Matching tra due coorti consecutive estratte da un registro di patologia

Dino Gibertoni

Dipartimento di Scienze Biomediche e Neuromotorie, Università di Bologna

Data Manager del Progetto PIRP, Azienda Ospedaliero-Universitaria S.Orsola-Malpighi, Bologna

XIV CONVEGNO ITALIANO DEGLI UTENTI DI STATA | 16 NOVEMBRE 2017

Modelli prognostici e validazione

- Modelli prognostici relativi a un determinato esito clinico vengono continuamente proposti, solitamente a partire da dati osservazionali raccolti in registri di patologia
- Affinché un modello prognostico possa avere un'applicazione clinica concreta è necessario che la sua utilità venga dimostrata in dati indipendenti da quelli con i quali il modello è stato sviluppato[1]
- Per validazione di un modello si intende "the *process of evaluating the performance of a model"*[1]
- La validazione può essere interna o esterna (temporale o su dataset ottenuti da diversi setting clinici)

[1] Royston and Altman, *External validation of a Cox prognostic model: principles and methods,* BMC Medical Research Methodology 2013, **13:33**

Il database del PIRP

- Il Progetto Insufficienza Renale Progressiva (PIRP) è attivo in Emilia-Romagna dal 2004 e ha l'obiettivo di rallentare il declino della funzionalità renale dei malati di insufficienza renale cronica (IRC), fornendo loro un trattamento tempestivo e specialistico ospedaliero
- Parte fondamentale del progetto è il database, che al 31 agosto 2017 contiene 24359 pazienti e 96478 visite e include dati degli esami di laboratorio, trattamento farmaceutico e dietologico
- Gli esiti morte ed RRT (renal replacement therapy, ovvero dialisi o trapianto) sono ottenuti mediante linkage a registri regionali di mortalità e dialisi, mentre i dati relativi ai ricoveri ospedalieri provengono dal linkage con il registro regionale SDO

Il modello CT-PIRP

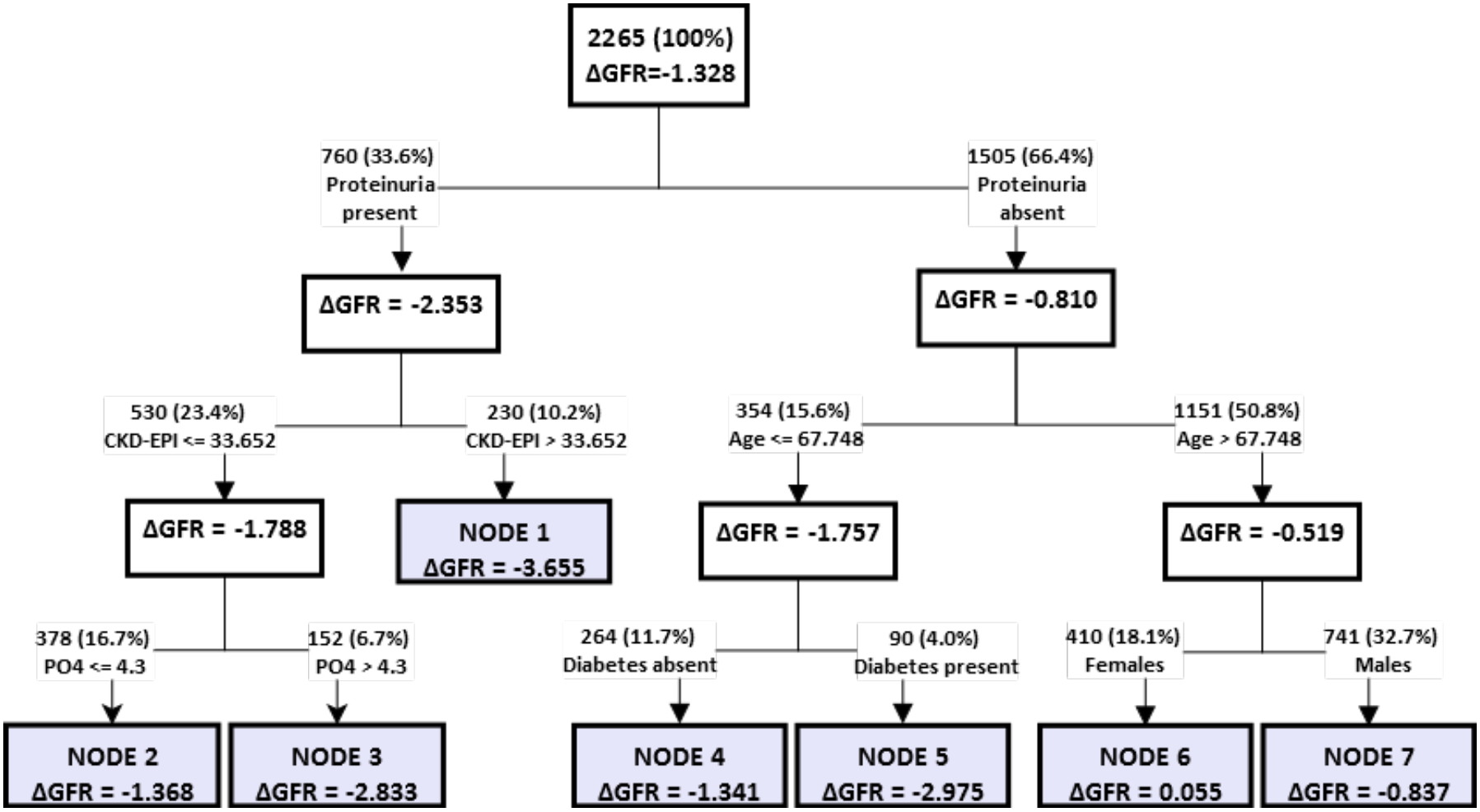
A clinical stratification tool for chronic kidney disease progression rate based on classification tree analysis

