UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea Triennale in Informatica

Uno strumento di analisi statistica delle prove d'esame di programmazione

Tesi di Laurea di:

Marco Cordoni

Matr. 855535

Relatore:

Prof. Massimo Santini

Anno Accademico 2016/2017

Indice

Capitolo 1

Descrizione del problema

L'analisi dei dati è il processo di studio e valutazione dei dati, precede la creazione dei primi calcolatori e si è sviluppata con essi per sfruttare al meglio le loro capacità di calcolo.

L'analisi si compone di diverse fasi, parte dalla raccolta delle informazioni necessarie all'attività da compiere e procede poi con la memorizzazione dei dati attraverso archivi cartacei o digitali e si conclude con la loro elaborazione e trasformazione per trarre conoscenze rilevanti come la distribuzione di un certo evento nel tempo o nello spazio.

Questo scienza ha trovato un largo riscontro in ogni ambito, per citarne alcuni: fisica, geografia e anche economia, basti pensare alla statistica economica sfruttata dalle imprese per analizzare il mercato e massimizzare il profitto. [1]

In questo progetto mostro come lo studio dei dati può supportare un docente nell'analisi degli esami di programmazione e fornire informazioni relative alla difficoltà degli esercizi assegnati e l'abilità degli studenti.

I dati da analizzare mi sono stati forniti dal docente, il quale li ha raccolti da esami e laboratori durante la sua attività di insegnante di Programmazione presso il corso di laurea triennale in Informatica Musicale.

Questi dati sono memorizzati all'interno di tabelle tridimensionali, ovvero strutture indicizzate da 3 valori: la matricola dello studente, il nome dell'esercizio e il criterio di valutazione.

Per ogni studente conosco se in ogni esercizio è stato effettuato l'upload della soluzione proposta, se compila senza presentare errori, se sono stati rilevati errori di esecuzione ed il numero di test case corretti ed errati.

Usufruendo di queste informazioni il mio obiettivo principale è di stimare le difficoltà degli esercizi e verificare se è possibile formulare delle ipotesi inerenti al livello di complessità riscontrato, per esempio se è sensato supporre l'esistenza di un livello crescente di difficoltà.

La finalità del progetto è lo sviluppo di uno strumento di analisi statistica delle prove d'esame e dei laboratori di Programmazione, cioè la creazione di una libreria di funzioni applicabili all'interno di Jupyter Notebook.

Capitolo 2

Strumenti impiegati

Precedentemente alla fase di realizzazione della tesi ho approfondito la conoscenza degli argomenti fondamentali e degli strumenti necessari per la creazione del progetto, le nozioni apprese sono state applicate in ogni aspetto della fase di sviluppo.

In questo capitolo provvederò a descrivere ed analizzare gli strumenti teorici e pratici che mi hanno permesso la realizzazione di questa tesi, le ragioni per le quali sono stati scelti, le alternative che avrei potuto adottare e i benefici che ne ho tratto.

2.1 Conoscenze teoriche

2.1.1 Durante il corso di studi

Durante il triennio ho potuto approcciarmi alla statistica per mezzo del corso "Statistica e analisi dei dati", erogato allo scopo di introdurre gli studenti alla disciplina e istruire all'uso di uno strumento di analisi statistica dei dati che prende il nome di R. [2]

Al termine di tale corso ho appreso molte conoscenze, seppur non sempre particolarmente approfondite, tra cui: la statistica descrittiva, elementi di probabilità, definizione di variabile aleatoria, valore atteso, alcuni modelli di variabili aleatorie e alcune nozioni basilari riguardanti la distribuzione delle statistiche campionarie e il concetto di stimatore.

Queste competenze mi sono state molto utili per la realizzazione di questo progetto anche se, in diverse occasioni, ho dovuto approfondire autonomamente o con il supporto del docente concetti più avanzati di quelli che sono stati affrontati e ricorrere a soluzioni ricercate per mezzo di libri ed internet.

2.1.2 Item response theory

La psicometria è una disciplina che ha lo scopo di misurare le abilità delle persone basandosi sulle loro conoscenze o sul loro comportamento.

Il paradigma il cui obiettivo è l'analisi delle risposte dei soggetti è detto "Item response theory" (IRT).

Le informazioni dei soggetti esaminati, da cui dedurre i dati, sono raccolte tramite test di difficoltà non uniforme; se questi test presentano risposte giudicabili unicamente come giuste o sbagliate allora sono riconducibili ad una forma di dati dicotomica. [3]

Questa forma è vantaggiosa perché è semplice da comprendere ma a scapito della perdita delle "conoscenze intermedie" nel caso vi fossero, cioè non vi sono distinzioni tra esercizi totalmente errati oppure parzialmente corretti.

Alternativamente, esiste la possibilità di trattare dati politomici, ovvero dati non vincolati dalla presenza di soli 2 risultati possibili, permettendo la conoscenza di informazioni sul livello di correttezza degli esercizi.

L'IRT permette di definire la probabilità con cui un soggetto può svolgere correttamente un certo quesito, tale viene definita attraverso varie funzioni matematiche che prevedono ciascuna la conoscenza dell'abilità del soggetto considerato e il livello di difficoltà del problema affrontato.

Esistono vari tipi di modelli per il calcolo di tale probabilità, di seguito presenterò i 3 più comuni descrivendo le loro differenze e motivando le scelte che ho compiuto:

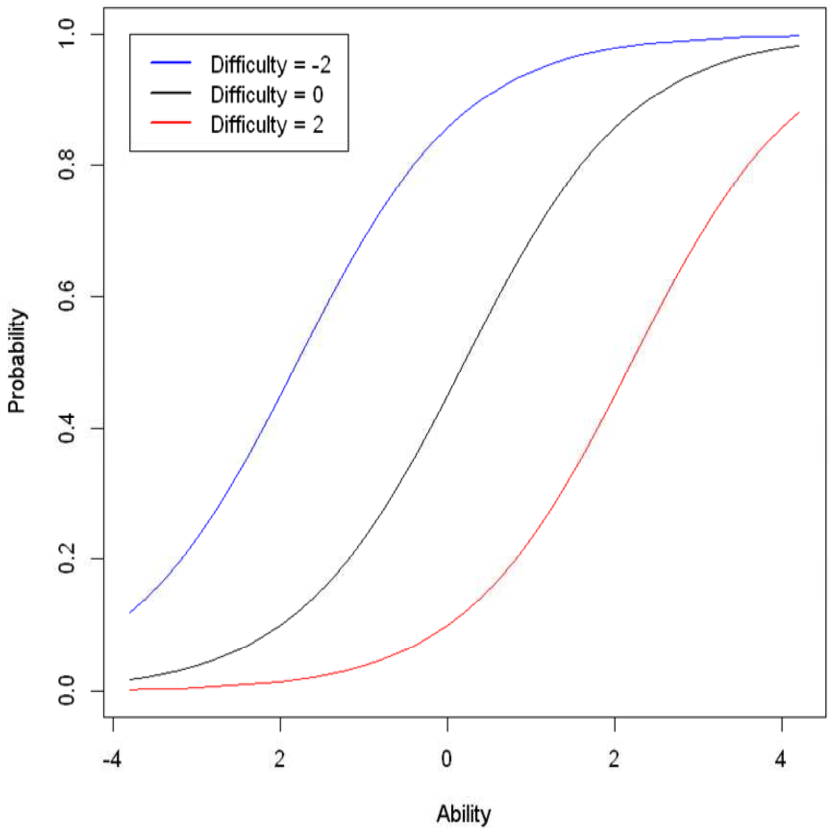
- Il modello che prevede la conoscenza solamente dell'abilità e della difficoltà prende il nome di "Modello ad 1 parametro" (1PL) o "Modello di Rasch" la cui formula è di seguito riportata:

$$p(x_{ij} = 1|\theta_i, \beta_j) = \frac{\mathrm{e}^{(\theta_i - \beta_j)}}{1 + \mathrm{e}^{(\theta_i - \beta_j)}} \tag{2.1}$$

  x_{ij} rappresenta la probabilità che il soggetto i-esimo superi il quesito j-esimo a cui viene sottoposto, θ_i è l'abilità dell'individuo considerato e β_j corrisponde alla difficoltà stimata per l'item j-esimo.

  θ_i e β_j devono appartenere alla stessa scala di valori, se così non fosse non avrei la possibilità di calcolare qualunque probabilità possibile e quindi ottenere ogni valore da 0 ad 1, se ad esempio le abilità appartenessero all'intervallo tra 5 e 10 e le difficoltà tra -2 e 2 anche in presenza della difficoltà massima e l'abilità minima avrei comunque una probabilità di successo prossima ad 1.

  Nel seguente grafico mostro l'incisione dell'abilità sul calcolo della probabilità assumendo 3 diversi livelli di difficoltà, valori bassi di difficoltà determinano un esercizio facile e alti valori di abilità denotano uno studente con grandi competenze.



Noto che discostandosi di solamente 2 unità i grafici subiscono una forte traslazione quando l'abilità è media e tornano quasi a coincidere in presenza di una capacità minima e massima.

Per la tesi ho scelto di adottare questo modello perché presenta tutte le informazioni che mi sono posto di stimare evitando, per ragioni di tempo, di analizzare altri parametri che non costituiscono l'obiettivo del progetto.

