



PROGETTO FINVALI 2005

Progetto 032: Il sistema scolastico come sistema complesso: qualità delle rivelazioni e modelli di interpretazione dei risultati



Istituto per le Applicazioni del Calcolo 'Mauro Picone' del Consiglio Nazionale delle Ricerche, Sede di Napoli



Dipartimento di Statistica e Matematica per la Ricerca Economica, Università degli Studi di Napoli 'Parthenope'

Con la partecipazione di



Dipartimento di Statistica, Università degli Studi di Milano Bicocca



Dipartimento di Scienze della Terra, Università degli Studi di Napoli 'Federico II'



Dipartimento di Matematica e Applicazioni, Università degli Studi di Napoli 'Federico II'



PROGETTO FINVALI 2005

Progetto 032: Il sistema scolastico come sistema complesso: qualità delle rivelazioni e modelli di interpretazione dei risultati

Report dicembre 2006: Lettura dati INVALSI e sviluppo di software

Umberto Amato⁽¹⁾, Annamaria Caputo⁽²⁾



⁽¹⁾Istituto per le Applicazioni del Calcolo 'Mauro Picone' del Consiglio Nazionale delle Ricerche, Sede di Napoli



⁽²⁾Istituto Nazionale per la Valutazione del Sistema Educativo di Istruzione e di Formazione, Frascati

Abstract: Il report contiene l'attività svolta nel primo semestre del progetto riguardo la lettura del database INVALSI relativo ai questionari di valutazione, la conversione in formato leggibile da Matlab, il software Matlab per la risoluzione dei modelli ANOVA a più fattori ed il software MATLAB per la stima dell'andamento frattale delle abilità.

Si ringraziano la Dott.ssa Letizia Marcucci⁽²⁾ e la Dott.ssa Francesca Fortini⁽²⁾ per la disponibilità e competenza dimostrate nel mettere a disposizione i dati INVALSI

Indice

1	<i>anovan – Software MATLAB per modelli ANOVA</i>	2
1.1	Introduzione	2
1.2	Anovan – Analisi della varianza a N fattori (ANOVA)	2
1.3	Spiegazione dei parametri di ingresso	4
1.3.1	Tipi di modello	4
1.3.2	Tipi di somma dei quadrati	6
1.4	Spiegazione parametri di uscita	7
1.4.1	La struttura <code>stats</code>	7
1.5	Esempi	8
1.5.1	Esempio di modello ANOVA a 3 fattori	8
1.5.2	Esempio su un dataset più ampio	10
2	<i>Lettura dei dati INVALSI e conversione in formato Matlab</i>	16
2.1	SPSS ed esportazione di dati	16
3	<i>Codice per la gestione dei questionari e l'esecuzione dei modelli ANOVA</i>	19
3.1	Introduzione	19
3.2	Codice per l'analisi ANOVA	19
3.3	Codice per la lettura dei questionari di valutazione	30
3.4	Codice per la lettura dei dati delle scuole	36
4.	<i>Codice per la stima del comportamento frattale del sistema scolastico</i>	38
4.1	Introduzione	38
4.2	Codice per la stima del comportamento frattale	38

1 anovan – Software MATLAB per modelli ANOVA

1.1 Introduzione

Matlab è un ambiente di lavoro estremamente potente e largamente in uso presso la comunità scientifica in grado di risolvere svariati problemi di matematica e statistica, nonché fornire un'ampia gamma di tipi di grafici per la visualizzazione dei risultati. Matlab è costituito da numerosi toolboxes in grado di risolvere numerosi problemi matematici di base; in più la comunità scientifica contribuisce a Matlab rilasciando toolboxes sviluppati per particolari applicazioni e largamente testati dalla stessa comunità.

Nel toolbox `Statistics` Matlab include una routine per il calcolo dell'analisi della varianza mediante modelli ANOVA di tipo estremamente generali (modelli a N fattori, a effetti fissi o misti, con numero di campioni generalmente diverso per ogni cella, con scelta arbitraria degli effetti da considerare). Data la sua generalità, la risoluzione del modello avviene formulando il modello ANOVA mediante effetti con un metodo di regressione e risolvendolo mediante il metodo dei minimi quadrati.

Il presente capitolo del report descrive la sintassi completa del comando `anovan` di Matlab, incluso le opzioni avanzate, con la spiegazione dei parametri di ingresso e di uscita del comando. Inoltre vengono effettuati dei test su due esperimenti a più fattori noti in letteratura ed utilizzati come benchmark.

1.2 Anovan – Analisi della varianza a N fattori (ANOVA)

Sintassi generale:

- `p = anovan(x,group)`
- `p = anovan(x,group,'Param1',val1,'Param2',val2,...)`
- `[p,table] = anovan(...)`
- `[p,table,stats] = anovan(...)`
- `[p,table,stats,terms] = anovan(...)`

Il comando

```
p = anovan(X,group)
```

esegue l'analisi di varianza (ANOVA) a più fattori nel caso di campionamento bilanciato (grandezza del campione uguale per tutte le celle del modello) e non bilanciato (grandezza del campione possibilmente diversa). Lo scopo dell'algoritmo è confrontare le medie delle osservazioni del vettore X rispetto a N fattori diversi. I fattori e i livelli dei fattori vengono assegnati nel cell array `group`. Ciascuna delle N celle di `group` contiene una lista dei livelli dei fattori che identificano le osservazioni in X rispetto a uno dei fattori. La lista all'interno di ciascuna cella può essere un vettore, un array di tipo character o a sua volta un cell array di stringhe e deve contenere lo stesso numero di elementi di X .

Il comando

```
p = anovan(x,group,'Param1',val1,'Param2',val2,...)
```

specifica una o più coppie di parametri/valori secondo la seguente tabella:

Parametro	Valori possibili
'sstype'	1, 2, or 3, specifica il tipo di somma di quadrati (default = 3)
'varnames'	Matrice di tipo character oppure cell array di stringhe contenenti i nomi delle variabili di raggruppamento, uno per ogni variabile. Se il parametro non viene specificato si assumono I nomi di default 'X1', 'X2', 'X3', ..., 'XN'.
'display'	'on' mostra in output la tabella ANOVA (default) 'off' non mostra in output la tabella ANOVA
'random'	Vettore di indici che indica quali variabili di raggruppamento sono effetti random (per default tutti gli effetti sono fissi, come nel caso del Sistema di Valutazione Scolastico considerato in questa fase).
'alpha'	Numero compreso tra 0 e 1 che specifica l'intervallo fiduciario $100(1 - \alpha)\%$ (il valore di default è 0.05 corrispondente ad un livello di significatività del 95%)

'model'

Tipo di modello ANOVA usato.

Il comando

```
[p,table] = anovan(...)
```

Fornisce in output nel cell array `table` la tabella ANOVA, incluso la denominazione dei fattori.

Il comando

```
[p,table,stats] = anovan(...)
```

Fornisce in output la struttura `stats` che può essere utilizzata per eseguire test di confronto multiplo mediante la routine `multcompare` a seguito dell'analisi ANOVA.

Il comando

```
[p,table,stats,terms] = anovan(...)
```

fornisce in output i termini principali e delle interazioni considerati nel modello ANOVA. Tali termini sono codificati nella matrice `terms` nel formato descritto di seguito a seconda del modello ANOVA impiegato. Nel caso in cui come formato della matrice venga indicato direttamente 'model', allora la matrice `terms` coinciderà con `model`.

1.3 Spiegazione dei parametri di ingresso

1.3.1 Tipi di modello

Nel comando

```
p = anovan(x,group,'model', modeltype)
```

l'argomento `modeltype` può assumere i seguenti valori:

- 'linear' (default): il modello ANOVA calcola i the p -values solo per le ipotesi nulle riguardanti gli N effetti principali (senza interazioni).
- 'interaction': Il modello ANOVA calcola i p -values per le ipotesi nulle riguardante gli N effetti principali e le $N(N+1)/2$ coppie di interazioni a 2 fattori..
- 'full': Il modello ANOVA calcola i p -values per le ipotesi nulle sugli N effetti principali e su tutte le interazioni a tutti i livelli.
- Un numero intero $k \leq N$: in questo caso il modello ANOVA calcola tutte le interazioni fino al livello k ; per esempio $k=1$ significa nessuna interazione (equivalente al caso linear); per $k=2$ vengono considerate solo le interazioni a coppie (equivalente al caso interaction); per $k=3$ le interazioni a coppie e quelle a 3 a 3. Il caso $k=N$ è equivalente al parametro full.

Una matrice dove ciascuna riga specifichi i termini desiderati nel modello attraverso una serie di 0 e 1 secondo il seguente schema:

Riga della matrice	Termine ANOVA corrispondente
[1 0 0]	Termine principale A
[0 1 0]	Termine principale B
[0 0 1]	Termine principale C
[1 1 0]	Termine di interazione AB
[0 1 1]	Termine di interazione BC
[1 0 1]	Termine di interazione AC
[1 1 1]	Termine di interazione ABC

Per esempio, se `modeltype` è la matrice `[0 1 0;0 0 1;0 1 1]`, il vettore di uscita `p` contiene nell'ordine i p -values corrispondenti alle ipotesi nulle sugli effetti principali B, C e sull'effetto dell'interazione tra B e C. Un modo semplice per generare la matrice consiste nel modificare l'array `terms` in uscita da una precedente chiamata ad `anovan`, che riporta i termini relativi agli effetti principali e di interazione nel modello ANOVA corrente secondo il formato della tabella. Se per

esempio l'uscita di `anovan` fornisce l'array `terms` come `[0 1 0;0 0 1;0 1 1]` e non risulta evidente alcuna interazione tra i fattori B e C, è consigliato ripetere l'analisi sui soli termini principali B and C specificando `modeltype` come `[0 1 0;0 0 1]`.

1.3.2 Tipi di somma dei quadrati

Nel comando `p = anovan(x,group,'sstype', integer)` il parametro `sstype` può assumere i valori 1, 2, o 3 (default) per indicare il tipo di somma di quadrati secondo schema riportato nella tabella seguente. Si osservi che il valore assegnato ha effetto solo sugli studi non bilanciati. La somma di quadrati per ogni termine è calcolata confrontando due modelli. La somma di Tipo 1 per una termine è la riduzione della somma residua di quadrati ottenuta aggiungendo quel termine al fit che già include i termini elencati precedentemente nell'ordine. La somma dei quadrati di Tipo 2 è la riduzione della somma residua di quadrati ottenuta aggiungendo quel termine ad un modello che include tutti gli altri termini che non contengono il termine in questione. Infine la somma dei quadrati di Tipo 3 è la riduzione della somma residua dei quadrati ottenuta aggiungendo quel termine ad un modello che include tutti gli altri termini, soggetta ai vincoli di compatibilità relativi alle somme nulle degli effetti. Per esempio consideriamo il caso in cui si proceda ad un fit di un modello che comprende due fattori e la loro interazione e i termini appaiano nell'ordine A, B, AB. Sia $R(\cdot)$ la somma residua dei quadrati per un modello (per esempio $R(A,B,AB)$ è la somma residua dei quadrati fittando l'intero modello; $R(A)$ è la somma residua dei quadrati fittando solo l'effetto principale A; $R(1)$ è la somma residua dei quadrati fittando solo il valore medio. Allora i tre tipi di somma dei quadrati sono

Termine	Tipo 1	Tipo 2	Tipo 3
A	$R(1)-R(A)$	$R(B)-R(A,B)$	$R(B,AB)-R(A,B,AB)$
B	$R(A)-R(A,B)$	$R(A)-R(A,B)$	$R(A,AB)-R(A,B,AB)$
AB	$R(A,B)-R(A,B,AB)$	$R(A,B)-R(A,B,AB)$	$R(A,B)-R(A,B,AB)$

Si osservi che il Tipo 3 impone delle restrizioni: per esempio nell'effettuare il fit di $R(B,AB)$, l'array degli effetti AB è vincolato ad avere somma nulla su A per ogni valore di B e somma nulla su B per ogni valore di A .

