

```
1  /*
2  AutoReg.
3  Procedura per la stima quasi automatica di un modello di regressione.
4  Versione 0.11.7 (28 Luglio 2013)
5  Copyright (C) Daniele Rampoldi
6
7  This program is free software: you can redistribute it and/or modify it under the terms
8  of the GNU General Public License, version 3, as published by the Free Software Foundation.
9  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
10 without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
11 See the GNU General Public License for more details.
12 You should have received a copy of the GNU General Public License version 3 along with this
13 program.
14 If not, see http://www.gnu.org/licenses/ .
15
16 Daniele Rampoldi
17 www.zxdr.it
18 rampoldid@zxdr.it
19 rampoldid@yahoo.it
20
21 La via del saggio è agire, ma non competere (Lao Tzu)
22 */
23
24
25 %macro simpson_c(vara, varb, utia, utib, prefisso, taglio);
26
27 proc sql;
28   create table &prefisso._bb11 as
29     select &vara, &varb, count(*) as num11
30     from &prefisso._pt
31     group by &vara, &varb;
32
33   create table &prefisso._bb10 as
34     select &vara, count(*) as num10
35     from &prefisso._pt
36     group by &vara;
37
38   create table &prefisso._bb01 as
39     select &varb, count(*) as num01
40     from &prefisso._pt
41     group by &varb;
42 quit;
43
44 proc sql;
45   create table &prefisso._cc2 as
46     select a.*, b.num10
47     from &prefisso._bb11 a left join &prefisso._bb10 b on
48       (a.&vara = b.&vara);
49
50   create table &prefisso._cc as
51     select a.*, b.num01
52     from &prefisso._cc2 a left join &prefisso._bb01 b on
53       (a.&varb = b.&varb);
54 quit;
55
56 data &prefisso._dd;
57 set &prefisso._cc;
58   format lambda10j 10.5;
59   format lambda01j 10.5;
60   if num10 = 1 and num11 = 1 then lambda10j = 1;
61   else lambda10j = (num11 * (num11 - 1)) / (num10 * (num10 - 1));
62   if num01 = 1 and num11 = 1 then lambda01j = 1;
63   else lambda01j = (num11 * (num11 - 1)) / (num01 * (num01 - 1));
64 run;
65
66 proc delete data=&prefisso._bb01 &prefisso._bb10 &prefisso._bb11
67   &prefisso._cc &prefisso._cc2;
68 run;
69
70 proc sql;
71   create table &prefisso._dd10 as
```

```
72     select &vara, num10, sum(lambda10j) as lambda10
73     from &prefisso._dd
74     group by &vara, num10;
75 quit;
76
77 proc sql;
78     create table &prefisso._dd01 as
79     select &varb, num01, sum(lambda01j) as lambda01
80     from &prefisso._dd
81     group by &varb, num01;
82 quit;
83
84 data &prefisso._dd10b;
85 set &prefisso._dd10;
86     nlambda10 = num10 * lambda10;
87 run;
88
89 data &prefisso._dd01b;
90 set &prefisso._dd01;
91     nlambda01 = num01 * lambda01;
92 run;
93
94 proc sql;
95     create table &prefisso._dd10c as
96     select 1 as incrocio, sum(nlambda10) / sum(num10) as lm10
97     from &prefisso._dd10b;
98
99     create table &prefisso._dd01c as
100    select 1 as incrocio, sum(nlambda01) / sum(num01) as lm01
101    from &prefisso._dd01b;
102 quit;
103
104 proc sql;
105     create table &prefisso._ee as
106     select lm10, lm01
107     from &prefisso._dd10c a inner join &prefisso._dd01c b on
108         (a.incrocio = b.incrocio);
109 quit;
110
111 data &prefisso._rit;
112 set &prefisso._ee;
113     simpson_plus = (lm10 + lm01) / 2;
114     if simpson_plus >= &taglio then val = simpson_plus;
115     else val = 0;
116 run;
117
118 proc delete data=&prefisso._dd &prefisso._dd01 &prefisso._dd01b &prefisso._dd01c
119         &prefisso._dd10 &prefisso._dd10b &prefisso._dd10c &prefisso._ee;
120 run;
121
122 %mend simpson_c;
123
124
125
126
127
128
129
130
131
132
133 %macro mod_b_meno_a(moda, modb, alpha, prefisso);
134
135 data &moda._dev;
136 set &moda;
137     if _n_ = 1;
138 run;
139
140 data &modb._dev;
141 set &modb;
142     if _n_ = 1;
143 run;
```

```
144
145 proc sql noprint;
146 create table &prefisso._c1 as
147   select criterion, df, value
148   from &moda._dev;
149
150 create table &prefisso._c2 as
151   select criterion, df, value
152   from &modb._dev;
153
154 create table &prefisso._c3 as
155   select a.df as dfa, a.value as vala,
156         b.df as dfb, b.value as valb
157   from &prefisso._c1 a, &prefisso._c2 b
158   where a.criterion = b.criterion;
159 quit;
160
161 data &prefisso._c3;
162 set &prefisso._c3;
163 ddev = valb - vala;
164 ddf = dfb - dfa;
165 if ddev > 0 and ddf > 0 then do;
166   p = cdf('CHISQUARE', ddev, ddf);
167   end;
168 else do;
169   p = .;
170   end;
171 p_inv = sum(1, -p);
172 run;
173
174 proc sql noprint;
175   select p_inv format=40.20 into :p
176   from &prefisso._c3;
177 quit;
178
179 %let p = &p;
180
181 proc delete data=&prefisso._c1 &prefisso._c2 &prefisso._c3;
182 run;
183
184 data &prefisso._rit;
185 format val 1.;
186 %if &p < &alpha %then %do;
187   val = 1;
188   %end;
189 %else %do;
190   val = 2;
191   %end;
192 run;
193
194 %mend mod_b_meno_a;
195
196
197
198
199
200
201
202
203
204
205 %macro classizz(in=a, taglio_correlazione=0.4, distribuzione=binomial, alfa=0.05, passo=10,
206               log_0=log, output_log=log, max_giri=0, max_format=$32767, passi=NIENTE,
                simpson=1);
207
208 %let versione_dr = 00 - v0.11.6 - 13.12.2010 - zxdr;
209
210 %put &versione_dr;
211
212 %put +++++ Parametri di input +++++;
213 %put 01 +&in+ (in);
214 %put 02 +&taglio_correlazione+ (taglio_correlazione);
```

```
215 %put 03 +&distribuzione+ (distribuzione);
216 %put 04 +&alfa+ (alfa);
217 %put 05 +&passo+ (passo);
218 %put 06 +&log_0+ (log_0);
219 %put 07 +&output_log+ (output_log);
220 %put 08 +&max_giri+ (max_giri);
221 %put 09 +&max_format+ (max_format);
222 %put 10 +&passi+ (passi);
223 %put 11 +&simpson+ (simpson);
224
225 proc printto log=&output_log; run;
226
227 %put &versione_dr;
228
229 %put +++++ Parametri di input +++++;
230 %put 01 +&in+ (in);
231 %put 02 +&taglio_correlazione+ (taglio_correlazione);
232 %put 03 +&distribuzione+ (distribuzione);
233 %put 04 +&alfa+ (alfa);
234 %put 05 +&passo+ (passo);
235 %put 06 +&log_0+ (log_0);
236 %put 07 +&output_log+ (output_log);
237 %put 08 +&max_giri+ (max_giri);
238 %put 09 +&max_format+ (max_format);
239 %put 10 +&passi+ (passi);
240 %put 11 +&simpson+ (simpson);
241
242 data &in._tesec;
243   inizio = datetime();
244   ahora = inizio;
245 run;
246
247 data &in._tesec;
248 set &in._tesec;
249   passoprec = ahora;
250   ahora = datetime();
251   delta = ahora - inizio;
252   deltagg = int(delta/(60*60*24));
253   deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
254   deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
255   deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
256   deltadec = deltasec + (delta - int(delta));
257   delta2 = ahora - passoprec;
258   deltagg2 = int(delta2/(60*60*24));
259   deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
260   deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
261   deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
262   deltadec2 = deltasec2 + (delta2 - int(delta2));
263 run;
264
265 proc sql noprint;
266   select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
267   from &in._tesec;
268
269   select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
270   from &in._tesec;
271 quit;
272
273 %let dg = &dg; %let dg2 = &dg2;
274 %let do = &do; %let do2 = &do2;
275 %let dm = &dm; %let dm2 = &dm2;
276 %let ds = &ds; %let ds2 = &ds2;
277 %let dd = &dd; %let dd2 = &dd2;
278
279 proc printto log=&log_0; run;
280
281 %put &versione_dr;
282
283 %put +++++ Parametri di input +++++;
284 %put 01 +&in+ (in);
285 %put 02 +&taglio_correlazione+ (taglio_correlazione);
286 %put 03 +&distribuzione+ (distribuzione);
```

```
287 %put 04 +&alfa+ (alfa);
288 %put 05 +&passo+ (passo);
289 %put 06 +&log_0+ (log_0);
290 %put 07 +&output_log+ (output_log);
291 %put 08 +&max_giri+ (max_giri);
292 %put 09 +&max_format+ (max_format);
293 %put 10 +&passi+ (passi);
294 %put 11 +&simpson+ (simpson);
295
296 %put 01 - Controllo dei parametri di input;
297 %put Tempo trascorso;;
298 %if &dg > 0 %then %do;
299     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
300 %end;
301 %else %if &do > 0 %then %do;
302     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
303 %end;
304 %else %if &dm > 0 %then %do;
305     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
306 %end;
307 %else %do;
308     %put &dd.s (&dd2.s);
309 %end;
310
311 %if %sysfunc(exist(&in)) = 0 %then %do; /* 01 */
312     %put La tabella &in non esiste.;
313     %goto theend;
314 %end; /* 01 */
315 %if %sysfunc(index(&taglio_correlazione, -)) ^= 0 or
316     %sysfunc(index(&alfa, -)) ^= 0 %then %do; /* 02 */
317     %put I valori di probabilita e correlazione (&tagliocorrelazione, &alfa) DEVONO essere
NON NEGATIVI.;
318     %goto theend;
319 %end; /* 02 */
320 %if %sysfunc(index(&taglio_correlazione, -)) > 1 or
321     %sysfunc(index(&alfa, -)) > 1 %then %do; /* 03 */
322     %put I valori di probabilita e correlazione (&tagliocorrelazione, &alfa) DEVONO essere
MINORI di 1.;
323     %goto theend;
324 %end; /* 03 */
325
326
327 proc printto log=&output_log; run;
328
329 data &in._tesec;
330 set &in._tesec;
331 passoprec = ahora;
332 ahora = datetime();
333 delta = ahora - inizio;
334 deltagg = int(delta/(60*60*24));
335 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
336 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
337 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
338 deltadec = deltasec + (delta - int(delta));
339 delta2 = ahora - passoprec;
340 deltagg2 = int(delta2/(60*60*24));
341 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
342 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
343 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
344 deltadec2 = deltasec2 + (delta2 - int(delta2));
345 run;
346
347 proc sql noprint;
348     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
349     from &in._tesec;
350
351     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
352     from &in._tesec;
353 quit;
354
355 %let dg = &dg; %let dg2 = &dg2;
356 %let do = &do; %let do2 = &do2;
```

```
357 %let dm = &dm; %let dm2 = &dm2;
358 %let ds = &ds; %let ds2 = &ds2;
359 %let dd = &dd; %let dd2 = &dd2;
360
361 proc printto log=&log_0; run;
362
363 %put 02 - Caricamento delle variabili da utilizzare;
364 %put Tempo trascorso;;
365 %if &dg > 0 %then %do;
366     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
367 %end;
368 %else %if &do > 0 %then %do;
369     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
370 %end;
371 %else %if &dm > 0 %then %do;
372     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
373 %end;
374 %else %do;
375     %put &dd.s (&dd2.s);
376 %end;
377
378 proc printto log=&output_log; run;
379
380 %let oncamm = 0;
381
382 %if "&passi" ^= "NIENTE" %then %do; /* 03b */
383     %if %sysfunc(exist(&passi)) = 0 %then %do; /* 03b.01 */
384         proc printto log=&log_0; run;
385         %put La tabella &passi non esiste.;
386         proc printto log=&output_log; run;
387         %goto theend;
388     %end; /* 03b.01 */
389
390 %let oncamm = 1;
391 %let pcamm = 0;
392
393 proc sql noprint;
394     select max(passo) into :mpass
395     from &passi;
396 quit;
397
398 %let mpass = &mpass;
399
400 data &in._s2;
401 set &passi;
402 %do i=1 %to &mpass; /* 03b.02 */
403     v&i = scan(modello, &i);
404 %end; /* 03b.02 */
405 word = %do i=1 %to &mpass; /* 03b.03 */
406     (v&i ^= '') +
407     %end; /* 03b.03 */
408     0;
409 run;
410
411 proc sort data=&in._s2;
412     by passo;
413 run;
414
415 proc sql noprint;
416     select max(word) into :mword
417     from &in._s2;
418 quit;
419
420 %let mword = &mword;
421
422 %if &mpass > &mword %then %do; /* 03b.04 */
423     data &in._s2;
424     set &in._s2;
425     drop %do i=%eval(&mword + 1) %to &mpass; /* 03b.04.01 */
426         v&i
427     %end; /* 03b.04.01 */
428     ;
```

```
429     run;
430 %end; /* 03b.04 */
431
432 data &in._s2;
433 set &in._s2;
434 if _N_ = 1 then do; /* 03b.05 */
435     %do i=1 %to &mword; /* 03b.05.01 */
436         retain vo&i;
437         vo&i = v&i;
438     %end; /* 03b.05.01 */
439     retain wordo;
440     wordo = word;
441     camm = v1;
442 end; /* 03b.05 */
443 %do i=1 %to &mword; /* 03b.06 */
444     if word > wordo then do; /* 03b.06.01 */
445         if v&i ^= '' then do; /* 03b.06.01.01 */
446             if v&i ^= vo&i then do; /* 03b.06.01.01.01 */
447                 camm = v&i;
448                 i = &mword + 1;
449                 wordo = word;
450             end; /* 03b.06.01.01.01 */
451         end; /* 03b.06.01.01 */
452     end; /* 03b.06.01 */
453 %end; /* 03b.06 */
454 %do i=1 %to &mword; /* 03b.07 */
455     vo&i = v&i;
456 %end; /* 03b.07 */
457     wordo = word;
458 run;
459
460 proc sql noprint;
461     select count(*) into :lcamm
462     from &in._s2
463     where camm ^= '';
464 quit;
465
466 %let lcamm = &lcamm;
467
468 proc sql noprint;
469     select compress(camm) into :camml - :camm&lcamm
470     from &in._s2
471     where camm ^= '';
472 quit;
473
474 proc delete data=&in._s2;
475 run;
476
477 %end; /* 03b */
478
479 data &in._cond;
480     format variabile $70.;
481     format condizione &max_format..;
482     format classe $100.;
483 set &in._cond;
484     if variabile ^= '';
485     variabile = upcase(compress(variabile));
486     keep variabile condizione classe;
487 run;
488
489 data &in._var ;
490     format name $70.;
491 set &in._var ;
492     if name ^= '';