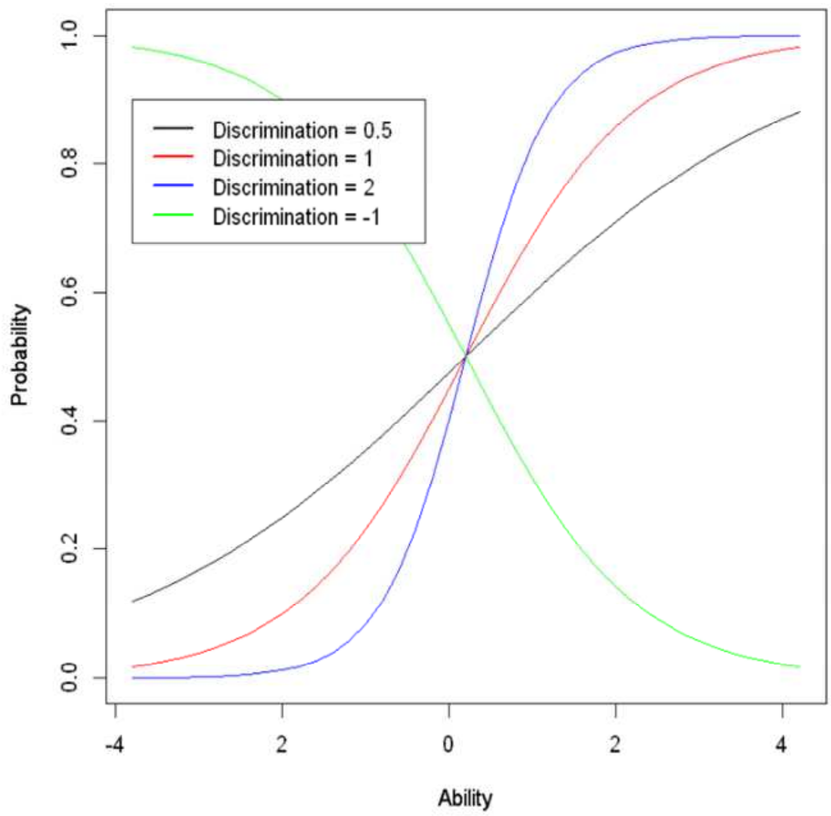
- Al precedente modello può essere aggiunto il parametro discriminatorio α_j ottenendo il "Modello a 2 parametri" (2PL).

  Il discriminante definisce la velocità di crescita della probabilità che il soggetto superi l'esercizio, cioè influisce sulla variazione della probabilità a basse modifiche del parametro dell'abilità.

$$p(x_{ij} = 1|\theta_i, \beta_j, \alpha_j) = \frac{e^{\alpha_j(\theta_i - \beta_j)}}{1 + e^{\alpha_j(\theta_i - \beta_j)}} \tag{2.2}$$

Quando α_j vale 1 i modelli a 1 e 2 parametri coincidono, se α_j assume valori compresi tra 0 e 1 la pendenza della curva di probabilità diminuisce e, viceversa, aumenta quando tale parametro e maggiore di 1.

α_j non può essere negativo altrimenti si otterrebbe l'irrealistico caso che a bassi valori di abilità corrispondono alte probabilità di superare l'esercizio e tale probabilità decrescerebbe all'aumentare dell'abilità dei soggetti.



Come si può leggere dal grafico, quando il discriminante è elevato (2) non vi è differenza se un soggetto possiede un livello di abilità basso (-4) o medio-basso (-2) ma in corrispondenza di una capacità media (0) è presente un alto grado di crescita della curva, quasi verticale.

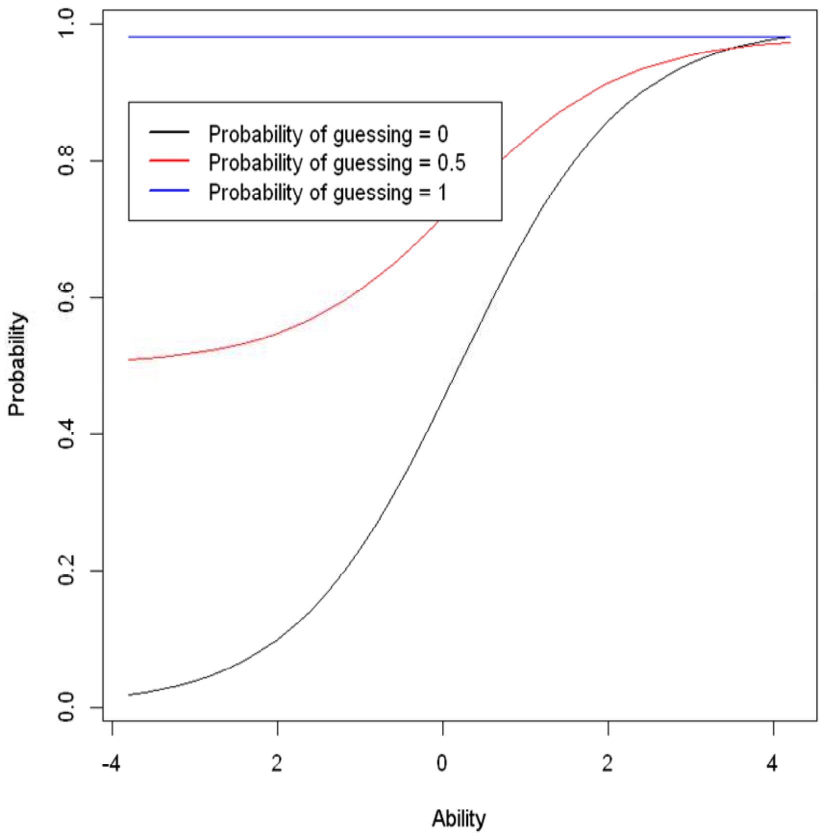
Non avendo ritenuto questa informazione rilevante per il docente ho preferito non adottare questo modello.

- Infine vi è il "Modello a 3 parametri" (3PL), il quale si differenzia da quello precedente per l'aggiunta del fattore χ_j, il quale determina la probabilità di indovinare ogni esercizio.

$$p(x_{ij} = 1|\theta_i, \beta_j, \alpha_j, \chi_j) = \chi_j + (1 - \chi_j)\frac{e^{\alpha_j(\theta_i - \beta_j)}}{1 + e^{\alpha_j(\theta_i - \beta_j)}} \tag{2.3}$$

Come si deduce dall'equazione se tale parametro vale 0 per ogni item, cioè non vi è la possibilità di indovinare alcun quesito, allora questo modello coincide con il precedente, invece se valesse sempre 1 si avrebbe sempre la possibilità di dare la risposta corretta con il 100% della probabilità.

χ_j essendo una probabilità assume solo valori compresi tra 0 ed 1.



Osservando il grafico è facile comprendere l'enorme impatto di χ_j sul calcolo della probabilità, arrivando ad essere costante quando assume valore 1.

Per una domanda a risposta chiusa χ_j assume valore pari a $\frac{1}{n}$, dove n è il numero di risposte possibili.

In presenza di una domanda a risposta aperta è più difficile definire il significato di tale parametro, anche se si potrebbe ipotizzare che il modo in cui viene posta la domanda stessa possa precludere un aiuto sulla riposta attesa.

Quindi, non ho ritenuto utile introdurre questo parametro perché è complicato definire il significato di "indovinare" relativamente ad un esame di programmazione, uno studente in possesso di esempi di input e output potrebbe definire funzioni che ritornano delle costanti e quindi superare i test case a sua disposizione ma anche in questo caso gli esercizi non risulterebbero corretti di fronte a test case differenti o al controllo diretto del codice da parte del docente, quindi l'esame non potrebbe essere ritenuto corretto.

Successivamente a questa fase è stato necessario definire gli stimatori di difficoltà e abilità, ovvero strumenti il cui scopo è quello di fornire una stima di un certo parametro, le stime derivano da variabili aleatorie e lo sono anche esse stesse. [4]

Un problema rilevante che ho dovuto affrontare è legato alla presenza di un campione di dati molto ristretto da cui dover trarne i risultati, dato che spesso un numero di studenti molto basso partecipa agli esami oppure partecipa attivamente ai laboratori e questo compromette la convergenza e attendibilità dell'algoritmo di stima.

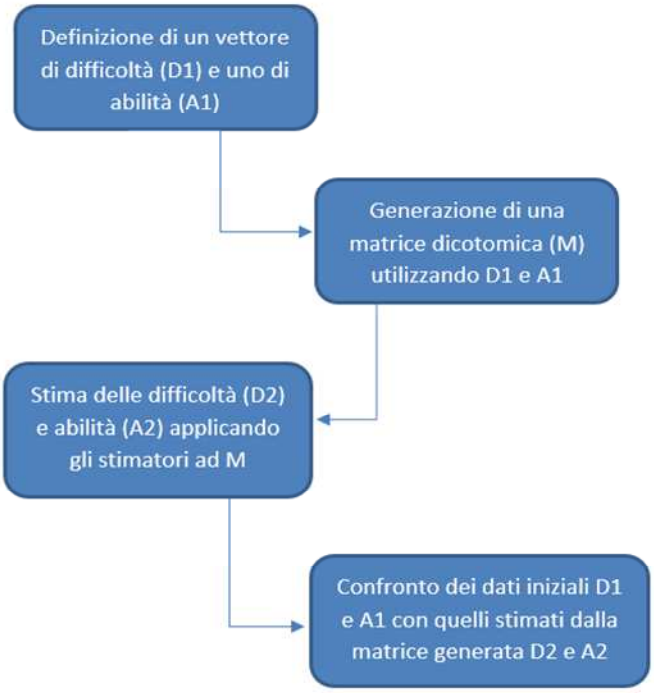
Per risolvere il problema ho usufruito di un metodo di ricampionamento detto bootstrap. [5]

Il bootstrap è una tecnica statistica di ricampionamento con reimmissione, quindi nel caso di un campione di n elementi vi sarà l'estrazione di m di essi dove $m > n$, senza l'eliminazione di un dato dall'insieme dopo la sua scelta.

