TP 4 marked survival

Olivier Gimenez

19/12/2020

On charge les packages `RMark` et `R2ucare`.

```
library(RMark)
library(R2ucare)
```

Partie 1 : Estimation de la survie, exemple du cincle plongeur

Les données

```
cincle <- convert.inp("dat/cincle-plongeur.inp")
head(cincle)
```

```
##        ch freq
## 1 0000010   23
## 2 0000011   23
## 3 0000100   16
## 4 0000110    9
## 5 0000111   16
## 6 0001000   16
```

Process the data

```
cincle.proc <- process.data(cincle,
                            begin.time = 1,
                            model = "CJS")
```

Create default design data

```
cincle.ddl <- make.design.data(cincle.proc)
```

Examine the design data

```
head(cincle.ddl$Phi)
```

```
##   par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1         1           1     1      1   0    1          1      0   0    0
## 2         2           2     1      1   1    2          1      0   1    1
## 3         3           3     1      1   2    3          1      0   2    2
## 4         4           4     1      1   3    4          1      0   3    3
## 5         5           5     1      1   4    5          1      0   4    4
## 6         6           6     1      1   5    6          1      0   5    5
```

```r
head(cincle.ddl$p)
```

```
##   par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1         1          22     1      1   1    2          1      0   1    0
## 2         2          23     1      1   2    3          1      0   2    1
## 3         3          24     1      1   3    4          1      0   3    2
## 4         4          25     1      1   4    5          1      0   4    3
## 5         5          26     1      1   5    6          1      0   5    4
## 6         6          27     1      1   6    7          1      0   6    5
```

On spécifie les effets sur les paramètres.

```r
phit <- list(formula=~time)
phi <- list(formula=~1)
pt <- list(formula=~time)
p <- list(formula=~1)
```

Fait tourner modèle CJS.

```r
cjs.cincle <- mark(cincle.proc,
                   cincle.ddl,
                   model.parameters = list(Phi = phit, p = pt))
```

```
##
## Output summary for CJS model
## Name : Phi(~time)p(~time)
##
## Npar :  12  (unadjusted=11)
## -2lnL:  656.9502
## AICc :  681.7057  (unadjusted=679.58789)
##
## Beta
##                    estimate          se          lcl         ucl
## Phi:(Intercept)  0.9354620   0.7685231   -0.5708434   2.4417673
## Phi:time2       -1.1982814   0.8706706   -2.9047958   0.5082331
## Phi:time3       -1.0228358   0.8049151   -2.6004694   0.5547979
## Phi:time4       -0.4198647   0.8091494   -2.0057977   1.1660682
## Phi:time5       -0.5361039   0.8031440   -2.1102663   1.0380584
## Phi:time6        0.2481383 345.6553900 -677.2364400 677.7327200
## p:(Intercept)    0.8292773   0.7837326   -0.7068387   2.3653932
## p:time3          1.6556293   1.2913761   -0.8754678   4.1867264
## p:time4          1.5220981   1.0729122   -0.5808098   3.6250060
## p:time5          1.3767466   0.9884788   -0.5606719   3.3141650
## p:time6          1.7950963   1.0688750   -0.2998987   3.8900914
## p:time7         -0.0147556 263.9821100 -517.4196900 517.3901800
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6
## 1 0.7181821 0.4346708 0.4781704 0.6261177 0.5985334 0.7655945
## 2           0.4346708 0.4781704 0.6261177 0.5985334 0.7655945
```

```
## 3                     0.4781704 0.6261177 0.5985334 0.7655945
## 4                               0.6261177 0.5985334 0.7655945
## 5                                         0.5985334 0.7655945
## 6                                                   0.7655945
##
##
## Real Parameter p
##
##           2         3         4         5         6         7
## 1 0.6962021 0.9230769 0.9130435 0.9007892 0.9324138 0.6930722
## 2           0.9230769 0.9130435 0.9007892 0.9324138 0.6930722
## 3                     0.9130435 0.9007892 0.9324138 0.6930722
## 4                               0.9007892 0.9324138 0.6930722
## 5                                         0.9324138 0.6930722
## 6                                                   0.6930722
```

```
cjs.cincle$results$real
```

```
##                   estimate         se           lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.7181821  0.1555465  3.610422e-01 0.9199573
## Phi g1 c1 a1 t2 0.4346708  0.0688290  3.075047e-01 0.5710587
## Phi g1 c1 a2 t3 0.4781704  0.0597091  3.643838e-01 0.5942685
## Phi g1 c1 a3 t4 0.6261177  0.0592656  5.048461e-01 0.7333741
## Phi g1 c1 a4 t5 0.5985334  0.0560517  4.855434e-01 0.7019411
## Phi g1 c1 a5 t6 0.7655945 62.0312990 1.930009e-294 1.0000000
## p g1 c1 a1 t2   0.6962021  0.1657632  3.302978e-01 0.9141500
## p g1 c1 a2 t3   0.9230769  0.0728777  6.161499e-01 0.9889758
## p g1 c1 a3 t4   0.9130435  0.0581758  7.140650e-01 0.9778505
## p g1 c1 a4 t5   0.9007892  0.0538330  7.360176e-01 0.9672855
## p g1 c1 a5 t6   0.9324138  0.0458025  7.684926e-01 0.9828579
## p g1 c1 a6 t7   0.6930722 56.1552750 4.435342e-225 1.0000000
```

