Gestione di processi complessi da Stata

Rosa Gini
rosa.gini@arsanita.toscana.it

Agenzia regionale di sanità
della Toscana

19 novembre 2009

- Stata non è pensato come un software che gestisce un database:

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,
  - non implementa il linguaggio SQL,

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,
  - non implementa il linguaggio SQL,
  - ha delle limitazioni nel gestire dataset molto grossi.

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,
  - non implementa il linguaggio SQL,
  - ha delle limitazioni nel gestire dataset molto grossi.
- Altri software statistici, per esempio SAS, sono in grado di svolgere questo ruolo, almeno in parte, e di integrare quindi gestione, data management e analisi in un'unica piattaforma.

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,
  - non implementa il linguaggio SQL,
  - ha delle limitazioni nel gestire dataset molto grossi.
- Altri software statistici, per esempio SAS, sono in grado di svolgere questo ruolo, almeno in parte, e di integrare quindi gestione, data management e analisi in un'unica piattaforma.
- Tuttavia, benché non sia nato per questo, anche Stata può integrare queste attività in un processo complesso.

- Stata non è pensato come un software che gestisce un database:
  - non conserva in memoria più dataset contemporaneamente,
  - non implementa il linguaggio SQL,
  - ha delle limitazioni nel gestire dataset molto grossi.
- Altri software statistici, per esempio SAS, sono in grado di svolgere questo ruolo, almeno in parte, e di integrare quindi gestione, data management e analisi in un'unica piattaforma.
- Tuttavia, benché non sia nato per questo, anche Stata può integrare queste attività in un processo complesso.
- In questa presentazione accenniamo come.

- La peculiarità di questa modalità di lavoro consiste nel concentrare la struttura dell'attività in un solo file do, chiamato `main.do`

- La peculiarità di questa modalità di lavoro consiste nel concentrare la struttura dell'attività in un solo file do, chiamato `main.do`
- Questo file

- La peculiarità di questa modalità di lavoro consiste nel concentrare la struttura dell'attività in un solo file do, chiamato `main.do`
- Questo file
  - definisce alcuni parametri fondamentali;

- La peculiarità di questa modalità di lavoro consiste nel concentrare la struttura dell'attività in un solo file do, chiamato `main.do`
- Questo file
  - definisce alcuni parametri fondamentali;
  - lancia alcuni file che stabiliscono dei parametri secondari;

- La peculiarità di questa modalità di lavoro consiste nel concentrare la struttura dell'attività in un solo file do, chiamato `main.do`
- Questo file
  - definisce alcuni parametri fondamentali;
  - lancia alcuni file che stabiliscono dei parametri secondari;
  - lancia i file do secondari che svolgono effettivamente i compiti di estrazione, elaborazione e pubblicazione.

Un tipico file main.do

```
set more off
set mem 500M

/*fissa i parametri principali */
global PROJECT "CRONICHE"
global patologie "IMA ictus scompenso"
global anno=2007
global dirpar "parametri/"

/*fissa l'albero delle directory (da un file a parte)*/
qui do ${dirpar}directory.do
/*fissa i parametri secondari (da un file a parte)*/
qui do ${dirpar}parametri.do

/*lancia i file do secondari*/
do ${dirdo}01_estrai.do
do ${dirdo}02_carica.do
do ${dirdo}03_data_management.do
do ${dirdo}04_analizza.do
do ${dirdo}05_genera_report.do
```

- Per poter lavorare in questo modo si utilizza come plancia di comando un editor di testo che

- Per poter lavorare in questo modo si utilizza come plancia di comando un editor di testo che
  - permetta di tenere aperti contemporaneamente tutti i file do interessati, in modo da poterli modificare se serve;

- Per poter lavorare in questo modo si utilizza come plancia di comando un editor di testo che
  - permetta di tenere aperti contemporaneamente tutti i file do interessati, in modo da poterli modificare se serve;
  - permetta di lanciare Stata dal proprio interno;

- Per poter lavorare in questo modo si utilizza come plancia di comando un editor di testo che
  - permetta di tenere aperti contemporaneamente tutti i file do interessati, in modo da poterli modificare se serve;
  - permetta di lanciare Stata dal proprio interno;
  - riconosca la sintassi di Stata (non indispensabile ma utile).