Ogni elemento può essere estratto con probabilità pari a $\frac{1}{n}$.

In questo modo è possibile ottenere campioni di dati con centinaia di soggetti partendo da poche decine, comunque, in presenza di un numero esiguo di osservazioni anche questo metodo perde di efficacia perché i dati che ne ricaveremmo potrebbero essere eccessivamente, se non totalmente, uniformi.

Per provare la correttezza degli strumenti di stima impiegati e quindi dimostrare agli utilizzatori la loro affidabilità ho considerato il seguente procedimento:



Quindi l'obiettivo successivo è stato quello di definire un generatore di matrici dicotomiche la cui distribuzione dei dati dipenda dall'aspetto degli input utilizzati.

Il metodo che ho utilizzato per calcolare se in ogni locazione m_{ij} della matrice M vada inserito 0 o 1 può essere definito in questo modo:

$$m_{ij} = rbinom(n = 1, size = 1, prob = pl1(ability_i, difficulty_j)) \tag{2.4}$$

In breve, calcolo la probabilità che il soggetto di abilità i-esima risolva l'esercizio di difficoltà j-esima mediante il modello adottato (2.1) e successivamente passo questo risultato come argomento alla funzione *rbinom* il cui compito è quello di generare randomicamente 1 valore (n) di taglia 1 (*size*), quindi un intero compreso tra 0 ed 1 estremi inclusi, basandosi sulla probabilità (*prob*) data.

Infine vi è la fase di validazione delle stime ottenute per dimostrare l'affidabilità degli strumenti utilizzati; nel progetto ho deciso di presentarne 2:

- Validazione per differenza: calcolo lo scostamento in valore assoluto di ogni valore iniziale con la sua corrispettiva stima attraverso una sottrazione, tralasciando il segno del risultato.

  Tanto più l'esito è prossimo allo 0 e maggiore è la precisione dello stimatore.

- Validazione "grafica", ottenibile confrontando i valori attraverso un diagramma cartesiano, maggiore è la sovrapposizione dei grafici e migliore potrà considerarsi lo stimatore impiegato.

2.2 Supporto software

2.2.1 Linguaggio utilizzato

Per la realizzazione del progetto ho scelto di utilizzare esclusivamente R, conoscevo già questo strumento ed ero al corrente delle sue potenzialità nell'analisi statistica dei dati.

R ha una sintassi semplice e dalla facile lettura e fornisce un'ampia varietà di tecniche statistiche e grafiche ed è altamente estensibile grazie alla sua natura Open Source.

Uno dei punti di forza di R è la facilità con cui è possibile produrre grafici di qualità includendo simboli e formule matematiche dove si ritiene necessario.

R è disponibile come Free Software sotto i termini della GNU General Public License della Free Software Foundation sotto forma di codice sorgente.

Questo linguaggio può essere compilato ed eseguito su una vasta gamma di piattaforme UNIX e sistemi simili (inclusi FreeBSD e Linux), Windows e MacOS. [6]

Per la realizzazione del progetto ho constatato personalmente l'incredibile supporto che esso riceve dagli sviluppatori che mettono a disposizione un enorme quantitativo di pacchetti per ogni esigenza e lavoro e della disponibilità degli utilizzatori che, attraverso i forum, offrono tempestivamente aiuto e consigli.

Nel progetto ho sfruttato la possibilità di vettorizzare le operazioni per aumentare la leggibilità del codice ed evitare di ricorrere a cicli o operazioni inutilmente complesse.

Mostro un esempio di tale caratteristica, creando una matrice(2x4) di numeri da 1 a 8 ed un vettore di valori da 1 a 4 e poi effettuo la divisione tra ogni colonna ed il valore nella corrispettiva posizione del vettore:

```
m <- matrix(data = c(1:8), nrow = 2)
m
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8

v <- c(1:4)
v
[1] 1 2 3 4

t((t(m)) / v)
     [,1] [,2]     [,3] [,4]
[1,]    1  1.5 1.666667 1.75
[2,]    2  2.0 2.000000 2.00
```

Ho potuto sfruttare le potenzialità del linguaggio anche quando ho avuto l'occasione di gestire delle matrici tridimensionali, ovvero tabelle indicizzate da 3 parametri: la matricola dello studente, il nome dell'esercizio e il parametro di valutazione.

- Esempio 1, mostro le valutazioni ottenute da uno studente in un certo esame

	uploaded	compiles	errors	diffs	oks
01-censura_la_prima	1	1	0	0	5
02-studiare_stanca	1	1	0	4	1
03-piramide_di_numeri	1	1	0	0	5
04-disegna_serpente	1	1	0	0	4
05-stampante_configurabile	1	1	0	5	0
06-righello_orizzontale	1	0	NA	NA	NA

- Esempio 2, nello stesso esame voglio conoscere il numero di test case passati per ogni studente:

	01-censura_la_prima	02-studiare_stanca	03-piramide_di_numeri	04-disegna_serpente	05-stampante_configurabile	06-righello_orizzontale
800000	0	NA	NA	NA	NA	NA
800001	5	NA	NA	NA	NA	NA
800002	5	5	5	4	1	NA
800003	NA	NA	NA	NA	NA	NA
800004	5	3	5	4	NA	NA
800005	NA	NA	NA	NA	NA	NA
800006	5	5	NA	0	NA	NA
800007	0	NA	NA	NA	NA	NA
800008	5	1	NA	3	NA	NA
800009	5	NA	NA	NA	NA	NA
800010	5	5	5	3	0	NA
800011	5	5	5	4	5	2
800012	5	5	5	4	0	NA
800013	5	2	NA	NA	NA	NA
800014	5	NA	NA	NA	NA	NA
800015	5	1	5	4	0	NA

Per ovvie ragioni negli esempi ho usato numeri di matricola fittizi.

2.2.2 Package adottati

Oltre alle funzioni già presenti in R e a quelle definite da me, ho ritenuto opportuno utilizzare il pacchetto *ltm*, questa libreria di funzioni sviluppata da Dimitris Rizopoulos, professore di biostatistica all'Erasmus Medical Center di Rotterdam, permette l'analisi di dati dicotomici e politomici attraverso l'approccio dell'Item response theory. [7]

Per l'analisi dei dati dicotomici sono state implementate diverse funzioni: Rasch, il modello logistico a 2 parametri ed il modello a 3 parametri di Birnbaum, invece per i dati politomici è disponibile il modello graduato di Samejima.

La stima dei parametri è ottenuta attraverso funzioni di massima verosimiglianza marginale utilizzando la regola di quadratura di Gauss-Hermite.

All'interno dello stesso pacchetto vi sono numerosi esempi eseguibili attraverso l'uso di dati reali resi disponibili.

2.2.3 Processo di sviluppo

In quanto programmatore è mio compito produrre un codice funzionante, efficiente ma soprattutto comprensibile agli utilizzatori che vorranno analizzarlo ed, eventualmente, modificarlo.
Per tale ragione, applicando i consigli del professore, ho provveduto ad adottare utili metodi di sviluppo:

- Coding style, cioè metodi di codifica come il *naming*, ovvero l'utilizzo di nomi espliciti e comunicativi per il compito assegnato a variabili e funzioni, l'eliminazione e la relativa trasformazione in valori parametrici per le funzioni di quelli che il docente stesso ha definito più volte come "magic number", ovvero valori costanti fissati all'interno di funzioni che non posso essere modificati all'esterno e che, oltre a poter alterare il comportamento di una funzione rispetto a ciò che l'utente si aspetterebbe, limitano le potenzialità del programma stesso.

Un altro fondamentale consiglio fornitomi è stato quello di applicare la modularità nello sviluppo delle funzionalità, cioè suddividere il percorso di sviluppo del software in moduli, essi sono componenti indipendenti tra loro. [8]

Questo paradigma di programmazione semplifica notevolmente lo sviluppo del progetto perché permette agli sviluppatori si scomporre problemi complessi in parti semplici e più facilmente risolvibili, migliora l'identificazione degli errori perché ogni modulo è testabile separatamente e agevola anche la fase di manutenzione del codice specialmente se eseguita da persone differenti dagli sviluppatori iniziali.

- Introdurre la documentazione prodotta tramite Roxygen, al fine di fornire informazioni precise e dalla facile lettura per gli utenti sulle funzioni realizzate. [9]

  Roxygen è un programma per la generazione automatica della documentazione a partire dal codice sorgente scritto in R.

  Per usarlo è sufficiente far precedere alle funzioni le informazioni relative ad esse come in questo esempio:

```
#' Add together two numbers
#'
#' @param x A number
#' @param y A number
#' @return The sum of x and y
#' @examples
```

```r
#' add(1, 1)
#' add(10, 1)
add <- function(x, y){
        x + y
}
```

Attraverso questo strumento verrà generato codice LaTex, all'interno di file .Rd, contenente tutte le informazioni descritte per ogni funzione definita.