PIM pour CJS.

```
PIMS(cjs.cincle,"Phi")
```

```
## group = Group 1
##    1  2  3  4  5  6
## 1  1  2  3  4  5  6
## 2     2  3  4  5  6
## 3        3  4  5  6
## 4           4  5  6
## 5              5  6
## 6                 6
```

Fait tourner modèle avec param constants.

```
phip.cincle <- mark(cincle.proc,
                    cincle.ddl,
                    model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
## -2lnL:  666.8377
## AICc :  670.866
##
## Beta
##                   estimate        se       lcl       ucl
## Phi:(Intercept) 0.2421484 0.1020127 0.0422034 0.4420933
## p:(Intercept)   2.2262660 0.3251094 1.5890516 2.8634804
##
##
## Real Parameter Phi
##
##          1        2        3        4        5        6
## 1 0.560243 0.560243 0.560243 0.560243 0.560243 0.560243
## 2          0.560243 0.560243 0.560243 0.560243 0.560243
## 3                   0.560243 0.560243 0.560243 0.560243
## 4                            0.560243 0.560243 0.560243
## 5                                     0.560243 0.560243
## 6                                              0.560243
##
##
## Real Parameter p
##
##           2         3         4         5         6         7
## 1 0.9025835 0.9025835 0.9025835 0.9025835 0.9025835 0.9025835
## 2           0.9025835 0.9025835 0.9025835 0.9025835 0.9025835
## 3                     0.9025835 0.9025835 0.9025835 0.9025835
## 4                               0.9025835 0.9025835 0.9025835
## 5                                         0.9025835 0.9025835
## 6                                                   0.9025835
```

```r
phip.cincle$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.5602430 0.0251330 0.5105493 0.6087577
## p g1 c1 a1 t2   0.9025835 0.0285857 0.8304826 0.9460113
```

PIM pour CJS.

```r
PIMS(phip.cincle,"Phi")
```

```
## group = Group 1
##    1  2  3  4  5  6
## 1  1  1  1  1  1  1
## 2     1  1  1  1  1
## 3        1  1  1  1
## 4           1  1  1
## 5              1  1
## 6                 1
```

```r
PIMS(phip.cincle,"p")
```

```
## group = Group 1
##    2  3  4  5  6  7
## 1  2  2  2  2  2  2
## 2     2  2  2  2  2
## 3        2  2  2  2
## 4           2  2  2
## 5              2  2
## 6                 2
```

On ajoute les covariables environnementales.

```r
cov.cincle <- readxl::read_xls("dat/covariables-environnementales-cincle-plongeur.xls")
cov.cincle
```

```
## # A tibble: 7 x 3
##   année `débit (l/sec)` `temperature hiver (°C)`
##   <dbl>           <dbl>                    <dbl>
## 1  1981             443                     -2.3
## 2  1982            1114                     -0.4
## 3  1983             529                     -1.2
## 4  1984             434                     -4.2
## 5  1985             627                     -3  
## 6  1986             466                     -2.8
## 7  1987             730                      0.1
```

On simplifie le nom des colonnes.

```r
cov.cincle <- janitor::clean_names(cov.cincle)
cov.cincle
```

```
## # A tibble: 7 x 3
##   annee debit_l_sec temperature_hiver_c
##   <dbl>       <dbl>               <dbl>
## 1  1981         443                -2.3
## 2  1982        1114                -0.4
## 3  1983         529                -1.2
## 4  1984         434                -4.2
## 5  1985         627                -3  
## 6  1986         466                -2.8
## 7  1987         730                 0.1
```

On a 7 occasions de capture, donc 6 paramètres de survie. Si on suppose que la première année de capture dans le jeu de données cincle est 1981, alors on peut estimer la survie entre 1981 et 1982, à laquelle on applique la valeur de covariable en 1981, etc… jusqu'à la survie entre 1986 et 1987 à laquelle s'applique la valeur de covariable de 1986, donc on n'a pas besoin de la dernière ligne dans le jeu de données.

```r
cov.cincle <- cov.cincle[!(cov.cincle$annee == "1987"),]
```