493     name = upcase(compress(name));
494     utilizzo = upcase(compress(utilizzo));
495     keep name type format formatl formatd utilizzo;
496 run;
497
498 %let ludecla = 0;
499
500 proc sql noprint;
```

```
501 select count(*) into :ludecla
502 from &in._var
503 where utilizzo = 'R';
504 quit;
505
506 proc printto log=&log_0; run;
507
508 %if &ludecla ^= 1 %then %do; /* 04 */
509 %put Non ci possono essere &ludecla variabili risposta.;
510 %goto theend;
511 %end; /* 04 */
512
513 %if &max_giri = 0 %then %do; /* 05 */
514 proc sql noprint;
515 select count(*) into :max_g
516 from &in._var
517 where utilizzo in ('O' 'C' 'Q' 'X' 'K');
518 quit;
519
520 %let max_g = %eval(&max_g*3);
521 %end; /* 05 */
522 %else %do; /* 06 */
523 %let max_g = &max_giri;
524 %end; /* 06 */
525
526 proc printto log=&output_log; run;
527
528
529 data &in._tesec;
530 set &in._tesec;
531 passoprec = ahora;
532 ahora = datetime();
533 delta = ahora - inizio;
534 deltagg = int(delta/(60*60*24));
535 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
536 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
537 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
538 deltadec = deltasec + (delta - int(delta));
539 delta2 = ahora - passoprec;
540 deltagg2 = int(delta2/(60*60*24));
541 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
542 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
543 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
544 deltadec2 = deltasec2 + (delta2 - int(delta2));
545 run;
546
547 proc sql noprint;
548 select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
549 from &in._tesec;
550
551 select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
552 from &in._tesec;
553 quit;
554
555 %let dg = &dg; %let dg2 = &dg2;
556 %let do = &do; %let do2 = &do2;
557 %let dm = &dm; %let dm2 = &dm2;
558 %let ds = &ds; %let ds2 = &ds2;
559 %let dd = &dd; %let dd2 = &dd2;
560
561 proc printto log=&log_0; run;
562
563 %put 03 - Copia dei file di input;
564 %put Tempo trascorso;;
565 %if &dg > 0 %then %do;
566 %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
567 %end;
568 %else %if &do > 0 %then %do;
569 %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
570 %end;
571 %else %if &dm > 0 %then %do;
572 %put &dm.m:&dd.s (&dm2.m:&dd2.s);
```



```
573     %end;
574     %else %do;
575         %put &dd.s (&dd2.s);
576     %end;
577
578 proc printto log=&output_log; run;
579
580 data &in._passi;
581     format passo 10.;
582     format modello &max_format.;
583     if passo > 0;
584 run;
585
586 data &in._kcl;
587     format var_orig $100.;
588     format cl_orig $100.;
589     format cl_nuova 20.;
590     if cl_nuova > 0;
591 run;
592
593 proc sql;
594     select name into :ludecla separated by ' '
595     from &in._var
596     where utilizzo in ('I' 'R' 'O' 'Q' 'X' 'C' 'K');
597 quit;
598
599 data &in._pt;
600 set &in ;
601     keep &ludecla ;
602 run;
603
604 proc sql noprint;
605     select count(*) into :n_vrk
606     from &in._var
607     where compress(utilizzo) = 'K';
608
609     select compress(name) into :vri
610     from &in._var
611     where compress(utilizzo) = 'R';
612 quit;
613
614 %let n_vrk = &n_vrk;
615 %let vri = &vri;
616
617 %if &n_vrk <= 0 %then %do;          /* 06b */
618     %goto finek;
619 %end; /* 06b */
620
621 proc sql noprint;
622     select compress(name), type, format, formatl, formatd into
623         :vrk1 - :vrk&n_vrk, :vrkt1 - :vrkt&n_vrk, :vrkfol - :vrkfo&n_vrk, :vrkfl1
624     - :vrkfl&n_vrk, :vrkfd1 - :vrkfd&n_vrk
625     from &in._var
626     where compress(utilizzo) = 'K';
627 quit;
628
629 %do i=1 %to &n_vrk; /* 06c */
630     data &in._escon;
631     set &in._escon;
632     var1 = upcase(compress(var1));
633     var2 = upcase(compress(var2));
634     run;
635
636     data &in._escon2;
637     format var1 $70.;
638     set &in._escon;
639     if compress(var1) = compress("&&vrk&i") then do; /* 06c.01a */
640         output;
641         var1 = compress('K_' || compress("&&vrk&i"));
642         output;
643     end; /* 06c.01a */
```

```
644     else do;          /* 06c.01a2 */
645         output;
646     end; /* 06c.01a2 */
647 run;
648
649 data &in._escon3;
650     format var2 $70.;
651 set &in._escon2;
652     if compress(var2) = compress("&&vrk&i") then do;          /* 06c.01b */
653         output;
654         var2 = compress('K_' || compress("&&vrk&i"));
655         output;
656     end; /* 06c.01b */
657     else do;          /* 06c.01b2 */
658         output;
659     end; /* 06c.01b2 */
660 run;
661
662 data &in._escon;
663 set &in._escon3;
664 run;
665
666 proc delete data=&in._escon2 &in._escon3;
667 run;
668
669 data &in._vrk;
670 set &in._pt;
671     keep &vri &&vrk&i;
672 run;
673
674 proc sql noprint;
675     create table &in._vrk2 as
676     select &&vrk&i, sum(&vri) / count(*) as incidenza
677     from &in._vrk
678     group by &&vrk&i;
679 quit;
680
681 data &in._conk;
682 set &in._cond;
683     if compress(variabale) = compress("&&vrk&i");
684 run;
685
686 proc sql noprint;
687     select count(*) into :nconk
688     from &in._conk;
689 quit;
690
691 %let nconk = &nconk;
692
693 proc sql noprint;
694     select condizione, classe into :conk1 - :conk&nconk, :clonk1 - :clonk&nconk
695     from &in._conk;
696 quit;
697
698 data &in._vrk2;
699 set &in._vrk2;
700     k_&&vrk&i = 1;
701     %if &nconk > 0 %then %do;          /* 06c.01 */
702         %do j=1 %to &nconk; /* 06c.01.01 */
703             if &&conk&j then do;          /* 06c.01.01.01 */
704                 k_&&vrk&i = -&j;
705             end;          /* 06c.01.01.01 */
706         %end;          /* 06c.01.01 */
707     %end;          /* 06c.01 */
708 run;
709
710 proc sort data=&in._vrk2;
711     by k_&&vrk&i incidenza;
712 run;
713
714 data &in._vrk2;
715 set &in._vrk2;
```

```
716 retain kk 0;
717 retain incid 0;
718 if kk <= 0 then do; /* 06c.02 */
719     kk = k_&&vrk&i;
720     incid = incidenza;
721 end; /* 06c.02 */
722 else if kk > 0 then do; /* 06c.03 */
723     if incidenza = incid then k_&&vrk&i = kk;
724     else do; /* 06c.03.01 */
725         kk = kk + 1;
726         k_&&vrk&i = kk;
727         incid = incidenza;
728     end; /* 06c.03.01 */
729 end; /* 06c.03 */
730 drop kk incid;
731 run;
732
733 proc sql noprint;
734     create table &in._ptk as
735     select a.*, b.k_&&vrk&i
736     from &in._pt a left join &in._vrk2 b on
737         (a.&&vrk&i = b.&&vrk&i);
738 quit;
739
740 data &in._pt;
741 set &in._ptk;
742 run;
743
744 proc delete data=&in._ptk;
745 run;
746
747 proc sql noprint;
748     select count(distinct k_&&vrk&i) into :nkk
749     from &in._vrk2;
750 quit;
751
752 %let nkk2 = %sysevalf(1 + (100/&passo), int);
753
754 %if &nkk2 >= &nkk %then %do; /* 06c.04 */
755     proc sql noprint;
756         insert into &in._var values ("K_&&vrk&i", 1, '', 0, 0, '0');
757     quit;
758 %end; /* 06c.04 */
759 %else %do; /* 06c.05 */
760     proc sql noprint;
761         insert into &in._var values ("K_&&vrk&i", 1, '', 0, 0, 'X');
762     quit;
763 %end; /* 06c.05 */
764
765 %if &nconk > 0 %then %do; /* 06c.06 */
766     %do j=1 %to &nconk; /* 06c.06.01 */
767         proc sql noprint;
768             insert into &in._cond values ("K_&&vrk&i", "K_&&vrk&i = -&j", "-&j");
769         quit;
770     %end; /* 06c.06.01 */
771 %end; /* 06c.06 */
772
773 %let vrkfo&i = &&vrkfo&i;
774 %let vrkfl&i = &&vrkfl&i;
775 %let vrkfd&i = &&vrkfd&i;
776
777 %if &&vrkt&i = 1 %then %do; /* 06c.07 */
778
779     %if "&&vrkfo&i" = "" and "&&vrkfl&i" = "0" and "&&vrkfd&i" = "0" %then %do; /* 06c.07.01 */
780         proc sql noprint;
781             create table &in._kcl0 as
782             select "&&vrk&i" as var_orig, compress(put(&&vrk&i, best.)) as cl_orig,
783                 k_&&vrk&i as cl_nuova
784             from &in._vrk2;
785         quit;
786     %end; /* 06c.07.01 */
787
```

```
788     %else %do;      /* 06c.07.02 */
789     proc sql noprint;
790         create table &in._kcl0 as
791         select "&&vrk&i" as var_orig, compress(put(&&vrk&i, &&vrkfo&i..&&vrkfl&i...&&vrkfd&i )) as
cl_orig,
792         k_&&vrk&i as cl_nuova
793         from &in._vrk2;
794     quit;
795     %end; /* 06c.07.02 */
796
797 %end; /* 06c.07 */
798
799 %else %if &&vrkt&i = 2 %then %do;      /* 06c.08 */
800     proc sql noprint;
801         create table &in._kcl0 as
802         select "&&vrk&i" as var_orig, strip(put(&&vrk&i, $100.)) as cl_orig,
803         k_&&vrk&i as cl_nuova
804         from &in._vrk2;
805     quit;
806     %end; /* 06c.08 */
807
808     data &in._kcl;
809     set &in._kcl &in._kcl0;
810     run;
811
812     proc delete data=&in._vrk &in._vrk2 &in._conk &in._kcl0;
813     run;
814
815 %end; /* 06c */
816
817 %finek:
818
819 data &in._tesec;
820 set &in._tesec;
821 passoprec = ahora;
822 ahora = datetime();
823 delta = ahora - inizio;
824 deltagg = int(delta/(60*60*24));
825 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
826 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
827 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
828 deltadec = deltasec + (delta - int(delta));
829 delta2 = ahora - passoprec;
830 deltagg2 = int(delta2/(60*60*24));
831 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
832 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
833 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
834 deltadec2 = deltasec2 + (delta2 - int(delta2));
835 run;
836
837 proc sql noprint;
838     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
839     from &in._tesec;
840
841     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
842     from &in._tesec;
843 quit;
844
845 %let dg = &dg; %let dg2 = &dg2;
846 %let do = &do; %let do2 = &do2;
847 %let dm = &dm; %let dm2 = &dm2;
848 %let ds = &ds; %let ds2 = &ds2;
849 %let dd = &dd; %let dd2 = &dd2;
850
851
852 proc printto log=&log_0; run;
853
854 %put 04a - Classificazione delle variabili 0;
855 %put Tempo trascorso;;
856 %if &dg > 0 %then %do;
857     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
858 %end;
```

```
859 %else %if &do > 0 %then %do;
860 %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
861 %end;
862 %else %if &dm > 0 %then %do;
863 %put &dm.m:&dd.s (&dm2.m:&dd2.s);
864 %end;
865 %else %do;
866 %put &dd.s (&dd2.s);
867 %end;
868
869
870 proc printto log=&output_log; run;
871
872 %let ludecla = 0;
873
874 proc sql noprint;
875 select max(length(classe)) into :ludecla
876 from &in._cond;
877 quit;
878
879 %if &ludecla = . %then %do;
880 %let ludecla = 0;
881 %end;
882
883 %let ludecla = %eval(&ludecla+2);
884
885 %if %eval(&ludecla < 50) %then %do; /* 07 */
886 %let ludecla = 50;
887 %end; /* 07 */
888
889
890
891 data &in._var_b;
892 set &in._var;
893 if utilizzo = '0';
894 run;
895
896 %let o_numvarcl = 0;
897
898 data _NULL_;
899 set &in._var_b end=fine;
900 call symputx('o_classizz'!!strip(_n_),name);
901 call symputx('o_varcl'!!strip(_n_),'CL_'!!strip(name));
902 if fine then call symputx('o_numvarcl',(_n_));
903 run;
904
905 proc delete data=&in._var_b ;
906 run;
907
908 %if %symexist(o_numvarcl) = 0 or &o_numvarcl = 0 %then %goto o_noclassizz;
909
910 %let num_condiz = 0;
911
912 data _NULL_;
913 set &in._cond end=fine;
914 call symputx('condiz'!!strip(_n_), condizione);
915 call symputx('v_condiz'!!strip(_n_), compress(variabile));
916 call symputx('cl_condiz'!!strip(_n_), strip(classe));
917 if fine then call symputx('num_condiz',(_n_));
918 run;
919
920
921 data &in._pt;
922 set &in._pt;
923 %do i=1 %to &o_numvarcl; /* 07b */
924 &&o_varcl&i.._z = 0;
925 format &&o_varcl&i $&ludecla.;
926 format &&o_varcl&i.._b 10.;
927 &&o_varcl&i = compress(put(&&o_classizz&i, best.));
928 &&o_varcl&i.._b = &&o_classizz&i;
929 if &&o_varcl&i.._b <= 0 then &&o_varcl&i.._z = 1;
930 %if &num_condiz > 0 %then %do; /* 07b.01 */
```

```
931     %do j=1 %to &num_condiz; /* 07b.01.01 */
932     %if %upcase(&&v_condiz&j) = %upcase(&&o_classizz&i) %then %do; /* 07b.01.01.01 */
933         if &&condiz&j then do; /* 07b.01.01.01.01 */
934             &&o_varcl&i = "&&cl_condiz&j";
935             &&o_varcl&i.._b = -&j;
936             &&o_varcl&i.._z = 0;
937         end; /* 07b.01.01.01.01 */
938     %end; /* 07b.01.01.01 */
939 %end; /* 07b.01.01 */
940 %end; /* 07b.01 */
941 %end; /* 07b */
942 run;
943
944
945 %do i=1 %to &o_numvarcl; /* 07c */
946     %let allarme = 0;
947
948     proc sql noprint;
949         select max(&&o_varcl&i.._z) into :allarme
950         from &in._pt;
951     quit;
952
953     %if &allarme = 1 %then %do; /* 07c.01 */
954         data &in._ptr;
955         set &in._pt;
956         if &&o_varcl&i.._z = 1 or &&o_varcl&i.._b > 0;
957         keep &&o_classizz&i;
958     run;
959
960     proc sort data=&in._ptr nodup;
961         by &&o_classizz&i;
962     run;
963
964     data &in._ptr;
965     set &in._ptr;
966     retain cla_new 1;
967     output;
968     cla_new = cla_new + 1;
969     run;
970
971     proc sql;
972     create table &in._ptr2 as
973     select a.*, b.cla_new
974     from &in._pt a left join &in._ptr b on
975     (a.&&o_classizz&i = b.&&o_classizz&i);
976     quit;
977
978     data &in._pt;
979     set &in._ptr2;
980     if cla_new ^= . then &&o_varcl&i.._b = cla_new;
981     drop cla_new;
982     run;
983
984     proc delete data=&in._ptr &in._ptr2;
985     run;
986 %end; /* 07c.01 */
987
988     data &in._pt;
989     set &in._pt;
990     drop &&o_varcl&i.._z;
991     run;
992 %end; /* 07c */
