UNIVERSITÀ DEGLI STUDI DI PADOVA

PsychoMetric

Progetto di Programmazione a Oggetti

A.A. 2021/2022

Studentessa: Valentina Caputo

Matricola: 1224456

1. Introduzione

1.1 Analisi preliminare del problema

L'esame di psicometria, al primo anno del corso di laurea di psicologia, è considerato spesso come un ostacolo "difficilmente sormontabile", frequentemente rimandato e vissuto con ansia da molti studenti. Come confermato da diversi sondaggi e dati statistici raccolti nelle varie università italiane (come ad esempio la Bicocca, UniCusano, l'università di Padova), esso rimane uno degli esami più difficili da superare.

I motivi, sottostanti queste difficoltà, possono essere ricondotti a pregiudizi verso la statistica e/o il ritenere la materia irrilevante rispetto al percorso di studi e per la professione (Primi e Chiesi, 2008).

In linea con quanto detto, l'approccio prediletto da alcuni professori si fonda su due principi: utilizzare un linguaggio semplice e ricorrere il più possibile ad esempi.

Ciò che maggiormente si riscontra da una preliminare analisi del web, è che mancano strumenti semplici e pratici in grado di fornire un concreto supporto allo studio. Il principale software utilizzato in fase di apprendimento è R, ma il suo utilizzo richiede già una buona conoscenza preliminare della psicometria.

1.2 Scopo del progetto

Alla luce di quanto emerso dall'analisi preliminare, l'obiettivo di PsychoMetric è fornire allo studente uno strumento semplice e pratico che consenta di acquisire alcuni dei concetti chiave di psicometria, in particolare: principi base della misurazione in psicometria e distribuzione di frequenza con una variabile.

Nota: essendo un progetto didattico con un tempo di lavoro limitato, sono stati selezionati i contenuti di psicometria ritenuti fondamentali all'apprendimento e in linea con le specifiche del progetto.

2. Progettazione

Il progetto è stato pensato e sviluppato secondo il modello MVC, in modo tale da mantenere separate la componente logica da quella grafica. Nello specifico, il modello è stato sviluppato a partire dalle conoscenze relative alla psicometria, in modo da descrivere e rappresentare al meglio questa realtà. In questa fase, maggior spazio e attenzione sono stati dedicati alla progettazione della scala di misura e della variabile. Il criterio guida per la parte logica è stato il raggiungimento dell'esaustività nei confronti della descrizione della realtà in esame: ovvero si è pensato più a descrivere bene la realtà che si vuole rappresentare, anziché focalizzarsi su ciò che poteva essere utile alla realizzazione e al funzionamento del programma. La parte grafica, invece, è stata progettata per rispondere alle esigenze del target a cui il programma è rivolto, motivo per cui si è scelto di inserire nel programma continui riferimenti alla teoria.

Tutto il programma è stato pensato, progettato e sviluppato per fornire risposte diverse sulla base del tipo di variabile, e quindi di scala di misura, utilizzata. Per questo motivo, il *core* del progetto sta nella progettazione e implementazione delle gerarchie di tipo relative alla scala di misura e alla variabile.

2.1 Gerarchie di tipo

2.1.1 Scala di misura

Tutta la psicometria si fonda sul concetto fondamentale di scala di misura. Gli oggetti di indagine nella ricerca psicologica, ma anche nell'ambito clinico, di sviluppo o di lavoro, sono misurati a partire dal tipo di scala di misura, ciascuno con delle specifiche proprietà, definite sulla base dell'attributo, del valore numerico e delle operazioni logiche e aritmetiche ammesse. In particolare:

1. *Scala di misura Nominale*: è utilizzata per la misurazione di oggetti di natura qualitativa, la cui caratteristica fondamentale è l'assenza di ordine tra gli oggetti. Viene definita tramite "attributi", ovvero parole che definiscono le proprietà degli oggetti. Ad esempio, per la misurazione della nazionalità si usa la scala nominale e gli attributi possono essere italiano, inglese o francese a seconda della nazione di provenienza della persona su cui si fanno le rilevazioni. All'attributo può essere associato un valore numerico, detto etichetta, utile per sostituire la parola con un numero, in caso di numerose misurazioni. Data la natura di questa scala di misura, le uniche operazioni concesse sono l'uguaglianza e la disuguaglianza.
2. *Scala di misura Ordinale*: molto simile alla scala di misura nominale, per quanto riguarda la natura qualitativa. Per cui viene assegnato un attributo e una possibile etichetta, ma il significato associato all'attributo implica una relazione di ordine con gli altri oggetti della misurazione. Ad esempio, la misurazione del dolore viene fatta spesso utilizzando quella che viene chiamata la "scala Likert", ovvero, indicando un numero "arbitrario"