On crée une survie qui depend du débit. Jetons un coup d'oeil à la structure sur la survie.

```
cincle.ddl$Phi
```

```
##    par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1          1           1     1      1   0    1          1      0   0    0
## 2          2           2     1      1   1    2          1      0   1    1
## 3          3           3     1      1   2    3          1      0   2    2
## 4          4           4     1      1   3    4          1      0   3    3
## 5          5           5     1      1   4    5          1      0   4    4
## 6          6           6     1      1   5    6          1      0   5    5
## 7          7           7     1      2   0    2          2      1   0    1
## 8          8           8     1      2   1    3          2      1   1    2
## 9          9           9     1      2   2    4          2      1   2    3
## 10        10          10     1      2   3    5          2      1   3    4
## 11        11          11     1      2   4    6          2      1   4    5
## 12        12          12     1      3   0    3          3      2   0    2
## 13        13          13     1      3   1    4          3      2   1    3
## 14        14          14     1      3   2    5          3      2   2    4
## 15        15          15     1      3   3    6          3      2   3    5
## 16        16          16     1      4   0    4          4      3   0    3
## 17        17          17     1      4   1    5          4      3   1    4
## 18        18          18     1      4   2    6          4      3   2    5
## 19        19          19     1      5   0    5          5      4   0    4
## 20        20          20     1      5   1    6          5      4   1    5
## 21        21          21     1      6   0    6          6      5   0    5
```

```
cincle.ddl$Phi$debit <- 0 # nv var mise a 0
for (i in 1:nrow(cov.cincle)){
   cincle.ddl$Phi$debit[cincle.ddl$Phi$time == i] <- as.numeric(cov.cincle[i, "debit_l_sec"])
}
```

On vérifie que ça a marché.

```
cincle.ddl$Phi
```

```
##    par.index model.index group cohort age time occ.cohort Cohort Age Time debit
## 1          1           1     1      1   0    1          1      0   0    0   443
## 2          2           2     1      1   1    2          1      0   1    1  1114
## 3          3           3     1      1   2    3          1      0   2    2   529
## 4          4           4     1      1   3    4          1      0   3    3   434
## 5          5           5     1      1   4    5          1      0   4    4   627
## 6          6           6     1      1   5    6          1      0   5    5   466
## 7          7           7     1      2   0    2          2      1   0    1  1114
## 8          8           8     1      2   1    3          2      1   1    2   529
## 9          9           9     1      2   2    4          2      1   2    3   434
## 10        10          10     1      2   3    5          2      1   3    4   627
## 11        11          11     1      2   4    6          2      1   4    5   466
## 12        12          12     1      3   0    3          3      2   0    2   529
## 13        13          13     1      3   1    4          3      2   1    3   434
## 14        14          14     1      3   2    5          3      2   2    4   627
## 15        15          15     1      3   3    6          3      2   3    5   466
## 16        16          16     1      4   0    4          4      3   0    3   434
## 17        17          17     1      4   1    5          4      3   1    4   627
```

```
## 18        18          18     1      4    2     6           4      3    2     5   466
## 19        19          19     1      5    0     5           5      4    0     4   627
## 20        20          20     1      5    1     6           5      4    1     5   466
## 21        21          21     1      6    0     6           6      5    0     5   466
```

Idem pour temperature.

```
cincle.ddl$Phi$temp <- 0 # nv var mise a 0
for (i in 1:nrow(cov.cincle)){
   cincle.ddl$Phi$temp[cincle.ddl$Phi$time == i] <- as.numeric(cov.cincle[i, "temperature_hiver_c"])
}
cincle.ddl$Phi
```

```
##    par.index model.index group cohort age time occ.cohort Cohort Age Time debit
## 1          1           1     1      1   0    1          1      0   0    0   443
## 2          2           2     1      1   1    2          1      0   1    1  1114
## 3          3           3     1      1   2    3          1      0   2    2   529
## 4          4           4     1      1   3    4          1      0   3    3   434
## 5          5           5     1      1   4    5          1      0   4    4   627
## 6          6           6     1      1   5    6          1      0   5    5   466
## 7          7           7     1      2   0    2          2      1   0    1  1114
## 8          8           8     1      2   1    3          2      1   1    2   529
## 9          9           9     1      2   2    4          2      1   2    3   434
## 10        10          10     1      2   3    5          2      1   3    4   627
## 11        11          11     1      2   4    6          2      1   4    5   466
## 12        12          12     1      3   0    3          3      2   0    2   529
## 13        13          13     1      3   1    4          3      2   1    3   434
## 14        14          14     1      3   2    5          3      2   2    4   627
## 15        15          15     1      3   3    6          3      2   3    5   466
## 16        16          16     1      4   0    4          4      3   0    3   434
## 17        17          17     1      4   1    5          4      3   1    4   627
## 18        18          18     1      4   2    6          4      3   2    5   466
## 19        19          19     1      5   0    5          5      4   0    4   627
## 20        20          20     1      5   1    6          5      4   1    5   466
## 21        21          21     1      6   0    6          6      5   0    5   466
##    temp
## 1  -2.3
## 2  -0.4
## 3  -1.2
## 4  -4.2
## 5  -3.0
## 6  -2.8
## 7  -0.4
## 8  -1.2
## 9  -4.2
## 10 -3.0
## 11 -2.8
## 12 -1.2
## 13 -4.2
## 14 -3.0
## 15 -2.8
## 16 -4.2
## 17 -3.0
```

```
## 18 -2.8
## 19 -3.0
## 20 -2.8
## 21 -2.8
```