993
994 %o_noclassizz:
995
996 data &in._tesec;
997 set &in._tesec;
998 passoprec = ahora;
999 ahora = datetime();
1000 delta = ahora - inizio;
1001 deltagg = int(delta/(60*60*24));
1002 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
```

```
1003   deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
1004   deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
1005   deltadec = deltasec + (delta - int(delta));
1006   delta2 = ahora - passoprec;
1007   deltagg2 = int(delta2/(60*60*24));
1008   deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
1009   deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
1010   deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
1011   deltadec2 = deltasec2 + (delta2 - int(delta2));
1012   run;
1013
1014   proc sql noprint;
1015     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
1016     from &in._tesec;
1017
1018     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
1019     from &in._tesec;
1020   quit;
1021
1022   %let dg = &dg;   %let dg2 = &dg2;
1023   %let do = &do;   %let do2 = &do2;
1024   %let dm = &dm;   %let dm2 = &dm2;
1025   %let ds = &ds;   %let ds2 = &ds2;
1026   %let dd = &dd;   %let dd2 = &dd2;
1027
1028
1029   proc printto log=&log_0; run;
1030
1031   %put 04b - Classificazione delle variabili X;
1032   %put Tempo trascorso;;
1033   %if &dg > 0 %then %do;
1034     %put &dg GG  &do.h:&dm.m:&dd.s (&dg2 GG  &do2.h:&dm2.m:&dd2.s);
1035   %end;
1036   %else %if &do > 0 %then %do;
1037     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
1038   %end;
1039   %else %if &dm > 0 %then %do;
1040     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
1041   %end;
1042   %else %do;
1043     %put &dd.s (&dd2.s);
1044   %end;
1045
1046
1047   proc printto log=&output_log; run;
1048
1049   data &in._var_b;
1050   set &in._var;
1051   if utilizzo = 'X';
1052   run;
1053
1054   %let numvarcl = 0;
1055
1056   data _NULL_;
1057   set &in._var_b end=fine;
1058   call symputx('classizz'!!strip(_n_),name);
1059   call symputx('varcl'!!strip(_n_),'CL_'!!strip(name));
1060   call symputx('formato'!!strip(_n_),format);
1061   call symputx('lform'!!strip(_n_),formatl);
1062   call symputx('dform'!!strip(_n_),formatd);
1063   if fine then call symputx('numvarcl',(_n_));
1064   run;
1065
1066   proc delete data=&in._var_b ;
1067   run;
1068
1069   %if %symexist(numvarcl) = 0 or &numvarcl = 0 %then %goto noclassizz;
1070
1071   %let num_condiz = 0;
1072
1073   data _NULL_;
1074   set &in._cond end=fine;
```

```
1075 call symputx('condiz'!!strip(_n_), condizione);
1076 call symputx('v_condiz'!!strip(_n_), compress(variabile));
1077 call symputx('cl_condiz'!!strip(_n_), strip(classe));
1078 if fine then call symputx('num_condiz',(_n_));
1079 run;
1080
1081
1082 %do i=1 %to &numvarcl ; /* 08 */
1083
1084 data &in._tesec;
1085 set &in._tesec;
1086 passoprec = ahora;
1087 ahora = datetime();
1088 delta = ahora - inizio;
1089 deltagg = int(delta/(60*60*24));
1090 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
1091 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
1092 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
1093 deltadec = deltasec + (delta - int(delta));
1094 delta2 = ahora - passoprec;
1095 deltagg2 = int(delta2/(60*60*24));
1096 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
1097 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
1098 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
1099 deltadec2 = deltasec2 + (delta2 - int(delta2));
1100 run;
1101
1102 proc sql noprint;
1103 select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
1104 from &in._tesec;
1105
1106 select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
1107 from &in._tesec;
1108 quit;
1109
1110 %let dg = &dg; %let dg2 = &dg2;
1111 %let do = &do; %let do2 = &do2;
1112 %let dm = &dm; %let dm2 = &dm2;
1113 %let ds = &ds; %let ds2 = &ds2;
1114 %let dd = &dd; %let dd2 = &dd2;
1115
1116 proc printto log=&log_0; run;
1117
1118
1119 %put 05.&i - Variabile &&classizz&i;
1120 %put Tempo trascorso:;
1121 %if &dg > 0 %then %do;
1122 %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
1123 %end;
1124 %else %if &do > 0 %then %do;
1125 %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
1126 %end;
1127 %else %if &dm > 0 %then %do;
1128 %put &dm.m:&dd.s (&dm2.m:&dd2.s);
1129 %end;
1130 %else %do;
1131 %put &dd.s (&dd2.s);
1132 %end;
1133
1134 proc printto log=&output_log; run;
1135
1136 data &in._pt_temp;
1137 set &in._pt;
1138 %if &num_condiz > 0 %then %do; /* 08.01 */
1139 %do j=1 %to &num_condiz; /* 08.01.01 */
1140 %if %upcase(&&v_condiz&j) = %upcase(&&classizz&i) %then %do; /* 08.01.01.01 */
1141 if &&condiz&j then delete;
1142 %end; /* 08.01.01.01 */
1143 %end; /* 08.01.01 */
1144 %end; /* 08.01 */
1145 keep &&classizz&i ;
1146 run;
```



```
1147
1148 proc sql noprint;
1149     select count(*) into :num_cl
1150     from &in._pt_temp;
1151 quit;
1152
1153 %if &num_cl = 0 %then %goto nonormali;
1154
1155 proc univariate data=&in._pt_temp noprint;
1156 var &&classizz&i ;
1157 output out=&in._pt_temp2
1158         pctlpre= p_
1159         pctlpts=0 to 100 by &passo ;
1160 run;
1161
1162
1163 proc transpose data=&in._pt_temp2 out=&in._pt_temp3 ;
1164 run;
1165
1166 proc sort data=&in._pt_temp3 ;
1167 by coll;
1168 run;
1169
1170 data &in._quanti ;
1171 set &in._pt_temp3 ;
1172 retain val_prec .;
1173 classe = _n_;
1174 output;
1175 val_prec = coll;
1176 run;
1177
1178 proc sort data=&in._quanti ;
1179 by classe;
1180 run;
1181
1182 proc sql noprint;
1183     select count(*) into :num_cl
1184     from &in._quanti ;
1185
1186     select coll into :max_cl1 - :max_cl%trim(&num_cl)
1187     from &in._quanti ;
1188
1189     select val_prec into :min_cl1 - :min_cl%trim(&num_cl)
1190     from &in._quanti ;
1191
1192     select classe into :nome_cl1 - :nome_cl%trim(&num_cl)
1193     from &in._quanti ;
1194 quit;
1195
1196 data _NULL_;
1197 set &in._quanti ;
1198 call symputx('min_classe'!!strip(_n_),val_prec);
1199 call symputx('max_classe'!!strip(_n_),coll);
1200 run;
1201
1202 proc delete data=&in._quanti ; run;
1203
1204 data &in._pt;
1205 set &in._pt;
1206 format &&varcl&i $&ludecla..;
1207 format &&varcl&i.._b 10.;
1208 &&varcl&i = '0';
1209 &&varcl&i.._b = -1;
1210 %do j=1 %to &num_cl ; /* 08.03 */
1211     %if &j = 1 %then %do; /* 08.03.01 */
1212         if &&classizz&i <= &&max_classe&j then do; /* 08.03.01.01 */
1213             &&varcl&i = "Min < &&classizz&i <= &&max_cl&j";
1214             &&varcl&i.._b = &&nome_cl&j;
1215             end; /* 08.03.01.01 */
1216         %end; /* 08.03.01 */
1217     %else %if &j = &num_cl %then %do; /* 08.03.02 */
1218         if &&classizz&i > &&min_classe&j then do; /* 08.03.02.01 */
```

```

1219         &&varcl&i = "&&min_cl&j < &&classizz&i <= Max";
1220         &&varcl&i.._b = &&nome_cl&j;
1221         end; /* 08.03.02.01 */
1222     %end; /* 08.03.02 */
1223     %else %do; /* 08.03.03 */
1224         if &&classizz&i > &&min_classe&j and &&classizz&i <= &&max_classe&j then
do; /* 08.03.03.01 */
1225             &&varcl&i = "&&min_cl&j < &&classizz&i <= &&max_cl&j";
1226             &&varcl&i.._b = &&nome_cl&j;
1227             end; /* 08.03.03.01 */
1228         %end; /* 08.03.03 */
1229     %end; /* 08.03 */
1230 run;
1231
1232 %nonormali:
1233
1234 %if %symexist(num_condiz) or num_condiz ^= 0 %then %do; /* 08.04 */
1235     data &in._pt;
1236     set &in._pt;
1237     format &&varcl&i $&ludecla..;
1238     format &&varcl&i.._b 10.;
1239     if &&varcl&i = '' then &&varcl&i = '0';
1240     if &&varcl&i.._b = . then &&varcl&i.._b = -1;
1241     %if &num_condiz > 0 %then %do; /* 08.04.01 */
1242         %do j=1 %to &num_condiz ; /* 08.04.01.01 */
1243             %if %upcase(&&v_condiz&j) = %upcase(&&classizz&i) %then %do; /* 08.04.01.01.01 */
1244                 if &&condiz&j then do; /* 08.04.01.01.01.01 */
1245                     &&varcl&i = "&&cl_condiz&j";
1246                     &&varcl&i.._b = -&j;
1247                 end; /* 08.04.01.01.01.01 */
1248             %end; /* 08.04.01.01.01 */
1249         %end; /* 08.04.01.01 */
1250     %end; /* 08.04.01 */
1251 run;
1252 %end; /* 08.04 */
1253
1254 %end; /* 08 */
1255
1256
1257
1258 %if %sysfunc(exist(&in._pt_temp)) %then %do; /* 09 */
1259     proc delete data=&in._pt_temp;
1260     run;
1261 %end; /* 09 */
1262
1263 %if %sysfunc(exist(&in._pt_temp2)) %then %do; /* 10 */
1264     proc delete data=&in._pt_temp2;
1265     run;
1266 %end; /* 10 */
1267
1268 %if %sysfunc(exist(&in._pt_temp3)) %then %do; /* 11 */
1269     proc delete data=&in._pt_temp3;
1270     run;
1271 %end; /* 11 */
1272
1273
1274 %noclassizz:
1275
1276 %let ludecla = 0;
1277
1278 proc sql noprint;
1279     select count(*) into :ludecla
1280     from &in._var
1281     where utilizzo in ('0' 'Q' 'X' 'C');
1282 quit;
1283
1284 %let ludecla = &ludecla;
1285
1286 proc sql noprint;
1287     select name, utilizzo into :variabl - :variab&ludecla, :utiliz1 - :utiliz&ludecla
1288     from &in._var
1289     where utilizzo in ('0' 'Q' 'X' 'C');

```

```
1290 quit;
1291
1292 %do i=1 %to &ludecla; /* 11b */
1293 %let variab&i = &&variab&i;
1294 %let utiliz&i = &&utiliz&i;
1295
1296 %if "&&utiliz&i" = "X" or "&&utiliz&i" = "0" %then %do; /* 11b.01 */
1297 proc sql noprint;
1298 select count(distinct CL_&&variab&i.._b) format=40.0 into :ludecla2
1299 from &in._pt;
1300 quit;
1301 %end; /* 11b.01 */
1302 %else %do; /* 11b.02 */
1303 proc sql noprint;
1304 select count(distinct &&variab&i) format=40.0 into :ludecla2
1305 from &in._pt;
1306 quit;
1307 %end; /* 11b.02 */
1308
1309 %let ludecla2 = &ludecla2;
1310
1311 %if &ludecla2 <= 1 %then %do; /* 11b.03 */
1312 data &in._var ;
1313 set &in._var ;
1314 if name = "&&variab&i" then utilizzo = 'N';
1315 run;
1316
1317 proc printto log=&log_0; run;
1318 %put Variabile &&variab&i eliminata dalla lista delle potenziali (ha solo &ludecla2
modalita);
1319 proc printto log=&output_log; run;
1320
1321 data &in._pt;
1322 set &in._pt;
1323 drop
1324 %if "&&utiliz&i" = "X" or "&&utiliz&i" = "0" %then %do; /* 11b.03.01 */
1325 CL_&&variab&i CL_&&variab&i.._b
1326 %end; /* 11b.03.01 */
1327 %else %do; /* 11b.03.02 */
1328 &&variab&i
1329 %end; /* 11b.03.02 */
1330 ;
1331 run;
1332 %end; /* 11b.03 */
1333 %end; /* 11b */
1334
1335 data &in._tesec;
1336 set &in._tesec;
1337 passoprec = ahora;
1338 ahora = datetime();
1339 delta = ahora - inizio;
1340 deltagg = int(delta/(60*60*24));
1341 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
1342 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
1343 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
1344 deltadec = deltasec + (delta - int(delta));
1345 delta2 = ahora - passoprec;
1346 deltagg2 = int(delta2/(60*60*24));
1347 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
1348 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
1349 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
1350 deltadec2 = deltasec2 + (delta2 - int(delta2));
1351 run;
1352
1353 proc sql noprint;
1354 select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
1355 from &in._tesec;
1356
1357 select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
1358 from &in._tesec;
1359 quit;
1360
```

```
1361 %let dg = &dg; %let dg2 = &dg2;
1362 %let do = &do; %let do2 = &do2;
1363 %let dm = &dm; %let dm2 = &dm2;
1364 %let ds = &ds; %let ds2 = &ds2;
1365 %let dd = &dd; %let dd2 = &dd2;
1366
1367
1368 proc printto log=&log_0; run;
1369
1370 %put 06 - Analisi della correlazione (o indipendenza) fra le variabili;
1371 %put Tempo trascorso;;
1372 %if &dg > 0 %then %do;
1373     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
1374 %end;
1375 %else %if &do > 0 %then %do;
1376     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
1377 %end;
1378 %else %if &dm > 0 %then %do;
1379     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
1380 %end;
1381 %else %do;
1382     %put &dd.s (&dd2.s);
1383 %end;
1384
1385 proc printto log=&output_log; run;
1386
1387 data &in._var_b;
1388 set &in._var;
1389 if utilizzo = 'Q' or utilizzo = '0' or utilizzo = 'X';
1390 run;
1391
1392 data _null_;
1393 set &in._var_b end=fine;
1394 call symputx('varaltre'!!strip(_n_),name);
1395 call symputx('utilaltre'!!strip(_n_),utilizzo);
1396 if fine then call symputx('numvaraltre',(_n_));
1397 run;
1398
1399
1400 data &in._temp &in._corr &in._corrs &in._corr2 &in._corr3 &in._corr3b;
1401 set _NULL_;
1402 run;
1403
1404 data &in._corr4;
1405 set _NULL_;
1406 format v1 $50.;
1407 format v2 $50.;
1408 format corr best12.;
1409 format tipo_corr $1.;
1410 format ut_v1 $2.;
1411 format ut_v2 $2.;
1412 format corr2 best12.;
1413 run;
1414
1415 %if %symexist(numvarcl) = 0 %then %let numvarcl = 0;
1416 %if %symexist(numvaraltre) = 0 %then %let numvaraltre = 0;
1417
1418
1419 proc sql noprint;
1420     select count(*) into :num_corr
1421     from &in._var_b;
1422 quit;
1423
1424 %if &num_corr < 2 %then %do; /* 11b */
1425     proc delete data=&in._var_b ;
1426     run;
1427
1428     %goto passol;
1429 %end; /* 11b */
1430
1431 proc delete data=&in._var_b ;
1432 run;
```

```
1433
1434
1435 data &in._pt_temp;
1436 set &in._pt;
1437 keep
1438 %do i=1 %to &numvaraltre;          /* 12 */
1439     &&varaltre&i
1440 %end; /* 12 */
1441 ;
1442 run;
1443
1444 proc corr data=&in._pt_temp outp=&in._corr out=&in._corrs noprint;
1445 run;
1446
1447 data &in._corr ;
1448 set &in._corr ;
1449 if _TYPE_ = 'CORR';