Paola Rucci[1], Marcora Mandreoli[2], Dino Gibertoni[1], Alessandro Zuccalà[3], Maria Pia Fantini[1], Jacopo Lenzi[1], Antonio Santoro[2] for the Prevention of Renal Insufficiency Progression (PIRP) Project*

[1]Department of Biomedical and Neuromotor Sciences, Alma Mater Studiorum – University of Bologna, Bologna, Italy, [2]Division of Nephrology, Dialysis and Hypertension, Policlinico S. Orsola-Malpighi, Bologna, Italy and [3]Division of Nephrology and Dialysis, Ospedale S. Maria della Scaletta, Imola, Italy

- Il modello CT-PIRP (Nephrol Dial Transplant (2014) 29: 603–610) classifica i pazienti IRC in sette gruppi omogenei che hanno rischi diversi di progressione della malattia usando sei caratteristiche cliniche e socio-demografiche
- L'esito del modello era la variazione media annua del filtrato glomerulare (GFR)

Il modello CT-PIRP

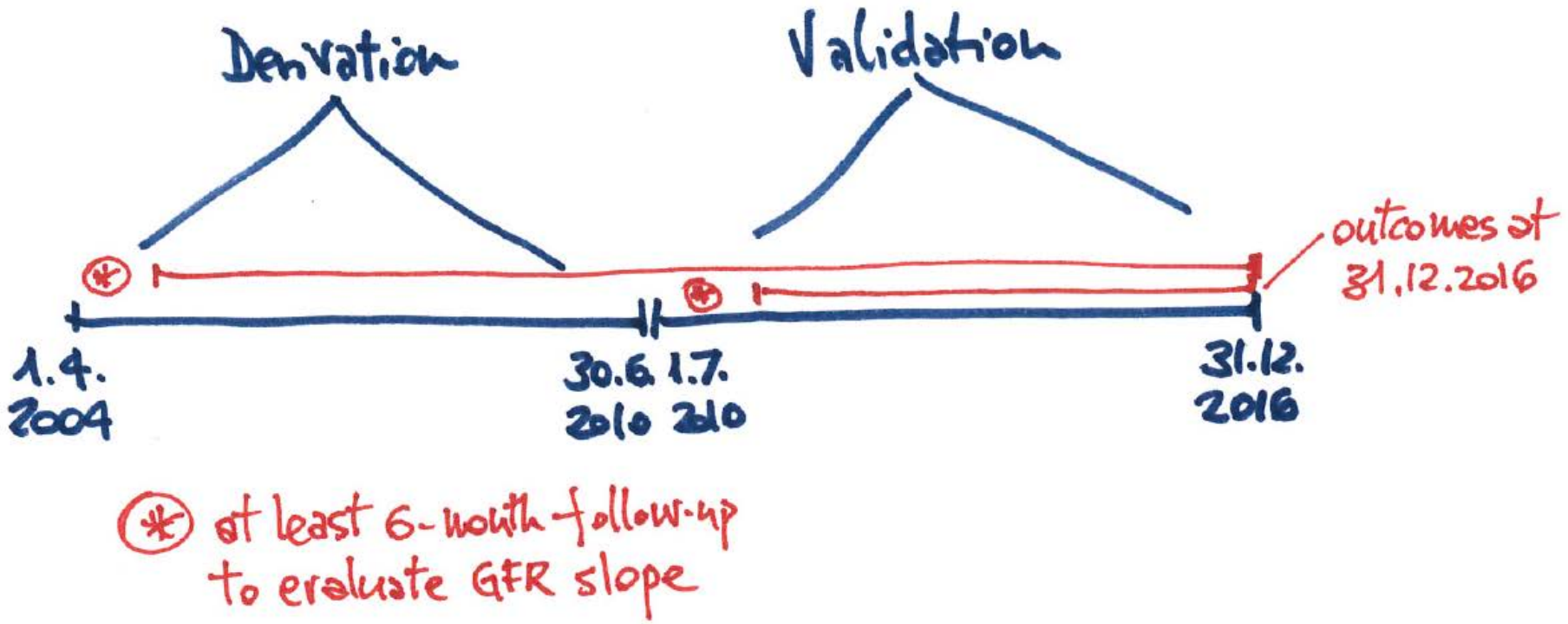


Cosa vogliamo verificare

1. Verificare se l'appartenenza a un nodo è anche predittiva degli esiti della malattia renale, RRT e morte
2. Verificare se in una popolazione indipendente (**coorte di validazione, CV**) da quella su cui si è ottenuto il modello (**coorte di derivazione, CD**):
   - **raggruppando i pazienti nei 7 nodi del modello CT-PIRP si ottengono valori medi annui di variazione del GFR simili a quelli della CD**
   - **gli esiti RRT e morte si presentano con rischi simili**