Crimson Editor - [P:\osservatori\epidemiologia\area_geriatrica\rosa_gini\0406_Stata\0910_ItUsMeeting\sug2009ita\annex\main.do]

File Edit Search View Document Project Tools Macros Window Help

main.do | 01_estrai.do | 02_carica.do | 03_data_management.do | 04_analizza.do | 05_genera_report.do | parametri.do

```
1  set more off
2  set mem 500M
3
4  /*fissa i parametri principali */
5  global PROJECT "CRONICHE"
6  global patologie "IMA ictus scompenso"
7  global anno=2007
8  global dirpar "parametri/"
9
10 /*fissa l'albero delle directory (da un file a parte)*/
11 qui do ${dirpar}directory.do
12 /*fissa i parametri secondari (da un file a parte)*/
13 qui do ${dirpar}parametri.do
14
15 /*lancia i file do secondari*/
16 do ${dirdo}01_estrai.do
17 do ${dirdo}02_carica.do
18 do ${dirdo}03_data_management.do
19 do ${dirdo}04_analizza.do
20 do ${dirdo}05_genera_report.do
21
```

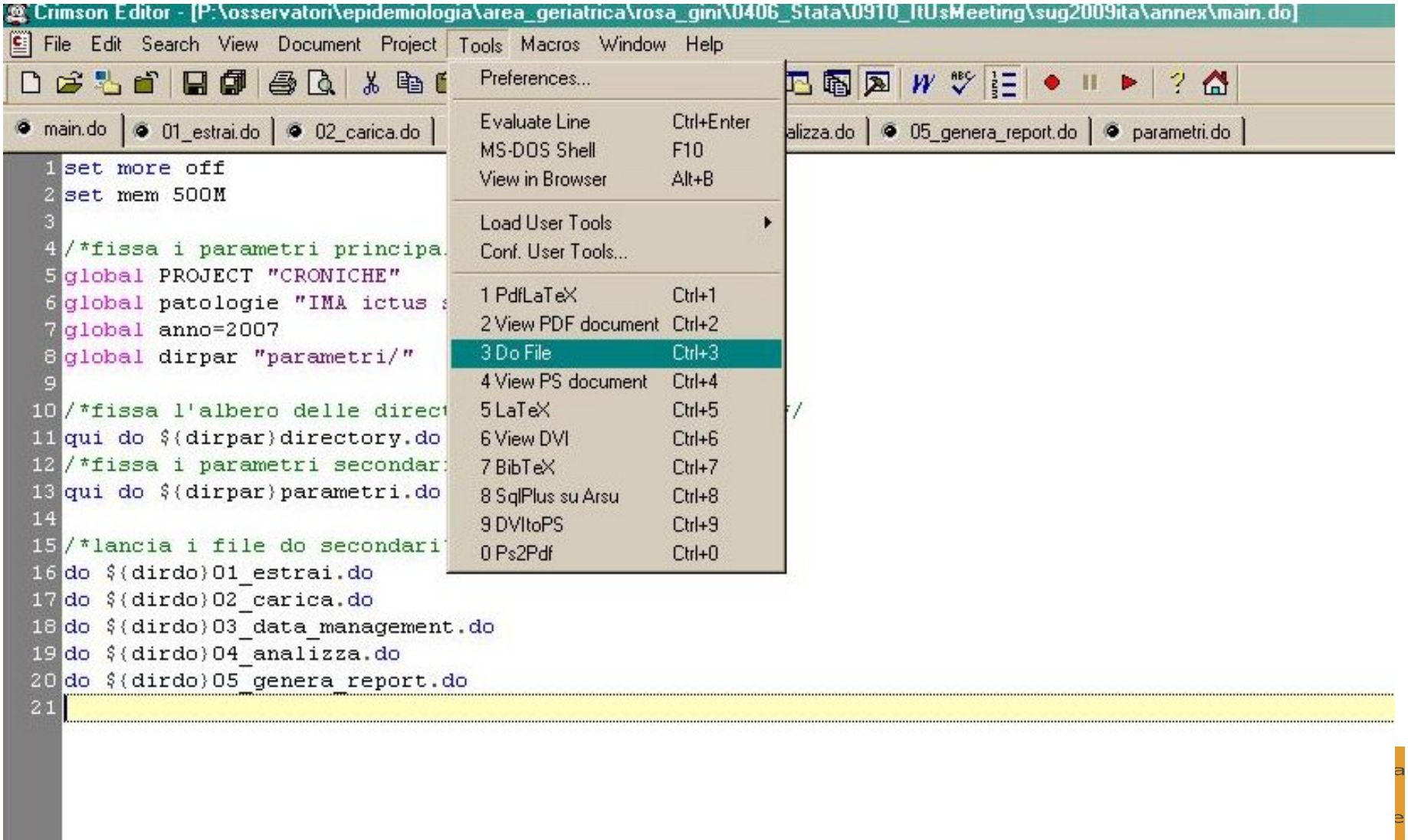
Crimson Editor - [P:\osservatori\epidemiologia\area_geriatrica\rosa_gini\0406_Stata\0910_ItUsMeeting\sug2009ita\annex\parametri\p