On définit les effets.

```r
phi.debitptemp <- list(formula =~ debit + temp)
```

On ajuste le modèle.

```r
phicov.cincle <- mark(cincle.proc,
                      cincle.ddl,
                      model.parameters = list(Phi = phi.debitptemp, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~debit + temp)p(~1)
##
## Npar :  4
## -2lnL:  660.53
## AICc :  668.625
##
## Beta
##                      estimate           se        lcl       ucl
## Phi:(Intercept) -2.883130e-01 0.6383614000 -1.5395013 0.9628753
## Phi:debit        3.500863e-05 0.0006604602 -0.0012595 0.0013295
## Phi:temp        -2.095951e-01 0.1170473000 -0.4390078 0.0198176
## p:(Intercept)    2.235035e+00 0.3250918000  1.5978548 2.8722148
##
##
## Real Parameter Phi
##
##          1         2         3         4         5         6
## 1 0.552126 0.4587253 0.4954303 0.6472973 0.5896267 0.5780728
## 2          0.4587253 0.4954303 0.6472973 0.5896267 0.5780728
## 3                    0.4954303 0.6472973 0.5896267 0.5780728
## 4                              0.6472973 0.5896267 0.5780728
## 5                                        0.5896267 0.5780728
## 6                                                  0.5780728
##
##
## Real Parameter p
##
##           2         3         4         5         6         7
## 1 0.9033518 0.9033518 0.9033518 0.9033518 0.9033518 0.9033518
## 2           0.9033518 0.9033518 0.9033518 0.9033518 0.9033518
## 3                     0.9033518 0.9033518 0.9033518 0.9033518
## 4                               0.9033518 0.9033518 0.9033518
## 5                                         0.9033518 0.9033518
## 6                                                   0.9033518
```

Les paramètres estimés.

```
phicov.cincle$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.5521260 0.0380925 0.4768511 0.6250856
## Phi g1 c1 a1 t2 0.4587253 0.0640472 0.3382616 0.5842148
## Phi g1 c1 a2 t3 0.4954303 0.0515129 0.3959965 0.5952268
## Phi g1 c1 a3 t4 0.6472973 0.0421823 0.5609559 0.7249835
## Phi g1 c1 a4 t5 0.5896267 0.0319296 0.5259227 0.6504599
## Phi g1 c1 a5 t6 0.5780728 0.0299045 0.5186305 0.6353361
## p g1 c1 a1 t2   0.9033518 0.0283829 0.8317184 0.9464557
```

Visualisons relation survie débit pour une valeur moyenne de température.

```
min.debit <- min(cov.cincle$debit_l_sec)
max.debit <- max(cov.cincle$debit_l_sec)
debit.values <- seq(from = min.debit, to = max.debit, by = 5)
temp.values <- c(quantile(cov.cincle$temperature_hiver_c, 0.05),
                 mean(cov.cincle$temperature_hiver_c),
                 quantile(cov.cincle$temperature_hiver_c, 0.95))
```

Construit le jeu de données.

```
pred.dat <- expand.grid(debit = debit.values,
                        temp = temp.values)
pred.dat <- cbind(1, pred.dat)
pred.dat <- as.matrix(pred.dat)
```

Make prediction sur échelle logit.

```
betas.phi <- phicov.cincle$results$beta[1:3,1]
pred.surv.logit <- pred.dat %*% betas.phi
```

On back-transforme et on arrange.

```
pred.surv <- plogis(pred.surv.logit)
pred.df <- cbind(pred.dat[,-1], pred.surv)
colnames(pred.df) <- c("debit", "temp", "survie")
pred.df <- as.data.frame(pred.df)
head(pred.df)
```

```
##   debit temp    survie
## 1   434 -3.9 0.6328125
## 2   439 -3.9 0.6328532
## 3   444 -3.9 0.6328939
## 4   449 -3.9 0.6329345
## 5   454 -3.9 0.6329752
## 6   459 -3.9 0.6330159
```