- La coorte di validazione è stata individuata tra i pazienti entrati in PIRP successivamente a quelli da cui è stata tratta la coorte di derivazione (**validazione temporale**)

Validazione del modello CT-PIRP



Procedimento di matching e validazione

1. Applicazione alla CV dei criteri di eleggibilità dello studio (almeno 4 visite in almeno 6 mesi di follow-up)
2. Attribuzione dei soggetti eleggibili della CV ad un nodo, sulla base delle caratteristiche che descrivono i nodi
3. Matching 1:1 tra le due coorti rispetto a nodo e tempo di osservazione della slope del GFR
4. Eliminazione dei soggetti eccedenti il matching, in entrambe le coorti
5. Confronto caratteristiche dei pazienti delle due coorti e applicazione dei metodi di validazione

Procedimento (1)

```
(…)
tempfile paper
save `paper'                // Derivation cohort

*** ora ottengo la COORTE DI VALIDAZIONE: prendo il main dataset e con
il merge elimino i pz del paper (opzione keep(master))
use "main.dta", clear
merge m:1 id using "`paper'", keep(master) keepusing(id) nogen

*** SELEZIONE PAZIENTI CONFRONTABILI NELLA COORTE DI VALIDAZIONE
drop if data_1vis < 18444       // prima visita > 1.07.2010
drop if data_visita > 20819     // via le visite > 31.12.2016
sort id data_visita
by id: replace progr=_n
by id: replace num_visite=_N
by id: gen data_ultvis=data_visita[_N]
by id: gen obs_time=data_ultvis - data_1vis

drop if num_visite < 4          // almeno 4 visite disponibili
drop if obs_time<=182          // follow-up di almeno 6 mesi
```

Procedimento (2)