Questa documentazione potrà essere consultata installando il relativo package e richiamandola attraverso il comando *help(nomeFunzione)*, oppure, nel caso si usasse RStudio, con il relativo gestore di pacchetti.

- Uso di *Git* per condividere lo svolgimento del progetto con il professore e per tutelarmi dalla possibile perdita del lavoro.

  Avendo lavorato singolarmente per questa tesi non ho potuto sfruttare completamente le sue potenzialità.

  Git è un servizio il cui obiettivo è fornire supporto di controllo versione ai programmi in sviluppo e permette la facile creazione di gruppi di lavoro, la definizione dei diritti di lettura e scrittura sul software e soprattutto la realizzazione di branch di sviluppo paralleli sui quali sviluppare nuove funzionalità e successivamente integrarle al ramo di lavoro principale senza lavorare direttamente e costantemente su esso e quindi rischiando di generare conflitti ed errori sul codice prodotto.

Capitolo 3

Un esempio di analisi svolta tramite lo strumento

In questo capitolo mostrerò e analizzerò le fasi dello sviluppo del progetto, spiegherò le funzioni implementate, i problemi affrontati e le conclusioni tratte dall'analisi dei dati.

3.1 Dati simulati

3.1.1 Implementazione del modello

Nella prima fase di progettazione ho definito la funzione *pl1* che corrisponde all'implementazione dell'analogo modello (2.1), essa assume come dati in input difficoltà e abilità dello studente e determina con quale probabilità riuscirà a risolvere correttamente un determinato esercizio.

Definita questa funzione è stato semplice sfruttarla all'interno di *calcProbabilities* per generare una tabella di probabilità, passati come input: un vettore di difficoltà, un vettore di abilità e un dataframe da cui estrapolare i nomi delle colonne e delle righe per facilitare la lettura dell'output.

	Esercizio 1	Esercizio 2	Esercizio 3	Esercizio 4	Esercizio 5
Studente 1	0.1977711	0.7549859	0.5616473	0.3717282	0.5910636
Studente 2	0.1939622	0.7504849	0.5556847	0.3660979	0.5852056
Studente 3	0.2739647	0.8250679	0.6622943	0.4752389	0.6886999
Studente 4	0.3652185	0.8779204	0.7493877	0.5799794	0.7713346
Studente 5	0.2716985	0.8234131	0.6597347	0.4723910	0.6862457
Studente 6	0.2470063	0.8039274	0.6302970	0.4404911	0.6579123
Studente 7	0.3523574	0.8718007	0.7387422	0.5663025	0.7613245
Studente 8	0.2399032	0.7977769	0.6212658	0.4310089	0.6491802
Studente 9	0.1007087	0.5832897	0.3678991	0.2118348	0.3963428
Studente 10	0.1538992	0.6945179	0.4859522	0.3038848	0.5160713

Per migliorare la lettura della tabella ho introdotto la funzione *showProbabilities*.

	Esercizio.1	Esercizio.2	Esercizio.3	Esercizio.4	Esercizio.5
Studente 1	19 %	75 %	56 %	37 %	59 %
Studente 2	19 %	75 %	55 %	36 %	58 %
Studente 3	27 %	82 %	66 %	47 %	68 %
Studente 4	36 %	87 %	74 %	57 %	77 %
Studente 5	27 %	82 %	65 %	47 %	68 %
Studente 6	24 %	80 %	63 %	44 %	65 %
Studente 7	35 %	87 %	73 %	56 %	76 %
Studente 8	23 %	79 %	62 %	43 %	64 %
Studente 9	10 %	58 %	36 %	21 %	39 %
Studente 10	15 %	69 %	48 %	30 %	51 %

3.1.2 Simulazione dei dati

Successivamente è stato necessario possedere dei dati su cui applicare gli stimatori e iniziare ad effettuare i primi esperimenti.

Purtroppo non possedendo ancora i dati reali del docente e anche per poter disporre di valori precisi con cui confrontare le stime ottenute ho ritenuto necessario definire la funzione *generateDichot* il cui compito è simulare una tabella dicotomica basandosi sui valori di difficoltà e abilità passati: in ogni locazione della tabella ho assegnato un valore basandomi sulla formula definita nel capitolo precedente (2.4).

Per generare randomicamente i valori da passare alla funzione di simulazione dei dati ho scelto una distribuzione normale, questa scelta d'uso è comunemente utilizzata per simulare dei dati reali che si concentrano attorno a un singolo valor medio disperdendosi con una certa varianza. [3]

Esempio, simulo dei dati basandomi su una distribuzione normale:

X1	X2	X3	X4	X5
1	0	1	1	1
0	1	1	1	0
0	1	0	0	1
1	1	1	0	1
1	1	1	1	1
0	1	0	0	1
1	1	0	1	1
1	1	1	0	1
0	0	1	0	1
1	1	1	1	1

Tramite questa funzione posso compiere diversi tipi di esperimenti perché mi permette di simulare situazioni di vario tipo, ad esempio potrei assumere che l'abilità degli studenti è media e costante e la difficoltà degli esercizi crescente:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
1	1	1	0	1	0	0	0	1	0	0
1	1	1	1	0	0	0	1	0	0	0
1	0	1	0	0	0	1	0	0	0	0
1	1	1	1	1	0	1	1	0	0	0
1	1	1	1	1	0	1	0	0	1	0
0	1	0	0	1	0	0	0	1	0	0
1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	1	1	0	0	0	0
1	1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	1	0	0	1
1	1	1	1	0	0	1	0	0	0	0
1	1	1	1	1	1	1	0	1	0	0
0	1	1	1	1	1	1	0	1	0	0
1	0	0	1	0	1	1	0	0	1	0

Come previsto, ho rilevato una maggiore presenza di esercizi risolti correttamente nelle prime colonne e un alto numero di fallimenti nelle ultime.

3.1.3 Stima dei parametri

In questa fase ho realizzato gli strumenti principali del progetto, gli stimatori.

Essi sono di 2 tipi: stimatori di difficoltà e stimatori di abilità, della prima tipologia ho realizzato 2 versioni differenti per gestire l'impatto del discriminante sui valori ottenuti.

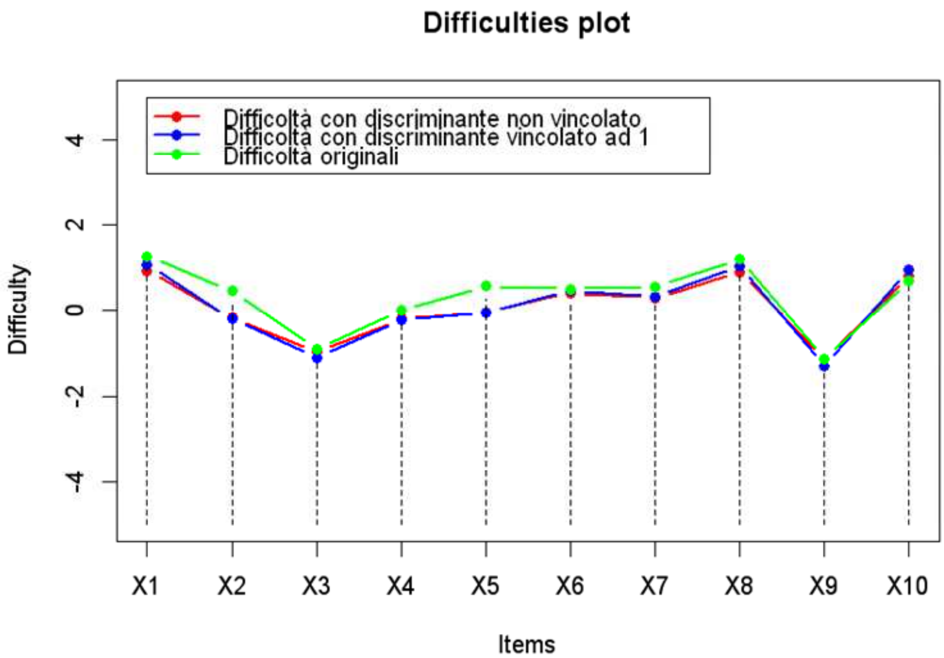
Nel seguente esempio genero una tabella dicotomica utilizzando dei valori di abilità e difficoltà ottenuti randomicamente da una distribuzione normale di media 0 e varianza 1 e confronto le stime calcolate vincolando il valore del discriminante ad 1 oppure lasciando che venga stimato e posto come valore uniforme per ogni esercizio; le funzioni si chiamano rispettivamente *estimateDifficultiesDichotOneDiscrim* e *estimateDifficultiesDichotFreeDiscrim*.

Prima di usare gli stimatori applico alla tabella la funzione *bootstrapping* ottenendone una di dimensioni 10 volte superiore a quella iniziale.

Ho ritenuto adeguato tale valore perché solitamente non più di 30-40 studenti partecipano ad esami e laboratori ed evito così di dover analizzare dati troppo piccoli che renderebbero le stime di dubbia affidabilità.