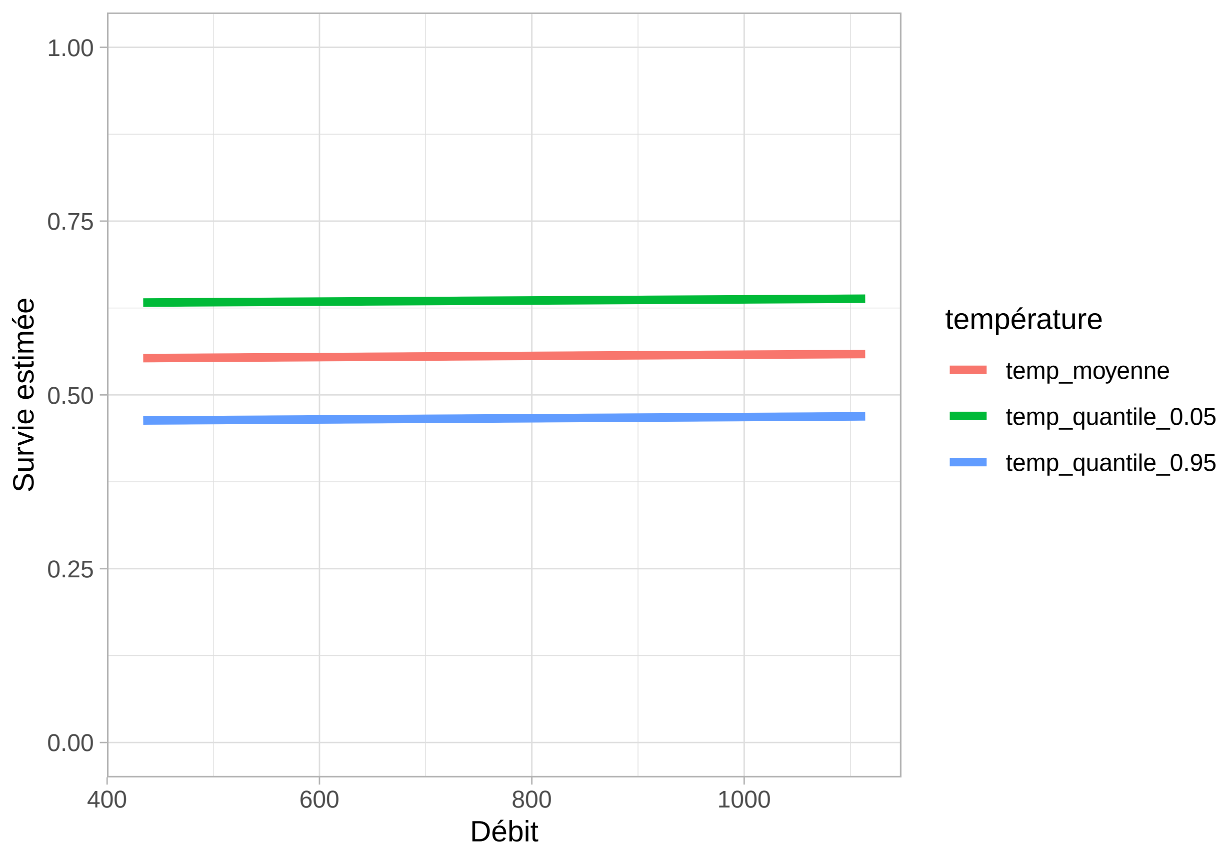
On prépare les données.

```r
pred.df$temp <- ifelse(pred.df$temp == mean(cov.cincle$temperature_hiver_c),
                       "temp_moyenne",
                       ifelse(pred.df$temp == quantile(cov.cincle$temperature_hiver_c, 0.05),
                              "temp_quantile_0.05",
                              "temp_quantile_0.95"))
head(pred.df)
```

```
##   debit               temp    survie
## 1   434 temp_quantile_0.05 0.6328125
## 2   439 temp_quantile_0.05 0.6328532
## 3   444 temp_quantile_0.05 0.6328939
## 4   449 temp_quantile_0.05 0.6329345
## 5   454 temp_quantile_0.05 0.6329752
## 6   459 temp_quantile_0.05 0.6330159
```

On visualise.

```r
library(ggplot2)
ggplot(pred.df, aes(x = debit, y = survie)) +
   geom_line(aes(color = temp), size = 1.5) +
   labs(x = "Débit",
        y = "Survie estimée",
        color = "température") +
    ylim(0, 1) +
   theme_light()
```



Partie 2 : Estimation de la survie, exemple du martinet noir

Les données

```
martinet <- convert.inp("dat/martinet-noir.inp",
                        group.df = data.frame(colonie = c("nord", "sud")),
                        covariates = NULL)
head(martinet)
```

```
##           ch freq colonie
## 1:1 00000001    7    nord
## 1:2 00000010    6    nord
## 1:3 00000011    1    nord
## 1:4 00000100    1    nord
## 1:8 00001000    1    nord
## 1:9 00001110    1    nord
```

Process the data

```
martinet.proc <- process.data(martinet,
                              begin.time = 1,
                              model = "CJS",
                              groups = ("colonie"))
```

Create default design data

```
martinet.ddl <- make.design.data(martinet.proc)
```

Examine the design data

```
head(martinet.ddl$Phi)
```

```
##   par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1         1           1  nord      1   0    1          1      0   0    0
## 2         2           2  nord      1   1    2          1      0   1    1
## 3         3           3  nord      1   2    3          1      0   2    2
## 4         4           4  nord      1   3    4          1      0   3    3
## 5         5           5  nord      1   4    5          1      0   4    4
## 6         6           6  nord      1   5    6          1      0   5    5
##   colonie
## 1    nord
## 2    nord
## 3    nord
## 4    nord
## 5    nord
## 6    nord
```

```
head(martinet.ddl$p)
```

```
##   par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1         1          57  nord      1   1    2          1      0   1    0
```

```
## 2          2           58  nord       1   2    3           1       0   2    1
## 3          3           59  nord       1   3    4           1       0   3    2
## 4          4           60  nord       1   4    5           1       0   4    3
## 5          5           61  nord       1   5    6           1       0   5    4
## 6          6           62  nord       1   6    7           1       0   6    5
##   colonie
## 1    nord
## 2    nord
## 3    nord
## 4    nord
## 5    nord
## 6    nord
```

On spécifie les effets sur les paramètres.

```r
phit <- list(formula=~time)
phi <- list(formula=~1)
pt <- list(formula=~time)
p <- list(formula=~1)
```