1450 drop _TYPE_;
1451 run;
1452
1453 data &in._corrs ;
1454 set &in._corrs ;
1455 if _TYPE_ = 'CORR';
1456 drop _TYPE_;
1457 run;
1458
1459
1460 data &in._corr2;
1461 set _NULL_;
1462 format v1 $50.;
1463 format v2 $50.;
1464 format corr best12.;
1465 format tipo_corr $1.;
1466 run;
1467
1468
1469 proc sql noprint;
1470 select count(*) into :num_corr
1471 from &in._corr;
1472
1473 select _NAME_ into :n_corr1 - :n_corr%trim(&num_corr)
1474 from &in._corr;
1475 quit;
1476
1477
1478 %if %symexist(num_corr) %then %do;          /* 14 */
1479 %if &num_corr > 1 %then %do;          /* 14.01 */
1480
1481 proc sql noprint;
1482 %do i=1 %to &num_corr; /* 14.01.01 */
1483 %do j=&i+1 %to &num_corr; /* 14.01.01.01 */
1484 insert into &in._corr2
1485     select "&n_corr&i", "&n_corr&j", &n_corr&j, "P"
1486     from &in._corr
1487     where _NAME_ = "&n_corr&i";
1488
1489 insert into &in._corr2
1490     select "&n_corr&i", "&n_corr&j", &n_corr&j, "S"
1491     from &in._corrs
1492     where _NAME_ = "&n_corr&i";
1493 %end; /* 14.01.01.01 */
1494 %end; /* 14.01.01 */
1495 quit;
1496
1497 %end; /* 14.01 */
1498 %end; /* 14 */
1499
1500
1501 data &in._temp;
1502 set &in._var;
1503 if utilizzo = 'X' or utilizzo = 'Q' or utilizzo = '0';
1504 keep name utilizzo;
```

```
1505 run;
1506
1507
1508 data &in._corr2;
1509 set &in._corr2;
1510 v1 = upcase(v1);
1511 v2 = upcase(v2);
1512 run;
1513
1514
1515 proc sql noprint;
1516 create table &in._corr3 as
1517 select a.*, b.utilizzo as ut_v1
1518 from &in._corr2 a, &in._temp b
1519 where a.v1 = b.name;
1520
1521 create table &in._corr3b as
1522 select a.*, b.utilizzo as ut_v2
1523 from &in._corr3 a, &in._temp b
1524 where a.v2 = b.name;
1525 quit;
1526
1527
1528 data &in._corr3b;
1529 set &in._corr3b;
1530 if ut_v1 = '0' or ut_v2 = '0' then do; /* 14b */
1531 if tipo_corr = 'P' then delete;
1532 end; /* 14b */
1533 else if tipo_corr = 'S' then delete;
1534 run;
1535
1536
1537 data &in._corr3b;
1538 set &in._corr3b;
1539 corr2=0;
1540 if corr < -&taglio_correlazione or corr > &taglio_correlazione then corr2=1;
1541 run;
1542
1543
1544 data &in._corr4;
1545 set &in._corr3b;
1546 if ut_v1 = 'X' or ut_v1 = '0' then v1 = 'CL_' || v1;
1547 if ut_v2 = 'X' or ut_v2 = '0' then v2 = 'CL_' || v2;
1548 run;
1549
1550
1551 proc sql noprint;
1552 create table &in._escon2 as
1553 select a.var1, a.var2, b.utilizzo as util1
1554 from &in._escon a left join &in._var b on
1555 (a.var1 = b.name);
1556 quit;
1557
1558
1559 proc sql noprint;
1560 create table &in._escon3 as
1561 select a.*, b.utilizzo as util2
1562 from &in._escon2 a left join &in._var b on
1563 (a.var2 = b.name);
1564 quit;
1565
1566
1567 data &in._escon2;
1568 set &in._escon3;
1569 format v1 $100.;
1570 format v2 $100.;
1571 if util1 in ('C' '0' 'Q' 'X') and util2 in ('C' '0' 'Q' 'X');
1572 if util1 in ('0' 'X') then v1 = 'CL_' || var1;
1573 else v1 = var1;
1574 if util2 in ('0' 'X') then v2 = 'CL_' || var2;
1575 else v2 = var2;
1576 corr = .;
```

```
1577 tipo_corr = 'E';
1578 ut_v1 = util1;
1579 ut_v2 = util2;
1580 corr2 = 1;
1581 keep v1 v2 corr tipo_corr ut_v1 ut_v2 corr2;
1582 run;
1583
1584 data &in._corr4;
1585 set &in._corr4 &in._esccon2;
1586 run;
1587
1588 proc delete data=&in._esccon2 &in._esccon3;
1589 run;
1590
1591
1592 proc delete data = &in._temp &in._corrp &in._corrs &in._corr2 &in._corr3 &in._corr3b
&in._pt_temp;
1593 run;
1594
1595
1596 %passol:
1597
1598 %if &simpson = 1 %then %do; /* 15a */
1599
1600 data &in._var_b;
1601 set &in._var;
1602 if utilizzo = 'Q' or utilizzo = 'O' or utilizzo = 'X' or utilizzo = 'C';
1603 run;
1604
1605 data _null_;
1606 set &in._var_b end=fine;
1607 call symputx('varaltec'!!strip(_n_), compress(name));
1608 call symputx('utilaltec'!!strip(_n_), compress(utilizzo));
1609 if fine then call symputx('numvaraltec',(_n_));
1610 run;
1611
1612 %if %symexist(numvaraltec) = 0 %then %let numvaraltec = 0;
1613
1614 %if &numvaraltec < 2 %then %do; /* 15b */
1615 proc delete data=&in._var_b ;
1616 run;
1617
1618 %goto passo2;
1619 %end; /* 15b */
1620
1621 proc delete data=&in._var_b ;
1622 run;
1623
1624 %do i=1 %to &numvaraltec; /* 16 */
1625 %if "&utilaltec&i" = "C" %then %do; /* 16.01 */
1626 %do j=1 %to &numvaraltec; /* 16.01.01 */
1627 %if ("&utilaltec&j" ^= "C" and &j < &i) or &j > &i %then %do; /* 16.01.01.01 */
1628 %simpson_c(vara=&varaltec&i, varb=&varaltec&j,
1629 utia=&utilaltec&i, utib=&utilaltec&j,
1630 prefisso=&in, taglio=&taglio_correlazione);
1631
1632 proc sql noprint;
1633 select val into :coor
1634 from &in._rit;
1635 quit;
1636
1637 proc delete data=&in._rit;
1638 run;
1639
1640 %if &coor ^= 0 %then %do; /* 16.01.01.01.01 */
1641 %if "&utilaltec&i" = "X" or "&utilaltec&i" = "0" %then %do; /* 16.01.01.01.01.01 */
1642 %let ivaraltec&i = CL_&varaltec&i;
1643 %end; /* 16.01.01.01.01.01 */
1644 %else %do; /* 16.01.01.01.01.02 */
1645 %let ivaraltec&i = &varaltec&i;
1646 %end; /* 16.01.01.01.01.02 */
1647
```

```
1648     %if "&&utilaltc&j" = "X" or "&&utilaltc&j" = "0" %then %do;          /* 16.01.01.01.01.03 */
1649         %let ivaraltc&j = CL &&varaltc&j;
1650     %end; /* 16.01.01.01.01.03 */
1651     %else %do; /* 16.01.01.01.01.04 */
1652         %let ivaraltc&j = &&varaltc&j;
1653     %end; /* 16.01.01.01.01.04 */
1654
1655     proc sql noprint;
1656         insert into &in._corr4 values
1657         ("&&ivaraltc&i", "&&ivaraltc&j", &coor, "C", "&&utilaltc&i", "&&utilaltc&j", 1);
1658     quit;
1659     %end; /* 16.01.01.01.01 */
1660 %end; /* 16.01.01.01 */
1661 %end; /* 16.01.01 */
1662 %end; /* 16.01 */
1663 %end; /* 16 */
1664 %end; /* 15a */
1665
1666
1667 %passo2:
1668
1669 data &in._tesec;
1670 set &in._tesec;
1671 passoprec = ahora;
1672 ahora = datetime();
1673 delta = ahora - inizio;
1674 deltagg = int(delta/(60*60*24));
1675 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
1676 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
1677 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
1678 deltadec = deltasec + (delta - int(delta));
1679 delta2 = ahora - passoprec;
1680 deltagg2 = int(delta2/(60*60*24));
1681 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
1682 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
1683 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
1684 deltadec2 = deltasec2 + (delta2 - int(delta2));
1685 run;
1686
1687 proc sql noprint;
1688     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
1689     from &in._tesec;
1690
1691     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
1692     from &in._tesec;
1693 quit;
1694
1695 %let dg = &dg; %let dg2 = &dg2;
1696 %let do = &do; %let do2 = &do2;
1697 %let dm = &dm; %let dm2 = &dm2;
1698 %let ds = &ds; %let ds2 = &ds2;
1699 %let dd = &dd; %let dd2 = &dd2;
1700
1701 proc printto log=&log_0; run;
1702
1703 %put 07 - Inizio regressione;
1704 %put Tempo trascorso;;
1705 %if &dg > 0 %then %do;
1706     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
1707 %end;
1708 %else %if &do > 0 %then %do;
1709     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
1710 %end;
1711 %else %if &dm > 0 %then %do;
1712     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
1713 %end;
1714 %else %do;
1715     %put &dd.s (&dd2.s);
1716 %end;
1717
1718 proc printto log=&output_log; run;
1719
```



```
1720 data &in._mod;
1721 set _NULL_;
1722   format nome $50.;
1723   format utilizzo $1.;
1724 run;
1725
1726 data &in._po;
1727 set &in._var;
1728   format nome $50.;
1729   format po 1.;
1730   if utilizzo in ('X' 'Q' 'C' '0');
1731   if utilizzo in ('X' '0') then nome = cats('CL_', name);
1732   else nome = name;
1733   po = 1;
1734   keep nome utilizzo po;
1735 run;
1736
1737
1738 proc sql noprint;
1739   select name into :v_risp
1740     from &in._var
1741     where utilizzo = 'R';
1742
1743
1744   select count(*) into :nin
1745     from &in._po;
1746
1747   select nome into :vin1 - :vin%trim(&nin)
1748     from &in._po;
1749
1750   select utilizzo into :uin1 - :uin%trim(&nin)
1751     from &in._po;
1752 quit;
1753
1754 %let nin = &nin;
1755
1756 ods listing close;
1757 ods output ParameterEstimates=&in._mcorr ModelFit=&in._smcorr ConvergenceStatus=&in._cocorr;
1758
1759 proc genmod data=&in._pt descending NAMELEN=50;
1760   model &v_risp = / dist = &distribuzione ;
1761   output out=&in._dcorr predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
1762 run;
1763
1764 ods output close;
1765 ods listing;
1766
1767 proc sql noprint;
1768   select status into :stcorr
1769     from &in._cocorr;
1770 quit;
1771
1772 %let stcorr = &stcorr;
1773
1774 proc delete data=&in._cocorr;
1775 run;
1776
1777 data &in._smcorr;
1778 set &in._smcorr;
1779   format df 20.0;
1780 run;
1781
1782 data &in._mcorr;
1783 set &in._mcorr;
1784   format level1 $30.;
1785   format estimate 20.10;
1786   parameter = upcase(parameter);
1787 run;
1788
1789 proc sql noprint;
1790   select count(*) into :n_param
1791     from &in._mcorr
```

```
1792     where df ^= 0;
1793 quit;
1794
1795 data &in._smcorr;
1796 set &in._smcorr;
1797   if _n_ = 5 then do;    /* 17 */
1798     output;
1799     criterion = 'AIC';
1800     value = (2 * &n_param) - 2 * value;
1801   end;    /* 17 */
1802   output;
1803 run;
1804
1805
1806 %let attivo = 0;
1807 %let vold = ;
1808 %let vcold = ;
1809
1810 %do %while(&attivo < 2);    /* 18 */
1811
1812 %let nuovo = 0;
1813
1814   %do i=1 %to &nin;    /* 18.01 */
1815
1816   %if &oncamm = 1 %then %do;    /* 18.01b */
1817     %if &i = 1 %then %do; /* 18.01b.01 */
1818       %let pcamm = %eval(&pcamm + 1);
1819     %end; /* 18.01b.01 */
1820     %if &pcamm <= &lcamm %then %do;    /* 18.01b.02 */
1821       %if "&&vin&i" ^= "&&camm&pcamm" %then %goto goloop;
1822     %end; /* 18.01b.02 */
1823     %else %do;    /* 18.01b.03 */
1824       %let oncamm = 0;
1825     %end; /* 18.01b.03 */
1826   %end; /* 18.01b */
1827
1828 data &in._tesec;
1829 set &in._tesec;
1830 passoprec = ahora;
1831 ahora = datetime();
1832 delta = ahora - inizio;
1833 deltagg = int(delta/(60*60*24));
1834 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
1835 deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
1836 deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
1837 deltadec = deltasec + (delta - int(delta));
1838 delta2 = ahora - passoprec;
1839 deltagg2 = int(delta2/(60*60*24));
1840 deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
1841 deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
1842 deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
1843 deltadec2 = deltasec2 + (delta2 - int(delta2));
1844 run;
1845
1846 proc sql noprint;
1847   select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
1848   from &in._tesec;
1849
1850   select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
1851   from &in._tesec;
1852 quit;
1853
1854 %let dg = &dg; %let dg2 = &dg2;
1855 %let do = &do; %let do2 = &do2;
1856 %let dm = &dm; %let dm2 = &dm2;
1857 %let ds = &ds; %let ds2 = &ds2;
1858 %let dd = &dd; %let dd2 = &dd2;
1859
1860 proc printto log=&log_0; run;
1861
1862 %put 07 (&i./&nin.) - Provo ad inserire la variabile &&vin&i (modello: &vold);
1863 %put Tempo trascorso;;
```

```
1864 %if &dg > 0 %then %do;
1865 %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
1866 %end;
1867 %else %if &do > 0 %then %do;
1868 %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
1869 %end;
1870 %else %if &dm > 0 %then %do;
1871 %put &dm.m:&dd.s (&dm2.m:&dd2.s);
1872 %end;
1873 %else %do;
1874 %put &dd.s (&dd2.s);
1875 %end;
1876
1877 proc printto log=&output_log; run;
1878
1879 proc sort data=&in._pt;
1880 by &&vin&i;
1881 run;
1882
1883 %if %upcase(&&uin&i) = X or %upcase(&&uin&i) = 0 %then %do; /* 18.01.01 */
1884
1885 data &in._k1;
1886 set &in._pt;
1887 if &&vin&i.._b >= 0;
1888 keep &&vin&i.._b;
1889 run;
1890
1891 proc sort data=&in._k1 nodup;
1892 by &&vin&i.._b;
1893 run;
1894
1895 proc sql noprint;
1896 select count(*) into :limite_giri_k
1897 from &in._k1;
1898 quit;
1899
1900 data &in._k2;
1901 set &in._k1;
1902 classe_orig = &&vin&i.._b;
1903 classe_der = classe_orig;
1904 keep classe_orig classe_der;
1905 run;
1906
1907 data &in._ptk;
1908 set &in._pt;
1909 &&vin&i.._c = &&vin&i.._b;
1910 run;
1911
1912 ods listing close;
1913 ods output ParameterEstimates=&in._mbestk ModelFit=&in._smbestk ConvergenceStatus=&in._cobestk;
1914
1915 proc genmod data=&in._ptk descending NAMELEN=50;
1916 class &vcold
1917 %if %upcase(&&uin&i) = C %then %do; /* 18.01.01.01a */
1918 &&vin&i
1919 %end; /* 18.01.01.01a */
1920 %if %upcase(&&uin&i) = X or %upcase(&&uin&i) = 0 %then %do; /* 18.01.01.01b */
1921 &&vin&i.._c
1922 %end; /* 18.01.01.01b */
1923 / missing ;
1924 model &v_risp = &vold &&vin&i.._c / dist = &distribuzione ;
1925 output out=&in._dbestk predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
1926 run;
1927
1928 ods output close;
1929 ods listing;
1930
1931 proc sql noprint;
1932 select status into :stbestk
1933 from &in._cobestk;
1934 quit;
1935
```

```
1936 %let stbestk = &stbestk;
1937
1938 proc delete data=&in._cobestk;
1939 run;
1940
1941 data &in._smbestk;
1942 set &in._smbestk;
1943 format df 20.0;
1944 run;
1945
1946 data &in._mbestk;
1947 set &in._mbestk;
1948 format level1 $30.;
1949 format estimate 20.10;
1950 parameter = upcase(parameter);
1951 run;
1952
1953 proc sql noprint;
1954 select count(*) into :n_param
1955 from &in._mbestk
1956 where df ^= 0;
1957 quit;
1958
1959 data &in._smbestk;
1960 set &in._smbestk;