1.4 Spiegazione parametri di uscita

1.4.1 La struttura `stats`

La routine `anovan` testa l'ipotesi che livelli differenti di un fattore (o in generale di un termine) hanno lo stesso effetto, contro l'ipotesi alternative che essi non abbiano lo stesso effetto. In genere risulta preferibile effettuare un test per verificare quail coppie di livelli sono statisticamente significative. A tale scopo è possibile utilizzare la routine `multicompare` che richiede in input proprio la struttura `stats`. Tale analisi sarà demandata alla seconda fase del progetto.

La struttura `stats` contiene I campi listati di seguito, oltre ad altri campi richiesti per i confronti multipli:

Campo di <code>stats</code>	Significato
<code>coeffs</code>	Coefficienti stimati
<code>coeffnames</code>	Nome del termine per ciascun coefficiente
<code>vars</code>	Matrice di raggruppamento delle variabili per ciascun termine
<code>resid</code>	Residui rispetto al modello fittato

Nel caso di presenza di effetti misti (che non si riferisce allo studio effettuato nella prima fase del progetto) sono altresì presenti i seguenti campi:

Campo di <code>stats</code>	Significato
<code>ems</code>	Medie quadrate attese
<code>denom</code>	Definizione del denominatore
<code>rtnames</code>	Nome dei termini casuali

varest	Varianza delle componenti stimate (una per ogni termine casuale)
varci	Intervalli di confidenza per le componenti della varianza

1.5 Esempi

1.5.1 Esempio di modello ANOVA a 3 fattori

Consideriamo i vettori y e $group$ seguenti:

- $y = [52.7 \ 57.5 \ 45.9 \ 44.5 \ 53.0 \ 57.0 \ 45.9 \ 44.0]'$;
- $g1 = [1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2]'$;
- $g2 = \{ 'hi'; 'hi'; 'lo'; 'lo'; 'hi'; 'hi'; 'lo'; 'lo' \}'$;
- $g3 = \{ 'may'; 'may'; 'may'; 'may'; 'june'; 'june'; 'june'; 'june' \}'$;

In pratica è uno studio ANOVA a 3 fattori con 2 livelli ciascuno. Ciascuna osservazione in y è identificata da una certa combinazione di livelli dei fattori. Se i fattori sono A, B e C, l'osservazione $y(1)$ è associata a

- Livello 1 del fattore A
- Livello 'hi' del fattore B
- Livello 'may' del fattore C

Allo stesso modo per esempio l'osservazione $y(6)$ è associata a

- Livello 2 del fattore A
- Livello 'hi' del fattore B
- Livello 'june' del fattore C

Per calcolare il modello ANOVA va eseguito il comando

```
p = anovan(y, {g1 g2 g3})
```

e l'uscita risulta

```
p = [0.4174 0.0028 0.9140]
```

In pratica il vettore \mathbf{p} contiene i p -values relativi all'ipotesi nulla sugli N effetti principali: l'elemento $\mathbf{p}(1)$ contiene il p -value per l'ipotesi nulla, H_{0A} , che i campioni per tutti i livelli del fattore A sono estratti dalla stessa popolazione; l'elemento $\mathbf{p}(2)$ contiene il p -value per l'ipotesi nulla, H_{0B} , che i campioni per tutti i livelli del fattore A sono estratti dalla stessa popolazione; e così via.

Se qualche p -value è vicino allo 0, sussistono dubbi sull'ipotesi nulla corrispondente, vale a dire per esempio che un p -value piccolo per l'ipotesi H_{0A} suggerisce che almeno un valore medio del campione di A per un livello sia significativamente differente dalle medie degli altri livelli del fattore; in altre parole sussiste un effetto principale per il fattore A . Va scelta pertanto una soglia del p -value per determinare se un risultato va considerato significativo dal punto di vista statistico. Generalmente si assume che un risultato è significativo se il p -value è minore di 0.05 o 0.01.

Il codice `anovan` produce anche una figura che mostra la tabella standard ANOVA che per default ripartisce la variabilità dei dati x nella

- Variabilità dovuta alle differenze tra i livelli di ciascun fattore considerato nel modello (una riga per ogni fattore)
- Variabilità restante non spiegata da nessuna sorgente sistematica

La tabella ANOVA presenta 6 colonne nell'ordine:

- La sorgente della variabilità
- La somma dei quadrati di ciascuna sorgente (SS)
- I gradi di libertà associate a ciascuna sorgente
- Le medie quadratiche (MS), vale a dire i rapporti tra SS e il numero di gradi di libertà
- Il valore dello statistic F , calcolato come il rapporto tra le medie quadratiche
- I p -values corrispondenti allo statistic F .

La tabella viene mostrata nella seguente figura:

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	3.781	1	3.781	0.82	0.4174
X2	199.001	1	199.001	42.95	0.0028
X3	0.061	1	0.061	0.01	0.914
Error	18.535	4	4.634		
Total	221.379	7			

Constrained (Type III) sums of squares.

Per default `anovan` calcola i p-values solo per i tre effetti principali. Come spiegato in precedenza per calcolare i p-values per le interazioni a due fattori, $X1 * X2$, $X1 * X3$ e $X2 * X3$, è sufficiente aggiungere la coppia parametro/valore 'model', 'interaction' come argomenti di ingresso alla chiamata di `anovan`:

```
p = anovan(y, {g1 g2 g3}, 'model', 'interaction')
```

L'uscita in questo caso sarà

```
p = [0.0347 0.0048 0.2578 0.0158 0.1444 0.5000]
```

I primi tre valori di p rendono conto degli effetti principali. Gli ultimi tre forniscono i p-values relativi alle interazioni a due fattori. La corrispondente tabella di uscita fornisce chiaramente l'indicazione delle interazioni considerate:

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	3.781	1	3.781	336.11	0.0347
X2	199.001	1	199.001	17689	0.0048
X3	0.061	1	0.061	5.44	0.2578
X1*X2	18.301	1	18.301	1626.78	0.0158
X1*X3	0.211	1	0.211	18.78	0.1444
X2*X3	0.011	1	0.011	1	0.5
Error	0.011	1	0.011		
Total	221.379	7			

Constrained (Type III) sums of squares.

1.5.2 Esempio su un dataset più ampio

Consideriamo un dataset classico di test relative a 406 automobili costruite tra il 1970 e il 1982.

Innanzitutto si carichi il dataset e si guardino le variabili presenti:

```
load carbig
whos
```

Name	Size	Bytes	Class
Acceleration	406x1	3248	double array
Cylinders	406x1	3248	double array
Displacement	406x1	3248	double array
Horsepower	406x1	3248	double array
MPG	406x1	3248	double array
Model	406x36	29232	char array
Model_Year	406x1	3248	double array
Origin	406x7	5684	char array
Weight	406x1	3248	double array
cyl4	406x5	4060	char array
org	406x7	5684	char array
when	406x5	4060	char array

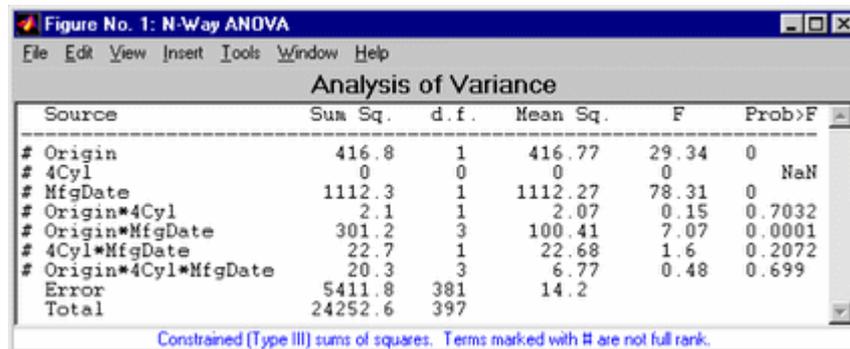
In questo esempio considereremo quattro variabili: MPG (il numero di miglia percorse per gallone per ciascuna delle 406 automobili; i valori mancanti sono indicati con NaN); cyl4 (se l'automobile è a Quattro cilindri o non lo è), org (se l'origine dell'automobile è l'Europa, il Giappone o gli Stati Uniti d'America), when (se l'automobile è stata costruita all'inizio del periodo considerato, nel mezzo o alla fine).