Fait tourner modèle CJS.

```r
cjs.martinet <- mark(martinet.proc,
                     martinet.ddl,
                     model.parameters = list(Phi = phit, p = pt))
```

```
##
## Output summary for CJS model
## Name : Phi(~time)p(~time)
##
## Npar :  14  (unadjusted=13)
## -2lnL:  354.9445
## AICc :  385.1905  (unadjusted=382.88072)
##
## Beta
##                    estimate          se           lcl          ucl
## Phi:(Intercept)   1.7439702   0.8654846     0.0476203    3.4403201
## Phi:time2        -0.9670003   1.0306467    -2.9870678    1.0530671
## Phi:time3        -0.5738993   1.1624645    -2.8523298    1.7045312
## Phi:time4        -0.8957165   1.0338542    -2.9220708    1.1306379
## Phi:time5        -0.9809820   0.9802262    -2.9022254    0.9402615
## Phi:time6        -0.6912528   1.0551063    -2.7592611    1.3767555
## Phi:time7        -1.8256786 344.3625800  -676.7763500  673.1250000
## p:(Intercept)     2.0030659   1.0495327    -0.0540181    4.0601500
## p:time3          -0.9689927   1.1966925    -3.3145101    1.3765247
## p:time4          -1.9340732   1.1630601    -4.2136711    0.3455247
## p:time5          -1.2041742   1.1750334    -3.5072397    1.0988912
## p:time6          -0.0882468   1.2916754    -2.6199307    2.4434370
## p:time7          -0.0861421   1.4799741    -2.9868913    2.8146071
## p:time8          -1.1127893 615.7357200 -1207.9548000 1205.7293000
##
##
## Real Parameter Phi
```

```
## Group:colonienord
##           1         2         3         4         5         6         7
## 1 0.8511907 0.6850267 0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 2           0.6850267 0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 3                     0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 4                               0.7002007 0.6820022 0.7412964 0.4795843
## 5                                         0.6820022 0.7412964 0.4795843
## 6                                                   0.7412964 0.4795843
## 7                                                             0.4795843
##
## Group:coloniesud
##           1         2         3         4         5         6         7
## 1 0.8511907 0.6850267 0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 2           0.6850267 0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 3                     0.7631578 0.7002007 0.6820022 0.7412964 0.4795843
## 4                               0.7002007 0.6820022 0.7412964 0.4795843
## 5                                         0.6820022 0.7412964 0.4795843
## 6                                                   0.7412964 0.4795843
## 7                                                             0.4795843
##
##
## Real Parameter p
## Group:colonienord
##           2         3         4         5         6        7         8
## 1 0.8811186 0.7377048 0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 2           0.7377048 0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 3                     0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 4                               0.6897374 0.8715596 0.871795 0.7089473
## 5                                         0.8715596 0.871795 0.7089473
## 6                                                   0.871795 0.7089473
## 7                                                            0.7089473
##
## Group:coloniesud
##           2         3         4         5         6        7         8
## 1 0.8811186 0.7377048 0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 2           0.7377048 0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 3                     0.5172413 0.6897374 0.8715596 0.871795 0.7089473
## 4                               0.6897374 0.8715596 0.871795 0.7089473
## 5                                         0.8715596 0.871795 0.7089473
## 6                                                   0.871795 0.7089473
## 7                                                            0.7089473
```

```r
cjs.martinet$results$real
```

```
##                      estimate          se           lcl       ucl fixed note
## Phi gnord c1 a0 t1 0.8511907   0.1096267  5.119028e-01 0.9689411
## Phi gnord c1 a1 t2 0.6850267   0.1013890  4.640514e-01 0.8452711
## Phi gnord c1 a2 t3 0.7631578   0.1402703  4.131408e-01 0.9365017
## Phi gnord c1 a3 t4 0.7002007   0.1187097  4.353324e-01 0.8761683
## Phi gnord c1 a4 t5 0.6820022   0.0998051  4.653068e-01 0.8409044
## Phi gnord c1 a5 t6 0.7412964   0.1157329  4.675202e-01 0.9033957
## Phi gnord c1 a6 t7 0.4795843  85.9469270 6.883379e-294 1.0000000
## p gnord c1 a1 t2   0.8811186   0.1099371  4.864987e-01 0.9830460
## p gnord c1 a2 t3   0.7377048   0.1112487  4.768147e-01 0.8966880
```

```
## p gnord c1 a3 t4   0.5172413   0.1251483  2.863173e-01 0.7410289
## p gnord c1 a4 t5   0.6897374   0.1130732  4.410916e-01 0.8622989
## p gnord c1 a5 t6   0.8715596   0.0842865  6.080354e-01 0.9674087
## p gnord c1 a6 t7   0.8717950   0.1166264  4.679772e-01 0.9813323
## p gnord c1 a7 t8   0.7089473 127.0513500 1.354961e-308 1.0000000
```