1961 if _n_ = 5 then do; /* 18.01.01.02 */
1962 output;
1963 criterion = 'AIC';
1964 value = (2 * &n_param) - 2 * value;
1965 end; /* 18.01.01.02 */
1966 output;
1967 run;
1968
1969 proc sql noprint;
1970 select value format=40.20 into :aic_bestk
1971 from &in._smbestk
1972 where criterion = 'AIC';
1973 quit;
1974
1975 %let aic_bestk = &aic_bestk;
1976
1977 proc delete data=&in._k1;
1978 run;
1979
1980 %let contatore = 1;
1981 %let limite_giri_k = %eval(&limite_giri_k*3);
1982 %let ok_k = 4;
1983
1984 %iniziowhile:
1985
1986 proc sql noprint;
1987 select count(distinct classe_der) into :nccc
1988 from &in._k2;
1989 quit;
1990
1991 %let nccc = &nccc;
1992
1993 proc sql noprint;
1994 select distinct classe_der into :cl_der1 - :cl_der&nccc
1995 from &in._k2;
1996 quit;
1997
1998 %do j=2 %to &nccc; /* 18.01.01.03 */
1999
2000 %let j2 = %eval(&j - 1);
2001
2002 data &in._k4;
2003 set &in._k2;
2004 if classe_der = &&cl_der&j then classe_der = &&cl_der&j2;
2005 run;
2006
2007 proc sql noprint;
```

```
2008 create table &in._ptk as
2009 select a.*, b.classe_der as &&vin&i.._c
2010 from &in._pt a left join &in._k4 b on
2011 (a.&&vin&i.._b = b.classe_orig);
2012 quit;
2013
2014 data &in._ptk;
2015 set &in._ptk;
2016 if &&vin&i.._c = . then &&vin&i.._c = &&vin&i.._b;
2017 run;
2018
2019 ods listing close;
2020 ods output ParameterEstimates=&in._mnewk ModelFit=&in._smnewk ConvergenceStatus=&in._conewk;
2021
2022 proc genmod data=&in._ptk descending NAMELEN=50;
2023 class &vcold
2024 %if %upcase(&&uin&i) = C %then %do; /* 18.01.01.03.01a */
2025 &&vin&i
2026 %end; /* 18.01.01.03.01a */
2027 %if %upcase(&&uin&i) = X or %upcase(&&uin&i) = 0 %then %do; /* 18.01.01.03.01b */
2028 &&vin&i.._c
2029 %end; /* 18.01.01.03.01b */
2030 / missing ;
2031 model &v_risp = &vold &&vin&i.._c / dist = &distribuzione ;
2032 output out=&in._dnewk predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
2033 run;
2034
2035 ods output close;
2036 ods listing;
2037
2038 proc sql noprint;
2039 select status into :stnewk
2040 from &in._conewk;
2041 quit;
2042
2043 %let stnewk = &stnewk;
2044
2045 proc delete data=&in._conewk;
2046 run;
2047
2048 data &in._smnewk;
2049 set &in._smnewk;
2050 format df 20.0;
2051 run;
2052
2053 data &in._mnewk;
2054 set &in._mnewk;
2055 format level1 $30.;
2056 format estimate 20.10;
2057 parameter = upcase(parameter);
2058 run;
2059
2060 proc sql noprint;
2061 select count(*) into :n_param
2062 from &in._mnewk
2063 where df ^= 0;
2064 quit;
2065
2066 data &in._smnewk;
2067 set &in._smnewk;
2068 if _n_ = 5 then do; /* 18.01.01.03.02 */
2069 output;
2070 criterion = 'AIC';
2071 value = (2 * &n_param) - 2 * value;
2072 end; /* 18.01.01.03.02 */
2073 output;
2074 run;
2075
2076 proc sql noprint;
2077 select value format=40.20 into :aic_newk
2078 from &in._smnewk
2079 where criterion = 'AIC';
```

```
2080 quit;
2081
2082 %let aic_newk = &aic_newk;
2083
2084 %if &stnewk = 0 %then %do; /* 18.01.01.03.02b */
2085 %if &stbestk ^= 0 or %sysevalf(&aic_bestk > &aic_newk) %then %do; /* 18.01.01.03.03 */
2086
2087 %let stbestk = &stnewk;
2088 %let aic_bestk = &aic_newk;
2089 %let ok_k = 1;
2090
2091 data &in._mbestk;
2092 set &in._mnewk;
2093 run;
2094
2095 data &in._smbestk;
2096 set &in._smnewk;
2097 run;
2098
2099 data &in._dbestk;
2100 set &in._dnewk;
2101 run;
2102
2103 data &in._k4_ok;
2104 set &in._k4;
2105 run;
2106
2107 %end; /* 18.01.01.03.03 */
2108 %end; /* 18.01.01.03.02b */
2109 %end; /* 18.01.01.03 */
2110
2111 %if &ok_k ^= 1 %then %do; /* 18.01.01.04 */
2112
2113 %let contatore = %eval(&contatore+1);
2114 %if &contatore >= 2 %then %goto finewhilek;
2115 %end; /* 18.01.01.04 */
2116
2117 %else %do; /* 18.01.01.05 */
2118 data &in._k2;
2119 set &in._k4_ok;
2120 run;
2121
2122 proc delete data=&in._k4_ok;
2123 run;
2124
2125 %let contatore = 0;
2126 %end; /* 18.01.01.05 */
2127
2128 %let ok_k = 1;
2129
2130 proc sql noprint;
2131 create table &in._kriass as
2132 select classe_der, min(classe_orig) as min, max(classe_orig) as max
2133 from &in._k2
2134 group by classe_der;
2135 quit;
2136
2137 data &in._kriass2;
2138 set &in._kriass;
2139 if min ^= max;
2140 run;
2141
2142 proc sql noprint;
2143 select count(*) into :nncc
2144 from &in._kriass2;
2145 quit;
2146
2147 %let nncc = &nncc;
2148
2149 proc sql noprint;
2150 select classe_der, min, max into :col - :co&nncc, :mi1 - :mi&nncc, :ma1 - :ma&nncc
2151 from &in._kriass2;
```

```
2152 quit;
2153
2154 proc delete data=&in._kriass &in._kriass2;
2155 run;
2156
2157 %do j=1 %to &nnc; /* 18.01.01.06 */
2158
2159 proc sql noprint;
2160 select min(classe_orig) into :nuova_classe_k
2161 from &in._k2
2162 where classe_der = &&co&j and classe_orig ^= &&mi&j;
2163 quit;
2164
2165 data &in._k3;
2166 set &in._k2;
2167 if classe_der = &&co&j and classe_orig ^= &&mi&j then classe_der = &nuova_classe_k;
2168 run;
2169
2170 proc sql noprint;
2171 create table &in._ptk as
2172 select a.*, b.classe_der as &&vin&i.._c
2173 from &in._pt a left join &in._k3 b on
2174 (a.&&vin&i.._b = b.classe_orig);
2175 quit;
2176
2177 data &in._ptk;
2178 set &in._ptk;
2179 if &&vin&i.._c = . then &&vin&i.._c = &&vin&i.._b;
2180 run;
2181
2182 ods listing close;
2183 ods output ParameterEstimates=&in._mnewk ModelFit=&in._smnewk ConvergenceStatus=&in._conewk;
2184
2185 proc genmod data=&in._ptk descending NAMELEN=50;
2186 class &vcold
2187 %if %upcase(&&uin&i) = C %then %do; /* 18.01.01.06.01a */
2188 &&vin&i
2189 %end; /* 18.01.01.06.01a */
2190 %if %upcase(&&uin&i) = X or %upcase(&&uin&i) = 0 %then %do; /* 18.01.01.06.01b */
2191 &&vin&i.._c
2192 %end; /* 18.01.01.06.01b */
2193 / missing ;
2194 model &v_out = &vold &&vin&i.._c / dist = &istribuzione ;
2195 output out=&in._dnewk predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
2196 run;
2197
2198 ods output close;
2199 ods listing;
2200
2201 proc sql noprint;
2202 select status into :stnewk
2203 from &in._conewk;
2204 quit;
2205
2206 %let stnewk = &stnewk;
2207
2208 proc delete data=&in._conewk;
2209 run;
2210
2211 data &in._smnewk;
2212 set &in._smnewk;
2213 format df 20.0;
2214 run;
2215
2216 data &in._mnewk;
2217 set &in._mnewk;
2218 format level1 $30.;
2219 format estimate 20.10;
2220 parameter = upcase(parameter);
2221 run;
2222
2223 proc sql noprint;
```

```

2224     select count(*) into :n_param
2225         from &in._mnewk
2226         where df ^= 0;
2227 quit;
2228
2229 data &in._smnewk;
2230 set &in._smnewk;
2231 if _n_ = 5 then do;    /* 18.01.01.06.02 */
2232     output;
2233     criterion = 'AIC';
2234     value = (2 * &n_param) - 2 * value;
2235     end;    /* 18.01.01.06.02 */
2236 output;
2237 run;
2238
2239 proc sql noprint;
2240     select value format=40.20 into :aic_newk
2241         from &in._smnewk
2242         where criterion = 'AIC';
2243 quit;
2244
2245 %let aic_newk = &aic_newk;
2246
2247 %if &stnewk = 0 %then %do;    /* 18.01.01.06.02b */
2248     %if &stbestk ^= 0 or %sysevalf(&aic_bestk > &aic_newk) %then %do;    /* 18.01.01.06.03 */
2249
2250         %let stbestk = &stnewk;
2251         %let aic_bestk = &aic_newk;
2252         %let ok_k = 2;
2253
2254         data &in._mbestk;
2255         set &in._mnewk;
2256         run;
2257
2258         data &in._smbestk;
2259         set &in._smnewk;
2260         run;
2261
2262         data &in._dbestk;
2263         set &in._dnewk;
2264         run;
2265
2266         data &in._k4_ok;
2267         set &in._k3;
2268         run;
2269
2270     %end;    /* 18.01.01.06.03 */
2271 %end;    /* 18.01.01.06.02b */
2272
2273 data &in._k4;
2274 set &in._k2;
2275     if classe_orig = &&ma&j then classe_der = &&ma&j;
2276 run;
2277
2278 proc sql noprint;
2279     create table &in._ptk as
2280     select a.*, b.classe_der as &&vin&i.._c
2281     from &in._pt a left join &in._k4 b on
2282     (a.&&vin&i.._b = b.classe_orig);
2283 quit;
2284
2285 data &in._ptk;
2286 set &in._ptk;
2287     if &&vin&i.._c = . then &&vin&i.._c = &&vin&i.._b;
2288 run;
2289
2290 ods listing close;
2291 ods output ParameterEstimates=&in._mnewk ModelFit=&in._smnewk ConvergenceStatus=&in._conewk;
2292
2293 proc genmod data=&in._ptk descending NAMELEN=50;
2294 class &vcold
2295     %if %upcase(&&uin&i) = C %then %do; /* 18.01.01.06.04a */

```



```
2296         &&vin&i
2297     %end;          /* 18.01.01.06.04a */
2298     %if %upcase(&&uin&i) = X or %upcase(&&uin&i) = 0 %then %do; /* 18.01.01.06.04b */
2299         &&vin&i._c
2300     %end;          /* 18.01.01.06.04b */
2301     / missing ;
2302     model &v_risp = &vold &&vin&i._c / dist = &distribuzione ;
2303     output out=&in._dnewk predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
2304     run;
2305
2306     ods output close;
2307     ods listing;
2308
2309     proc sql noprint;
2310         select status into :stnewk
2311         from &in._conewk;
2312     quit;
2313
2314     %let stnewk = &stnewk;
2315
2316     proc delete data=&in._conewk;
2317     run;
2318
2319     data &in._smnewk;
2320     set &in._smnewk;
2321     format df 20.0;
2322     run;
2323
2324     data &in._mnewk;
2325     set &in._mnewk;
2326     format level1 $30.;
2327     format estimate 20.10;
2328     parameter = upcase(parameter);
2329     run;
2330
2331     proc sql noprint;
2332         select count(*) into :n_param
2333         from &in._mnewk
2334         where df ^= 0;
2335     quit;
2336
2337     data &in._smnewk;
2338     set &in._smnewk;
2339     if _n_ = 5 then do; /* 18.01.01.06.05 */
2340         output;
2341         criterion = 'AIC';
2342         value = (2 * &n_param) - 2 * value;
2343     end; /* 18.01.01.06.05 */
2344     output;
2345     run;
2346
2347     proc sql noprint;
2348         select value format=40.20 into :aic_newk
2349         from &in._smnewk
2350         where criterion = 'AIC';
2351     quit;
2352
2353     %let aic_newk = &aic_newk;
2354
2355     %if &stnewk = 0 %then %do; /* 18.01.01.06.05b */
2356     %if &stbestk ^= 0 or %sysevalf(&aic_bestk > &aic_newk) %then %do; /* 18.01.01.06.06 */
2357
2358         %let stbestk = &stnewk;
2359         %let aic_bestk = &aic_newk;
2360         %let ok_k = 3;
2361
2362         data &in._mbestk;
2363         set &in._mnewk;
2364         run;
2365
2366         data &in._smbestk;
2367         set &in._smnewk;
```

```
2368     run;
2369
2370     data &in._dbestk;
2371     set &in._dnewk;
2372     run;
2373
2374     data &in._k4_ok;
2375     set &in._k4;
2376     run;
2377     %end; /* 18.01.01.06.06 */
2378     %end; /* 18.01.01.06.05b */
2379     %end; /* 18.01.01.06 */
2380
2381
2382     %if &ok_k < 2 %then %do; /* 18.01.01.07 */
2383
2384     %let contatore = %eval(&contatore+1);
2385     %if &contatore >= 2 %then %goto finewhilek;
2386
2387     %end; /* 18.01.01.07 */
2388     %else %do; /* 18.01.01.08 */
2389     %if &ok_k = 2 %then %do; /* 18.01.01.08.01 */
2390
2391     data &in._k2;
2392     set &in._k4_ok;
2393     run;
2394
2395     proc delete data=&in._k4_ok;
2396     run;
2397
2398     %end; /* 18.01.01.08.01 */
2399     %else %do; /* 18.01.01.08.02 */
2400
2401     data &in._k2;
2402     set &in._k4_ok;
2403     run;
2404
2405     proc delete data=&in._k4_ok;
2406     run;
2407
2408     %end; /* 18.01.01.08.02 */
2409
2410     %let contatore = 0;
2411     %end; /* 18.01.01.08 */
2412
2413     %let ok_k = 4;
2414
2415     %let limite_giri_k = %eval(&limite_giri_k-1);
2416     %if &limite_giri_k <= 0 %then %goto finewhilek;
2417
2418     %goto iniziowhile;
2419
2420     %finewhilek:
2421
2422     %mod_b_meno_a(moda=&in._smbestk, modb=&in._smcorr, alpha=&alfa, prefisso=&in);
2423
2424     proc sql noprint;
2425     select val into :meglio
2426     from &in._rit;
2427     quit;
2428
2429     proc delete data=&in._rit;
2430     run;
2431
2432     %if &stbestk ^= 0 %then %do;
2433     %let meglio = 0;
2434     %end;
2435     %if &stcorr ^= 0 and &stbestk = 0 %then %do;
2436     %let meglio = 1;
2437     %end;
2438
2439     %if &meglio = 1 %then %do; /* 18.01.01.09 */
```

```

2440
2441 proc sql noprint;
2442   select value format=40.20 into :aic_new
2443     from &in._smbestk
2444     where criterion = 'AIC';
2445 quit;
2446
2447 %let aic_new = &aic_new;
2448
2449 %if &nuovo = 0 %then %do;      /* 18.01.01.09.01 */
2450
2451   %let stbest = &stbestk;
2452   %let nuovo = 1;
2453   %let aic_best = &aic_new;
2454   %let v_inserita = &&vin&i;
2455
2456   data &in._mbest;
2457   set &in._mbestk;
2458   run;
2459
2460   data &in._smbest;
2461   set &in._smbestk;
2462   run;
2463
2464   data &in._dbest;
2465   set &in._dbestk;
2466   run;
2467
2468   %end;      /* 18.01.01.09.01 */
2469
2470 %else %do;   /* 18.01.01.09.02 */
2471
2472   %if %sysevalf(&aic_best > &aic_new) %then %do;      /* 18.01.01.09.02.01 */
2473
2474     %let stbest = &stbestk;
2475     %let aic_best = &aic_new;
2476     %let v_inserita = &&vin&i;
2477
2478     data &in._mbest;
2479     set &in._mbestk;
2480     run;
2481
2482     data &in._smbest;
2483     set &in._smbestk;
2484     run;
2485
2486     data &in._dbest;
2487     set &in._dbestk;
2488     run;
2489
2490     %end;      /* 18.01.01.09.02.01 */
2491
2492   %end;      /* 18.01.01.09.02 */
2493
2494 %end;      /* 18.01.01.09 */
2495
2496 %end; /* 18.01.01 */
2497
2498 %else %do;   /* 18.01.02 */
2499
2500 ods listing close;
2501 ods output ParameterEstimates=&in._mnew ModelFit=&in._smnew ConvergenceStatus=&in._conew;
2502
2503 proc genmod data=&in._pt descending NAMELEN=50;
2504 class &vcold