Come “rimpolpare” il dataset: sostituzione dei dati mancanti al baseline con quelli eventualmente presenti nelle visite immediatamente successive

```
*** RECUPERO DATI MANCANTI AL BASELINE
*** quando il baseline è missing prendo il valore della prima
visita successiva purché sia entro 1 anno dal baseline
quietly summ progr_sk
local visite=r(max)    // max num. Visits in the dataset
foreach var of varlist PO4 proteinuric {
        forvalues i=`visite'(-1)1 {
        by id: replace `var' = `var'[_n+`i'] if (`var'==. &
`var'[_n+`i']<.) & (data_visita[_n+`i'] - data_visita) <= 365
                }
        }


*** PO4 from 17299 to 20707 nonmissing;
*** proteinuric from 17681 to 20915
```

Procedimento (3)

Calcolo delle slopes individuali di GFR nella CV

```
*** individual slopes estimated by PARMBY (Roger Newson)
by id: gen time=(data_visita - data_1vis)/365.25
quietly parmby "regress CKD_EPI time", by(id) saving(parmby,
        replace) stars(0.05 0.01 0.001) escal(N)

drop time obs_time
reshape wide data_visita data_esame CKD_EPI creatinina proteinuric
        diabete PO4, i(id) j(progr)
merge 1:m id using "parmby.dta", keep(match master)
        keepusing(estimate parmseq) nogen
drop if parmseq==2  /* elimino le righe con la costante; */
```

Procedimento (4)

Assegnazione dei nodi nella CV e unione dei dataset CD e CV

```
gen nodo = cond(prot==1 & CKD_EPI>33.652, 1,
   cond(prot==1 & (CKD_EPI==.|CKD_EPI<=33.652) & (PO4==.|PO4<=4.3),2,
    cond(prot==1 & (CKD_EPI==.|CKD_EPI<=33.652) & PO4>4.3,3,
     cond(prot==0 & age<=67.748 & (diabetes==. | diabetes==0),4,
      cond(prot==0 & age<=67.748 & diabetes==1,5,
       cond(prot==0 & age>67.748 & males==0,6,7))))))

keep id GFR_prog prot CKD_EPI PO4 age diabetes males nodo centro
        data_exit data_1vis data_ultvis esito

append using "`paper'", gen(deriv)
recode deriv (0=1) (1=0), gen(valid)
label define deriv 0 "Valid" 1 "Deriv"
label values deriv deriv
label define valid 0 "Deriv" 1 "Valid"
label values valid valid
```

Procedimento (5)

MATCHING STEP 1 (Deterministico)

```
gen traj_time = round((data_ultvis-data_1vis)/30)
egen groups = group(nodo traj_time)

*** MATCHING 1:1
bysort groups: gen num_gr=_N
bysort groups: egen num_der=total(deriv)
bysort groups: egen num_val=total(valid)
gen matched = 1                          // indicatore 0/1 del matching
replace matched=0 if num_gr==1           // gruppi con un solo soggetto
replace matched=0 if num_der==0          // gruppi senza soggetti della CD
replace matched=0 if num_val==0          // gruppi senza soggetti della CV
```

30 è un 'tuning parameter' del matching: si appaiano soggetti dello stesso nodo con lo stesso numero di **mesi** di follow-up; usare 7 per avere lo stesso numero di settimane, 15 per metà mese, ecc…

	id	nodo	traj_time	valid	groups	num_gr	num_der	num_val	matched
2	P17940	4	7	Deriv	2	1	1	0	0
3	P35272	4	8	Valid	3	1	0	1	0
4	P22316	4	9	Deriv	4	6	3	3	1
5	P17177	4	9	Deriv	4	6	3	3	1
6	P32407	4	9	Valid	4	6	3	3	1
7	P25857	4	9	Valid	4	6	3	3	1
8	P24577	4	9	Deriv	4	6	3	3	1
9	P35797	4	9	Valid	4	6	3	3	1
10	P19102	4	10	Deriv	5	3	1	2	1
11	P34265	4	10	Valid	5	3	1	2	1
12	P25557	4	10	Valid	5	3	1	2	1
13	P25640	4	11	Valid	6	3	0	3	0
14	P36290	4	11	Valid	6	3	0	3	0
15	P36071	4	11	Valid	6	3	0	3	0
16	P36938	4	12	Valid	7	7	1	6	1
17	P35590	4	12	Valid	7	7	1	6	1
18	P35220	4	12	Valid	7	7	1	6	1
19	P26818	4	12	Valid	7	7	1	6	1
20	P27466	4	12	Valid	7	7	1	6	1
21	P17254	4	12	Deriv	7	7	1	6	1
22	P35953	4	12	Valid	7	7	1	6	1
23	P15286	4	13	Deriv	8	7	4	3	1
24	P31553	4	13	Valid	8	7	4	3	1
25	P15677	4	13	Deriv	8	7	4	3	1
26	P25835	4	13	Valid	8	7	4	3	1
27	P32489	4	13	Valid	8	7	4	3	1
28	P18190	4	13	Deriv	8	7	4	3	1
29	P25513	4	13	Deriv	8	7	4	3	1