Come prima cosa eseguiamo il modello ANOVA completo, ma richiedendo interazioni fino a 3 fattori, e sommatorie dei quadrati di Tipo 3:

```
varnames = {'Origin'; '4Cyl'; 'MfgDate'};
anovan(MPG, {org cyl4 when}, 3, 3, varnames)
```

```
ans =
    0.0000
      NaN
        0
```

0.7032
 0.0001
 0.2072
 0.6990



Si osservi che diversi termini sono identificati con il simbolo # che sta ad indicare l'assenza di rango pieno; uno di loro ha 0 gradi di libertà e pertanto il corrispondente *p*-value è assente. Tali situazioni capitano ci sono combinazioni di livelli di fattore senza campioni ed il modello contiene termini di interazione di ordine alto. In tal caso può essere di ausilio una tavola incrociata ottenuta mediante la routine `crosstab`; nell'esempio si osserva che non sono presenti nel campione automobile costruite in Europa nel primo periodo con motori diversi dal 4 cilindri (elemento (2,1,1) di `table`):

```
[table, chi2, p, factorvals] = crosstab(org,when,cyl4)
```

```
table(:, :, 1) =
```

```
82    75    25
 0     4     3
 3     3     4
```

```
table(:, :, 2) =
```

```
12    22    38
23    26    17
12    25    32
```

chi2 =

207.7689

p =

0

factorvals =

'USA'	'Early'	'Other'
'Europe'	'Mid'	'Four'
'Japan'	'Late'	['']

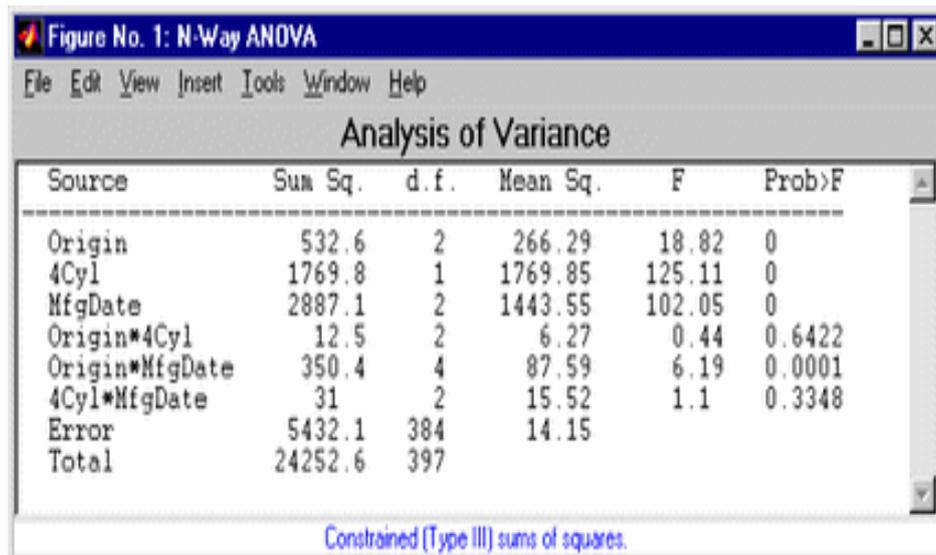
Pertanto risulta impossibile stimare gli effetti dell'interazione a tre fattori e la sua inclusione nel modello lo rende singolare e pertanto non risolvibile. Esaminiamo quindi un modello ANOVA con interazioni a due fattori:

```
[p,tbl,stats,terms] = anovan(MPG,{org cyl4 when},
2,3,varnames);
```

terms

terms =

1	0	0
0	1	0
0	0	1
1	1	0
1	0	1
0	1	1



Ora tutti i termini sono stimabili. I p-values del termine di interazione 4 (Origin*4Cyl) e del termine di interazione 6 (4Cyl*MfgDate) sono molto maggiori del valore tipico di soglia 0.05, il che sta ad indicare che tali termini non sono significativi. Pertanto risulta conveniente omettere completamente tali termini dal modello e raggruppare i loro effetti nel termine di errore. La variabile di uscita `terms` restituisce una matrice di codici; ciascuna riga rappresenta il termine analizzato mediante una rappresentazione di bits 0-1. Per eliminare un termine sarà sufficiente cancellare la corrispondente riga della matrice `terms` ed eseguire di nuovo `anovan`, fornendo in input come argomento di `model` la matrice appena modificata:

```
terms([4 6], :) = []
```

```
terms =
```

```

1      0      0
0      1      0
0      0      1
1      0      1
```

```
anovan(MPG, {org cyl4 when}, terms, 3, varnames)
```

ans =

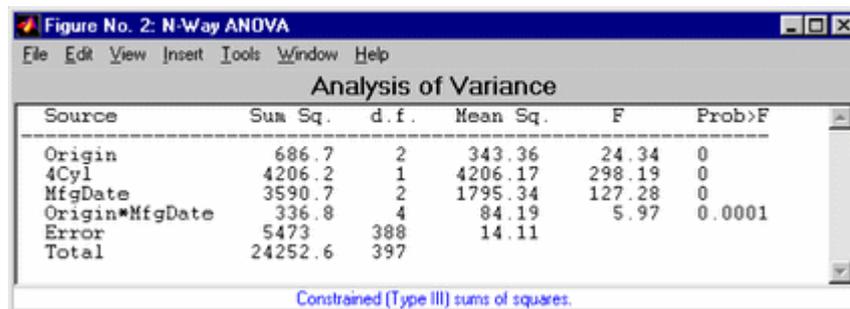
1.0e-003 *

0.0000

0

0

0.1140



Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Origin	686.7	2	343.36	24.34	0
4Cyl	4206.2	1	4206.17	298.19	0
MfgDate	3590.7	2	1795.34	127.28	0
Origin*MfgDate	336.8	4	84.19	5.97	0.0001
Error	5473	388	14.11		
Total	24252.6	397			

Constrained (Type III) sums of squares.

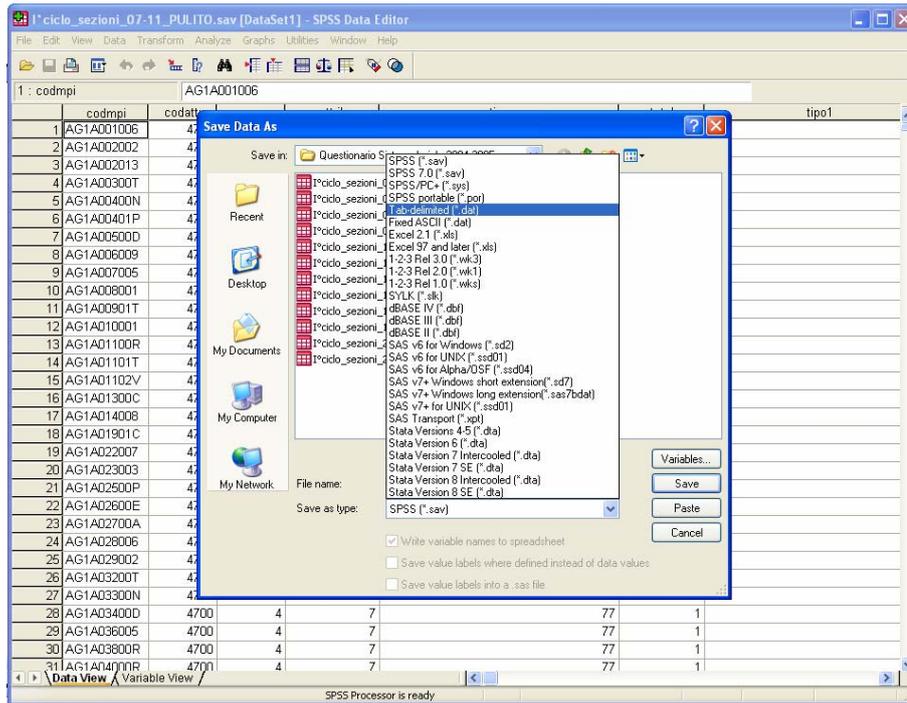
Si ottiene pertanto un modello maggiormente compatto da cui si evince che il consumo specifico sembra legato a tutti i 3 fattori e che l'effetto della data di costruzione dipende da dove sono state costruite le automobili.

2 Lettura dei dati INVALSI e conversione in formato Matlab

I dati INVALSI vengono elaborati mediante la piattaforma SPSS e sono memorizzati in un formato proprietario SPSS non leggibile da altri ambienti di sviluppo. Pertanto è stata sviluppata una procedura per la lettura dei dati mediante SPSS e la loro conversione in formato Matlab

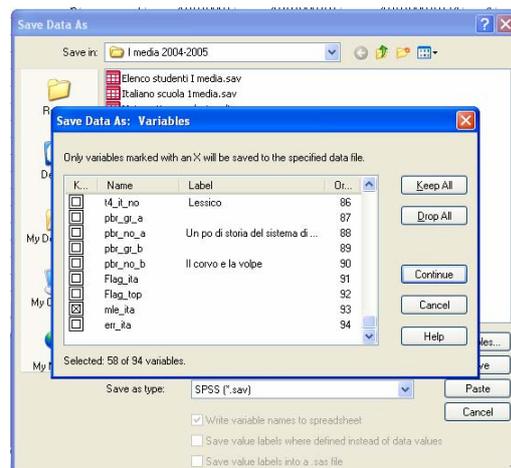
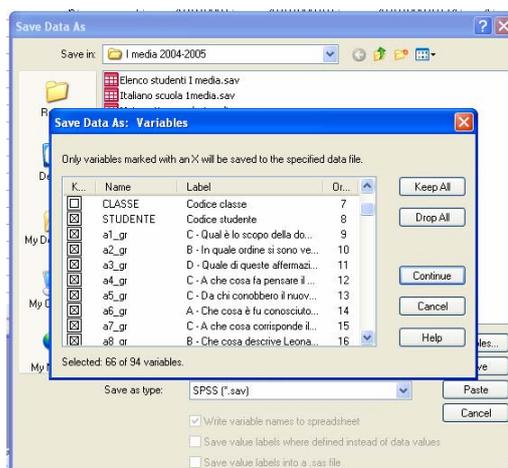
2.1 SPSS ed esportazione di dati

Per la lettura dei dati è stato utilizzato il prodotto SPSS nella versione 14.0. SPSS offre diversi formati per salvare i files di dati, oltre al formato `.sav` proprietario, per esempio `ascii`, `Excel`, `1-2-3`, `dBase`, `SAS`, `Stata`. Per la lettura dei dati mediante Matlab si sono provate due differenti opzioni potenzialmente in grado di permettere lo scambio con Matlab: `ascii` e `Excel`, utilizzando le script Matlab disponibili per la lettura di tali formati. Tuttavia il formato `Excel` si è rilevato maggiormente farraginoso nella gestione di campi misti numerici ed alfanumerici, oltre a creare alcuni problemi di compatibilità nella gestione della virgola come carattere separatore tra parte intera e decimale dei numeri reali. Pertanto si è preferito optare per il formato `ascii` (suffisso `.dat`), in particolare con delimitazione dei campi mediante il caratter `TAB` (`tab-delimited`).



Per consentire una gestione più veloce dei dati da parte di Matlab, si è scelto in questa prima fase del progetto di selezionare un numero limitato di variabili da trasferire al nuovo ambiente di sviluppo.

Nel caso del questionario di valutazione tali campi sono: Codice Studente, Sesso, Indice della risposta ad ogni domanda del questionario, calcolo dell'abilità dello studente mediante Item Response Theory.



Si osservi che il Codice studente è a sua volta formato da tre sottocampi e comprende all'interno sia il codice della Scuola, sia il codice della sezione relativo alla scuola. Si è scelto di riportare nel nuovo formato la risposta fornita alla singola domanda del

questionario per consentire nel prosieguo della ricerca analisi più dettagliate riguardo il raggruppamento delle domande in sottodiscipline omogenee (per esempio quelle previste già in partenza dai questionari). L'abilità dello studente valutata con Item Response Theory a 3 parametri è stata inserita nel nuovo formato perché la sua stima può avvenire con diverse metodologie; pertanto riportando direttamente il campo dal database originale INVALSI si è sicuri di lavorare con dati omogenei. Infine sono stati intenzionalmente omessi tutti i dati relativi alle Scuole, perché vengono desunti direttamente dal database delle Scuole.