2505   %if %upcase(&&uin&i) = C or %upcase(&&uin&i) = X or
2506     %upcase(&&uin&i) = 0 %then %do; /* 18.01.02.01 */
2507     &&vin&i
2508   %end;      /* 18.01.02.01 */
2509 / missing ;
2510 model &v_risp = &vold &&vin&i / dist = &istribuzione ;
2511 output out=&in._dnew predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;

```

```
2512     run;
2513
2514     ods output close;
2515     ods listing;
2516
2517     proc sql noprint;
2518         select status into :stnew
2519         from &in._conew;
2520     quit;
2521
2522     %let stnew = &stnew;
2523
2524     proc delete data=&in._conew;
2525     run;
2526
2527     data &in._smnew;
2528     set &in._smnew;
2529     format df 20.0;
2530     run;
2531
2532     data &in._mnew;
2533     set &in._mnew;
2534     format level1 $30.;
2535     format estimate 20.10;
2536     parameter = upcase(parameter);
2537     run;
2538
2539     proc sql noprint;
2540         select count(*) into :n_param
2541         from &in._mnew
2542         where df ^= 0;
2543     quit;
2544
2545     data &in._smnew;
2546     set &in._smnew;
2547     if _n_ = 5 then do;    /* 18.01.02.02 */
2548         output;
2549         criterion = 'AIC';
2550         value = (2 * &n_param) - 2 * value;
2551     end;    /* 18.01.02.02 */
2552     output;
2553     run;
2554
2555     %mod_b_meno_a(moda=&in._smnew, modb=&in._smcorr, alpha=&alfa, prefisso=&in);
2556
2557     proc sql noprint;
2558         select val into :meglio
2559         from &in._rit;
2560     quit;
2561
2562     proc delete data=&in._rit;
2563     run;
2564
2565     %if &stnew ^= 0 %then %do;
2566         %let meglio = 0;
2567     %end;
2568     %if &stcorr ^= 0 and &stnew = 0 %then %do;
2569         %let meglio = 1;
2570     %end;
2571
2572     %if &meglio = 1 %then %do;    /* 18.01.02.03 */
2573
2574         proc sql noprint;
2575             select value format=40.20 into :aic_new
2576             from &in._smnew
2577             where criterion = 'AIC';
2578         quit;
2579
2580         %let aic_new = &aic_new;
2581
2582         %if &nuovo = 0 %then %do;    /* 18.01.02.03.01 */
2583
```

```
2584     %let stbest = &stnew;
2585     %let nuovo = 1;
2586     %let aic_best = &aic_new;
2587     %let v_inserita = &&vin&i;
2588
2589     data &in._mbest;
2590     set &in._mnew;
2591     run;
2592
2593     data &in._smbest;
2594     set &in._smnew;
2595     run;
2596
2597     data &in._dbest;
2598     set &in._dnew;
2599     run;
2600
2601     %end;          /* 18.01.02.03.01 */
2602
2603 %else %do;      /* 18.01.02.03.02 */
2604
2605     %if %sysevalf(&aic_best > &aic_new) %then %do;          /* 18.01.02.03.02.01 */
2606
2607         %let stbest = &stnew;
2608         %let aic_best = &aic_new;
2609         %let v_inserita = &&vin&i;
2610
2611         data &in._mbest;
2612         set &in._mnew;
2613         run;
2614
2615         data &in._smbest;
2616         set &in._smnew;
2617         run;
2618
2619         data &in._dbest;
2620         set &in._dnew;
2621         run;
2622
2623         %end;          /* 18.01.02.03.02.01 */
2624
2625     %end;          /* 18.01.02.03.02 */
2626
2627 %end;          /* 18.01.02.03 */
2628
2629 %end; /* 18.01.02 */
2630
2631 %goloop:
2632
2633 %end; /* 18.01 */
2634
2635
2636 %if &nuovo = 0 %then %do;          /* 18.02 */
2637     %let attivo = %eval(&attivo + 1);
2638
2639     data &in._tesec;
2640     set &in._tesec;
2641     passoprec = ahora;
2642     ahora = datetime();
2643     delta = ahora - inizio;
2644     deltagg = int(delta/(60*60*24));
2645     deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
2646     deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
2647     deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
2648     deltadec = deltasec + (delta - int(delta));
2649     delta2 = ahora - passoprec;
2650     deltagg2 = int(delta2/(60*60*24));
2651     deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
2652     deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
2653     deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
2654     deltadec2 = deltasec2 + (delta2 - int(delta2));
2655     run;
```

```
2656
2657 proc sql noprint;
2658     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
2659     from &in._tesec;
2660
2661     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
2662     from &in._tesec;
2663 quit;
2664
2665 %let dg = &dg; %let dg2 = &dg2;
2666 %let do = &do; %let do2 = &do2;
2667 %let dm = &dm; %let dm2 = &dm2;
2668 %let ds = &ds; %let ds2 = &ds2;
2669 %let dd = &dd; %let dd2 = &dd2;
2670
2671 proc printto log=&log_0; run;
2672
2673 %put 08 - Nessuna variabile aggiunta al modello;
2674 %put Tempo trascorso;;
2675 %if &dg > 0 %then %do;
2676     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
2677 %end;
2678 %else %if &do > 0 %then %do;
2679     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
2680 %end;
2681 %else %if &dm > 0 %then %do;
2682     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
2683 %end;
2684 %else %do;
2685     %put &dd.s (&dd2.s);
2686 %end;
2687
2688 proc printto log=&output_log; run;
2689
2690 %if &attivo = 2 %then %goto finemegaciclo;
2691 %end; /* 18.02 */
2692
2693 %else %do; /* 18.03 */
2694     %let attivo = 0;
2695
2696     data &in._tesec;
2697     set &in._tesec;
2698     passoprec = ahora;
2699     ahora = datetime();
2700     delta = ahora - inizio;
2701     deltagg = int(delta/(60*60*24));
2702     deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
2703     deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
2704     deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
2705     deltadec = deltasec + (delta - int(delta));
2706     delta2 = ahora - passoprec;
2707     deltagg2 = int(delta2/(60*60*24));
2708     deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
2709     deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
2710     deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
2711     deltadec2 = deltasec2 + (delta2 - int(delta2));
2712 run;
2713
2714 proc sql noprint;
2715     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
2716     from &in._tesec;
2717
2718     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
2719     from &in._tesec;
2720 quit;
2721
2722 %let dg = &dg; %let dg2 = &dg2;
2723 %let do = &do; %let do2 = &do2;
2724 %let dm = &dm; %let dm2 = &dm2;
2725 %let ds = &ds; %let ds2 = &ds2;
2726 %let dd = &dd; %let dd2 = &dd2;
2727
```

```
2728 proc printto log=&log_0; run;
2729
2730 %put 08 - Variabile &v_inserita aggiunta al modello;
2731 %put Tempo trascorso;;
2732 %if &dg > 0 %then %do;
2733     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
2734 %end;
2735 %else %if &do > 0 %then %do;
2736     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
2737 %end;
2738 %else %if &dm > 0 %then %do;
2739     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
2740 %end;
2741 %else %do;
2742     %put &dd.s (&dd2.s);
2743 %end;
2744
2745 proc printto log=&output_log; run;
2746
2747 proc sql noprint;
2748     insert into &in._mod
2749     select nome, utilizzo
2750     from &in._po
2751     where nome = "&v_inserita";
2752 quit;
2753
2754 data &in._corr5;
2755 set &in._corr4;
2756 if v1 = "&v_inserita" or v2 = "&v_inserita";
2757 if corr2 ^= 0;
2758 if v1 = "&v_inserita" then var = v2;
2759 if v2 = "&v_inserita" then var = v1;
2760 keep var;
2761 run;
2762
2763 proc sort data=&in._corr5 nodup;
2764     by var;
2765 run;
2766
2767 proc sql noprint;
2768     insert into &in._corr5
2769     set var = "&v_inserita";
2770
2771     create table &in._po2 as
2772     select a.*, b.var
2773     from &in._po a left join &in._corr5 b on
2774     (a.nome = b.var);
2775 quit;
2776
2777 data &in._po;
2778 set &in._po2;
2779 if var ^= '' then po = 0;
2780 drop var;
2781 run;
2782
2783 proc delete data=&in._po2 &in._corr5;
2784 run;
2785
2786
2787 proc sql noprint;
2788     select utilizzo into :uinno
2789     from &in._mod
2790     where nome = "&v_inserita";
2791 quit;
2792
2793
2794 %if %upcase(&uinno) = X or %upcase(&uinno) = 0 %then %do; /* 18.03.01 */
2795
2796 data &in._nuoco;
2797 set &in._dbest;
2798 keep &v_inserita._b
2799     &v_inserita
```

```
2800         &v_inserita._c;
2801     run;
2802
2803     proc sort data=&in._nuoco nodup;
2804         by    &v_inserita._b
2805             &v_inserita
2806             &v_inserita._c;
2807     run;
2808
2809     proc sql noprint;
2810         create table &in._nuoco2 as
2811             select &v_inserita._c, min(&v_inserita._b) as minimo, max(&v_inserita._b) as massimo
2812             from &in._nuoco
2813             group by &v_inserita._c;
2814     quit;
2815
2816     data &in._nuoco2;
2817     set &in._nuoco2;
2818     if minimo ^= massimo;
2819     run;
2820
2821     %let nuori = 0;
2822
2823     proc sql noprint;
2824         select count(*) into :nuori
2825         from &in._nuoco2;
2826     quit;
2827
2828     %let nuori = &nuori;
2829
2830     proc sql noprint;
2831     select &v_inserita._c, minimo, massimo into :nuov1 - :nuov&nuori, :nuomi1
- :nuomi&nuori, :nuoma1 - :nuoma&nuori
2832     from &in._nuoco2;
2833     quit;
2834
2835     proc delete data=&in._nuoco2;
2836
2837     %do j=1 %to &nuori;    /* 18.03.01.01 */
2838     proc sql noprint;
2839         select &v_inserita into :minni
2840         from &in._nuoco
2841         where &v_inserita._b = &&nuomi&j;
2842
2843         select &v_inserita into :massi
2844         from &in._nuoco
2845         where &v_inserita._b = &&nuoma&j;
2846     quit;
2847
2848     %let minni = &minni;
2849     %let massi = &massi;
2850
2851     %if %upcase(&uinno) = X %then %do;    /* 18.03.01.01.01 */
2852         %let nucla&j =
2853             %qscan(&minni, 1, ' <=> ') < %qscan(&minni, 2, ' <=> ') <= %qscan(&massi, 3, ' <=> ');
2854         %end;    /* 18.03.01.01.01 */
2855     %else %if %upcase(&uinno) = 0 %then %do;    /* 18.03.01.01.02 */
2856         %let nucla&j = &minni <= &v_inserita <= &massi;
2857         %end;    /* 18.03.01.01.02 */
2858     %end; /* 18.03.01.01 */
2859
2860
2861     data &in._nuoco;
2862     set &in._nuoco;
2863     &v_inserita._d = &v_inserita;
2864     %do j=1 %to &nuori;    /* 18.03.01.02 */
2865         if &v_inserita._c = &&nuov&j then &v_inserita._d = "&&nucla&j";
2866     %end;    /* 18.03.01.02 */
2867     run;
2868
2869     proc sql noprint;
2870     create table &in._ptk as
```



```
2871     select a.*, b.&v_inserita._c, b.&v_inserita._d
2872     from &in._pt a left join &in._nuoco b on
2873         (a.&v_inserita._b = b.&v_inserita._b);
2874 quit;
2875
2876 data &in._pt;
2877 set &in._ptk;
2878 run;
2879
2880 proc delete data=&in._ptk;
2881 run;
2882
2883 data &in._nuoco;
2884 set &in._nuoco;
2885     format var $40.;
2886     var = "&v_inserita";
2887 run;
2888
2889 %if %sysfunc(exist(&in._kvar)) = 0 %then %do; /* 18.03.01.03 */
2890
2891     data &in._kvar;
2892     set &in._nuoco;
2893     giro = 1;
2894     kvar = &v_inserita;
2895     kvar_b = &v_inserita._b;
2896     kvar_c = &v_inserita._c;
2897     kvar_d = &v_inserita._d;
2898     keep var giro kvar kvar_b kvar_c kvar_d;
2899     run;
2900
2901 %end;          /* 18.03.01.03 */
2902
2903 %else %do;    /* 18.03.01.04 */
2904
2905     %let ggg = 0;
2906
2907     proc sql noprint;
2908         select max(giro) into :ggg
2909         from &in._kvar;
2910     quit;
2911
2912     data &in._nuoco2;
2913     set &in._nuoco;
2914     giro = sum(&ggg, 1);
2915     kvar = &v_inserita;
2916     kvar_b = &v_inserita._b;
2917     kvar_c = &v_inserita._c;
2918     kvar_d = &v_inserita._d;
2919     keep var giro kvar kvar_b kvar_c kvar_d;
2920     run;
2921
2922     data &in._kvar;
2923     set &in._kvar &in._nuoco2;
2924     run;
2925
2926     proc delete data=&in._nuoco2;
2927     run;
2928
2929 %end;          /* 18.03.01.04 */
2930
2931 %end; /* 18.03.01 */
2932
2933 %let stcorr = &stbest;
2934
2935 data &in._mcorr;
2936 set &in._mbest;
2937 run;
2938
2939 data &in._smcorr;
2940 set &in._smbest;
2941 run;
2942
```

```
2943 data &in._dcorr;
2944 set &in._dbest;
2945 run;
2946
2947 %let ggg = 0;
2948 %let noloop = ;
2949
2950 proc sql noprint;
2951   select max(passo) into :ggg
2952   from &in._passi;
2953
2954   select nome into :noloop separated by ' '
2955   from &in._mod
2956   order by nome;
2957 quit;
2958
2959 %let noloop = &noloop;
2960
2961 data &in._nnnnome;
2962   format modello &max_format.;
2963   modello = "&noloop";
2964   output;
2965 run;
2966
2967 data &in._nnnnome;
2968 set &in._nnnnome;
2969   passo = sum(&ggg, 1);
2970 run;
2971
2972 data &in._passi;
2973 set &in._passi &in._nnnnome;
2974 run;
2975
2976 proc delete data=&in._nnnnome;
2977 run;
2978
2979 %end; /* 18.03 */
2980
2981
2982 %let nuovo = 0;
2983 %let nin = 0;
2984
2985 proc sql noprint;
2986   select count(*) into :nin
2987   from &in._mod;
2988 quit;
2989
2990 %let nin = &nin;
2991
2992 proc sql noprint;
2993   select nome, utilizzo into :vin1 - :vin&nin, :uin1 - :uin&nin
2994   from &in._mod;
2995 quit;
2996
2997
2998 %do i=1 %to &nin; /* 18.04 */
2999
3000 ods listing close;
3001 ods output ParameterEstimates=&in._mnew ModelFit=&in._smnew ConvergenceStatus=&in._conew;
3002
3003 proc genmod data=&in._pt descending NAMELEN=50;
3004 class
3005   %do j=1 %to &nin; /* 18.04.01 */
3006     %if &j ^= &i %then %do; /* 18.04.01.01 */
3007       %if %upcase(&&uin&j) = C %then %do; /* 18.04.01.01.01a */
3008         &&vin&j