Procedimento (6)

MATCHING STEP 2 (Probabilistico)

```
sort id          // must be done before set seed for replicability

set seed 19021966

gen shuffle = runiform()       // creates random number between 0-1

bysort groups deriv (shuffle): replace matched=0 if deriv & _n>num_val
        // removes exceeding records in CD

bysort groups valid (shuffle): replace matched=0 if valid & _n>num_der
        // removes exceeding records in CV
```

	id	nodo	traj_time	valid	groups	num_gr	num_der	num_val	matched	shuffle
2	P17940	4	7	Deriv	2	1	1	0	0	.9142439
3	P35272	4	8	Valid	3	1	0	1	0	.6350135
4	P22316	4	9	Deriv	4	6	3	3	1	.0806678
5	P24577	4	9	Deriv	4	6	3	3	1	.4065889
6	P17177	4	9	Deriv	4	6	3	3	1	.5300566
7	P32407	4	9	Valid	4	6	3	3	1	.7198889
8	P25857	4	9	Valid	4	6	3	3	1	.9630874
9	P35797	4	9	Valid	4	6	3	3	1	.9975709
10	P19102	4	10	Deriv	5	3	1	2	1	.0658045
11	P34265	4	10	Valid	5	3	1	2	1	.2858194
12	P25557	4	10	Valid	5	3	1	2	0	.5711487
13	P36071	4	11	Valid	6	3	0	3	0	.0954485
14	P25640	4	11	Valid	6	3	0	3	0	.3694083
15	P36290	4	11	Valid	6	3	0	3	0	.7278211
16	P17254	4	12	Deriv	7	7	1	6	1	.3773385
17	P27466	4	12	Valid	7	7	1	6	1	.0961416
18	P36938	4	12	Valid	7	7	1	6	0	.4179251
19	P35220	4	12	Valid	7	7	1	6	0	.5640218
20	P35590	4	12	Valid	7	7	1	6	0	.7189272
21	P26818	4	12	Valid	7	7	1	6	0	.8455352
22	P35953	4	12	Valid	7	7	1	6	0	.9923445
23	P25513	4	13	Deriv	8	7	4	3	1	.2274183
24	P15677	4	13	Deriv	8	7	4	3	1	.2813778
25	P15286	4	13	Deriv	8	7	4	3	1	.4871168
26	P18190	4	13	Deriv	8	7	4	3	0	.7817312
27	P32489	4	13	Valid	8	7	4	3	1	.5079011
28	P25835	4	13	Valid	8	7	4	3	1	.7816603
29	P31553	4	13	Valid	8	7	4	3	1	.8490198

Verifica del matching

```
. tab deriv nodo if matched, row nokey
```

deriv	nodo 4	7	8	9	10	11	12	Total
Valid	213	344	100	215	73	385	694	2,024
	10.52	17.00	4.94	10.62	3.61	19.02	34.29	100.00
Deriv	213	344	100	215	73	385	694	2,024
	10.52	17.00	4.94	10.62	3.61	19.02	34.29	100.00
Total	426	688	200	430	146	770	1,388	4,048
	10.52	17.00	4.94	10.62	3.61	19.02	34.29	100.00

Verifica del matching

```
. tab matched nodo if deriv, col nokey
```

matched	nodo 4	7	8	9	10	11	12	Total
0	17	34	52	49	17	25	47	241
	7.39	8.99	34.21	18.56	18.89	6.10	6.34	10.64
1	213	344	100	215	73	385	694	2,024
	92.61	91.01	65.79	81.44	81.11	93.90	93.66	89.36
Total	230	378	152	264	90	410	741	2,265
	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Matching 1:2