Inoltre i dati relativi agli Istituti scolastico sono riportati nel file Database SNV 2004-05.xls reso disponibile direttamente da INVALSI in formato Excel. In particolare sono stati considerati i seguenti campi: Attributo (relativo al tipo di scuola se statale o non), codice di Regione, codice di provincia, CAP, codice di scuola elementare, media inferiore o media superiore; nello stesso file sono state aggiunte latitudine e longitudine di numerose scuole della Campania, che saranno utilizzate per l'analisi degli andamenti frattali della abilità degli studenti. Tali campi sono accessibili ad esterni per motivi di riservatezza dei dati.

3 Codice per la gestione dei questionari e l'esecuzione dei modelli ANOVA

3.1 Introduzione

Nel presente capitolo riportiamo i codici sviluppati per l'analisi dei questionari di valutazione mediante modelli ANOVA. In particolare il paragrafo 3.2 conterrà il codice generale che guida la lettura di tutti i dati ed organizza l'analisi ANOVA divisa per diverse tipologie, producendo i relativi grafici e tabelle. Il paragrafo 3.3 sarà riferito alla lettura dei dati relative ai questionari di valutazione, convertiti nel formato Matlab. Infine il paragrafo 3.4 conterrà il codice per la lettura dei dati relativi agli Istituti scolastici.

3.2 Codice per l'analisi ANOVA

La variabile principale del codice è `School`, di tipo struttura, che consiste dei seguenti campi:

- `code`: indica il codice della scuola come fornito da INVALSI
- `tipo`: indica il tipo di scuola; nella presente analisi vengono considerati esclusivamente il tipo `Statale` e `Privata`, equiparando pertanto tutte le scuole paritarie e non paritarie
- `ordine`: indica se si tratta di scuola elementare, media inferiore o media superiore
- `regione`: indica mediante un codice compreso tra 1 e 20 la regione di appartenenza della scuola
- `latitudine`: indica la latitudine della scuola (dato sensibile e non disponibile)
- `longitudine`: indica la longitudine della scuola (dato sensibile e non disponibile)
- `perc`: percentuale di risposte esatte per la scuola, calcolata su tutte le domande di tutte le materie oggetto del questionario e ripartita per sesso; valore medio

- perc: percentuale di risposte esatte per la scuola, calcolata su tutte le domande di tutte le materie oggetto del questionario e ripartita per sesso; valore mediano
- percmaterie: percentuale di risposte esatte per la scuola, calcolata su tutte le domande, ma divisa per materia (Italiano, Matematica, Scienze) e per sesso; valore medio
- percmaterie: percentuale di risposte esatte per la scuola, calcolata su tutte le domande, ma divisa per materia (Italiano, Matematica, Scienze) e per sesso; valore mediano
- mle: indice di abilità valutato mediante Item Response Theory per la scuola, calcolato su tutte le domande di tutte le materie oggetto del questionario e ripartito per sesso; valore medio
- mlemed: indice di abilità valutato mediante Item Response Theory per la scuola, calcolato su tutte le domande di tutte le materie oggetto del questionario e ripartito per sesso; valore mediano
- mlematerie: indice di abilità valutato mediante Item Response Theory per la scuola, calcolato su tutte le domande, ma diviso per materia (Italiano, Matematica, Scienze) e per sesso; valore medio
- mlemedmaterie: indice di abilità valutato mediante Item Response Theory per la scuola, calcolato su tutte le domande, ma diviso per materia (Italiano, Matematica, Scienze) e per sesso; valore mediano

La prima fase del codice comprende la lettura dei dati di valutazione dei singoli Istituti (richiamando la script `readvalutazione` descritta nel seguito); quindi si procede alla lettura dei dati di Istituto, mediante un incrocio dei corrispondenti database. Infine nella terza fase si procede all'analisi ANOVA. In particolare verranno considerati separatamente i casi delle singole materie (Italiano, Matematica e Scienze) ed il caso di tutte le materie (includendo cioè tutte le risposte ottenute nei questionari di Italiano, Matematica e Scienze). Per ognuno di questi verranno analizzate sia le percentuali di risposte esatte fornite dagli studenti, sia l'indice di abilità stimato con IRT. Per ognuno di questi casi verrà eseguita l'analisi di varianza a 1 fattore per i seguenti fattori: Tipo (con 2 livelli: Istituto pubblico o privato); Regione (con 20 livelli dati dalle regioni); Ordine (con due livelli: Scuola Elementare e Media di primo

grado); Sesso (due livelli: Maschi e Femmine). Il codice contiene anche alcune analisi a più fattori che verranno discusse in dettaglio nel report sulle analisi ANOVA.

```

verbose=0;
parent=fullfile('d','data','invalsi','matlab');

Maxschool=15000;;
School.code=zeros(Maxschool,1)+NaN;
School.tipo=zeros(Maxschool,1)+NaN;
School.ordine=zeros(Maxschool,1)+NaN;
School.regione=zeros(Maxschool,1)+NaN;
School.latitudine=zeros(Maxschool,1)+NaN;
School.longitudine=zeros(Maxschool,1)+NaN;
School.perc=zeros(Maxschool,3)+NaN; % totale sulle discipline
School.percmed=zeros(Maxschool,3)+NaN; % totale sulle discipline
School.percmaterie=zeros(Maxschool,3,3)+NaN; % (scuole,materie,sesso)
School.percmedmaterie=zeros(Maxschool,3,3)+NaN; % (scuole,materie,sesso)
School.mle=zeros(Maxschool,3)+NaN;
School.mlemed=zeros(Maxschool,3)+NaN;
School.mlematerie=zeros(Maxschool,3,3)+NaN;
School.mlemedmaterie=zeros(Maxschool,3,3)+NaN;

Nschool=0;

for iordine=1:2
    switch iordine
        case 1,
            %SCUOLA ELEMENTARE
                M=6;
                group={1 4; 2 5; 3 6}; % elementare
                if verbose>=0, fprintf('Analisi delle scuole elementari\n'), end

            %2 elementare
                fname=fullfile(parent,'ita2elem0405.dat');
                order='cplsm'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[14];
                lab={'Italiano'};
                readvalutazione
                totschooll(1)=school;

                fname=fullfile(parent,'mat2elem0405.dat');
                order='csp2m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[12 4];
                lab={'Algebra','Geometria'};
                readvalutazione
                totschooll(2)=school;

                fname=fullfile(parent,'sci2elem0405.dat');
                order='csp1m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[10];
                lab={'Scienze'};
                readvalutazione
                totschooll(3)=school;

            %4 elementare
                fname=fullfile(parent,'ita4elem0405.dat');
                order='csp4m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[7 12 4 7];
                lab={'Comprensione globale','Comprensione particolare','Lessico','Altro'};
                readvalutazione
                totschooll(4)=school;

                fname=fullfile(parent,'mat4elem0405.dat');
                order='csp3m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[14 8 6];
                lab={'Algebra','Geometria','Misura e dati'};
                readvalutazione
                totschooll(5)=school;

                fname=fullfile(parent,'sci4elem0405.dat');
                order='csp4m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
                answ=[9 8 5 6];
    
```

```

        lab={'Uomo/Ambiente', 'Viventi/Non viventi', 'Elementi di metodo
sperimentale', 'Trasformazioni'};
        readvalutazione
        totscool(6)=school;

        case 2
%SCUOLA MEDIA

        M=3;
        group={1;2;3}; % elementare
        if verbose>=0, fprintf('Analisi delle scuole medie\n'), end

%I media

        fname=fullfile(parent, 'italmedia0405.dat');
        order='csp4m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
        answ=[6 9 9 6];
        lab={'Comprensione globale', 'Comprensione particolare', 'Morfologia,
sintassi, semantica', 'Lessico'};
        readvalutazione
        totscool(1)=school;

        fname=fullfile(parent, 'matlmedia0405.dat');
        order='csp3m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
        answ=[14 8 6];
        lab={'Algebra', 'Geometria', 'Misura e dati'};
        readvalutazione
        totscool(2)=school;

        fname=fullfile(parent, 'scilmedia0405.dat');
        order='csp4m'; % c: codestud; pn: perc with n groups; s: sex; m: mle
        answ=[7 7 7 7];
        lab={'Uomo/Ambiente', 'Viventi/Non viventi', 'Elementi di metodo
sperimentale', 'Trasformazioni'};
        readvalutazione
        totscool(3)=school;

    end

    schoolmax=zeros(M,1);
    for m=1:M
        schoolmax(m)=length(totscool(m).code);
    end

    i=ones(M,1);

    while max(i)>0,
        Nschool=Nschool+1;
        if mod(Nschool,1000)==0 & verbose>=1, fprintf('%i.', Nschool); end
%cerca il minimo di codeschool(i)
        actschool=1e30;
        ind=zeros(M,1);
        for m=1:M
            if i(m)>0,
                if totscool(m).code(i(m)) < actschool,
                    actschool=totscool(m).code(i(m));
                end
            end
        end
        for m=1:M
            if i(m)>0,
                if totscool(m).code(i(m)) == actschool,
                    ind(m)=1;
                end
            end
        end
    end

    School.code(Nschool)=actschool;

    pp2=zeros(1,3); mm2=zeros(1,3); pp2med=zeros(1,3); mm2med=zeros(1,3);
    totstud2=zeros(1,3, 'int32');
    for imateria=1:3
        pp=zeros(1,3); mm=zeros(1,3); ppmed=zeros(1,3); mmmmed=zeros(1,3);
        totstud=zeros(1,3, 'int32');
        for igroup=1:length(group(imateria,:))
            indgroup=group{imateria, igroup};