File Edit Search View Document Project Tools Macros Window Help

main.do | 01_estrai.do | 02_carica.do | 03_data_management.do | 04_analizza.do | 05_genera_report.do | parametri.do

```
1  /*connessione*/
2  global username "anna.rossi"
3  global psswd "nom3_del_g4ttO"
4  global ODBC "oracle_odbc"
5  global servername "HAL"
6
7  /*parametri patologie*/
8
9  global cod_IMA "410"
10 global cod_ictus "430 431 432 433 434 436"
11 global cod_scompenso "428"
12
13 /*parametri SQL per l'estrazione*/
14 global cond_anni "SDO.ANNO=$anno"
15 global cond_sdo "SDO.REG_RESIDENZA='LAZIO'"
16
17 foreach pat of global patologie{
18 »   foreach cod of global cod_`pat'{
19 »   »   local or=cond("${condspec_sdo_`pat'}"=="","","OR")
20 »       global condspec_sdo_`pat' `"${condspec_SDO_`pat'} `or' SDO.DIADIMI LIKE ``cod'%'"'
21 »   »   }
22 »   }
23 »
24 /*parametri analisi*/
25
26 global fasce_eta "45,65,70,75,80,85,150"
27 global K=1000
28 global fmt "%9,1f"
29
30 /*per pubblicazione*/
31
32 global NomeIMA "Infarto del miocardio"
33 global Nomeictus "Ictus"
34 global Nomescompenso "Insufficienza cardiaca"
35
```

Agenzia Regionale Sanità Toscana


Crimson Editor - [P:\osservatori\epidemiologia\area_geriatrica\rosa_gini\0406_Stata\0910_ItUsMeeting\sug2009ita\annex\main.do]
File Edit Search View Document Project Tools Macros Window Help
Preferences...
Evaluate Line Ctrl+Enter
MS-DOS Shell F10
View in Browser Alt+B
Load User Tools
Conf. User Tools...
1 PdfLaTeX Ctrl+1
2 View PDF document Ctrl+2
3 Do File Ctrl+3
4 View PS document Ctrl+4
5 LaTeX Ctrl+5
6 View DVI Ctrl+6
7 BibTeX Ctrl+7
8 SqlPlus su Arsu Ctrl+8
9 DVItoPS Ctrl+9
0 Ps2Pdf Ctrl+0
main.do
01_estrai.do
02_carica.do
05_genera_report.do
parametri.do
1 set more off
2 set mem 500M
3
4 /*fissa i parametri principa
5 global PROJECT "CRONICHE"
6 global patologie "IMA ictus
7 global anno=2007
8 global dirpar "parametri/"
9
10 /*fissa l'albero delle direct
11 qui do ${dirpar}directory.do
12 /*fissa i parametri secondar
13 qui do ${dirpar}parametri.do
14
15 /*lancia i file do secondari
16 do ${dirdo}01_estrai.do
17 do ${dirdo}02_carica.do
18 do ${dirdo}03_data_management.do
19 do ${dirdo}04_analizza.do
20 do ${dirdo}05_genera_report.do
21

Sono da raccomandare i seguenti due programmi, entrambi gratuiti e Open Source.

Crimson Editor. Un programma molto piccolo ed estremamente flessibile. Non fa altro che l'essenziale. La sintassi di Stata è automaticamente riconosciuta, mentre è necessario programmare un Tool per lanciare Stata su un file aperto: l'operazione è tuttavia semplice e ben documentata. Dal dicembre 2006 è divenuto un progetto Open Source, conosciuto anche come Emerald Editor, ed è stato rilasciato con licenza GPL. Purtroppo gira solo sotto Windows.