STEP 1 (Deterministico)

```
bysort groups: gen num_gr=_N
bysort groups: egen num_der=total(deriv)
bysort groups: egen num_val=total(valid)
gen matched=1

replace matched=0 if num_gr<=2          // min group size must be 3
replace matched=0 if num_der==0
replace matched=0 if num_val<=1         // min valid records must be 2
```

Matching 1:2

STEP 2 (Probabilistico)

Per ogni gruppo vanno considerate separatamente queste tre possibilità:

Surplus di record nella CD:

ad es: num_der=4 e num_val=2

num_der > num_val/2

Surplus di record nella CV:

ad es: num_der=1 e num_val=6

num_val > 2*num_der

Surplus di record in entrambe le coorti:

ad es: num_der=3 e num_val=3; num_der=4 e num_val=5

(num_der > num_val/2) & (num_val > 2*max_der)

Matching 1:2

STEP 2 (Probabilistico)

```
sort id
set seed 19021966
gen shuffle = runiform()      // creates random number between 0-1

gen max_der = floor(num_val/2)          // max possible rec CD per group

*** Indicators of surplus
gen srpl_der = (num_der > num_val/2)
gen srpl_val = (num_val > 2*num_der)
gen srpl_both = (num_der > num_val/2) & (num_val > 2*max_der)


bysort groups deriv (shuffle):
       replace matched=0 if srpl_val& (valid & _n > (num_der*2))
bysort groups deriv (shuffle):
       replace matched=0 if srpl_der& (deriv & _n > (num_val/2))
bysort groups deriv (shuffle):
       replace matched=0 if srpl_both & (valid & _n > (max_der*2))
```

	id	nodo	deriv	groups	num_gr	num_der	num_val	max_der	matched	srpl_der	srpl_val	srpl_both	shuffle
2	P17940	1	Deriv	2	1	1	0	0	0	1	0	0	.3188426
3	P35272	1	Valid	3	1	0	1	0	0	0	1	0	.7377269
4	P32407	1	Valid	4	6	3	3	1	1	1	0	1	.2050566
5	P25857	1	Valid	4	6	3	3	1	1	1	0	1	.7438174
6	P35797	1	Valid	4	6	3	3	1	0	1	0	1	.9305531
7	P24577	1	Deriv	4	6	3	3	1	1	1	0	1	.1478051
8	P22316	1	Deriv	4	6	3	3	1	0	1	0	1	.4182868
9	P17177	1	Deriv	4	6	3	3	1	0	1	0	1	.6302797
10	P34265	1	Valid	5	3	1	2	1	1	0	0	0	.1621066
11	P25557	1	Valid	5	3	1	2	1	1	0	0	0	.9450775
12	P19102	1	Deriv	5	3	1	2	1	1	0	0	0	.6616318
13	P25640	1	Valid	6	3	0	3	1	0	0	1	0	.0104858
14	P36290	1	Valid	6	3	0	3	1	0	0	1	0	.2780766
15	P36071	1	Valid	6	3	0	3	1	0	0	1	0	.9815893
16	P26818	1	Valid	7	7	1	6	3	1	0	1	0	.1343792
17	P35590	1	Valid	7	7	1	6	3	1	0	1	0	.1449912
18	P35953	1	Valid	7	7	1	6	3	0	0	1	0	.2562879
19	P27466	1	Valid	7	7	1	6	3	0	0	1	0	.6191685
20	P35220	1	Valid	7	7	1	6	3	0	0	1	0	.7679403
21	P36938	1	Valid	7	7	1	6	3	0	0	1	0	.9983293
22	P17254	1	Deriv	7	7	1	6	3	1	0	1	0	.7343181
23	P25835	1	Valid	8	7	4	3	1	1	1	0	1	.5625523
24	P31553	1	Valid	8	7	4	3	1	1	1	0	1	.9018886
25	P32489	1	Valid	8	7	4	3	1	0	1	0	1	.9462936
26	P15286	1	Deriv	8	7	4	3	1	1	1	0	1	.1943687
27	P18190	1	Deriv	8	7	4	3	1	0	1	0	1	.6712633
28	P15677	1	Deriv	8	7	4	3	1	0	1	0	1	.9412326
29	P25513	1	Deriv	8	7	4	3	1	0	1	0	1	.9649813