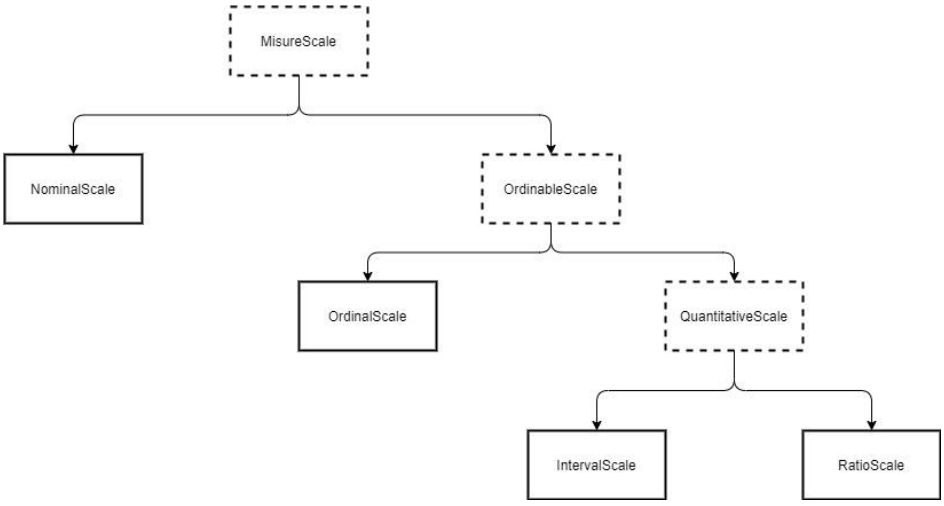
compreso tra 1 e 5, dove 1 indica livello basso, 3 livello medio e 5 livello alto. Gli attributi sono “basso”, “medio”, “alto” e i numeri associati sono etichette. Dato che gli attributi sono ordinabili, rispetto alla scala nominale, sono ammesse anche operazioni di minore e maggiore.

3. *Scala di misura a Intervalli*: è utilizzata per la misurazione di fenomeni di natura quantitativa. Con la scala a intervalli si adottano delle *convenzioni* che consentono l’assegnazione di una quantità numerica all’oggetto della misurazione. Caratteristiche salienti sono: lo zero non indica una reale assenza (quantità nulla) e i valori possono essere negativi. Esempio classico è la temperatura, che si può misurare con i gradi Celsius o con i gradi Fahrenheit. Oltre alle operazioni logiche, sono ammesse anche operazioni di somma e sottrazione; non quelle di moltiplicazione e divisione.
4. *Scala di misura a Rapporti*: molto simile alla scala di misura a intervalli, si differenzia da questa per il carattere “assoluto” dello zero. Il numero associato indica, infatti, una reale quantità, per cui 0 implica un’assenza e il valore negativo perde completamente di significato. Tutte le operazioni, logiche e aritmetiche, sono ammesse.

Le caratteristiche salienti prese in esame per definire le corrispettive classi sono riassunte nella tabella sottostante.

	Nominale	**Ordinale**	**A Intervalli**	**A Rapporti**
Attributo	Descrive la proprietà in esame	Descrive la proprietà in esame	*coincide con il valore numerico*	*coincide con il valore numerico*
Valore numerico	OPZIONALE: etichetta assegnata, non ha alcuna valenza numerica	OBBLIGATORIO: Indica il livello dell'attributo rispetto agli altri	valore numerico effettivo, può essere intero o reale, positivo o negativo	valore numerico effettivo, può essere intero o reale, ma può essere solamente un numero >=0
operazioni logiche				
uguaglianza	x	x	x	x
“< ” o “>”	-	x	x	x
operazioni aritmetiche				
“+” o “-“	-	-	x	x
“*” o “/”	-	-	-	x

Sulla base di questi riferimenti teorici è stata definita la prima gerarchia di tipo: MisureScale.