```

```

    if i(indgroup)>0 %& totschool(indgroup).N(i(m),3) > 0,
        totstud=totstud+totschool(indgroup).N(i(indgroup),:);

pp=pp+totschool(indgroup).perc(i(indgroup),:).*double(totschool(indgroup).N(i(indgroup),:));

ppmed=ppmed+totschool(indgroup).percmed(i(indgroup),:).*double(totschool(indgroup).N(i(indgroup),:));

mm=mm+totschool(indgroup).mle(i(indgroup),:).*double(totschool(indgroup).N(i(indgroup),:));

mmmed=mmmed+totschool(indgroup).mlemed(i(indgroup),:).*double(totschool(indgroup).N(i(indgroup),:));
    end
    end
    pp2=pp2+pp; pp2med=pp2med+ppmed; mm2=mm2+mm; mm2med=mm2med+mmmed;
totstud2=totstud2+totstud;
    indd=find(totstud>0);
    School.percmaterie(Nschool,imateria,indd)=pp(indd)./double(totstud(indd));

School.percmedmaterie(Nschool,imateria,indd)=ppmed(indd)./double(totstud(indd));
    School.mlematerie(Nschool,imateria,indd)=mm(indd)./double(totstud(indd));

School.mlemedmaterie(Nschool,imateria,indd)=mmmed(indd)./double(totstud(indd));
    end
    indd2=find(totstud2>0);
    School.perc(Nschool,indd2)=pp2(indd2)./double(totstud2(indd2));
    School.percmed(Nschool,indd2)=pp2med(indd2)./double(totstud2(indd2));
    School.mle(Nschool,indd2)=mm2(indd2)./double(totstud2(indd2));
    School.mlemed(Nschool,indd2)=mm2med(indd2)./double(totstud2(indd2));
    School.ordine(Nschool)=iordine; % elementare o media

    for m=1:M
        if ind(m)>0, i(m)=i(m)+1; end
    end
    for m=1:M
        if(i(m)>schoolmax(m)),
            i(m)=-1; % Ended school of group m
        end
    end
end
end % of while
if verbose>=0, fprintf('Done.\n'), end

clear totscool

clear school;

%if verbose>=0, fprintf('Done.\n'), end

end
School.code(Nschool+1:Maxschool)=[ ];
School.tipo(Nschool+1:Maxschool)=[ ];
School.ordine(Nschool+1:Maxschool)=[ ];
School.regione(Nschool+1:Maxschool)=[ ];
School.latitudine(Nschool+1:Maxschool)=[ ];
School.longitudine(Nschool+1:Maxschool)=[ ];
School.perc(Nschool+1:Maxschool,:)=[];
School.percmed(Nschool+1:Maxschool,:)=[];
School.percmaterie(Nschool+1:Maxschool,,:)=[];
School.percmedmaterie(Nschool+1:Maxschool,,:)=[];
School.mle(Nschool+1:Maxschool,:)=[];
School.mlemed(Nschool+1:Maxschool,:)=[];
School.mlematerie(Nschool+1:Maxschool,,:)=[];
School.mlemedmaterie(Nschool+1:Maxschool,,:)=[];

readscuole;

Ndataschool=length(Dataschool.attributo);
indelem=find(~isnan(Dataschool.codelem));
nelem=length(indelem);
indmedie=find(~isnan(Dataschool.codmedie));
nmedie=length(indmedie);

```

```

if verbose>=0, fprintf('Numero di scuole elementari nel database delle scuole:
%i\n',nelem), end
if verbose>=0, fprintf('Numero di scuole medie nel database delle scuole:
%i\n',nmedie), end
if verbose>=0, fprintf('Merging database della valutazione e database delle
scuole...'); end

j0=0;

for i=1:Nschool
    if mod(i,1000)==0 & verbose>=1, , fprintf('%i..',i), end
    actcode=School.code(i);
    switch School.ordine(i)
        case 1 % elementari
            found=0;
            for j=min(j0+1,nelem):nelem
                if actcode == Dataschool.codelem(indelem(j)),
                    School.tipo(i)=Dataschool.attributo(indelem(j));
                    School.regione(i)=Dataschool.regione(indelem(j));
                    School.latitudine(i)=Dataschool.latitudine(indelem(j));
                    School.longitudine(i)=Dataschool.longitudine(indelem(j));
                    j0=j; found=1;
                    break
                end
            end
            if found==0;
                for j=1:j0
                    if actcode == Dataschool.codelem(indelem(j)),
                        School.tipo(i)=Dataschool.attributo(indelem(j));
                        School.regione(i)=Dataschool.regione(indelem(j));
                        School.latitudine(i)=Dataschool.latitudine(indelem(j));
                        School.longitudine(i)=Dataschool.longitudine(indelem(j));
                        j0=j; found=1;
                        break
                    end
                end
            end % of if found==0;
            if found==0,
                if verbose>=2, fprintf('Codice della scuola %i non
trovato\n',actcode); end
            end
            case 2 % medie
                found=0;
                for j=min(j0+1,nmedie):nmedie
                    if actcode == Dataschool.codmedie(indmedie(j)),
                        School.tipo(i)=Dataschool.attributo(indmedie(j));
                        School.regione(i)=Dataschool.regione(indmedie(j));
                        School.latitudine(i)=Dataschool.latitudine(indmedie(j));
                        School.longitudine(i)=Dataschool.longitudine(indmedie(j));
                        j0=j; found=1;
                        break
                    end
                end
                if found==0;
                    for j=1:j0
                        if actcode == Dataschool.codmedie(indmedie(j)),
                            School.tipo(i)=Dataschool.attributo(indmedie(j));
                            School.regione(i)=Dataschool.regione(indmedie(j));
                            School.latitudine(i)=Dataschool.latitudine(indmedie(j));
                            School.longitudine(i)=Dataschool.longitudine(indmedie(j));
                            j0=j; found=1;
                            break
                        end
                    end
                end
            end % of if found==0;
            if found==0,
                if verbose>=2, fprintf('Codice della scuola %i non
trovato\n',actcode); end
            end
            otherwise
                if verbose>=1, fprintf('School.tipo non trovato per scuola %i\n',actcode);
            end
        end % of switch School.code
    end % of for i=1:Nschool
    if verbose>=0, fprintf('Fatto.\n'); end

```

```

ind=find(School.tipo>=1 & School.tipo<=3 & School.ordine>=1 & School.ordine<=3 &
School.regione>=1 & School.regione<=20);
ind0=1:Nschool;
ind0(ind)=[];
School.code(ind0)=[];
School.tipo(ind0)=[];
School.ordine(ind0)=[];
School.regione(ind0)=[];
School.latitudine(ind0)=[];
School.longitudine(ind0)=[];
School.perc(ind0,:)=[];
School.percmed(ind0,:)=[];
School.mle(ind0,:)=[];
School.mlemed(ind0,:)=[];
School.percmaterie(ind0,:)=[];
School.percmedmaterie(ind0,:)=[];
School.mlematerie(ind0,:)=[];
School.mlemedmaterie(ind0,:)=[];

% Merge scuole paritarie (tipo=2) e non paritarie (tipo=3)
School.tipo(find(School.tipo==3))=2;
Nschool=length(School.tipo);
if verbose>=0, fprintf('Numero di scuole dopo filtraggio dati mancanti:
%i\n',Nschool), end

clear totschoole indelem ind Dataschoole

%ANOVA
fattore={'Tipo','Regione','Ordine','Sesso'};

ntipo=2;
legtipo={'Pubblico','Privato'};

nordine=2;
legordine={'Elementare','Media I grado'};

nregione=20;
legregione={'Valle d''Aosta','Piemonte','Liguria','Lombardia','Trentino',...
'Veneto','Friuli','Emilia','Toscana','Umbria','Marche','Lazio','Abruzzo',...
'Molise','Campania','Puglia','Basilicata','Calabria','Sicilia','Sardegna'};

nsesto=2;
legsesto={'Femmine','Maschi'};

for imat=1:4

    for iabil=1:2
        switch iabil
            case 1
                tit01='Percentuale';
            case 2
                tit01='Abilità';
        end
        switch imat
            case 1,
                switch iabil
                    case 1,
                        target=School.perc(:,3); % Italiano-Matematica-Scienze
                        indfemale=find(~isnan(School.perc(:,1)));
                        indmale=find(~isnan(School.perc(:,2)));
                        targetfemale=School.perc(indfemale,1);
                        targetmale=School.perc(indmale,2);
                    case 2,
                        target=School.mle(:,3); % Italiano-Matematica-Scienze
                        indfemale=find(~isnan(School.mle(:,1)));
                        indmale=find(~isnan(School.mle(:,2)));
                        targetfemale=School.mle(indfemale,1);
                        targetmale=School.mle(indmale,2);
                end
                tit00='Italiano-Matematica-Scienze';
            case 2,
                switch iabil
                    case 1,
                        target=School.percmaterie(:,1,3); % Italiano
                        indfemale=find(~isnan(School.percmaterie(:,1,1)));
                        indmale=find(~isnan(School.percmaterie(:,1,2)));

```

```

        targetfemale=School.percmaterie(indfemale,1,1);
        targetmale=School.percmaterie(indmale,1,2);
    case 2,
        target=School.mlematerie(:,1,3); % Italiano
        indfemale=find(~isnan(School.mlematerie(:,1,1)));
        indmale=find(~isnan(School.mlematerie(:,1,2)));
        targetfemale=School.mlematerie(indfemale,1,1);
        targetmale=School.mlematerie(indmale,1,2);
    end
    tit00='Italiano';
case 3,
    switch iabil
    case 1,
        target=School.percmaterie(:,2,3); % Matematica
        indfemale=find(~isnan(School.percmaterie(:,2,1)));
        indmale=find(~isnan(School.percmaterie(:,2,2)));
        targetfemale=School.percmaterie(indfemale,2,1);
        targetmale=School.percmaterie(indmale,2,2);
    case 2,
        target=School.mlematerie(:,2,3); % Matematica
        indfemale=find(~isnan(School.mlematerie(:,2,1)));
        indmale=find(~isnan(School.mlematerie(:,2,2)));
        targetfemale=School.mlematerie(indfemale,2,1);
        targetmale=School.mlematerie(indmale,2,2);
    end
    tit00='Matematica';
case 4,
    switch iabil
    case 1,
        target=School.percmaterie(:,3,3); % Scienze
        indfemale=find(~isnan(School.mlematerie(:,3,1)));
        indmale=find(~isnan(School.mlematerie(:,3,2)));
        targetfemale=School.percmaterie(indfemale,3,1);
        targetmale=School.percmaterie(indmale,3,2);
    case 2,
        target=School.mlematerie(:,3,3); % Scienze
        indfemale=find(~isnan(School.mlematerie(:,3,1)));
        indmale=find(~isnan(School.mlematerie(:,3,2)));
        targetfemale=School.mlematerie(indfemale,3,1);
        targetmale=School.mlematerie(indmale,3,2);
    end
    tit00='Scienze';
end
tit0=[tit01 ' ' tit00];
fprintf('\n\n ===== %s =====\n',tit0);

%1-way analysis on 'tipo'
fact=fattore{1};
nleg=ntipo; leg=legtipo;
[p,table,stats,terms]=anovan(target,{School.tipo},...
    'varnames',{fact},'display','off',...
    'model','full'... % 'full', 'linear', 'interaction', integer, matrix
);
stats.grpnames={leg};
[c,m,h,gnames]=multcompare(stats,'display','off');
figure(imat*100+(iabil-1)*50+1);
tit=['1-way ANOVA su ' fact ' - ' tit0];
set(gcf,'Name',tit);
plot(m(:,1),[1:nleg],'o')
hold on
for i=1:nleg
    plot([m(i,1)-m(i,2),m(i,1)+m(i,2)],[i,i])
end
axis([min(m(:,1)-m(:,2)) max(m(:,1)+m(:,2)) 0.5 nleg+0.5]);
title(tit)
xlabel(tit0)
set(gca,'Ytick',[1:nleg])
set(gca,'YtickLabel',leg)
fprintf('\n** 1-way ANOVA - Fattore: tipo\n');
fprintf('Grandmean: %6.2f\n',stats.coeffs(1));
fprintf('F-test: %d - Liv. signif.: %i\n',table{2,6},table{2,7});
for i=1:nleg
    fprintf('%s %i: %s - Estimated mean: %6.2f +/-
    %6.2f\n',fact,i,leg{i},m(i,1),m(i,2));
end