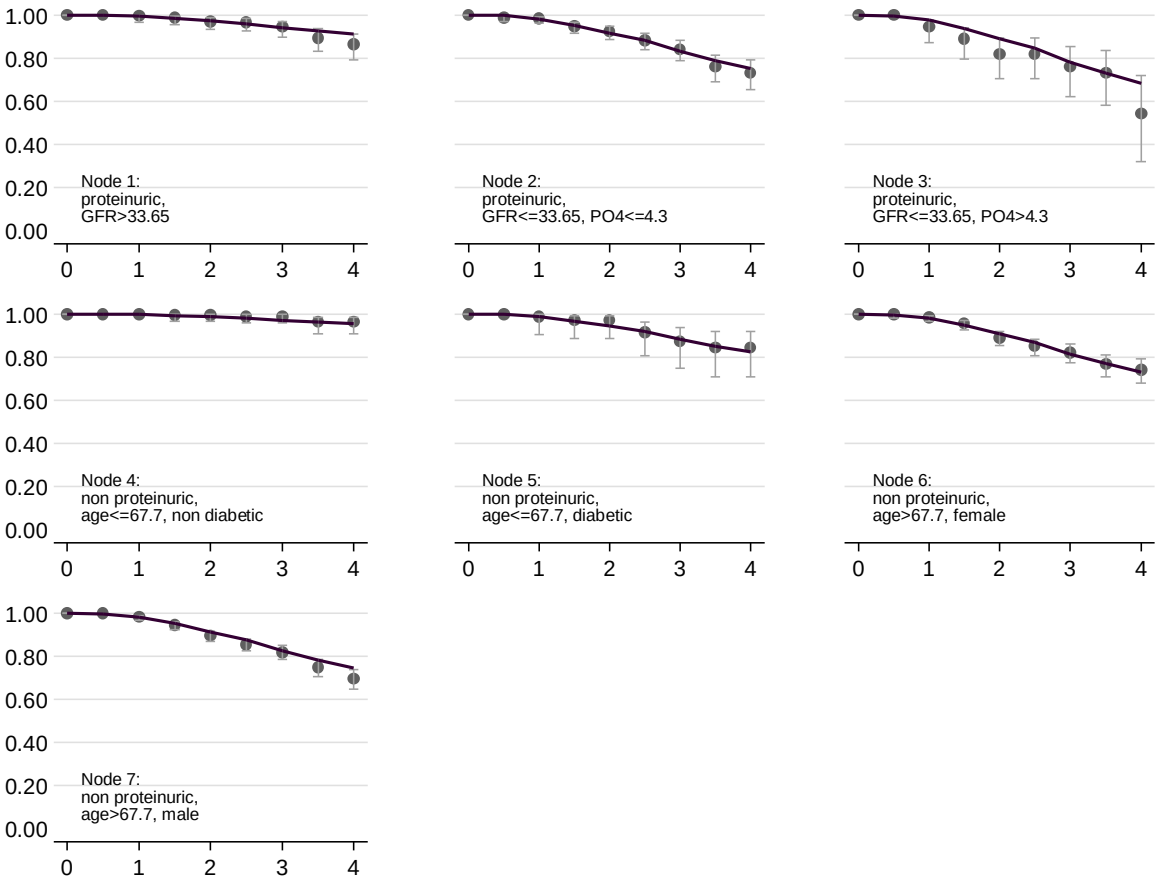
Verifica del matching

```
. tab deriv nodo if matched, row nokey

|                                    nodo
     deriv |         1          2          3          4          5          6          7 |     Total
-----------+-----------------------------------------------------------------------------+----------
     Valid |       306        532        124        286         92        544        940 |     2,824
           |     10.84      18.84       4.39      10.13       3.26      19.26      33.29 |    100.00
-----------+-----------------------------------------------------------------------------+----------
     Deriv |       153        266         62        143         46        272        470 |     1,412
           |     10.84      18.84       4.39      10.13       3.26      19.26      33.29 |    100.00
-----------+-----------------------------------------------------------------------------+----------
     Total |       459        798        186        429        138        816      1,410 |     4,236
           |     10.84      18.84       4.39      10.13       3.26      19.26      33.29 |    100.00
```

Come finisce questa storia?