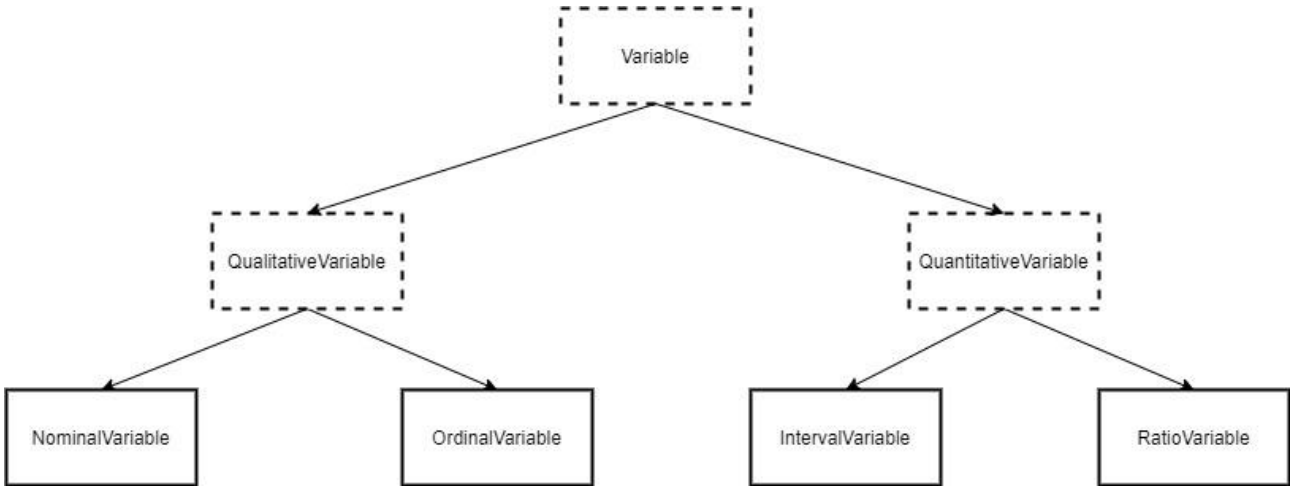


Partendo dalle definizioni date alle diverse scale di misura, si è scelto di creare una gerarchia di tipo che rispettasse la natura gerarchica già esistente, per certi aspetti simile ad una matrioska. Per cui i metodi fondamentali, comuni a tutte le scale di misura vengono definite nella classe base astratta **MisureScale**. Man mano che si scende nella gerarchia, le classe si specializzano e definiscono ulteriori metodi comuni alle sottoclassi. Per questo sono state definite le classi astratte **OrdinableScale** e **QuantitativeScale**, dove sono definiti rispettivamente i metodi che garantiscono operazioni di minore e maggiore, e le operazioni aritmetiche comuni di somma e sottrazione. Solo ad ultimo sono definite le classi concrete, dove vengono implementate le caratteristiche peculiari di ogni scala di misura. Si è scelto di non definire subito gli attributi nella MisureScale per migliorare le prestazioni di memoria: scale di misura quantitative non sono definite da attributi, mentre la scala nominale non richiede necessariamente un valore numerico.

In MisureScale viene definito il metodo statico: static bool **checkDigit** (string digit); che serve a verificare e implementare correttamente tutte le classi concrete derivate; essendo un metodo necessario esclusivamente alla definizione di tutte le classi derivate, è stato dichiarato *protected*.

2.1.2 Variabile

La scala di misura è il sistema di regole che determina l'attribuzione di un valore numerico ad un oggetto (singolo dato), mentre ciò che interessa il ricercatore è la definizione della variabile, ovvero l'insieme di dati raccolti che variano da individuo a individuo. Sulla base del sistema di misurazione scelto, la variabile può essere: qualitativa su scala nominale, qualitativa su scala ordinale, quantitativa su scala a intervalli o quantitativa su scala a rapporti. Per questo è stata definita la seconda, più importante, gerarchia di tipi, basata sulla prima gerarchia.



In psicometria, la variabile viene definita sulla base di tre elementi: il nome assegnato (es. genere), la scala di misura (es. nominale) e l'insieme di valori che costituiscono i dati grezzi della variabile (es. maschio, maschio, femmina). Sulla base di questi elementi, la classe Variable viene definita dai corrispettivi attributi: name (nome della variabile), scale (scala di misura che la definisce) e rawData (dati grezzi). Si è scelto di rappresentare i dati grezzi attraverso il template std::list. L'utilizzo di questo contenitore è stato preferito a std::vector, in quanto ritenuto più flessibile ed efficiente nel gestire le ripetute operazioni di inserimento, cancellazione e di modifica dei dati. Nel momento in cui si riteneva più opportuno dare la priorità alla posizione degli elementi, si è scelto di utilizzare come contenitore std::vector.
Le classi derivate definiscono in automatico la propria scala di appartenenza e sono implementate in modo da assegnare a rawData oggetti appartenenti alla corretta scala di misura. Le funzioni ausiliare bool **checkCoerency** (MisureScale *toCheck) const; virtual bool ***checkScaleCoerency***() const; definite nella parte protetta, garantiscono la coerenza all'interno dei dati della variabile, in modo tale che i dati siano sempre appartenenti alla stessa scala di misura, coincidente con quella indicata dalla variabile. Da notare che:

1. bool **checkCoerency**(MisureScale *toCheck) const; non è dichiarata virtuale perché il suo unico scopo è controllare che tutti i dati della variabile siano stati definiti con la stessa scala di misura e non è logico prevedere sue future modifiche da parte di classi derivate.
2. virtual bool ***checkScaleCoerency***() const; dichiarata virtuale perché l'obiettivo è valutare la coerenza dei dati in base alla scala di misura della variabile, per cui determinate sottoclassi potrebbero trovarsi a dover aggiungere o modificare il codice per rispondere meglio alle proprie esigenze. Come è stato fatto in QualitativeVariable e in NominalVariable.
3. virtual bool ***checkConsistency***(MisureScale* toCheck) const; specifico della classe QualitativeVariable, in quanto devono tener conto della consistenza dei dati tra attributo ed etichetta numerica, caratteristica esclusiva delle variabili qualitative. Questo metodo è dichiarato virtuale per adattare meglio la risposta della funzione alle esigenze specifiche, e potenzialmente diverse, delle singole variabili qualitative.

Particolare attenzione è stata posta alla implementazione della *rule of three*, in quanto la lista che rappresenta i dati grezzi è costituita da puntatori, per cui la copia, l'assegnazione e il distruttore devono prima agire sull'oggetto puntato dal singolo elemento della lista, poi sull'elemento e infine sull'intera lista. Per fare questo si è definita la funzione ausiliaria virtual void ***deepCopy***(list<MisureScale*> data) =0; da implementare nelle classi derivate concrete, in modo tale da fare una copia profonda specifica per il tipo di scala di misura della variabile; e la funzione void **clearData**(); per non lasciare garbage in fase di distruzione della variabile.

Infine, i metodi disponibili riguardano azioni di modifica, ricerca, inserimento ed eliminazione dei dati. La classe Variable implementa già tutti i metodi definibili, l'overriding avviene solo quando i cambiamenti nella variabile potrebbero portare a incoerenze specifiche per la scala di misura.

2.1.2 ChartView

Al di fuori del modello, è stato utile definire una nuova gerarchia di tipo per creare la vista relativa alla rappresentazione dei grafici. La classe base astratta è ChartView, mentre le classi derivate sono PieChartView, BarChartView e LinearChartView. Questa gerarchia di tipo ha reso la scrittura del codice più fluida ed efficiente, rendendo possibile poi al controller generare il grafico tramite un'unica chiamata.

2.2 Chiamate polimorfe

Le principali chiamate polimorfe vengono eseguite nella classe Variable e in tutte le classi che gestiscono una variabile. In particolare, la classe Variable utilizza le chiamate polimorfe alle funzioni

- virtual string ***getAttribute***() const;
- virtual void ***changeAttribute***(const string & att);
- virtual double ***getNumberValue***() const;
- virtual void ***changeNumberValue***(double nv);

per cercare e/o modificare i dati all'interno della rawData, senza conoscere la scala di misura con cui il dato è stato definito; mentre tutte le classi che gestiscono una variabile utilizzano continuamente chiamate polimorfe per modificare, sostituire, inserire, aggiungere o fare il reset dei dati all'interno della variabile, senza conoscere come questa è stata definita (nominale, ordinale, a intervalli o a rapporti). I metodi virtuali a cui fanno riferimento sono:

- virtual void ***replaceData***(MisureScale* oldValue, MisureScale* newValue);
- virtual void ***replaceAttribute***(const string& oldAtt, const string& newAtt);
- virtual void ***replaceNumberValue***(double oldNV, double newNV);

- virtual void ***changeData***(unsigned int pos, MisureScale* newData);
- virtual void ***changeAttribute***(unsigned int pos, const string& newAttribute);
- virtual void ***changeNumberValue***(unsigned int pos, double newNumberValue);

- virtual void ***sortData***();
- virtual void ***addData***(MisureScale* toAdd);
- virtual void ***insertData***(unsigned int pos, MisureScale* toInsert);
- virtual void ***resetData***();

Sono dichiarati virtuali soltanto questi metodi perché prevedono una implementazione specifica sulla base della scala di misura che definisce la variabile.

2.3 Gestione I/O

La gestione dell'input/output è stata sviluppata tramite la classe FileManagement, la quale definisce i metodi per la lettura da file e per la scrittura su file; il cui formato è XML, in quanto ritenuto più semplice e adatto allo scopo del progetto.
L'importazione dei dati va a buon fine se è presente l'elemento Variable nel file XML e tutti gli attributi sono definiti correttamente. Nel caso in cui un attributo del file sia scritto in maniera errata, il dato verrà letto con valori standard (stringa vuota per l'attributo della scala di misura e valore di default 0 per il valore numerico della scala di misura) che possono generare errore in fase di creazione della variabile, per mancanza di coerenza nei dati.