3009       %end; /* 18.04.01.01.01a */
3010       %if %upcase(&&uin&j) = X or %upcase(&&uin&j) = 0 %then %do; /* 18.04.01.01.01b */
3011         &&vin&j.._c
3012       %end; /* 18.04.01.01.01b */
3013     %end; /* 18.04.01.01 */
3014   %end; /* 18.04.01 */
```

```
3015 / missing ;
3016 model &v_risp =
3017 %do j=1 %to &nin; /* 18.04.02 */
3018 %if &j ^= &i %then %do; /* 18.04.02.01 */
3019 %if %upcase(&&uin&j) = X or %upcase(&&uin&j) = 0 %then %do; /* 18.04.02.01.01 */
3020 &&vin&j.._c
3021 %end; /* 18.04.02.01.01 */
3022 %else %do;
3023 &&vin&j
3024 %end;
3025 %end; /* 18.04.02.01 */
3026 %end; /* 18.04.02 */
3027 / dist = &distribuzione ;
3028 output out=&in._dnew predicted=predetti stdreschi=residui lower=inf upper=sup xbeta=xbet;
3029 run;
3030
3031 ods output close;
3032 ods listing;
3033
3034 proc sql noprint;
3035 select status into :stnew
3036 from &in._conew;
3037 quit;
3038
3039 %let stnew = &stnew;
3040
3041 proc delete data=&in._conew;
3042 run;
3043
3044 data &in._smnew;
3045 set &in._smnew;
3046 format df 20.0;
3047 run;
3048
3049 data &in._mnew;
3050 set &in._mnew;
3051 format level1 $30.;
3052 format estimate 20.10;
3053 parameter = upcase(parameter);
3054 run;
3055
3056 proc sql noprint;
3057 select count(*) into :n_param
3058 from &in._mnew
3059 where df ^= 0;
3060 quit;
3061
3062 data &in._smnew;
3063 set &in._smnew;
3064 if _n_ = 5 then do; /* 18.04.03 */
3065 output;
3066 criterion = 'AIC';
3067 value = (2 * &n_param) - 2 * value;
3068 end; /* 18.04.03 */
3069 output;
3070 run;
3071
3072 %mod_b_meno_a(moda=&in._smcorr, modb=&in._smnew, alpha=&alfa, prefisso=&in);
3073
3074 proc sql noprint;
3075 select val into :meglio
3076 from &in._rit;
3077 quit;
3078
3079 proc delete data=&in._rit;
3080 run;
3081
3082 %if &stnew ^= 0 %then %do;
3083 %let meglio = 0;
3084 %end;
3085 %if &stcorr ^= 0 and &stnew = 0 %then %do;
3086 %let meglio = 2;
```

```
3087 %end;
3088
3089 %if &meglio = 2 %then %do; /* 18.04.04 */
3090
3091 proc sql noprint;
3092 select value format=40.20 into :aic_new
3093 from &in._smnew
3094 where criterion = 'AIC';
3095 quit;
3096
3097 %let aic_new = &aic_new;
3098
3099 %if &nuovo = 0 %then %do; /* 18.04.04.01 */
3100
3101 %let stbest = &stnew;
3102 %let nuovo = 1;
3103 %let aic_best = &aic_new;
3104 %let v_inserita = &&vin&i;
3105
3106 data &in._mbest;
3107 set &in._mnew;
3108 run;
3109
3110 data &in._smbest;
3111 set &in._smnew;
3112 run;
3113
3114 data &in._dbest;
3115 set &in._dnew;
3116 run;
3117
3118 %end; /* 18.04.04.01 */
3119
3120 %else %do; /* 18.04.04.02 */
3121
3122 %if %sysevalf(&aic_best > &aic_new) %then %do; /* 18.04.04.02.01 */
3123
3124 %let stbest = &stnew;
3125 %let aic_best = &aic_new;
3126 %let v_inserita = &&vin&i;
3127
3128 data &in._mbest;
3129 set &in._mnew;
3130 run;
3131
3132 data &in._smbest;
3133 set &in._smnew;
3134 run;
3135
3136 data &in._dbest;
3137 set &in._dnew;
3138 run;
3139
3140 %end; /* 18.04.04.02.01 */
3141
3142 %end; /* 18.04.04.02 */
3143
3144 %end; /* 18.04.04 */
3145
3146 %end; /* 18.04 */
3147
3148 %if &nuovo = 0 %then %do; /* 18.05 */
3149 %let attivo = %eval(&attivo + 1);
3150
3151 data &in._tesec;
3152 set &in._tesec;
3153 passoprec = ahora;
3154 ahora = datetime();
3155 delta = ahora - inizio;
3156 deltagg = int(delta/(60*60*24));
3157 deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
3158 deltamain = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
```

```
3159     deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
3160     deltadec = deltasec + (delta - int(delta));
3161     delta2 = ahora - passoprec;
3162     deltagg2 = int(delta2/(60*60*24));
3163     deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
3164     deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
3165     deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
3166     deltadec2 = deltasec2 + (delta2 - int(delta2));
3167     run;
3168
3169 proc sql noprint;
3170     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
3171     from &in._tesec;
3172
3173     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
3174     from &in._tesec;
3175 quit;
3176
3177 %let dg = &dg; %let dg2 = &dg2;
3178 %let do = &do; %let do2 = &do2;
3179 %let dm = &dm; %let dm2 = &dm2;
3180 %let ds = &ds; %let ds2 = &ds2;
3181 %let dd = &dd; %let dd2 = &dd2;
3182
3183 proc printto log=&log_0; run;
3184
3185 %put 10 - Nessuna variabile eliminata dal modello;
3186 %put Tempo trascorso;;
3187 %if &dg > 0 %then %do;
3188     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
3189 %end;
3190 %else %if &do > 0 %then %do;
3191     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
3192 %end;
3193 %else %if &dm > 0 %then %do;
3194     %put &dm.m:&dd.s (&dm2.m:&dd2.s);
3195 %end;
3196 %else %do;
3197     %put &dd.s (&dd2.s);
3198 %end;
3199
3200 proc printto log=&output_log; run;
3201
3202 %if &attivo = 2 %then %goto finemegaciclo;
3203 %end; /* 18.05 */
3204
3205 %else %do; /* 18.06 */
3206     %let attivo = 0;
3207     %let nuovo = 0;
3208     %let vin1 = 0;
3209     %let stcorr = &stbest;
3210
3211     data &in._mcorr;
3212     set &in._mbest;
3213     run;
3214
3215     data &in._smcorr;
3216     set &in._smbest;
3217     run;
3218
3219     data &in._dcorr;
3220     set &in._dbest;
3221     run;
3222
3223
3224     data &in._mod;
3225     set &in._mod;
3226     if nome = "&v_inserita" then delete;
3227     run;
3228
3229 %let ggg = 0;
3230 %let noloop = ;
```

```
3231
3232 proc sql noprint;
3233     select max(passo) into :ggg
3234     from &in._passi;
3235
3236     select nome into :nolooop separated by ' '
3237     from &in._mod
3238     order by nome;
3239 quit;
3240
3241 %let nolooop = &nolooop;
3242
3243 data &in._nnnnome;
3244     format modello &max_format.;
3245     modello = "&nolooop";
3246     output;
3247 run;
3248
3249 data &in._nnnnome;
3250 set &in._nnnnome;
3251     passo = sum(&ggg, 1);
3252 run;
3253
3254 data &in._passi;
3255 set &in._passi &in._nnnnome;
3256 run;
3257
3258 proc delete data=&in._nnnnome;
3259 run;
3260
3261 data &in._tesec;
3262 set &in._tesec;
3263     passoprec = ahora;
3264     ahora = datetime();
3265     delta = ahora - inizio;
3266     deltagg = int(delta/(60*60*24));
3267     deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
3268     deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
3269     deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
3270     deltadec = deltasec + (delta - int(delta));
3271     delta2 = ahora - passoprec;
3272     deltagg2 = int(delta2/(60*60*24));
3273     deltaore2 = int((delta2 - deltagg2*(60*60*24)) / (60*60));
3274     deltamin2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60)) / (60));
3275     deltasec2 = int((delta2 - deltagg2*(60*60*24) - deltaore2*(60*60) - deltamin2*60) / (1));
3276     deltadec2 = deltasec2 + (delta2 - int(delta2));
3277 run;
3278
3279 proc sql noprint;
3280     select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
3281     from &in._tesec;
3282
3283     select deltagg2, deltaore2, deltamin2, deltasec2, deltadec2 into :dg2, :do2, :dm2, :ds2, :dd2
3284     from &in._tesec;
3285 quit;
3286
3287 %let dg = &dg; %let dg2 = &dg2;
3288 %let do = &do; %let do2 = &do2;
3289 %let dm = &dm; %let dm2 = &dm2;
3290 %let ds = &ds; %let ds2 = &ds2;
3291 %let dd = &dd; %let dd2 = &dd2;
3292
3293 proc printto log=&log_0; run;
3294
3295 %put 10 - Variabile &v_inserita eliminata dal modello;
3296 %put Tempo trascorso;;
3297 %if &dg > 0 %then %do;
3298     %put &dg GG &do.h:&dm.m:&dd.s (&dg2 GG &do2.h:&dm2.m:&dd2.s);
3299 %end;
3300 %else %if &do > 0 %then %do;
3301     %put &do.h:&dm.m:&dd.s (&do2.h:&dm2.m:&dd2.s);
3302 %end;
```

```
3303     %else %if &dm > 0 %then %do;
3304         %put &dm.m:&dd.s (&dm2.m:&dd2.s);
3305     %end;
3306     %else %do;
3307         %put &dd.s (&dd2.s);
3308     %end;
3309
3310 proc printto log=&output_log; run;
3311
3312 proc sql noprint;
3313     select nome into :vin1 - :vin%eval(&nin-1)
3314     from &in._mod;
3315 quit;
3316
3317 %if %symexist(vin1) = 0 or &vin1 = 0 %then %do;           /* 18.06.01 */
3318     data &in._po;
3319     set &in._po;
3320     po = 1;
3321     run;
3322 %end; /* 18.06.01 */
3323
3324 %else %do; /* 18.06.02 */
3325     data &in._corr5;
3326     set &in._corr4;
3327     if v1 in (
3328         %do j=1 %to %eval(&nin-1);           /* 18.06.02.01 */
3329             "&&vin&j"
3330         %end; /* 18.06.02.01 */
3331     ) or
3332     v2 in (
3333         %do j=1 %to %eval(&nin-1);           /* 18.06.02.02 */
3334             "&&vin&j"
3335         %end; /* 18.06.02.02 */
3336     );
3337     if corr2 ^= 0;
3338     keep v1 v2;
3339     run;
3340
3341     data &in._corr6;
3342     set &in._corr5;
3343     var = v1;
3344     output;
3345     var = v2;
3346     output;
3347     keep var;
3348     run;
3349
3350 proc sql noprint;
3351     %do j=1 %to %eval(&nin-1); /* 18.06.02.03 */
3352         insert into &in._corr6 values ("&&vin&j");
3353     %end; /* 18.06.02.03 */
3354 quit;
3355
3356 proc sort data=&in._corr6 nodup;
3357     by var;
3358     run;
3359
3360 proc sql noprint;
3361     create table &in._po2 as
3362     select a.*, b.var
3363     from &in._po a left join &in._corr6 b on
3364     (a.nome = b.var);
3365 quit;
3366
3367 data &in._po;
3368 set &in._po2;
3369 if var ^= '' then po = 0;
3370 else po = 1;
3371 drop var;
3372 run;
3373
3374 proc delete data=&in._corr5 &in._corr6 &in._po2;
```

```
3375     run;
3376
3377     %end; /* 18.06.02 */
3378
3379     proc sql noprint;
3380         select utilizzo into :uinno
3381             from &in._po
3382             where nome = "&v_inserita";
3383     quit;
3384
3385     %if %upcase(&uinno) = X or %upcase(&uinno) = 0 %then %do; /* 18.06.03 */
3386         data &in._pt;
3387         set &in._pt;
3388         drop &v_inserita._c &v_inserita._d;
3389     run;
3390     %end; /* 18.06.03 */
3391
3392 %end; /* 18.06 */
3393
3394 %let vold_b = ;
3395 %let vold_c = ;
3396 %let vcold_b = ;
3397 %let vcold_c = ;
3398
3399 proc sql noprint;
3400     select nome into :vold_b separated by ' '
3401         from &in._mod
3402         where utilizzo not in ('X' '0');
3403
3404     select compress(nome)||'_c' into :vold_c separated by ' '
3405         from &in._mod
3406         where utilizzo in ('X' '0');
3407
3408     select nome into :vcold_b separated by ' '
3409         from &in._mod
3410         where utilizzo in ('C');
3411
3412     select compress(nome)||'_c' into :vcold_c separated by ' '
3413         from &in._mod
3414         where utilizzo in ('X' '0');
3415
3416     select count(*) into :nin
3417         from &in._po
3418         where po = 1;
3419     quit;
3420
3421 %let vold = &vold_b &vold_c;
3422 %let vcold = &vcold_b &vcold_c;
3423
3424 %let nin = &nin;
3425
3426 proc sql noprint;
3427     select nome, utilizzo into :vin1 - :vin&nin, :uin1 - :uin&nin
3428         from &in._po
3429         where po = 1;
3430     quit;
3431
3432 %finemegaciclo:
3433
3434 %if &max_g < -10 %then %do; /* 18.07 */
3435 %end; /* 18.07 */
3436 %else %do; /* 18.08 */
3437     %let max_g = %eval(&max_g-1);
3438
3439     %if max_g <= 0 %then %do; /* 18.08.01 */
3440         %let attivo = 2;
3441         %end; /* 18.08.01 */
3442     %end; /* 18.08 */
3443
3444 %end; /* 18 */
3445
3446 %let vold_b = ;
```



```
3447 %let vold_c = ;
3448
3449 proc sql noprint;
3450   select nome into :vold_b separated by ', '
3451   from &in._mod
3452   where utilizzo not in ('X' '0');
3453
3454   select compress(nome)||'_c' into :vold_c separated by ', '
3455   from &in._mod
3456   where utilizzo in ('X' '0');
3457 quit;
3458
3459 proc printto log=&log_0; run;
3460 %put ;
3461 %put ll - Modello completo. Variabili inserite: ;
3462 %put @@@ &vold_b , &vold_c @@@;
3463 %if &stcorr ^= 0 %then %do;
3464   %put @@@ ATTENZIONE: SEMBRA NON ESSERCI CONVERGENZA DELL ALGORITMO PER IL MODELLO TROVATO @@@;
3465 %end;
3466 %put Tabelle di riferimento;;
3467 %put &in._cond (Condizioni imposte per le classi);
3468 %put &in._corr4 (Correlazioni tra le variabili);
3469 %put &in._dcorr (Dati e stime per il modello);
3470 %put &in._escon (Correlazioni imposte tra le variabili);
3471 %put &in._kvar (Classi individuate per le variabili di modello);
3472 %put &in._mcorr (Stima dei parametri per il modello);
3473 %put &in._mod (variabili di modello);
3474 %put &in._passi (Descrizione flusso stepwise);
3475 %put &in._po (Elenco variabili potenziali);
3476 %put &in._pt (Dataset Post Trattamento);
3477 %put &in._smcorr (Valori riassuntivi per il modello);
3478 %put &in._var (Variabili nel dataset di input);
3479 %put &in._zgri (Stime dei parametri del modello);
3480 proc printto log=&output_log; run;
3481
3482
3483 data &in._tesec;
3484 set &in._tesec;
3485   fine = datetime();
3486   delta = fine - inizio;
3487   deltagg = int(delta/(60*60*24));
3488   deltaore = int((delta - deltagg*(60*60*24)) / (60*60));
3489   deltamin = int((delta - deltagg*(60*60*24) - deltaore*(60*60)) / (60));
3490   deltasec = int((delta - deltagg*(60*60*24) - deltaore*(60*60) - deltamin*60) / (1));
3491   deltadec = deltasec + (delta - int(delta));
3492 run;
3493
3494 proc sql noprint;
3495   select deltagg, deltaore, deltamin, deltasec, deltadec into :dg, :do, :dm, :ds, :dd
3496   from &in._tesec;
3497 quit;
3498
3499 %let dg = &dg;
3500 %let do = &do;
3501 %let dm = &dm;
3502 %let ds = &ds;
3503 %let dd = &dd;
3504
3505
3506 proc delete data=&in._tesec; run;
3507