Geany. Uno strumento più moderno e completo, quindi un po' più difficile all'inizio. La sintassi di Stata non viene riconosciuta automaticamente ed è necessario programmare un Tool per lanciare Stata su un file aperto. Gira su qualsiasi sistema operativo.

- IL trucco è scrivere alcuni *template* di query del proprio database, ovvero delle query SQL in cui alcuni elementi chiave (il nome della tabella, i criteri di selezione,. . .) sono scritti sotto forma di parametro.

[1] La routine `rewrite`, scaricabile da `scc`, riscrive un file di testo sostituendo alle macro che vi sono contenute i valori che quelle macro hanno al momento in cui la routine è lanciata. La sintassi è la stessa di `copy`, ma va specificato se il file da riscrivere deve rimpiazzare un eventuale file esistente con lo stesso nome (`replace`), oppure accodarvisi (`append`).

- IL trucco è scrivere alcuni *template* di query del proprio database, ovvero delle query SQL in cui alcuni elementi chiave (il nome della tabella, i criteri di selezione,. . .) sono scritti sotto forma di parametro.
- Nel momento in cui serve fare un'estrazione dal database sarà sufficiente definire i paramtri che interessano, riscrivere[1] il template usando qui parametri e lanciare la query tramite una sorgente ODBC o il client del DBMS (ad esempio Oracle)

[1] La routine `rewrite`, scaricabile da `scc`, riscrive un file di testo sostituendo alle macro che vi sono contenute i valori che quelle macro hanno al momento in cui la routine è lanciata. La sintassi è la stessa di `copy`, ma va specificato se il file da riscrivere deve rimpiazzare un eventuale file esistente con lo stesso nome (`replace`), oppure accodarvisi (`append`).

Un template di query

sdo.sql

```
DROP TABLE ${PROJECT}${anno}${nome};
CREATE TABLE ${PROJECT}${anno}${nome} AS
SELECT ID_SDO, ID_PAZIENTE, ANNO, SESSO, ASL_RES, DIADIMI, REG_RESIDENZA, DRG, ETA, GG
FROM FLUSSI.SDO SDO
WHERE  ${cond_sdo} AND (${condspec_sdo}) AND ${anni_sdo};
```

Un file 01_estrai.do

```
1  /*genera il file sql*/
2  foreach pat of global patologie{
3  global nome =upper("`pat'")
4  global condspec_sdo "${condspec_sdo_`pat'}"
5  rewrite sdo.sql using estrai$anno.sql,append
6  }
7  /*lancia su Oracle il file sql generato*/
8  set debug on
9  shell sqlplus $username/${psswd}@$server @estrai$anno.sql
10 set debug off
```

Se i parametri nel file `parametri.do` erano

```
/*parametri patologie*/
global cod_IMA "410"
global cod_ictus "430 431 432 433 434 436"
global cod_scompenso "428"
/*parametri SQL per l'estrazione*/
global cond_anni "SDO.ANNO=$anno"
global cond_sdo "SDO.REG_RESIDENZA='LAZIO'"
foreach pat of global patologie{
foreach cod of global cod_`pat'{
local or=cond("${condspec_sdo_`pat'}"==""," ","OR")
    global condspec_sdo_`pat' `"${condspec_SDO_`pat'} `or' SDO.DIADIMI LIKE ``cod'%'"'
}
}
```

allora il file SQL riscritto sarà come segue

estrai2007.sql

```
-- DROP TABLE CRONICHE2007IMA;
CREATE TABLE CRONICHE2007IMA AS
SELECT ID_SDO, ID_PAZIENTE, ANNO, SESSO, ASL_RES,  DIADIMI, REG_RESIDENZA, DRG, ETA, GG
FROM FLUSSI.SDO SDO
WHERE  SDO.REG_RESIDENZA='LAZIO' AND (  SDO.DIADIMI LIKE '410%') AND SDO.ANNO=2007;
-- DROP TABLE CRONICHE2007ICTUS;
CREATE TABLE CRONICHE2007ICTUS AS
SELECT ID_SDO, ID_PAZIENTE, ANNO, SESSO, ASL_RES,  DIADIMI, REG_RESIDENZA, DRG, ETA, GG
FROM FLUSSI.SDO SDO
WHERE  SDO.REG_RESIDENZA='LAZIO' AND ( OR SDO.DIADIMI LIKE '436%') AND SDO.ANNO=2007;
-- DROP TABLE CRONICHE2007SCOMPENSO;
CREATE TABLE CRONICHE2007SCOMPENSO AS
SELECT ID_SDO, ID_PAZIENTE, ANNO, SESSO, ASL_RES,  DIADIMI, REG_RESIDENZA, DRG, ETA, GG
FROM FLUSSI.SDO SDO
WHERE  SDO.REG_RESIDENZA='LAZIO' AND (  SDO.DIADIMI LIKE '428%') AND SDO.ANNO=2007 ;
```