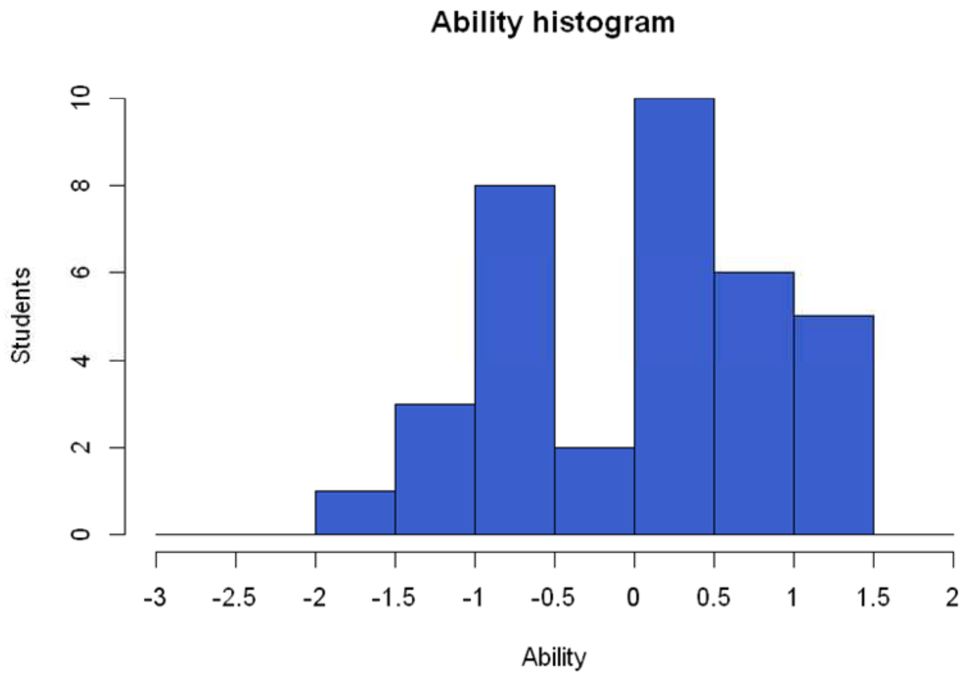
Infine confronto graficamente i valori corretti di difficoltà, ovvero quelli con cui ho generato i dati dicotomici, con le stime calcolate.

Per permettere una migliore visualizzazione grafica ho definito la funzione *plotDifficultiesDichot* che, oltre a disegnare un diagramma cartesiano, pone sull'asse delle ascisse i nomi degli esercizi e disegna delle linee tratteggiate per facilitare l'associazione dei nomi con i punti.



Le stime calcolate con entrambi i metodi sono pressoché equivalenti ed entrambe si avvicinano molto ai valori reali.

Per visualizzare graficamente le stime delle abilità ottenute ho usufruito di un istogramma che è un grafico costituito da più rettangoli adiacenti, ognuno dei quali ha per base un certo intervallo della variabile e un'altezza che rappresenta il numero di frequenze registrate.



Poiché per generare i dati ho utilizzato una distribuzione normale è comprensibile che il grafico delle difficoltà e l'istogramma presentino l'aspetto ottenuto.

Prima di procedere con la fase successiva ho iterato le precedenti e ciò è stato necessario per perfezionare gli strumenti definiti.

3.1.4 Validazione

Una volta prodotti gli stimatori ho definito le funzioni per implementare le tecniche di validazione descritte nel capitolo precedente.

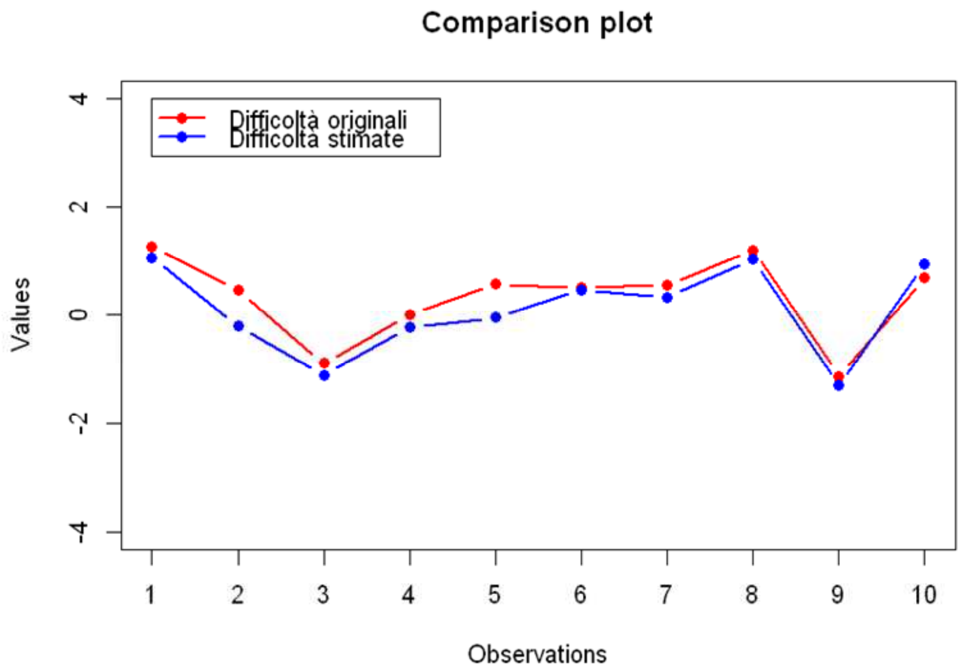
- **Validazione per differenza**

Difficoltà originali	Difficoltà stimate	Scostamento
1.267897305	1.0685922	0.20
0.457093314	-0.1988227	0.66
-0.884828120	-1.1035202	0.22
0.003832331	-0.2275306	0.23
0.569932522	-0.0418142	0.61
0.509092729	0.4725934	0.04
0.555349234	0.3283189	0.23
1.206293440	1.0360098	0.17
-1.123092404	-1.2966803	0.17
0.701728770	0.9399469	0.24

Dalla tabella rilevo che su 10 esercizi 2 presentano uno scostamento di circa 0.6 e i rimanenti si differenziano per meno di 0.3; data la bassa differenza posso affermare che il mio stimatore è valido.

Questa tecnica è semplice e di facile lettura se gli esercizi considerati sono in numero non eccessivamente grande.

- **Validazione grafica**



Il confronto grafico è il metodo più immediato per valutare la validità dei dati ottenuti, esso è applicabile sia in presenza di una grande quantità di esercizi che di uno scarso numero di essi.

Dal grafico posso dedurre che le linee dei valori tendono a coincidere, confermando nuovamente la correttezza delle stime ottenute.

3.1.5 Gestione di dati mancanti

Procedendo nello sviluppo del progetto mi sono imbattuto in un problema che sorge frequentemente quando si ha a che fare con questo tipo di informazioni, ovvero la presenza di dati assenti dovuti al mancato invio dell'esercizio da parte dello studente.

Per simulare questa situazione ho definito la funzione *replaceRandomZeroWithNA* il cui compito è rimpiazzare alcuni valori 0 con il dato assente NA, la quantità di rimpiazzamenti compiuta è basata su una probabilità passata come argomento alla funzione, se essa è 0 non verrà compiuta alcuna sostituzione, viceversa se essa è 1 ogni valore 0 sparirà.

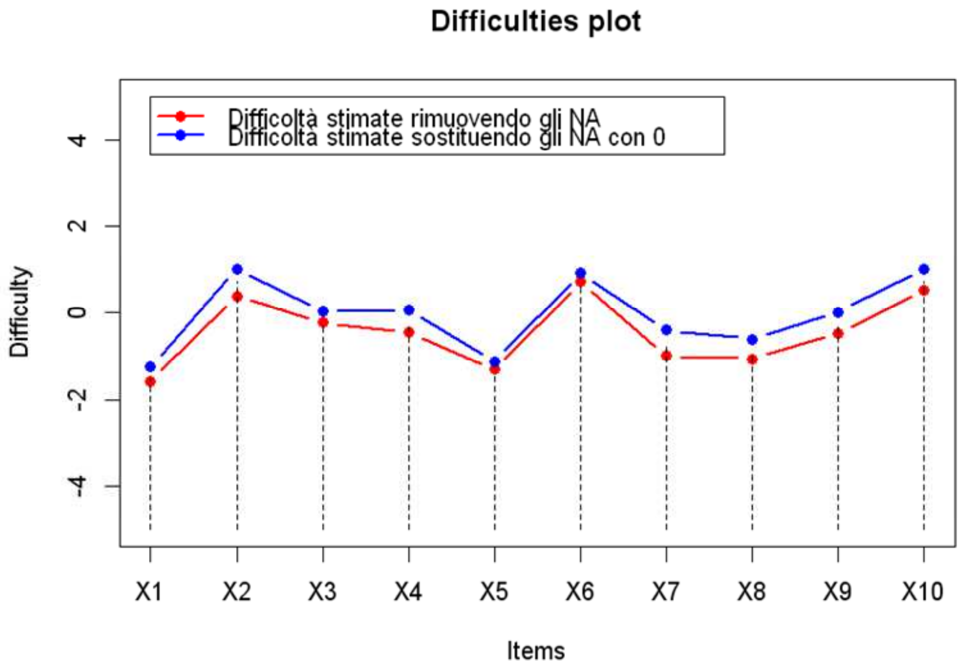
Ho scelto di effettuare il rimpiazzamento solo sugli 0 perché ho supposto che uno studente in possesso di un esercizio corretto non avrebbe motivo di non effettuare l'upload, invece nel caso sia consapevole dell'inesattezza di esso potrebbe scegliere ugualmente di caricarlo oppure di non farlo dato che non influirebbe sul voto finale dell'esame o, peggio, potrebbe abbassarlo.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1	1	0	0	1	0	1	1	0	1
1	0	0	1	0	0	0	0	1	0
1	0	1	1	1	0	0	0	0	0
1	0	1	1	1	0	1	1	0	0
1	0	1	0	1	1	1	0	0	1
1	1	0	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	0
1	1	1	1	1	0	0	1	1	0
1	0	1	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1	1	0

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1	1	0	NA	1	0	1	1	0	1
1	0	0	1	0	0	NA	NA	1	NA
1	NA	1	1	1	NA	0	0	0	0
1	NA	1	1	1	0	1	1	NA	0
1	0	1	0	1	1	1	NA	0	1
1	1	0	1	1	1	1	1	1	1
1	1	1	NA	1	0	1	1	1	NA
1	1	1	1	1	0	NA	1	1	0
1	NA	1	1	1	1	1	1	1	1
NA	0	1	1	1	NA	1	1	1	NA

Adesso che possiedo dei dati che rispecchiano il problema, procedo utilizzando 2 tecniche che lo trattano in modo differente.