Nella cartella "resources/esempi_PsychoMetric" sono forniti 4 esempi di file, uno per ogni tipo di variabile.

3. Manuale Utente

Il programma è diviso in tre sezioni:

1. *Definizione della variabile*: utile per acquisire i concetti chiave relativi alla scala di misura,
2. *Analisi delle frequenze*: con possibilità di avere la tabella e completarla come esercizio, o accedere direttamente alla tabella con i risultati
3. *Rappresentazione grafica*: con possibilità di scegliere tra tre diverse forme, sulla base della variabile analizzata.

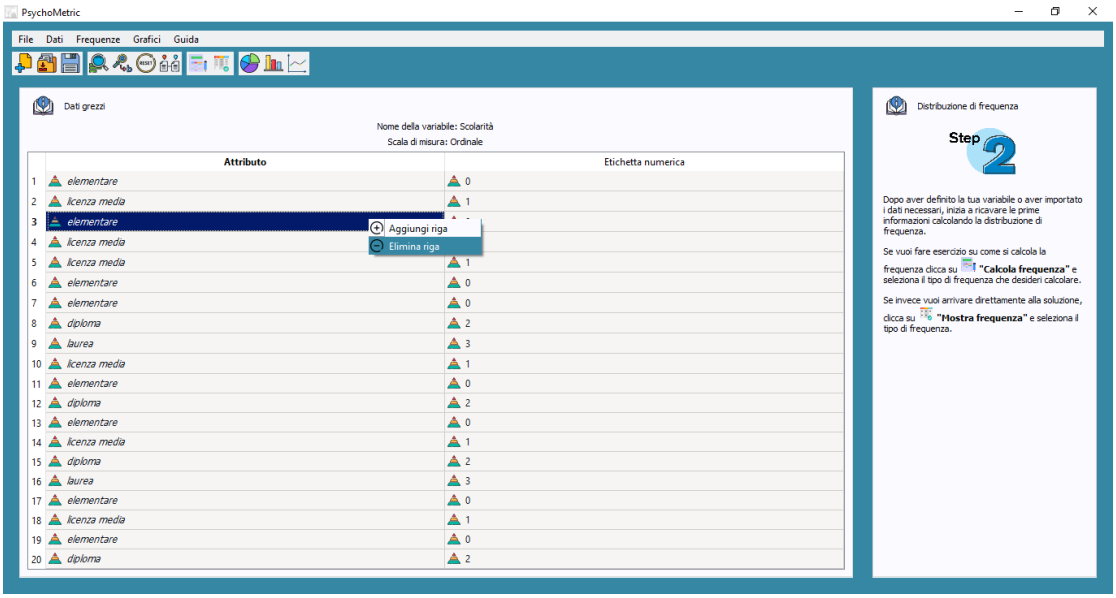
Le prime due parti sono corredate da una guida che serve a facilitare lo studente nell'apprendimento.
Tutto il materiale è tratto dai primi due capitoli del libro "Introduzione alla psicometria" di Caterina Primi e Francesca Chiesi (Editori Laterza, 2008).

3.1 La variabile

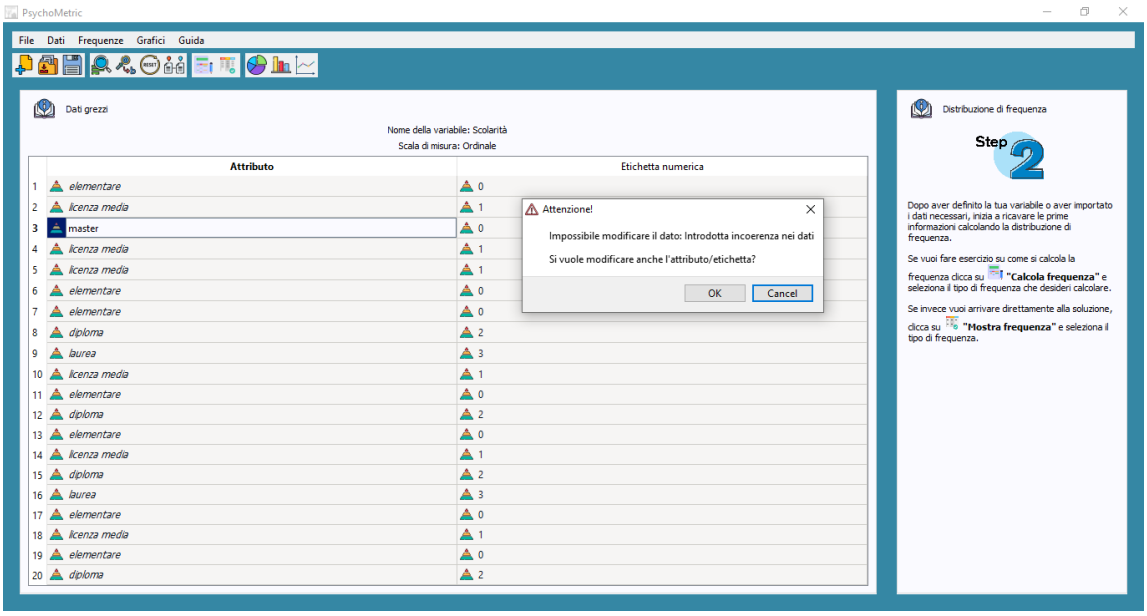
Dalla Home è possibile creare una nuova variabile o importare i dati della variabile da un file .xml, selezionando la corrispettiva azione dal menù a tendina o cliccando il relativo pulsante nel menù a icone, appena sotto alla barra del menù. Senza prima aver fatto questa azione non sarà possibile fare niente altro nel programma.
A seconda del tipo di variabile creata o importata, viene creata una tabella con una o due colonne e i dati contengono una icona specifica per la scala di misura a cui appartengono.