Calibration dell'outcome morte



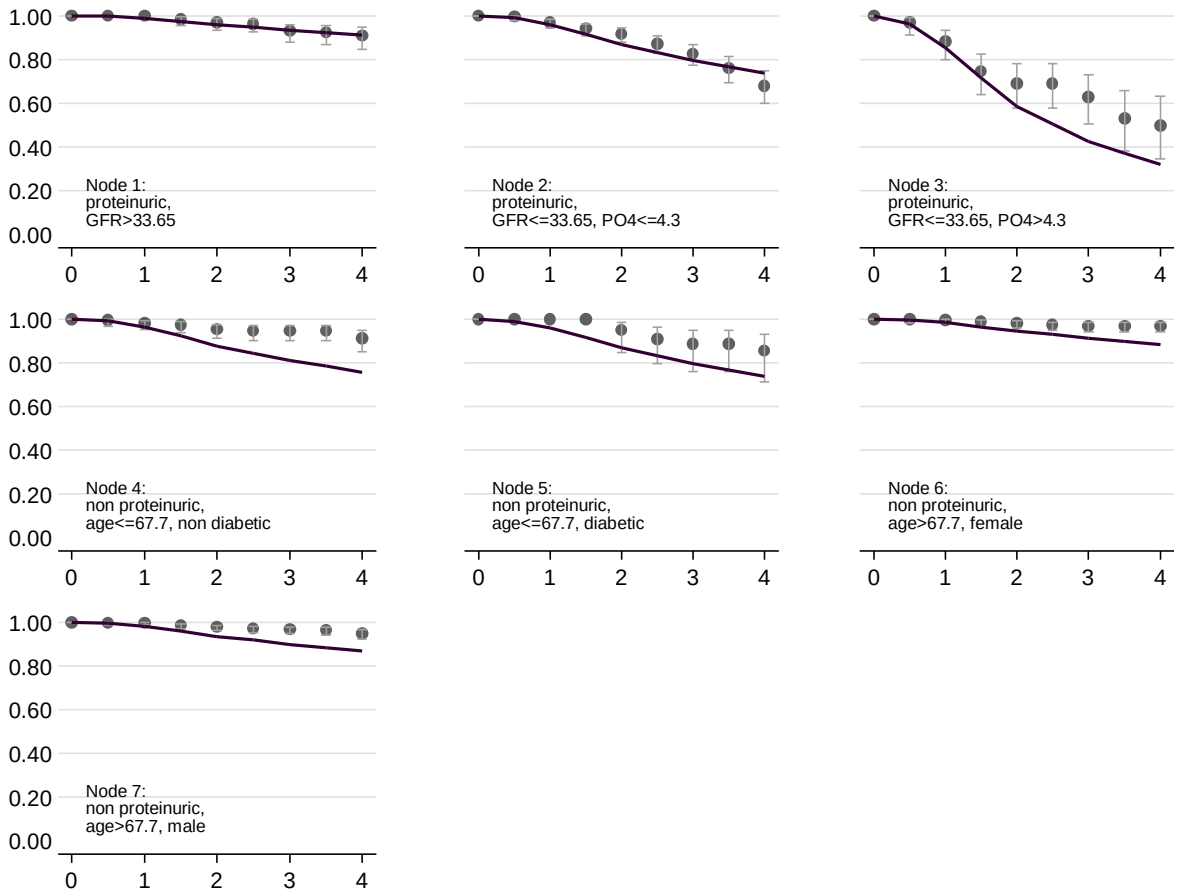
Linea = survival attesa (baseline survival della CD)

Punti con IC: survival osservata nella CV

With a big help from `stcoxgrp` (Royston, Stata Journal, 2015;15:275–91)

Come finisce questa storia?

Calibration dell'outcome RRT



Linea = survival attesa (baseline survival della CD)

Punti con IC: survival osservata nella CV

With a big help from `stcoxgrp` (Royston, Stata Journal, 2015;15:275–91)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Grazie per l’attenzione!

dino.gibertoni2@unibo.it

www.unibo.it