- Il template era costruito in modo tale che tutte le tabelle generate all'interno del progetto `CRONICHE` avessero questa stringa all'inizio del nome, quindi per caricare le tabelle generate basta caricare tutte le tabelle il cui nome comincia così.

- Il template era costruito in modo tale che tutte le tabelle generate all'interno del progetto `CRONICHE` avessero questa stringa all'inizio del nome, quindi per caricare le tabelle generate basta caricare tutte le tabelle il cui nome comincia così.
- Il file `02_estrai.do` che segue è costruito per Oracle ma può essere adattato ad altri DBMS.

02_carica.do

```
set debug on
qui odbc load,exec("SELECT SEGMENT_NAME,SEGMENT_TYPE FROM USER_SEGMENTS") dsn("$ODBC")
> user($username) p($psswd) dialog(noprompt) clear
keep if SEGMENT_TYPE=="TABLE" & strmatch(SEGMENT_NAME,"${PROJECT}*${anno}*")

list
local quanti_ogg=_N
forvalues s=1/‘quanti_ogg’{
local nometab‘s’=SEGMENT_NAME[‘s’]
}
forvalues s=1/‘quanti_ogg’{
qui odbc load,t(‘"‘nometab‘s’’"’) dsn("$ODBC") user($username) p($psswd)
> dialog(noprompt) clear
if _caller()>=10{
qui ds *, has(format %tc)
local listdate ‘"‘r(varlist)’"’
foreach var of local listdate{
replace sogg‘pat’=dofc(sogg‘pat’)
format sogg‘pat’ %td
}
}
qui do rinomina.do
qui compress
sort iduni
local nomearc=subinstr(‘"‘nometab‘s’’"’,"${PROJECT}","",1)
qui count
di in ye ‘"‘nomearc’ ha ‘r(N)’ righe"’
save ${dirsql}‘nomearc’.dta,replace
}
set debug off
```

Attenzione al file `rinomina.do`!

- Le tabelle caricate prima di essere salvate in formato dta subiscono due trasformazioni

Attenzione al file `rinomina.do`!

- Le tabelle caricate prima di essere salvate in formato dta subiscono due trasformazioni
  - Le variabili che contengono informazioni temporali sono messe in formato giorni (per Stata superiore a versione 9)

Attenzione al file `rinomina.do`!

- Le tabelle caricate prima di essere salvate in formato dta subiscono due trasformazioni
  - Le variabili che contengono informazioni temporali sono messe in formato giorni (per Stata superiore a versione 9)
  - Le variabili vengono rinominate tramite un file `rinomina.do`: questo passaggio è irrilevante se il progetto coinvolge un'istituzione sola, ma è cruciale se invece si vuole mettere in opera l'interoperabilità tra database di istituzioni diverse (vedi più avanti)

- Questi file contengono ordinari comandi di data management e analisi

- Questi file contengono ordinari comandi di data management e analisi
- La pubblicazione automatica (in csv, html, pdf, . . .) è oggetto di grande attenzione (vedi la prima conferenza di stamani)

- Questi file contengono ordinari comandi di data management e analisi
- La pubblicazione automatica (in csv, html, pdf, . . .) è oggetto di grande attenzione (vedi la prima conferenza di stamani)
- L'aspetto specifico di questa presentazione è che le attività di data management, analisi e pubblicazione devono essere programmate, incardinate sui parametri fondamentali del progetto, predisposte per generare output che il ricercatore esaminerà alla fine della procedura invece che durante il suo svolgimento.