In ogni caso è possibile:

- cercare un dato nella variabile, cliccando sulla voce "Cerca attributo" del menù o sul relativo pulsante nella barra sottostante a quella del menù;
- sostituire un certo attributo (o valore numerico, in caso di variabile quantitativa) con un altro indicato dall'utente, cliccando sulla voce "Sostituisci attributo" del menù o sul relativo pulsante nella barra sottostante a quella del menù;
- inserire o eliminare le righe della tabella (e quindi il dato della variabile), cliccando con il tasto destro del mouse in corrispondenza della riga che si vuole eliminare o dove si vuole inserire una nuova riga;

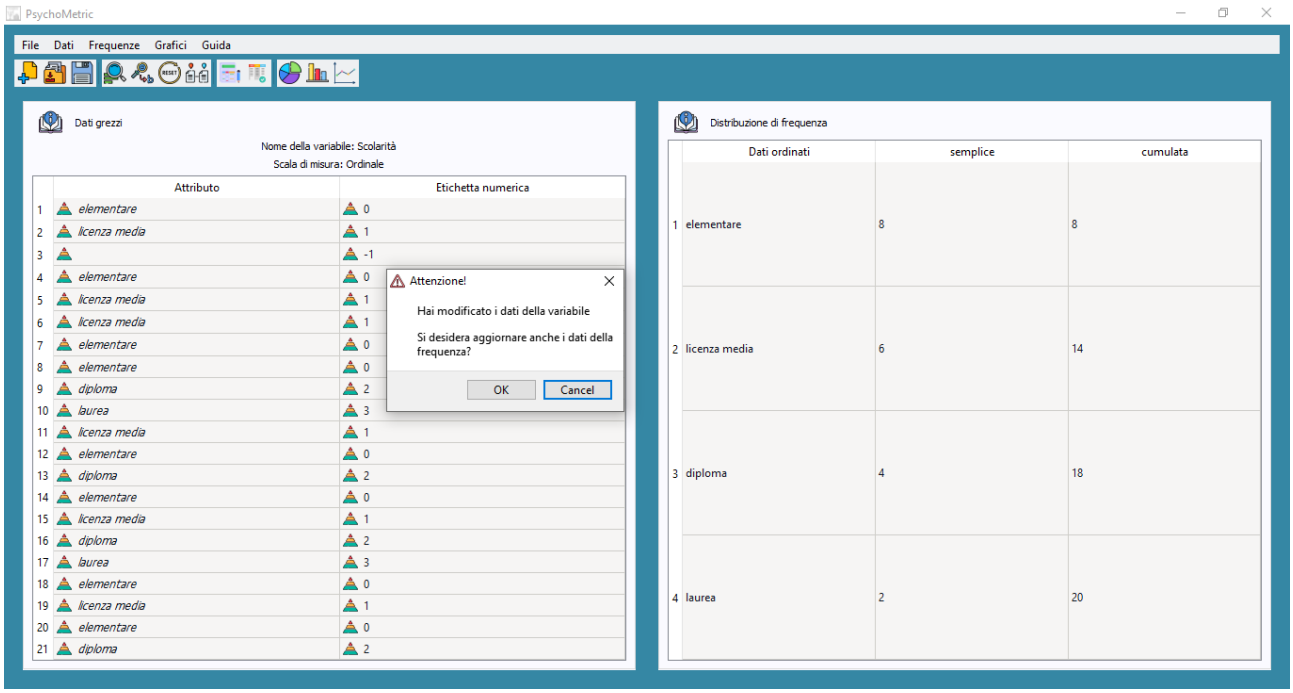


- modificare i singoli valori direttamente con il doppio click sulla cella: in questo caso è stata posta particolare attenzione al mantenimento della coerenza nei dati, per cui cambiamenti non ammessi per il tipo di variabile generano una finestra di dialogo che consente di porre rimedio all'incoerenza e quindi consentire il cambiamento del dato nella variabile, come mostrato nell'immagine sottostante;



