	<b>8.12%</b>

## Riferimenti bibliografici

- [1] Gareth Parry, Janet Tucker, William Tarnow-Mordi, UK Neonatal Staffing Study Collaborative Group, et al. Crib ii: an update of the clinical risk index for babies score. *The Lancet*, 361(9371):1789–1791, 2003.
- [2] Michael G Pinette, Yuqun Pan, Sheila G Pinette, Jacquelyn Blackstone, John Garrett, and Angelina Cartin. Estimation of fetal weight: mean value from multiple formulas. *Journal of ultrasound in medicine*, 18(12):813–817, 1999.