Il primo metodo rimuove i valori assenti prima della computazione passando *NULL* al parametro *forNA* della funzione di stima, il secondo sostituisce ogni valore *NA* con 0 per mezzo della funzione *replaceNaWithZero*, assumendo che ogni esercizio non pervenuto sia errato.



Dato che il secondo metodo registra più esercizi non superati il grafico presenta una difficoltà superiore ma non eccessiva, almeno per la presenza del 30% in più di esercizi falliti.

Quindi posso constatare che per un numero limitato di valori assenti non noto grandi differenze nell'uso dei metodi definiti ma in situazioni diverse potrei doverli utilizzare valutando i rischi derivanti dall'uso di uno piuttosto che l'altro.

3.2 Dati reali

3.2.1 Utilizzo di dati reali

Ricevuti i dati da parte del docente ho potuto iniziare la fase di analisi effettiva.

Questi dati hanno un aspetto politomico perché sono memorizzati all'interno di tabelle tridimensionali indicizzate dalle matricole, i nomi degli esercizi ed i criteri valutativi.

Per ogni studente ed ogni esercizio conosco se è stato effettuato l'upload, se compila, il numero di errori in esecuzione, quanti sono i test case corretti e quanti presentano differenze dal risultato atteso.

Nonostante la loro forma sia già predisposta per un'analisi di tipo politomico, ho ritenuto necessario trasformarli in forma dicotomica, semplificando il loro studio a causa della natura del progetto, cioè una tesi di laurea triennale, e dei tempi ristretti.

Per rendere possibile questo passaggio ho creato una funzione che, definita una soglia, assegna 0 agli esercizi che non raggiungono il grado di correttezza indicato ed 1 a quelli che lo raggiungono.

Questi valori indicano rispettivamente il fallimento e superamento dell'esercizio.

Ad esempio, se la soglia fosse il 50% dei test case corretti la funzione attribuirebbe 0 quando non è stato effettuato l'upload, non compila, vi sono errori di esecuzione o il numero di test case che non presentano differenze dall'output aspettato è inferiore alla metà del totale per l'esercizio considerato.

La funzione per effettuare questo passaggio prende il nome di *evaluateSource*.

Di seguito mostro un esempio della sua applicazione valutando l'esame del 16 febbraio 2017 impostando come soglia di superamento di ogni esercizio il 50% dei test case corretti.

01-abbastanza_risparmi	02-parole_incatenate	03-u_inscatolate	04-nave_piu_lunga
5	5	5	NA
2	NA	NA	NA
5	5	5	9
0	NA	NA	NA
5	5	5	0
3	3	0	NA
5	5	5	9
2	NA	NA	NA
5	5	NA	NA
5	5	5	NA
5	5	5	NA
5	5	5	9

01-abbastanza_risparmi	02-parole_incatenate	03-u_inscatolate	04-nave_piu_lunga
1	1	1	0
0	0	0	0
1	1	1	1
0	0	0	0
1	1	1	0
1	1	0	0
1	1	1	1
0	0	0	0
1	1	0	0
1	1	1	0
1	1	1	0
1	1	1	1

Adesso posso applicare gli stimatori definiti sulle tabelle dicotomiche ottenute.

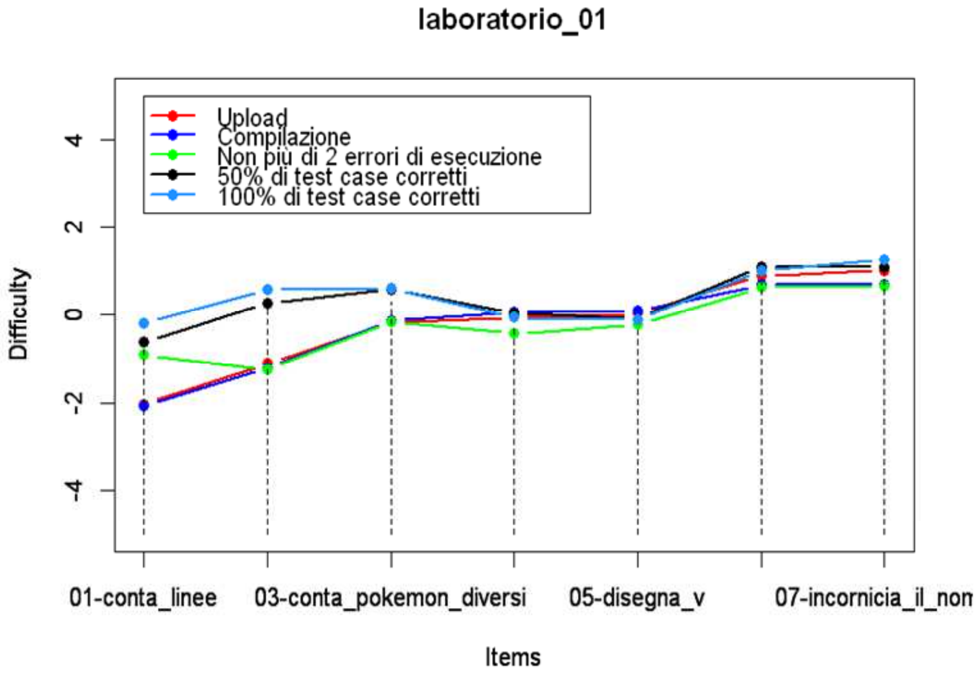
3.2.2 Formulazione di ipotesi sotto l'approccio dell'IRT

Come passo successivo visualizzo e confronto i grafici relativi alle difficoltà stimate considerando 5 casi differenti:

- E' stato effettuato l'upload
- La compilazione non presenta errori
- Sono stati registrati meno di 2 errori di esecuzione
- Almeno il 50% dei test case sono corretti
- Tutti i test case sono stati superati con successo.

Avendo notato che le sessioni relative agli esami presentano un elevatissimo numero di valori assenti ed esercizi incorretti ho preferito concentrare l'analisi unicamente sui laboratori.

Valuto i dati del primo laboratori attraverso i 5 criteri definiti, stimo le difficoltà delle altrettante tabelle dicotomiche ottenute e confronto graficamente le stime calcolate.



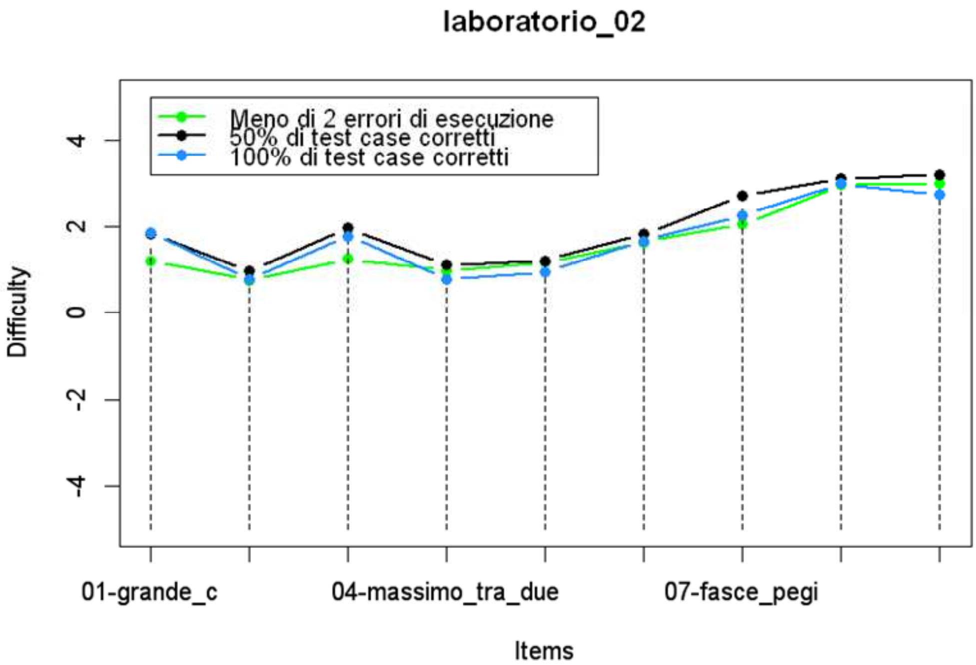
Dal grafico si possono trarre diverse conclusioni:

- Noto che le stime legate ai primi 3 criteri hanno un'elevata sovrapposizione grafica, da ciò deduco che gli studenti consegnano gli esercizi solo quando sono fortemente convinti della loro correttezza.