- la banca dati delle patologie croniche **MaCro**, che classifica annualmente tutti gli assistibili della Toscana a seconda che risultino o no affetti da una patologia cronica (diabete, insufficienza cardiaca, pregresso ictus, . . .) e verifica che le linee guida dignostico–terapeutiche per la gestione di ciascuna patologia siano seguite;

- la banca dati delle patologie croniche **MaCro**, che classifica annualmente tutti gli assistibili della Toscana a seconda che risultino o no affetti da una patologia cronica (diabete, insufficienza cardiaca, pregresso ictus, . . .) e verifica che le linee guida dignostico–terapeutiche per la gestione di ciascuna patologia siano seguite;
- una procedura esplorativa per la classificazione di ospedalizzazioni e decessi che possono essere dovuti a reazioni avverse da farmaci (all'interno del progetto europeo EU-ADR);

- la banca dati delle patologie croniche **MaCro**, che classifica annualmente tutti gli assistibili della Toscana a seconda che risultino o no affetti da una patologia cronica (diabete, insufficienza cardiaca, pregresso ictus, . . .) e verifica che le linee guida dignostico–terapeutiche per la gestione di ciascuna patologia siano seguite;
- una procedura esplorativa per la classificazione di ospedalizzazioni e decessi che possono essere dovuti a reazioni avverse da farmaci (all'interno del progetto europeo EU-ADR);
- la generazione automatica di report dalle indagini periodiche Multiscopo dell'ISTAT.

- Questa procedura si presta a essere condivisa tra database dai contenuti simili ma che sono mantenuti presso istituzioni diverse.

- Questa procedura si presta a essere condivisa tra database dai contenuti simili ma che sono mantenuti presso istituzioni diverse.
- Infatti è sufficiente che ciascuna istituzioni sviluppi i propri template di query e il proprio file `rinomina.do` e che tutti i nomi di campo così ottenuti siano uguali.

- Questa procedura si presta a essere condivisa tra database dai contenuti simili ma che sono mantenuti presso istituzioni diverse.
- Infatti è sufficiente che ciascuna istituzioni sviluppi i propri template di query e il proprio file `rinomina.do` e che tutti i nomi di campo così ottenuti siano uguali.
- I file `parametri.do`, `01_estrai.do` e `02_carica.do` possono dover subire piccoli adattatamenti legati ai diversi software di gestione database, e il resto della procedura è uguale per tutti.

Esempio di interoperabilità

- I database sanitari (ospedalizzazioni, consumo di farmaci, specialistica ambulatoriale...) hanno un tracciato record unico regolato da normativa nazionale, ma, per ragioni legate alla normativa sulla riservatezza, i dati individuali sono conservati presso ciascuna Regione e che quando vengono trasmessi al Ministero vengono anonimizzati, perdendo la possibilità di essere incrociati tra di loro.

Esempio di interoperabilità

- I database sanitari (ospedalizzazioni, consumo di farmaci, specialistica ambulatoriale...) hanno un tracciato record unico regolato da normativa nazionale, ma, per ragioni legate alla normativa sulla riservatezza, i dati individuali sono conservati presso ciascuna Regione e che quando vengono trasmessi al Ministero vengono anonimizzati, perdendo la possibilità di essere incrociati tra di loro.
- Quindi un'analisi che richieda l'incrocio di dati tra più flussi (ad esempio il calcolo della mortalità dopo infarto, o del controllo dell'emoglobina glicata tra i diabetici) può essere eseguita a livello nazionale solo condividendo la metodologia di calcolo tra le diverse istituzioni.

Esempio di interoperabilità

- La tipologia di procedura illustrata finora consente di implementare l'interoperabilità intraregionale utilizzando Stata, che è un software presente in molte delle istituzioni interessate.

[2] Gini R, Capon A, Roti L, Mastromattei A, Buiatti E. Le fratture di femore tra gli anziani del Lazio e della Toscana: analisi del fenomeno nel periodo 1999-2003. *Epidemiol Prev*. 2007, 31(4) 194-203.

Esempio di interoperabilità

- La tipologia di procedura illustrata finora consente di implementare l'interoperabilità intraregionale utilizzando Stata, che è un software presente in molte delle istituzioni interessate.
- Una sperimentazione è stata eseguita tra Toscana e Lazio,[2] e si è estesa a Veneto e Puglia.

[2] Gini R, Capon A, Roti L, Mastromattei A, Buiatti E. Le fratture di femore tra gli anziani del Lazio e della Toscana: analisi del fenomeno nel periodo 1999-2003. *Epidemiol Prev*. 2007, 31(4) 194-203.

Grazie per l'attenzione!