```

```

%1-way analysis on 'regione'
fact=fattore{2};
nleg=nregione; leg=legregione;
[p,table,stats,terms]=anovan(target,{School.regione},...
    'varnames',{'Regione'},'display','off',...
    'model','full'... % 'full', 'linear', 'interaction', integer, matrix
);
stats.grpnames={leg};
[c,m,h,gnames]=multcompare(stats,'display','off');
figure(imat*100+(iabil-1)*50+2);
tit=['1-way ANOVA su ' fact ' - ' tit0];
set(gcf,'Name',tit);
plot(m(:,1),[1:nleg],'o')
hold on
for i=1:nleg
    plot([m(i,1)-m(i,2),m(i,1)+m(i,2)],[i,i])
end
axis([min(m(:,1)-m(:,2)) max(m(:,1)+m(:,2)) 0.5 nleg+0.5]);
title(tit)
xlabel(tit0)
set(gca,'Ytick',[1:nleg])
set(gca,'YtickLabel',leg)
fprintf('\n** 1-way ANOVA - Fattore: regione\n');
fprintf('Grandmean: %6.2f\n',stats.coeffs(1));
fprintf('F-test: %d - Liv. signif.: %i\n',table{2,6},table{2,7});
for i=1:nleg
    fprintf('%s %i: %s - Estimated mean: %6.2f +/-
%6.2f\n',fact,i,leg{i},m(i,1),m(i,2));
end

matdiffreg=zeros(nregione,nregione);
matdiffreg2=zeros(nregione,nregione);
for i=1:size(c,1)
    matdiffreg(c(i,1),c(i,2))=c(i,4);
    matdiffreg(c(i,2),c(i,1))=matdiffreg(c(i,1),c(i,2));
    matdiffreg2(c(i,1),c(i,2))=c(i,4)/(c(i,5)-c(i,3));
    matdiffreg2(c(i,2),c(i,1))=matdiffreg2(c(i,1),c(i,2));
end

figure(imat*100+(iabil-1)*50+3)
tit=['Diff medie tra regioni - ' tit0];
set(gcf,'Name',tit);
imagesc(matdiffreg)
colorbar
title(tit);

figure(imat*100+(iabil-1)*50+4)
tit=['Diff medie normalizzate tra regioni - ' tit0];
set(gcf,'Name',tit);
imagesc(matdiffreg2)
colorbar
title(tit);

%1-way analisi su 'ordine'
fact=fattore{3};
nleg=nordine; leg=legordine;
[p,table,stats,terms]=anovan(target,{School.ordine},...
    'varnames',{'Ordine'},'display','off',...
    'model','full'... % 'full', 'linear', 'interaction', integer, matrix
);
stats.grpnames={leg};
[c,m,h,gnames]=multcompare(stats,'display','off');
figure(imat*100+(iabil-1)*50+5);
tit=['1-way ANOVA su ' fact ' - ' tit0];
set(gcf,'Name',tit);
plot(m(:,1),[1:nleg],'o')
hold on
for i=1:nleg
    plot([m(i,1)-m(i,2),m(i,1)+m(i,2)],[i,i])
end
axis([min(m(:,1)-m(:,2)) max(m(:,1)+m(:,2)) 0.5 nleg+0.5]);
title(tit)
xlabel(tit0)
set(gca,'Ytick',[1:nleg])
set(gca,'YtickLabel',leg)
fprintf('\n** 1-way ANOVA - Fattore: ordine\n');
    
```

```

fprintf('Grandmean: %6.2f\n',stats.coeffs(1));
fprintf('F-test: %d - Liv. signif.: %i\n',table{2,6},table{2,7});
for i=1:nleg
    fprintf('%s %i: %s - Estimated mean: %6.2f +/-
%6.2f\n',fact,i,leg{i},m(i,1),m(i,2));
end

%1-way analisi su 'sesso'
fact=fattore{4};
nleg=nsesso; leg=legsesso;
[p,table,stats,terms]=anovan([targetfemale;targetmale],...
    {[School.ordine(indfemale);School.ordine(indmale)]},...
    'varnames',{'Ordine'},'display','off',...
    'model','full'... % 'full', 'linear', 'interaction', integer, matrix
);
stats.grpnames={leg};
[c,m,h,gnames]=multcompare(stats,'display','off');
figure(imat*100+(iabil-1)*50+6);
tit=['1-way ANOVA su ' fact ' - ' tit0];
set(gcf,'Name',tit);
plot(m(:,1),[1:nleg],'o')
hold on
for i=1:nleg
    plot([m(i,1)-m(i,2),m(i,1)+m(i,2)],[i,i])
end
axis([min(m(:,1)-m(:,2)) max(m(:,1)+m(:,2)) 0.5 nleg+0.5]);
title(tit)
xlabel(tit0)
set(gca,'Ytick',[1:nleg])
set(gca,'YtickLabel',leg)
fprintf('\n** 1-way ANOVA - Fattore: sesso\n');
fprintf('Grandmean: %6.2f\n',stats.coeffs(1));
fprintf('F-test: %d - Liv. signif.: %i\n',table{2,6},table{2,7});
for i=1:nleg
    fprintf('%s %i: %s - Estimated mean: %6.2f +/-
%6.2f\n',fact,i,leg{i},m(i,1),m(i,2));
end

% matr=zeros(2,20);
% for i=1:Nschool
%     matr(School.tipo(i),School.regione(i))=matr(School.tipo(i),School.regione(i))+1;
% end
%
% Drop Valle D'Aosta because has no private elementary schools
% ind=find(School.regione~=0);
% fprintf('Number of schools after removal of Valle d'Aosta (without private
schools): %i\n',length(ind));
%
% [p,table,stats,terms]=anovan(target(ind),{School.tipo(ind), School.regione(ind)},...
%     'varnames',{'Tipo', 'Regione'},'display','on',...
%     'model','full'... % 'full', 'linear', 'interaction', integer, matrix
% );

%2-way analysis su tipo - regione
% [p,table,stats,terms]=anovan(target,{School.tipo, School.regione},...
%     'varnames',{'Tipo', 'Regione'},'display','on',...
%     'model','full'... % 'full', 'linear', 'interaction', integer, matrix
% );
%
% c=multcompare(stats);

    end % of for iabil=1:2
end % of for imat=1:4

%Exmaple on 2 factors
%[p,table,stats,terms]=anovan(target,{School.tipo
School.ordine},'varnames',{'Tipo','Ordine'},'display','on','model','full');
%stats.grpnames={legtipo legordine};
%[c,m,h,gnames]=multcompare(stats,'dim',[1 2],'display','on');

%Analisi per gruppi di regioni
%ind1=find(School.regione<=13 | School.regione==20);
%[p,table,stats,terms]=anovan(target(ind1),{School.regione(ind1)},'varnames',{'Regione
'},'display','on','model','full');

```

```
%[c,m,h,gnames]=multcompare(stats,'display','on');  
%ind2=find(School.regione>13 & School.regione<20);  
%[p,table,stats,terms]=anovan(target(ind2),{School.regione(ind2)},'varnames',{'Regione'  
'},'display','on','model','full');  
%[c,m,h,gnames]=multcompare(stats,'display','on');
```

3.3 Codice per la lettura dei questionari di valutazione

Il codice si prefigge lo scopo di leggere l'intero database di risposte degli studenti e di fornire un dato di percentuale media di risposte esatte e abilità dello studente secondo l'IRT aggregato per singolo Istituto. I dati sono memorizzati nella struttura `school` (da non confondere con la struttura `School` del codice ANOVA descritta nella sezione precedente), di cui condivide i campi relativi alla valutazione. La lettura del file è resa farraginosa dal fatto che l'ordine delle variabili nei database SPSS originari INVALSI non è sempre uguale a seconda del database (per classi elementari o medie). Inoltre vengono ricavati alcuni indicatori statistici sul numero di studenti e sul numero di scuole presenti.

La lettura effettiva dei dati avviene utilizzando la potente script `textscan` di Matlab, in grado di leggere database ascii ordinati per campi, impostando la delimitazione degli stessi mediante il carattere TAB. Si osservi che la configurazione scelta per la conversione del database da SPSS a Matlab e la lettura successiva in ambiente Matlab ha consentito di leggere il 100% dei dati originari SPSS correttamente.

Tutte le analisi dei campi, anche numerici, sono effettuate nel codice mediante opportune conversioni; in particolare in questo modo si è risolto il problema dell'interpretazione delle virgole usate come carattere separatore tra la parte intera e quella decimale dei numeri reali.

```

NA=-999;
Nmax=30000; % Maximum number of schools

if verbose>=0, fprintf('\nDatabase %s\n',fname); end
fid=fopen(fname,'r');
wh=strfind(order,'p');
ngroup=max(1,str2num(order(wh+1)));
nfields=3+ngroup;

% Read heading
a=textscan(fid,'%s',nfields);

% Non funziona causa le virgole presenti nei file di uscita SPSS
%b=textscan(fid,'%d%d%n%d%n','delimiter','\t');

% legge numeri in virgola mobile come caratteri, sostituisce virgola con
% punto e converte in decimali
if verbose>0, fprintf('Lettura dati...'); end
format=[];
i=0;
for ifields=1:4
    i=i+1;

```

```

switch order(i)
    case 'c',
        format=[format '%n'];
    case 's',
        format=[format '%d8'];
    case 'p',
        for j=1:ngroup
            format=[format '%s'];
        end
        i=i+1;
    case 'm'
        format=[format '%s'];
end
end

b=textscan(fid,format,'delimiter','\t');
if verbose >0, fprintf('Done.\n'), end
N=size(b{1},1);
if verbose>=0, fprintf(' Numero di studenti: %i\n',N); end

label=0;
ii=0;
for ifields=1:4,
    ii=ii+1;
    switch order(ii)
        case 'c',
            if verbose >0, fprintf('Analisi di codestud..'), end
            label=label+1;
            codestud=b{label};
            schoolcode=round(codestud/1e4);
            class=round((codestud-schoolcode*1e4)/1e2);
            student=codestud-schoolcode*1e4-class*1e2;
            clear codestud;
        case 's',
            if verbose>0, fprintf('sesso..'), end
            label=label+1;
            sesso=b{label};
        case 'p',
            ii=ii+1;
            percgroup=zeros(N,ngroup)+NA;
            perc=zeros(N,1);
            for j=1:ngroup
                label=label+1;
                if verbose>0, fprintf('perc-%i..',j), end
                percs=b{label};
                percs=strrep(percs,',','.');
                percs=char(percs);
                maxpercs=max(percs,[],2);
                minpercs=min(percs,[],2);
                ind=find(~(maxpercs == 32 & minpercs == 32));
                % Conversione da fare a pezzi perche' per intero richiede troppo
            end
            % Contattare Mathworks
            %perc(ind)=str2num(percs);
            block=400;
            Nind=length(ind);
            Nloop=ceil(N/block);
            k2=0;
            for loop=1:Nloop
                i1=(loop-1)*block+1;
                i2=min(loop*block,N);
                dum=str2num(percs(i1:i2,:));
                ndum=length(dum);
                k1=k2+1;
                k2=k1+ndum-1;
                percgroup(ind(k1:k2),j)=dum;
            end % of for loop=1:Nloop
            perc(ind)=perc(ind)+percgroup(ind,j);
            percgroup(ind,j)=100*percgroup(ind,j)/answ(j);
        end % of for j=1:ngroup
        perc(ind)=perc(ind)*100/sum(answ);
    case 'm'
        if verbose>0, fprintf('mle..'), end
        label=label+1;
        mles=b{label};
        mle=zeros(N,1)+NA;
    end
end