- Le difficoltà legate alle percentuali di test case corretti non hanno un andamento crescente, anzi subiscono basse variazioni nell'itero laboratorio, quindi il docente potrebbe non aver assegnato gli esercizi prevedendo un aumento dell'impegno da parte degli studenti.

- Nella fase iniziale le stime per test case si discostano abbastanza dalle altre, sovrapponendosi ad esse successivamente, questo potrebbe essere indicativo del fatto che gli studenti all'inizio sono meno abili, forse a causa di spiegazioni poco chiare del docente.

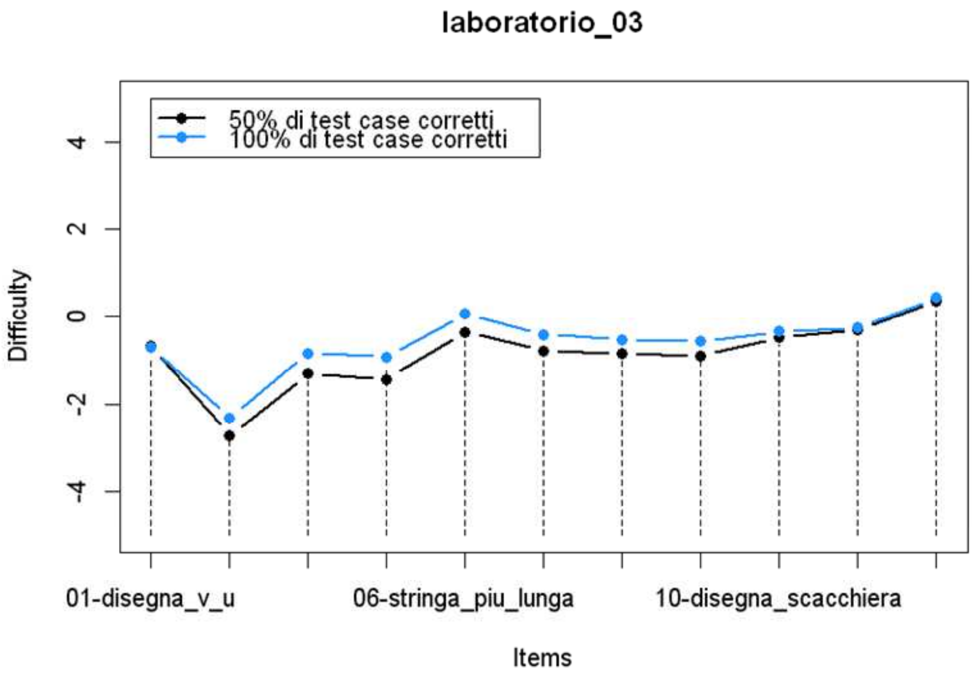
Analizzo la seconda sessione di laboratorio tralasciando i criteri di valutazione meno rilevanti.



Anche nel secondo laboratorio si può notare una sovrapposizione tra le stime considerando la presenza di meno di 2 errori di esecuzione e il raggiungimento del 50% e 100% di test case corretti, noto che il grafico non ha un andamento crescente ma è quasi sempre fisso su una difficoltà medio-alta.

Possiamo dedurre che gli studenti hanno riscontrato particolari difficoltà in tutti gli esercizi, negando nuovamente l'ipotesi sul livello di difficoltà crescente con l'avanzare del numero dell'esercizio.

Analizzo la terza sessione di laboratorio.

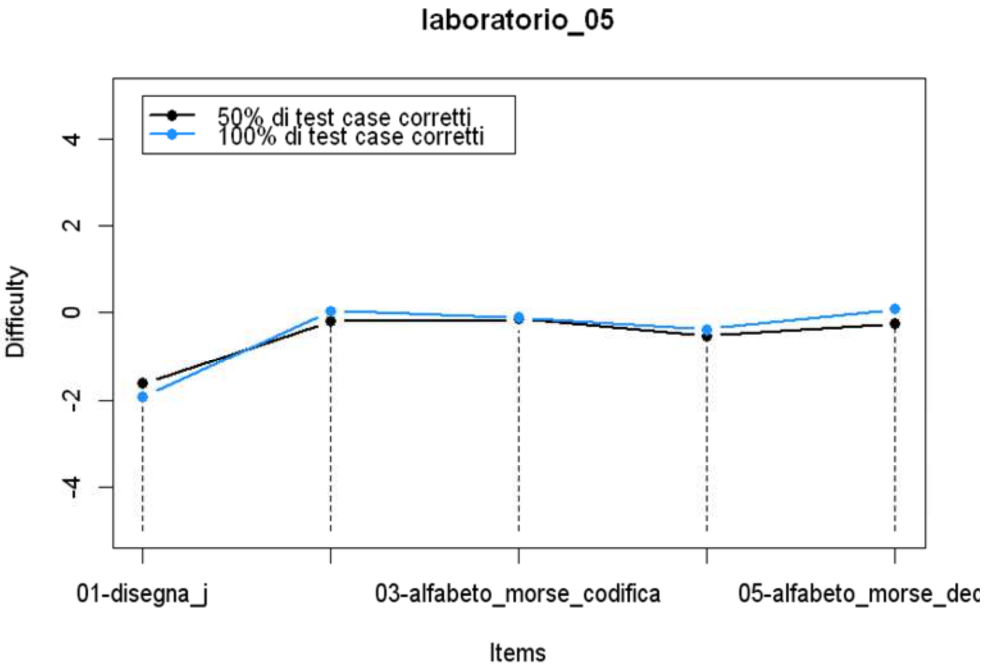


In questa sessione è nuovamente presente una sovrapposizione tra i grafici delle stime, entrambe registrano un calo di difficoltà dopo il primo esercizio e una successiva crescita fino al quinto.

Dopodiché vi è un leggero calo di difficoltà che si assesta su un livello quasi costante fino al penultimo esercizio, terminando con una leggera crescita.

Il livello delle difficoltà riscontrato rileva che ad eccezione dei primi esercizi, in particolar modo del secondo, è stato mantenuto un grado costante di difficoltà.

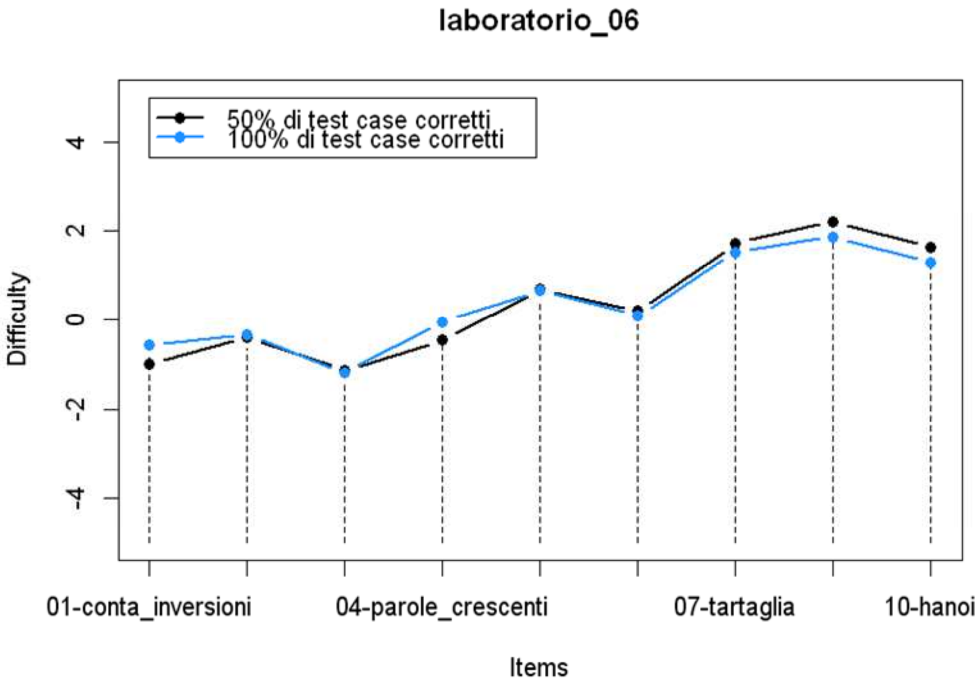
Analizzo la quinta sessione di laboratorio.



Ad eccezione del primo esercizio che si rivela essere più semplice degli altri, nell'intera sessione di laboratorio è visibile un livello medio e costante di difficoltà, senza la presenza né di picchi né di drastici cali.

Potrei ipotizzare che questa situazione è indicativa di un corretto livello di difficoltà degli esercizi assegnati da parte del docente che non causa impedimenti agli studenti in possesso di un basso livello di abilità o eccessive agevolazioni per gli studenti più capaci.

Analizzo la sesta ed ultima sessione di laboratorio.



In quest'ultima analisi rilevo, oltre ad una ancor più marcata sovrapposizione dei grafici, un livello di difficoltà altalenante con numerose crescite e diminuzioni.

Questa situazioni mi impedisce di formulare ipotesi precise sull'accaduto dato che non è presente un livello di difficoltà sempre crescente o costante.

In conclusione alle analisi applicate posso asserire che gli studenti consegnano gli esercizi principalmente quando questi risultano corretti e adeguatamente testati, cioè quando passano la maggior parte dei test case.

Inoltre non posso affermare con certezza né che gli esercizi siano assegnati con un ordine crescente di difficoltà né che siano stati concepiti per essere ugualmente complessi.