- fare il reset dei dati, cliccando sulla voce “Reset dei dati” del menù o sul relativo pulsante nella barra sottostante a quella del menù; essendo questa operazione irreversibile, per poter proseguire viene chiesto conferma all’utente;
- soltanto se la variabile è quantitativa, è possibile trasformare la variabile in esame da quantitativa a qualitativa, selezionando l’azione “Converti in variabile ordinale” dal menù oppure cliccando il pulsante nella barra sottostante a quella del menù.

Tutte le modifiche apportate ai dati possono avere delle conseguenze diverse a seconda del fatto che sia stata già creata la tabella delle frequenze. In particolare, nel caso in cui l’utente abbia già creato la tabella delle frequenze, ogni tipo di modifica apportata sui dati genera una finestra di dialogo che consente di aggiornare i dati della frequenza.



Se l’utente sceglie di non aggiornare i dati, la tabella delle frequenze verrà nascosta, in modo tale da non creare confusione nell’utente.
Per lo stesso principio, se l’utente sceglie di creare o aprire una nuova variabile ed era già stata creata la tabella delle frequenze, quest’ultima verrà rimossa dalla vista.

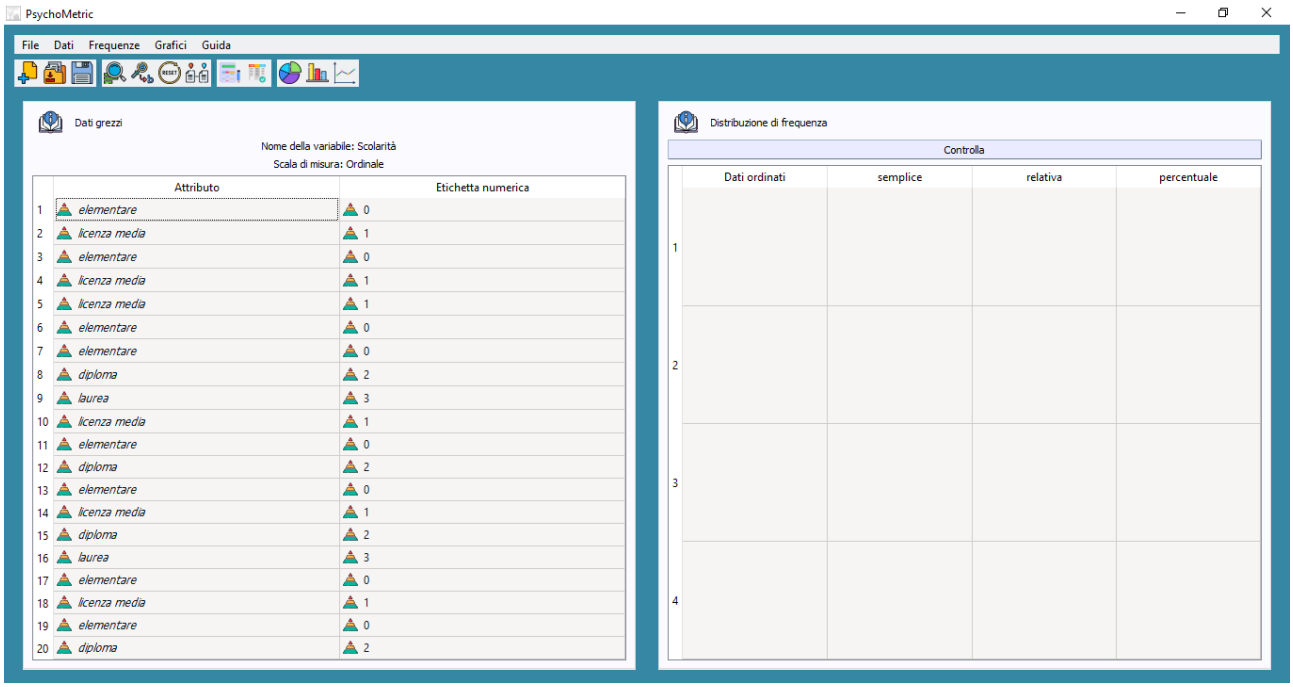
3.2 Le frequenze

Gli esercizi iniziali di psicometria si incentrano maggiormente sul calcolo delle frequenze. In linea con lo scopo del progetto, è stata creata una sezione dedicata allo svolgimento di questi esercizi.