```

```

mles=strrep(mles,',','.');
mles=char(mles);
nchar=size(mles,2);
nchar2=min(nchar,6);
mles=mles(:,1:nchar2);
maxmles=max(mles,[],2);
minmles=min(mles,[],2);
ind=find(~(maxmles == 32 & minmles == 32));
% Conversione da fare a pezzi perche' per intero richiede troppo tempo!
% Contattare Mathworks
% mle(ind)=str2num(mles3);
block=400;
Nind=length(ind);
Nloop=ceil(N/block);
k2=0;
for loop=1:Nloop
    i1=(loop-1)*block+1;
    i2=min(loop*block,N);
    dum=str2num(mles(i1:i2,:));
    ndum=length(dum);
    k1=k2+1;
    k2=k1+ndum-1;
    mle(ind(k1:k2))=dum;
end
end % of switch order(i)
end % of for ifields=1:4,

if verbose>0, fprintf('Fatto.\n'), end
fclose(fid);

clear b minmles maxmles minpercs maxpercs mles percs;

%% Processing data school
if verbose>=0, fprintf(' Analisi dei dati di studente...'), end
if verbose>=2, fprintf(' Scuola '), end

student=[student;-1];
class=[class;-1];
school.perc=zeros(Nmax,3,'single')+NaN;
school.percstd=zeros(Nmax,3,'single')+NaN;
school.percmed=zeros(Nmax,3,'single')+NaN;
school.percgroup=zeros(Nmax,3,ngroup,'single')+NaN;
school.percgroupstd=zeros(Nmax,3,ngroup,'single')+NaN;
school.mle=zeros(Nmax,3,'single')+NaN;
school.mlestd=zeros(Nmax,3,'single')+NaN;
school.mlemed=zeros(Nmax,3,'single')+NaN;
school.code=zeros(Nmax,1,'int32');
school.N=zeros(Nmax,3,'int32'); % Number of students (total, male, female)
school.Nc=zeros(Nmax,1,'int32'); % Number of classes

nclass=0;
indexschool=0;
Nmalefemaleclass=zeros(1,2);
Nmalefemaleschool=zeros(1,3);
meanpercclass=zeros(1,2); stdpercclass=zeros(1,2);
meanpercgroupclass=zeros(2,ngroup); stdpercgroupclass=zeros(2,ngroup);
meanmleclass=zeros(1,2); stdmleclass=zeros(1,2);
dummeanpercclass=zeros(2,1); dumstdpercclass=zeros(2,1);
dummeanpercgroupclass=zeros(2,ngroup); dumstdpercgroupclass=zeros(2,ngroup);
dummeanmleclass=zeros(2,1); dumstdmleclass=zeros(2,1);

meanpercschool=zeros(1,3); stdpercschool=zeros(1,3);
meanpercgroupschool=zeros(ngroup,3); stdpercgroupschool=zeros(ngroup,3);
meanmleschool=zeros(1,3); stdmleschool=zeros(1,3);

percstud=zeros(10000,2);
mlestud=zeros(10000,2);
nstud=zeros(2,1);
oldschool=-1;

for n=1:N
    if percgroup(n,1)>NA & sesso(n)>=1 & sesso(n)<=2,
        sex=sesso(n);
        Nmalefemaleclass(sex)=Nmalefemaleclass(sex)+1;
        meanpercclass(sex)=meanpercclass(sex)+perc(n);
        meanpercgroupclass(sex,:)=meanpercgroupclass(sex,:)+percgroup(n,:);
    end
end

```

```

stdpercclass(sex)=stdpercclass(sex)+perc(n)*perc(n);

stdpercgroupclass(sex,:)=stdpercgroupclass(sex,:)+percgroup(n,:).*percgroup(n,:);
meanmleclass(sex)=meanmleclass(sex)+mle(n);
stdmleclass(sex)=stdmleclass(sex)+mle(n)*mle(n);
nstud(sex)=nstud(sex)+1;
percstud(nstud(sex),sex)=perc(n);
mlestud(nstud(sex),sex)=mle(n);
end % of if percgroup(n,1)>NA,
if student(n+1)<=student(n)
    nclass=nclass+1;
    if verbose>=4,
        fprintf('Class %i - Students: %i School:
%i\n',nclass,student(n),schoolcode(n));
    end
% Fine classe
if Nmalefemaleclass(1)>0,
    dummeanpercclass(1)=meanpercclass(1)/Nmalefemaleclass(1);
    dumstdpercclass(1)=sqrt(stdpercclass(1)/Nmalefemaleclass(1)-
meanpercclass(1)*meanpercclass(1)/Nmalefemaleclass(1)/Nmalefemaleclass(1));
    dummeanpercgroupclass(1,:)=meanpercgroupclass(1,:)/Nmalefemaleclass(1);
    dumstdpercgroupclass(1,:)=sqrt(stdpercgroupclass(1,:)/Nmalefemaleclass(1)-
meanpercgroupclass(1,:).*meanpercgroupclass(1,:)/Nmalefemaleclass(1)/Nmalefemaleclass(
1));
    dummeanmleclass(1)=meanmleclass(1)/Nmalefemaleclass(1);
    dumstdmleclass(1)=sqrt(stdmleclass(1)/Nmalefemaleclass(1)-
meanmleclass(1)*meanmleclass(1)/Nmalefemaleclass(1)/Nmalefemaleclass(1));
    else
        dummeanpercclass(1)=-999;
        dumstdpercclass(1)=-999;
        dummeanpercgroupclass(1,:)=-999;
        dumstdpercgroupclass(1,:)=-999;
        dummeanmleclass(1)=-999;
        dumstdmleclass(1)=-999;
    end % of if Nmalefemaleclass(1)>0,
    if Nmalefemaleclass(2)>0,
        dummeanpercclass(2)=meanpercclass(2)/Nmalefemaleclass(2);
        dumstdpercclass(2)=sqrt(stdpercclass(2)/Nmalefemaleclass(2)-
meanpercclass(2)*meanpercclass(2)/Nmalefemaleclass(2)/Nmalefemaleclass(2));
        dummeanpercgroupclass(2,:)=meanpercgroupclass(2,:)/Nmalefemaleclass(2);
        dumstdpercgroupclass(2,:)=sqrt(stdpercgroupclass(2,:)/Nmalefemaleclass(2)-
meanpercgroupclass(2,:).*meanpercgroupclass(2,:)/Nmalefemaleclass(2)/Nmalefemaleclass(
2));
        dummeanmleclass(2)=meanmleclass(2)/Nmalefemaleclass(2);
        dumstdmleclass(2)=sqrt(stdmleclass(2)/Nmalefemaleclass(2)-
meanmleclass(2)*meanmleclass(2)/Nmalefemaleclass(2)/Nmalefemaleclass(2));
        else
            dummeanpercclass(2)=-999;
            dumstdpercclass(2)=-999;
            dummeanpercgroupclass(2,:)=-999;
            dumstdpercgroupclass(2,:)=-999;
            dummeanmleclass(2)=-999;
            dumstdmleclass(2)=-999;
        end % of if Nmalefemaleclass(2)>0,
% check stdpercclass adeguato
if dumstdpercclass(1) < 0.001*dummeanpercclass(1) & Nmalefemaleclass(1)>1 &
verbose>=2,
    fprintf(' * %i - codestud %i: Femmine mean perc=%d - std=%d\n',...
        n,schoolcode(n),dummeanpercclass(1),dumstdpercclass(1))
end
if dumstdpercclass(2) < 0.001*dummeanpercclass(2) & Nmalefemaleclass(2)>1 &
verbose>=2,
    fprintf(' * %i - codestud %i: Maschi mean perc=%d - std=%d\n',...
        n,schoolcode(n),dummeanpercclass(2),dumstdpercclass(2))
end
if dumstdpercclass(1)+dumstdpercclass(2) ==0,
    fprintf(fidtemp,'%i\n',Nmalefemaleclass(1)+Nmalefemaleclass(2));
end
%
meanpercschool(1:2)=meanpercschool(1:2)+meanpercclass;
stdpercschool(1:2)=stdpercschool(1:2)+stdpercclass;
meanpercgroupschool(:,1:2)=meanpercgroupschool(:,1:2)+meanpercgroupclass';
stdpercgroupschool(:,1:2)=stdpercgroupschool(:,1:2)+stdpercgroupclass';
meanmleschool(1:2)=meanmleschool(1:2)+meanmleclass;
stdmleschool(1:2)=stdmleschool(1:2)+stdmleclass;
Nmalefemaleschool(1:2)=Nmalefemaleschool(1:2)+Nmalefemaleclass;
meanpercclass=zeros(1,2); stdpercclass=zeros(1,2);