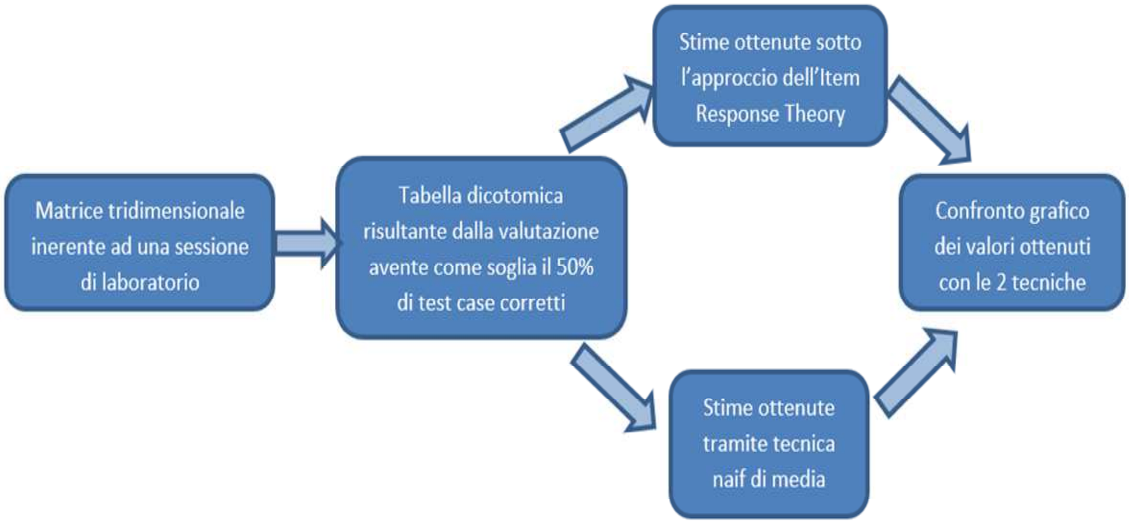
In ogni caso non è mai stato rilevato un livello di difficoltà totalmente decrescente nell'intera sessione di laboratorio.

3.2.3 Confronto IRT con tecnica naif

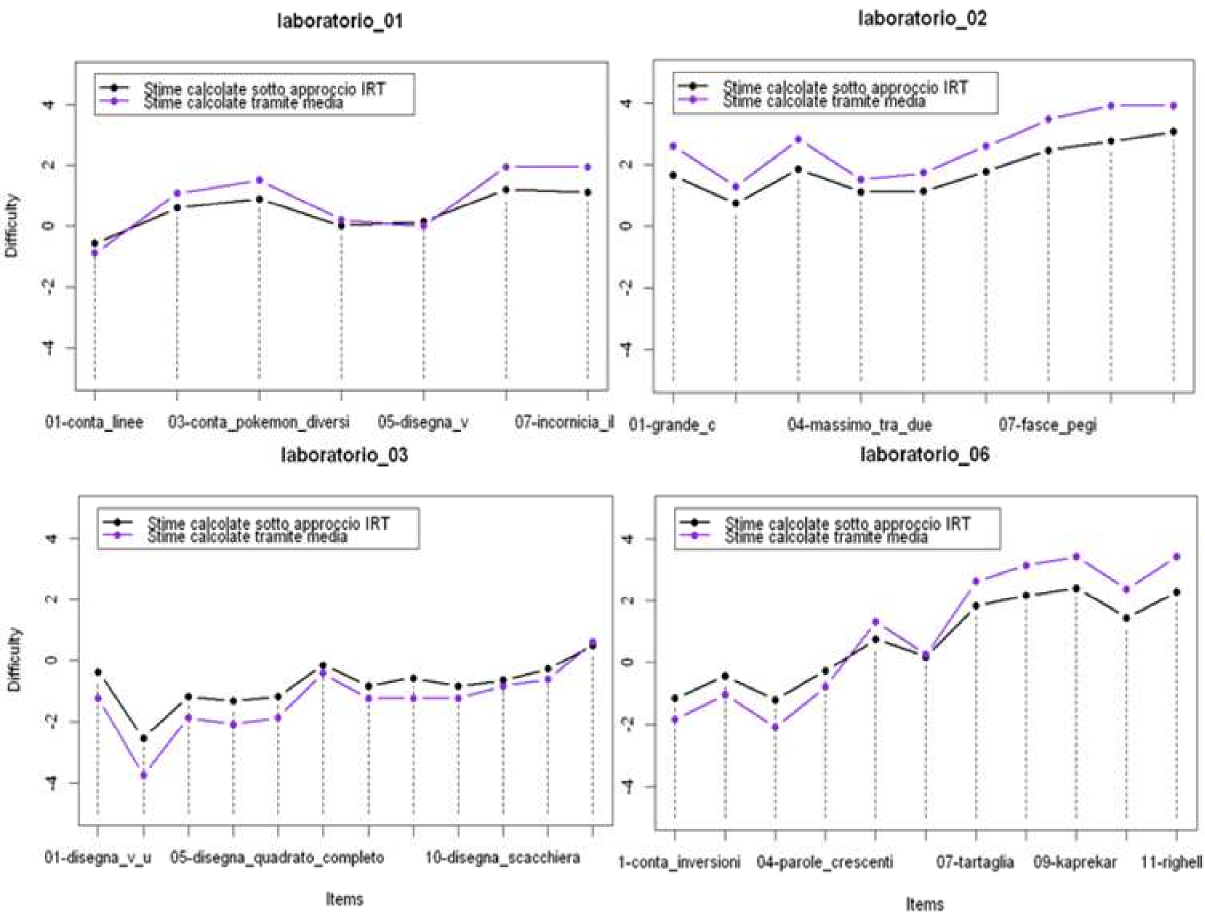
Eseguo un confronto tra le difficoltà stimate sotto l'approccio dell'Item Response Theory e un metodo di stima naif, effettuo ciò per osservare le differenze e i vantaggi ottenuti dalle tecniche di stima sfruttate.

Come metodo naif ho scelto di utilizzare la media, confronto graficamente le stime ottenute dalla valutazione delle sessioni di laboratorio, impostando come soglia di superamento almeno il 50% di test case corretti, con le medie calcolate dalle stesse matrici dicotomiche sommando i valori delle colonne e dividendoli per il totale degli studenti che vi hanno partecipato.

Lo schema è il seguente:



Visualizzo i grafici ottenuti confrontando le stime:



I valori ottenuti dalla tecnica naif (linee viola) tendono a coincidere con le stime calcolate sotto l'approccio dell'IRT (linee nere) anche se non sono mai totalmente equivalenti.

Questa similitudine potrebbe essere imputata all'uso di un modello poco complesso (2.1), quindi sfruttando tecniche di stima più efficienti si avrebbero risultati differenti perché conterrebbero un maggior numero di informazioni che vengono tralasciate nel semplice calcolo della media dei valori.

Considerando il sesto laboratorio è possibile notare un maggiore scostamento tra le stime degli ultimi esercizi, ciò avviene perché le stime calcolate sotto l'approccio dell'IRT considerano anche il livello di abilità degli studenti.

Questa evidente differenza è causata dal fatto che i soggetti che non sono stati in grado di superare quegli esercizi sono coloro che hanno dimostrato di aver uno scarso livello di abilità e quindi hanno avuto un impatto minore sulla misura della difficoltà, abbassando quindi le stime ottenute tramite IRT.

Capitolo 4

Sviluppi futuri e problemi aperti

Ci sono numerosi aspetti e funzionalità che possono essere implementate per fornire un maggiore supporto agli utenti.

In prima istanza si potrebbe introdurre nuovi stimatori per supportare i **modelli** a 2 e 3 paramenti, quindi estendere l'analisi al calcolo del discriminante (2.2) e della probabilità di indovinare la soluzione corretta (2.3).

Inoltre occorrerebbe implementare nuove tecniche di **validazione** in aggiunta a quelle già definite per rappresentare più dettagliatamente il grado di correttezza delle stime ottenute.

Infine sarebbe importante fornire un supporto diretto per l'analisi dei dati **politomici**, evitando di ricondurli ad una forma dicotomica e permettere così di trarne stime più valide perché derivanti dallo studio diretto di valori che presentano un livello di correttezza più preciso in quanto non si è più vincolati a considerare un esercizio o totalmente errato o perfetto.

Quest'ultima estensione è implementabile per mezzo del Graded Response

Model (GRM), questa famiglia di modelli matematici permette la valutazione dei dati politomici.

Il package ltm adottato dispone di funzioni di analisi dei dati politomici sotto l'approccio dell'Item Response Theory.

Bibliografia

[1] Wikipedia, the free encyclopedia. Statistica economica https://it.wikipedia.org/wiki/Statistica_economica, 2016.

[2] R Development Core Team. Linguaggio r https://www.r-project.org/.

[3] Margaret Wu, Hak Ping Tam, Tsung-Hau Jen. Educational measurement for applied, 2016.

[4] Sheldon M.Ross. *Probabilità e statistica per l'ingegneria e le scienze*. 2014.

[5] John K. Kruschke. Doing bayesian data analysis, second edition, 2014.

[6] R Development Core Team. What is r? https://www.r-project.org/about.html.

[7] Dimitris Rizopoulos. Documentazione ltm https://www.jstatsoft.org/article/view/v081i01/.

[8] Wikipedia, the free encyclopedia. Programmazione modulare https://it.wikipedia.org/wiki/Programmazione_modulare, 2016.

[9] Peter Danenberg, Manuel Eugster. Roxygen http://roxygen.org/, 2011.