PIM pour CJS.

```
PIMS(cjs.martinet,"Phi")
```

```
## group = colonienord
##    1  2  3  4  5  6  7
## 1  1  2  3  4  5  6  7
## 2     2  3  4  5  6  7
## 3        3  4  5  6  7
## 4           4  5  6  7
## 5              5  6  7
## 6                 6  7
## 7                    7
## group = coloniesud
##    1  2  3  4  5  6  7
## 1  1  2  3  4  5  6  7
## 2     2  3  4  5  6  7
## 3        3  4  5  6  7
## 4           4  5  6  7
## 5              5  6  7
## 6                 6  7
## 7                    7
```

Fait tourner modèle avec param constants.

```
phip.martinet <- mark(martinet.proc,
                      martinet.ddl,
                      model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
## -2lnL:  372.8533
## AICc :  376.9136
##
## Beta
##                    estimate        se       lcl      ucl
## Phi:(Intercept) 0.8524384 0.1753794 0.5086948 1.196182
## p:(Intercept)   0.8881233 0.2391869 0.4193170 1.356929
##
##
## Real Parameter Phi
## Group:colonienord
##           1          2          3          4          5          6          7
```

```
## 1 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 2           0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 3                     0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 4                               0.7010784 0.7010784 0.7010784 0.7010784
## 5                                         0.7010784 0.7010784 0.7010784
## 6                                                   0.7010784 0.7010784
## 7                                                             0.7010784
##
## Group:coloniesud
##           1         2         3         4         5         6         7
## 1 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 2           0.7010784 0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 3                     0.7010784 0.7010784 0.7010784 0.7010784 0.7010784
## 4                               0.7010784 0.7010784 0.7010784 0.7010784
## 5                                         0.7010784 0.7010784 0.7010784
## 6                                                   0.7010784 0.7010784
## 7                                                             0.7010784
##
##
## Real Parameter p
## Group:colonienord
##           2         3         4         5         6         7         8
## 1 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 2           0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 3                     0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 4                               0.7085027 0.7085027 0.7085027 0.7085027
## 5                                         0.7085027 0.7085027 0.7085027
## 6                                                   0.7085027 0.7085027
## 7                                                             0.7085027
##
## Group:coloniesud
##           2         3         4         5         6         7         8
## 1 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 2           0.7085027 0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 3                     0.7085027 0.7085027 0.7085027 0.7085027 0.7085027
## 4                               0.7085027 0.7085027 0.7085027 0.7085027
## 5                                         0.7085027 0.7085027 0.7085027
## 6                                                   0.7085027 0.7085027
## 7                                                             0.7085027
```

```
phip.martinet$results$real
```

```
##                       estimate        se       lcl       ucl fixed note
## Phi gnord c1 a0 t1 0.7010784 0.0367538 0.6245005 0.7678449
## p gnord c1 a1 t2   0.7085027 0.0493985 0.6033198 0.7952602
```

PIM pour CJS.

```
PIMS(phip.martinet,"Phi")
```

```
## group = colonienord
##    1  2  3  4  5  6  7
```

```
## 1  1  1  1  1  1  1  1
## 2     1  1  1  1  1  1
## 3        1  1  1  1  1
## 4           1  1  1  1
## 5              1  1  1
## 6                 1  1
## 7                    1
## group = coloniesud
##    1  2  3  4  5  6  7
## 1  1  1  1  1  1  1  1
## 2     1  1  1  1  1  1
## 3        1  1  1  1  1
## 4           1  1  1  1
## 5              1  1  1
## 6                 1  1
## 7                    1
```

Modèle avec 2 classes d'âge sur la survie.

```
# create 0, 1+ age variable
martinet.ddl <- add.design.data(martinet.proc,
                                martinet.ddl, # add 2 age-class structure to design matrix
                             "Phi",
                             type = "age",
                             bins = c(0, 1, 7),
                             name = "ageclass",
                             right = FALSE)
```

Parameter specification.

```
phi.age <- list(formula=~ageclass) # age effect on survival
```