```

```

meanpercgroupclass=zeros(2,ngroup); stdpercgroupclass=zeros(2,ngroup);
meanmleclass=zeros(1,2); stdmleclass=zeros(1,2);
Nmalefemaleschool=zeros(1,2);
if class(n+1)<=class(n)
% Fine scuola
    if n<N,
        if schoolcode(n+1)<=oldschool,
            fprintf('Attenzione! Codice scuola successiva (%i) non superiore a
quella attuale (%i)\n',...
                schoolcode(n+1),oldschool);
        end
    end
    if verbose>=4,
        fprintf('End of school\n')
    end
    Nmalefemaleschool(3)=Nmalefemaleschool(1)+Nmalefemaleschool(2);
    if verbose>1, fprintf('%i.',indexschool), end
    if Nmalefemaleschool(3)>0,
        indexschool=indexschool+1;
        school.code(indexschool)=schoolcode(n);
        school.N(indexschool,:)=Nmalefemaleschool; % Number of students (Male,
Female, Total)
        school.Nc(indexschool)=nclass; % Number of classes
        meanpercschool(3)=meanpercschool(1)+meanpercschool(2);
        stdpercschool(3)=stdpercschool(1)+stdpercschool(2);

meanpercgroupschool(:,3)=meanpercgroupschool(:,1)+meanpercgroupschool(:,2);

stdpercgroupschool(:,3)=stdpercgroupschool(:,1)+stdpercgroupschool(:,2);
    meanmleschool(3)=meanmleschool(1)+meanmleschool(2);
    stdmleschool(3)=stdmleschool(1)+stdmleschool(2);
    indd=find(Nmalefemaleschool>0);

school.perc(indexschool,indd)=meanpercschool(indd)./Nmalefemaleschool(indd);

school.percstd(indexschool,indd)=sqrt(stdpercschool(indd)./Nmalefemaleschool(indd)-
meanpercschool(indd).*meanpercschool(indd)./Nmalefemaleschool(indd)./Nmalefemaleschool
(indd));
        for ii=1:2
            if nstud(ii)>0,

school.percmed(indexschool,ii)=median(percstud(1:nstud(ii),ii));
                school.mlemed(indexschool,ii)=median(mlestud(1:nstud(ii),ii));
            end
        end

school.percmed(indexschool,3)=median([percstud(1:nstud(1),1);percstud(1:nstud(2),2)]);

school.mlemed(indexschool,3)=median([mlestud(1:nstud(1),1);mlestud(1:nstud(2),2)]);
        for j=1:ngroup

school.percgroup(indexschool,indd,j)=meanpercgroupschool(j,indd)./Nmalefemaleschool(in
dd);

school.percgroupstd(indexschool,indd,j)=sqrt(stdpercgroupschool(j,indd)./Nmalefemalesc
hool(indd)-
meanpercgroupschool(j,indd).*meanpercgroupschool(j,indd)./Nmalefemaleschool(indd)./Nma
lefemaleschool(indd));
        end

school.mle(indexschool,indd)=meanmleschool(indd)./Nmalefemaleschool(indd);

school.mlestd(indexschool,indd)=sqrt(stdmleschool(indd)./Nmalefemaleschool(indd)-
meanmleschool(indd).*meanmleschool(indd)./Nmalefemaleschool(indd)./Nmalefemaleschool(i
ndd));
        end % of if Nmalefemaleschool(3)>0,
        Nmalefemaleschool=zeros(1,3);
        meanpercschool=zeros(1,3); stdpercschool=zeros(1,3);
        meanpercgroupschool=zeros(ngroup,3); stdpercgroupschool=zeros(ngroup,3);
        meanmleschool=zeros(1,3); stdmleschool=zeros(1,3);
        nclass=0;
        nstud=zeros(2,1);
        oldschool=schoolcode(n);
    end % if class(n+1)<=class(n)
end % if istudent(n+1)<=student(n)
end % of for n=1:N
    
```

```
if verbose>=0, fprintf('Fatto.\n'), end

school.code(indexschool+1:Nmax)=[];
school.lab=lab;
school.quest=sum(answ);
school.questgroup=answ;
school.N(indexschool+1:Nmax,:)=[];
school.Nc(indexschool+1:Nmax)=[];
school.perc(indexschool+1:Nmax,:)=[];
school.percstd(indexschool+1:Nmax,:)=[];
school.percmed(indexschool+1:Nmax,:)=[];
school.percgroup(indexschool+1:Nmax,,:)=[];
school.percgroupstd(indexschool+1:Nmax,,:)=[];
school.mle(indexschool+1:Nmax,:)=[];
school.mlestd(indexschool+1:Nmax,:)=[];
school.mlemed(indexschool+1:Nmax,:)=[];
if verbose>=0, fprintf(' Numero di scuole: %i\n',indexschool), end

clear class ind mle perc percgroup schoolcode sesso student
```

3.4 Codice per la lettura dei dati delle scuole

Come discusso in precedenza la lettura dei dati delle scuole avviene interpellando direttamente il database Database SNV 2004-05.xls disponibile direttamente da INVALSI in formato Excel.

```

parent=fullfile('D:', 'data', 'invalsi', 'Database SNV 2004-05');
file=fullfile(parent, 'Database SNV 2004-05.xls');
if verbose>=0, fprintf('Read xcl datafile %s...', file), end
[anum, atxt, araw]=xlsread(file);
if verbose>=0, fprintf('Done.\n'), end
% Remove heading of the xcl file from txt (NOT from num)
atxt(1, :)=[];
N=size(anum, 1);
if verbose>=1, fprintf('Processing attributo..'), end
dumattributo=atxt(:, 6);
attributo=zeros(N, 1);
for i=1:N
    if length(dumattributo{i})>0,
        dum=dumattributo{i}(1:1);
        dum=lower(dum); % convert lower case
        switch dum
            case 's'
                attributo(i)=1;
            case 'p'
                attributo(i)=2;
            case 'n'
                attributo(i)=3;
            otherwise
                if verbose>=2, fprintf('i: %i - attributo: %s \n', i, attributo(i)),
end
                end
            else
                if verbose>=2, fprintf('i: %i - dumattributo: ''%s''
\n', i, char(dumattributo(i))), end
                end
end
if verbose>=1, fprintf('CAP..'), end
cap=anum(:, 11);
if verbose>=1, fprintf('Latitudine..'), end
dumlatitudine=atxt(:, 13);
latitudine=zeros(N, 1);
for i=1:N
    if length(dumlatitudine{i})>0,
        dum=dumlatitudine{i};
        k1=strfind(dum, 'o'); % decimal code 0176
        k2=strfind(dum, 'i'); % decimal code 039
        % k3=strfind(dum, '.'); % decimal code 046
        k4=strfind(dum, ' '); % decimal code 034
        latitudine(i)=str2num(dum(1:k1(1)-1))+... % gradi
            str2num(dum(k1(1)+1:k2(1)-1))/60+... % primi
            str2num(dum(k2(1)+1:k4(1)-1))/3600; % secondi
    end
end
if verbose>=1, fprintf('Longitudine..'), end
dumlongitudine=atxt(:, 14);
longitudine=zeros(N, 1);
for i=1:N
    if length(dumlongitudine{i})>0,
        dum=dumlongitudine{i};
        k1=strfind(dum, 'o'); % decimal code 0176
        k2=strfind(dum, 'i'); % decimal code 039
        % k3=strfind(dum, '.'); % decimal code 046
        k4=strfind(dum, ' '); % decimal code 034
        longitudine(i)=str2num(dum(1:k1(1)-1))+... % gradi
            str2num(dum(k1(1)+1:k2(1)-1))/60+... % primi
            str2num(dum(k2(1)+1:k4(1)-1))/3600; % secondi
    end
end

```

```
    end
end
if verbose>=1, fprintf('Regione..'), end
regione=zeros(N,1);
regione=anum(:,15);
if verbose>=1, fprintf('Provincia..'), end
provincia=atxt(:,17);
if verbose>=1, fprintf('cod_ele..'), end
codelem=anum(:,25);
if verbose>=1, fprintf('cod_medie..'), end
codmedie=anum(:,26);
if verbose>=1, fprintf('cod_sup..'), end
codsup=anum(:,27);
if verbose>=1, fprintf('Done.\n'), end

clear anum atxt araw dumattributo dumlatitudine dumlongitudine

Dataschool.attributo=attributo;
Dataschool.cap=cap;
Dataschool.latitudine=latitudine;
Dataschool.longitudine=longitudine;
Dataschool.regione=regione;
Dataschool.provincia=provincia;
Dataschool.codelem=codelem;
Dataschool.codmedie=codmedie;
Dataschool.codsup=codsup;

clear attributo cap latitudine longitudine regione provincia codelem codmedie codsup
```

4. Codice per la stima del comportamento frattale del sistema scolastico

4.1 Introduzione

Il codice riportato rappresenta il primo tentativo in assoluto per individuare l'eventuale presenza di un comportamento frattale nei dati relativi alla valutazione degli studenti. Il codice presuppone che sia già stata creata la struttura `School` mediante la script per l'analisi ANOVA, in modo da evitare di ripetere la lettura di tutti i database e l'incrocio degli stessi.

4.2 Codice per la stima del comportamento frattale

L'analisi del comportamento frattale viene eseguita innanzitutto su tutti i dati disponibili nel database, indipendentemente dalla regione, dal tipo (pubblica o privata), dall'ordine (elementare o media). In un secondo passo vengono effettuate analisi dettagliate ripartendo i dati per singola regione e per ordine delle Scuole. In ogni caso l'analisi viene effettuata separatamente per Maschi e Femmine separatamente, oltre che per Maschi e Femmine considerati un unico campione.

```
fattore={'Tipo','Regione','Ordine','Sesso'};

ntipo=2;
legtipo={'Pubblico','Privato'};

nordine=2;
legordine={'Elementare','Media I grado'};

nregione=20;
legregione={'Valle d'Aosta','Piemonte','Liguria','Lombardia','Trentino',...
            'Veneto','Friuli','Emilia','Toscana','Umbria','Marche','Lazio','Abruzzo',...
            'Molise','Campania','Puglia','Basilicata','Calabria','Sicilia','Sardegna','Italia'};

nsesto=2;
legsesto={'Femmine','Maschi'};

load School
N=size(School.perc,1);
perc=School.perc;
percord=zeros(N,3);
percord(:,1)=sort(perc(:,1));
percord(:,2)=sort(perc(:,2));
percord(:,3)=sort(perc(:,3));

nfig=1;
figure(nfig)
tit=['Tutte le scuole (' num2str(N) ')'];
set(gcf,'Name',tit)
set(nfig,'PaperType','A4','PaperUnits','centimeters',...
```

```

        'PaperPosition',[0 0 21 29.7]);
h=loglog(percord,[N:-1:1]);
set(gca,'Xtick',[0:10:100])
set(gca,'XtickLabel',num2str([0:10:100]'))
legend(h,'Femmine','Maschi','Totale','Location','SouthWest');
title(tit)

maxplots=6;
nfig=1;
in=0;
for iordine=1:nordine
    for iregione=1:nregione+1
        in=in+1;
        if iregione==1 | iregione==nregione+1 | in==maxplots+1,
            in=1;
            nfig=nfig+1;
            figure(nfig); clf;
            set(gcf,'Name',['Fractals plot - ' legordine{iordine} ' - '
legregione{iregione}])
            set(nfig,'PaperType','A4','PaperUnits','centimeters',...
                'PaperPosition',[0 0 21 29.7]);
        end
        if iregione <= nregione,
            ind=find(School.ordine==iordine&School.REGIONE==iregione);
            n1=3; n2=2;
            step=[0:10:100];
        else
            ind=find(School.ordine==iordine);
            n1=1; n2=1;
            step=[0:10:100];
        end
        N2=length(ind);
        percord=zeros(N2,3);
        percord(:,1)=sort(perc(ind,1));
        percord(:,2)=sort(perc(ind,2));
        percord(:,3)=sort(perc(ind,3));
        subplot(n1,n2,in)
        tit=[legordine{iordine} ' - ' legregione{iregione} ' (' num2str(N2) ')'];
        h=loglog(percord,[N2:-1:1]);
        set(gca,'Xtick',step)
        set(gca,'XtickLabel',num2str(step))
        legend(h,'Femmine','Maschi','Totale','Location','SouthWest');
        title(tit)
    end
end
end

```