3508 %if %sysfunc(exist(&in._dbest)) ^= 0 %then %do; /* 19 */
3509   proc delete data=&in._dbest; run;
3510   %end; /* 19 */
3511 %if %sysfunc(exist(&in._dbestk)) ^= 0 %then %do; /* 20 */
3512   proc delete data=&in._dbestk; run;
3513   %end; /* 20 */
3514 %if %sysfunc(exist(&in._dnew)) ^= 0 %then %do; /* 21 */
3515   proc delete data=&in._dnew; run;
3516   %end; /* 21 */
3517 %if %sysfunc(exist(&in._dnewk)) ^= 0 %then %do; /* 22 */
3518   proc delete data=&in._dnewk; run;
```

```
3519         %end; /* 22 */
3520 %if %sysfunc(exist(&in._k2)) ^= 0 %then %do; /* 23 */
3521     proc delete data=&in._k2; run;
3522     %end; /* 23 */
3523 %if %sysfunc(exist(&in._k3)) ^= 0 %then %do; /* 24 */
3524     proc delete data=&in._k3; run;
3525     %end; /* 24 */
3526 %if %sysfunc(exist(&in._k4)) ^= 0 %then %do; /* 25 */
3527     proc delete data=&in._k4; run;
3528     %end; /* 25 */
3529 %if %sysfunc(exist(&in._mbest)) ^= 0 %then %do; /* 26 */
3530     proc delete data=&in._mbest; run;
3531     %end; /* 26 */
3532 %if %sysfunc(exist(&in._mbestk)) ^= 0 %then %do; /* 27 */
3533     proc delete data=&in._mbestk; run;
3534     %end; /* 27 */
3535 %if %sysfunc(exist(&in._mnew)) ^= 0 %then %do; /* 28 */
3536     proc delete data=&in._mnew; run;
3537     %end; /* 28 */
3538 %if %sysfunc(exist(&in._mnewk)) ^= 0 %then %do; /* 29 */
3539     proc delete data=&in._mnewk; run;
3540     %end; /* 29 */
3541 %if %sysfunc(exist(&in._nuoco)) ^= 0 %then %do; /* 30 */
3542     proc delete data=&in._nuoco; run;
3543     %end; /* 30 */
3544 %if %sysfunc(exist(&in._smbest)) ^= 0 %then %do; /* 31 */
3545     proc delete data=&in._smbest; run;
3546     %end; /* 31 */
3547 %if %sysfunc(exist(&in._smbestk)) ^= 0 %then %do; /* 32 */
3548     proc delete data=&in._smbestk; run;
3549     %end; /* 32 */
3550 %if %sysfunc(exist(&in._smbestk_dev)) ^= 0 %then %do; /* 33 */
3551     proc delete data=&in._smbestk_dev; run;
3552     %end; /* 33 */
3553 %if %sysfunc(exist(&in._smcorr_dev)) ^= 0 %then %do; /* 34 */
3554     proc delete data=&in._smcorr_dev; run;
3555     %end; /* 34 */
3556 %if %sysfunc(exist(&in._smnew)) ^= 0 %then %do; /* 35 */
3557     proc delete data=&in._smnew; run;
3558     %end; /* 35 */
3559 %if %sysfunc(exist(&in._smnew_dev)) ^= 0 %then %do; /* 36 */
3560     proc delete data=&in._smnew_dev; run;
3561     %end; /* 36 */
3562 %if %sysfunc(exist(&in._smnewk)) ^= 0 %then %do; /* 37 */
3563     proc delete data=&in._smnewk; run;
3564     %end; /* 37 */
3565 %if %sysfunc(exist(&in._ptk)) ^= 0 %then %do; /* 38 */
3566     proc delete data=&in._ptk; run;
3567     %end; /* 38 */
3568 %if %sysfunc(exist(&in._temp)) ^= 0 %then %do; /* 38b */
3569     proc delete data=&in._temp; run;
3570     %end; /* 38b */
3571 %if %sysfunc(exist(&in._corrp)) ^= 0 %then %do; /* 38c */
3572     proc delete data=&in._corrp; run;
3573     %end; /* 38c */
3574 %if %sysfunc(exist(&in._corrs)) ^= 0 %then %do; /* 38d */
3575     proc delete data=&in._corrs; run;
3576     %end; /* 38d */
3577 %if %sysfunc(exist(&in._corr2)) ^= 0 %then %do; /* 38e */
3578     proc delete data=&in._corr2; run;
3579     %end; /* 38e */
3580 %if %sysfunc(exist(&in._corr3)) ^= 0 %then %do; /* 38f */
3581     proc delete data=&in._corr3; run;
3582     %end; /* 38f */
3583 %if %sysfunc(exist(&in._corr3b)) ^= 0 %then %do; /* 38g */
3584     proc delete data=&in._corr3b; run;
3585     %end; /* 38g */
3586 %if %sysfunc(exist(&in._pt_temp)) ^= 0 %then %do; /* 38h */
3587     proc delete data=&in._pt_temp; run;
3588     %end; /* 38h */
3589
3590 data &in._modk &in._modc;
```

```
3591 set &in._mod;
3592 if utilizzo in ('0' 'X') then output &in._modk;
3593 if utilizzo in ('Q' 'C') then output &in._modc;
3594 run;
3595
3596 data &in._modc;
3597 set &in._modc;
3598 format Condizione &max_format.;
3599 format kvar_d &max_format.;
3600 kvar_c = .;
3601 kvar_d = '';
3602 condizione = '';
3603 run;
3604
3605 %if %sysfunc(exist(&in._kvar)) ^= 0 %then %do; /* 39 */
3606 proc sql noprint;
3607 create table &in._um as
3608 select var, max(giro) as max_giro
3609 from &in._kvar
3610 group by var;
3611 quit;
3612
3613 proc sql noprint;
3614 create table &in._umk as
3615 select a.nome, a.utilizzo, b.max_giro
3616 from &in._modk a join &in._um b on
3617 (a.nome = b.var);
3618 quit;
3619
3620 proc sql noprint;
3621 create table &in._modk as
3622 select a.nome, a.utilizzo, b.kvar_c, b.kvar_d
3623 from &in._umk a join &in._kvar b on
3624 (a.nome = b.var and
3625 a.max_giro = b.giro);
3626 quit;
3627
3628 data &in._modk1 &in._modk2;
3629 set &in._modk;
3630 if kvar_c <= 0 then output &in._modk1;
3631 if kvar_c > 0 then output &in._modk2;
3632 run;
3633
3634 proc sql noprint;
3635 create table &in._condu as
3636 select a.*, b.utilizzo
3637 from &in._cond a left join &in._var b on
3638 (a.variabile = b.name);
3639 quit;
3640
3641 data &in._condu;
3642 set &in._condu;
3643 format nome $100.;
3644 if utilizzo in ('X' '0') then nome = cats('CL_', variabile);
3645 run;
3646
3647 proc sql noprint;
3648 create table &in._modk1c as
3649 select a.nome, a.utilizzo, a.kvar_c, a.kvar_d, b.condizione
3650 from &in._modk1 a join &in._condu b on
3651 (a.nome = b.nome and
3652 a.kvar_d = b.classe);
3653 quit;
3654
3655 data &in._modk2;
3656 set &in._modk2;
3657 condizione = kvar_d;
3658 run;
3659
3660 data &in._pgrgl;
3661 set &in._modk1c &in._modc &in._modk2;
3662 run;
```

```
3663
3664   proc delete data=&in._um &in._modk &in._modc &in._umk &in._modk1 &in._modk2 &in._modk1c
&in._condu;
3665   run;
3666 %end;   /* 39 */
3667
3668 %else %do;   /* 40 */
3669   data &in._pgrgl;
3670   set &in._modc;
3671   run;
3672
3673   proc delete data=&in._modk &in._modc;
3674   run;
3675 %end;   /* 40 */
3676
3677 proc sort data=&in._pgrgl nodup;
3678   by nome utilizzo kvar_c;
3679 run;
3680
3681 data &in._pgrgl;
3682 set &in._pgrgl;
3683   if utilizzo = '0' or utilizzo = 'X' then nome = cats(nome, '_C');
3684   kvar_cc = put(kvar_c, 20.);
3685   kvar_cc = compress(kvar_cc);
3686 run;
3687
3688 data &in._zgr0;
3689 set &in._mcorr;
3690   if _N_ = 1;
3691   nome = parameter;
3692   kvar_d = '';
3693   condizione = '1';
3694   utilizzo = '';
3695   kvar_c = .;
3696   keep nome kvar_d estimate condizione df levell1 utilizzo kvar_c;
3697 run;
3698
3699 proc sql noprint;
3700   create table &in._zgr1 as
3701     select a.nome, b.levell1, a.kvar_d, b.estimate, a.condizione, b.df, a.utilizzo, a.kvar_c
3702     from (select *
3703           from &in._pgrgl
3704           where utilizzo = '0' or utilizzo = 'X') a left join &in._mcorr b on
3705           (a.nome = b.parameter and a.kvar_cc = b.levell1);
3706 quit;
3707
3708 proc sql noprint;
3709   create table &in._zgr2 as
3710     select a.nome, b.levell1, a.kvar_d, b.estimate, a.condizione, b.df, a.utilizzo, a.kvar_c
3711     from (select *
3712           from &in._pgrgl
3713           where utilizzo = 'C') a left join &in._mcorr b on
3714           (a.nome = b.parameter);
3715 quit;
3716
3717 proc sql noprint;
3718   create table &in._zgr3 as
3719     select a.nome, b.levell1, a.kvar_d, b.estimate, a.condizione, b.df, a.utilizzo, a.kvar_c
3720     from (select *
3721           from &in._pgrgl
3722           where utilizzo = 'Q') a left join &in._mcorr b on
3723           (a.nome = b.parameter);
3724 quit;
3725
3726
3727 data &in._zgr1;
3728 set &in._zgr1 &in._zgr0 &in._zgr2 &in._zgr3;
3729 run;
3730
3731
3732 proc delete data=&in._zgr0 &in._zgr1 &in._zgr2 &in._zgr3 &in._pgrgl;
3733 run;
```

```
3734
3735
3736 proc sort data=&in._zgri nodup;
3737   by nome levell1;
3738 run;
3739
3740
3741 proc sql noprint;
3742   create table &in._zgri2 as
3743     select a.*, b.type
3744     from &in._zgri a left join &in._var b on
3745       (a.nome = b.name);
3746 quit;
3747
3748 data &in._zgri;
3749 set &in._zgri2;
3750 condizione = strip(condizione);
3751 if kvar_c >= 0 or kvar_c = . then do; /* 40b */
3752   if utilizzo = 'X' then do; /* 41 */
3753     mm = find(condizione, 'min <', 'i');
3754     if mm = 1 then do; /* 41.01 */
3755       condizione = substr(condizione, mm + 6);
3756     end; /* 41.01 */
3757     mm = find(condizione, ' max', 'i');
3758     if mm > 0 and (mm + 4 > length(condizione)) then do; /* 41.02 */
3759       condizione = substr(condizione, 1, mm-4);
3760     end; /* 41.02 */
3761   end; /* 41 */
3762   if utilizzo = '0' then do; /* 42 */
3763     mm = find(condizione, '<=', 'i');
3764     if mm > 0 then do; /* 42.02 */
3765       condizione = scan(condizione, 1, ' <=> ') || ' <=' || compress(substr(scan(condizione,
2, ' <=> '), 4)) || ' <=' || scan(condizione, 3, ' <=> ');
3766     end; /* 42.02 */
3767     else do; /* 42.03 */
3768       mm = length(nome);
3769       condizione = compress(substr(compress(nome), 4, mm-4-1)) || '=' || compress(kvar_d);
3770     end; /* 42.03 */
3771   end; /* 42 */
3772   if utilizzo = 'C' then do; /* 43 */
3773     if type = 1 then condizione = compress(nome) || '=' || compress(levell1);
3774     if type = 2 then condizione = compress(nome) || '=' || compress(levell1) || "";
3775   end; /* 43 */
3776 end; /* 40b */
3777 if utilizzo = '0' or utilizzo = 'X' then do; /* 44 */
3778   mm = length(nome);
3779   nome = compress(substr(compress(nome), 4, mm-4-1));
3780 end; /* 44 */
3781 drop type mm kvar_c;
3782 run;
3783
3784 proc delete data=&in._zgri2;
3785 run;
3786
3787 data &in._vak;
3788 set &in._var;
3789 if utilizzo = 'K';
3790 name = 'K_' || compress(name);
3791 name = compress(name);
3792 run;
3793
3794 proc sql noprint;
3795   select count(*) into :nukero
3796   from &in._vak;
3797 quit;
3798
3799 %let nukero = &nukero;
3800
3801 %if &nukero <= 0 %then %do; /* 45 */
3802   %goto finerik;
3803 %end; /* 45 */
3804
```

```
3805 proc sql noprint;
3806   create table &in._vak2 as
3807     select a.type, b.*
3808     from &in._vak a inner join &in._zgri b on
3809       (a.name = b.nome);
3810 quit;
3811
3812 proc sql noprint;
3813   select count(distinct nome) into :nukero
3814   from &in._vak2;
3815 quit;
3816
3817 %let nukero = &nukero;
3818
3819 %if &nukero <= 0 %then %do;      /* 46 */
3820   proc delete data=&in._vak2;
3821   run;
3822
3823   %goto finerik;
3824 %end; /* 46 */
3825
3826 proc sql noprint;
3827   select distinct nome into :kul - :ku&nukero
3828   from &in._vak2;
3829 quit;
3830
3831 %do j=1 %to &nukero;      /* 47 */
3832
3833   %let ku&j = &&ku&j;
3834
3835   data &in._vak3;
3836   set &in._vak2;
3837   if nome = "&&ku&j";
3838   nome_old = substr(nome, 3);
3839   run;
3840
3841   proc sort data=&in._vak3;
3842     by descending level1;
3843   run;
3844
3845   proc sql noprint;
3846     select count(*) into :nuker
3847     from &in._vak3;
3848   quit;
3849
3850   %let nuker = &nuker;
3851
3852   proc sql noprint;
3853     select type, level1, estimate, condizione, df, nome_old into
3854       :ktype1 - :ktype&nuker, :klev1 - :klev&nuker, :kest1 - :kest&nuker,
3855       :kcondi1 - :kcondi&nuker, :kdf1 - :kdf&nuker, :kuo1 - :kuo&nuker
3856     from &in._vak3;
3857   quit;
3858
3859   data &in._kcl2;
3860   set &in._kcl;
3861   if compress('K_' || compress(var_orig)) = "&&ku&j";
3862   &&ku&j = cl_nuova;
3863   run;
3864
3865   data &in._kcl2;
3866   set &in._kcl2;
3867   format level1 $30.;
3868   %do jj=1 %to &nuker; /* 47.01 */
3869     if &&kcondi&jj then do; /* 47.02 */
3870       level1 = compress("&&klev&jj");
3871     end; /* 47.02 */
3872   %end; /* 47.01 */
3873   run;
3874
3875   %do jj=1 %to &nuker; /* 47.03 */
3876     %let ktype&jj = &&ktype&jj;
```

```
3877 %let klev&jj = &&klev&jj;
3878 %let kest&jj = &&kest&jj;
3879 %let kcondi&jj = &&kcondi&jj;
3880 %let kdf&jj = &&kdf&jj;
3881 %let kuo&jj = &&kuo&jj;
3882
3883 %if &&ktype&jj = 1 %then %do; /* 47.03.01 */
3884
3885     proc sql noprint;
3886         select cl_orig into :k2cond&jj separated by ', '
3887         from &in._kcl2
3888         where compress(level1) = compress("&&klev&jj");
3889     quit;
3890
3891 %end; /* 47.03.01 */
3892
3893 %else %if &&ktype&jj = 2 %then %do; /* 47.03.02 */
3894
3895     proc sql noprint;
3896         select "" || compress(cl_orig) || "" into :k2cond&jj separated by ', '
3897         from &in._kcl2
3898         where compress(level1) = compress("&&klev&jj");
3899     quit;
3900
3901 %end; /* 47.03.02 */
3902
3903 %let k2cond&jj = &&kuo&jj in ( &&k2cond&jj );
3904
3905 %end; /* 47.03 */
3906
3907 data &in._zgri;
3908 set &in._zgri;
3909 if compress(nome) ^= compress("&&ku&jj");
3910 run;
3911
3912 proc sql noprint;
3913 %do jj=1 %to &nuker; /* 47.04 */
3914     insert into &in._zgri values ("&&kuo&jj", "&&klev&jj", "", &&kest&jj, "&&k2cond&jj",
&&kdf&jj, "K");
3915 %end; /* 47.04 */
3916 quit;
3917
3918 proc delete data=&in._vak3 &in._kcl2;
3919 run;
3920
3921 %end; /* 47 */
3922
3923 proc delete data=&in._vak2;
3924 run;
3925
3926 %finerik:
3927
3928 proc delete data=&in._vak;
3929 run;
3930
3931 proc sort data=&in._zgri;
3932 by nome descending level1;
3933 run;
3934
3935 %theend:
3936
3937 proc printto log=&log_0; run;
3938
3939 %put Tempo di esecuzione totale;;
3940 %if &dg > 0 %then %do;
3941     %put &dg GG &do.h:&dm.m:&ds.s.&dd;
3942 %end;
3943 %else %if &do > 0 %then %do;
3944     %put &do.h:&dm.m:&ds.s.&dd;
3945 %end;
3946 %else %if &dm > 0 %then %do;
3947     %put &dm.m:&ds.s.&dd;
```

```
3948     %end;
3949     %else %do;
3950         %put &dd.s;
3951     %end;
3952
3953 %put &versione_dr;
3954
3955 proc printto log=log; run;
3956
3957 %mend classizz;
3958
3959
3960
3961
```