In particolare, l'utente può decidere di svolgere gli esercizi selezionando dal menù principale: "Frequenze", "Calcola…" e il tipo di frequenza su cui si intende fare pratica; oppure cliccando sul pulsante sottostante la barra del menù e selezionare, dalla finestra di dialogo, il tipo di frequenza che si vuole calcolare. In alternativa, è possibile visualizzare direttamente i risultati, selezionando dal menù principale: "Frequenze", "Mostra…" e il tipo di frequenza che si vuole vedere; oppure cliccando sull'apposito pulsante sottostante la barra del menù e selezionare, dalla finestra di dialogo, il tipo di frequenza che si vuole visualizzare.

Come è possibile intuire, soltanto la tabella su cui si intende fare pratica è editabile ed ha a disposizione un pulsante "Controlla" che consente di verificare se si è svolto bene l'esercizio oppure no.

In entrambi i casi, la forma della tabella è pensata per scopi didattici, per cui il numero di colonne varia a seconda delle frequenze necessarie per calcolare quella desiderata. Per fare un esempio, se l'utente vuole calcolare la frequenza percentuale, esso avrà bisogno prima della frequenza relativa e, prima ancora, della frequenza semplice. Quindi il numero di colonne della tabella di frequenze è 4: una per la lista ordinata (solo se scala di misura ordinabile, altrimenti segue l'ordine con cui sono state indicate nella tabella dati) e una per ogni frequenza necessaria (semplice, relativa, percentuale).



3.3 La rappresentazione grafica

Una volta creata o aperta una variabile, è possibile visualizzare i dati sotto forma di grafico a torta, o a barre, o lineare (altrimenti detto poligono di frequenza), selezionando la corrispettiva voce dal menù a tendina o cliccando sul corrispettivo pulsante al di sotto della barra del menù. Il grafico verrà creato a partire dai dati della variabile e dal tipo di frequenza che si vuole visualizzare sottoforma di grafico.

Anche per la rappresentazione grafica, è opportuno selezionare il tipo di grafico a seconda del tipo di variabile: la variabile quantitativa non può essere rappresentata sottoforma di grafico a barre, mentre la variabile qualitativa non può essere rappresentata sottoforma di grafico lineare (poligono di frequenza).

4. Conclusione

4.1 Istruzioni compilazione ed esecuzione

Per compilare il progetto è necessario inserire da terminale i seguenti comandi:

- `qmake PsychoMetric.pro`
- `make`

Infine, il programma può essere eseguito su terminale tramite comando: `./PsychoMetric`

4.2 Tempo richiesto

Il tempo richiesto per lo svolgimento del progetto, riportato nella tabella sottostante, è risultato di un'approssimazione. Infatti, il programma era stato pensato, progettato e iniziato a implementare nel mese di maggio, tuttavia, la complessità dell'idea e la mia poca esperienza nella programmazione, mi hanno portato a sospendere il lavoro fatto per ricominciare di nuovo a luglio. Per cui le ore indicate sono calcolate a partire dal mese di luglio.

Analisi preliminare del problema	2 ore
progettazione	6 ore
Apprendimento libreria Qt	10 ore
Codifica modello	14 ore
Codifica GUI e controller	17 ore
Debugging & Testing	15 ore
Altro	*5 ore*
TOTALE	64 ore *(+5 ore)*

Complessivamente il progetto ha richiesto un quantitativo di ore superiore a 50. Questo perché il modello ha richiesto molto tempo, sia per la progettazione che per l'implementazione: ho dovuto riguardare bene i concetti chiave di psicometria che volevo ricreare nel programma, implementare bene le gerarchie ti tipo, fondamentali per la riuscita del programma, e la fase di testing ha portato via molto tempo. Inoltre, le numerose funzioni offerte dal programma hanno portato a dedicare molto tempo allo studio della libreria Qt e all'implementazione del codice sia della vista che del controller. Infine, sono state considerate come tempo aggiuntivo, indicate con *Altro* nella tabella, le ore dedicate al ripasso di psicometria e alla scrittura della guida. Per quello che è lo scopo del progetto, infatti, la presenza di una guida è fondamentale e doveva essere implementata a sufficienza. Questo perché, nel tempo a disposizione, non era possibile implementare una guida più accurata, ma era comunque importante definire l'idea di base e mostrare i dettagli necessari alla comprensione del programma e della psicometria.

4.3 Note su ambiente di sviluppo

Sistema Operativo: Windows 10 Enterprise
Compilatore: MSVC 2015
Libreria Qt: Qt 5.9.5

L'IDE con cui è stato sviluppato il programma è Qt Creator 4.5.2.

Il programma è stato testato anche sulla macchina virtuale, come esplicitamente indicato nelle specifiche del progetto.