Fit CJS model.

```
CJSage.martinet <- mark(martinet.proc,
                        martinet.ddl,
                        model.parameters = list(Phi = phi.age, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~ageclass)p(~1)
##
## Npar :  3
## -2lnL:  372.846
## AICc :  378.9672
##
## Beta
##                      estimate        se        lcl       ucl
## Phi:(Intercept)     0.8749554 0.3191399  0.2494412 1.5004696
## Phi:ageclass[1,7] -0.0339141 0.3988106 -0.8155828 0.7477547
## p:(Intercept)       0.8823122 0.2487229  0.3948154 1.3698091
##
```

```
##
## Real Parameter Phi
## Group:colonienord
##           1         2         3         4         5         6         7
## 1 0.7057758 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845
## 2           0.7057758 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845
## 3                     0.7057758 0.6986845 0.6986845 0.6986845 0.6986845
## 4                               0.7057758 0.6986845 0.6986845 0.6986845
## 5                                         0.7057758 0.6986845 0.6986845
## 6                                                   0.7057758 0.6986845
## 7                                                             0.7057758
##
## Group:coloniesud
##           1         2         3         4         5         6         7
## 1 0.7057758 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845
## 2           0.7057758 0.6986845 0.6986845 0.6986845 0.6986845 0.6986845
## 3                     0.7057758 0.6986845 0.6986845 0.6986845 0.6986845
## 4                               0.7057758 0.6986845 0.6986845 0.6986845
## 5                                         0.7057758 0.6986845 0.6986845
## 6                                                   0.7057758 0.6986845
## 7                                                             0.7057758
##
##
## Real Parameter p
## Group:colonienord
##           2         3         4         5         6         7         8
## 1 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 2           0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 3                     0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 4                               0.7073011 0.7073011 0.7073011 0.7073011
## 5                                         0.7073011 0.7073011 0.7073011
## 6                                                   0.7073011 0.7073011
## 7                                                             0.7073011
##
## Group:coloniesud
##           2         3         4         5         6         7         8
## 1 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 2           0.7073011 0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 3                     0.7073011 0.7073011 0.7073011 0.7073011 0.7073011
## 4                               0.7073011 0.7073011 0.7073011 0.7073011
## 5                                         0.7073011 0.7073011 0.7073011
## 6                                                   0.7073011 0.7073011
## 7                                                             0.7073011
```

```
CJSage.martinet$results$real
```

```
##                      estimate        se       lcl       ucl fixed note
## Phi gnord c1 a0 t1 0.7057758 0.0662714 0.5620390 0.8176445
## Phi gnord c1 a1 t2 0.6986845 0.0463273 0.6010232 0.7811452
## p gnord c1 a1 t2   0.7073011 0.0514922 0.5974414 0.7973493
```

PIM pour CJS avec âge.

```r
PIMS(CJSage.martinet,"Phi")
```

```
## group = colonienord
##    1  2  3  4  5  6  7
## 1  1  2  2  2  2  2  2
## 2     1  2  2  2  2  2
## 3        1  2  2  2  2
## 4           1  2  2  2
## 5              1  2  2
## 6                 1  2
## 7                    1
## group = coloniesud
##    1  2  3  4  5  6  7
## 1  1  2  2  2  2  2  2
## 2     1  2  2  2  2  2
## 3        1  2  2  2  2
## 4           1  2  2  2
## 5              1  2  2
## 6                 1  2
## 7                    1
```

Maintenant gros modèle $phi(a.g), p(g.t)$.

Parameter specification.

```r
phi.a.g <- list(formula=~ageclass*colonie) # age and colonie effect on survival
p.g.t <- list(formula=~colonie*time) # age and colonie effect on survival
```

Fit CJS model.

```r
gros.mod <- mark(martinet.proc,
                        martinet.ddl,
                        model.parameters = list(Phi = phi.a.g, p = p.g.t))
```

```
##
## Output summary for CJS model
## Name : Phi(~ageclass * colonie)p(~colonie * time)
##
## Npar :  18  (unadjusted=16)
## -2lnL:  340.7324
## AICc :  380.4701  (unadjusted=375.67296)
##
## Beta
##                                estimate          se          lcl         ucl
## Phi:(Intercept)               0.1691839   0.5256402   -0.8610709   1.1994386
## Phi:ageclass[1,7]             0.4792946   0.7462046   -0.9832664   1.9418555
## Phi:coloniesud                1.4022291   0.7054837    0.0194810   2.7849771
## Phi:ageclass[1,7]:coloniesud -1.0377625   0.9299044   -2.8603752   0.7848503
## p:(Intercept)                16.4434260 171.6759400 -320.0414100 352.9282700
## p:coloniesud                -14.5239570 171.6768500 -351.0105900 321.9626700
## p:time3                     -15.5464860 171.6787800 -352.0368900 320.9439200
## p:time4                     -16.6873740 171.6778700 -353.1760100 319.8012600
```

```
## p:time5                        -17.8446000 171.6801800 -354.3377500 318.6485500
## p:time6                        -16.3966710 171.6801700 -352.8898100 320.0964700
## p:time7                          9.9834043   0.0000000    9.9834043   9.9834043
## p:time8                        -17.0637770 171.6756800 -353.5481200 319.4205700
## p:coloniesud:time3              14.6035410 171.6806700 -321.8905800 351.0976700
## p:coloniesud:time4              14.9333520 171.6793700 -321.5582300 351.4249300
## p:coloniesud:time5              17.2557740 171.6822200 -319.2413900 353.7529400
## p:coloniesud:time6              16.8234600 171.6842000 -319.6775700 353.3244900
## p:coloniesud:time7             -10.2100830   0.0000000  -10.2100830 -10.2100830
## p:coloniesud:time8              15.0287470 171.6768400 -321.4578600 351.5153600
##
##
## Real Parameter Phi
## Group:colonienord
##           1         2         3         4         5         6         7
## 1 0.5421954 0.6566675 0.6566675 0.6566675 0.6566675 0.6566675 0.6566675
## 2           0.5421954 0.6566675 0.6566675 0.6566675 0.6566675 0.6566675
## 3                     0.5421954 0.6566675 0.6566675 0.6566675 0.6566675
## 4                               0.5421954 0.6566675 0.6566675 0.6566675
## 5                                         0.5421954 0.6566675 0.6566675
## 6                                                   0.5421954 0.6566675
## 7                                                             0.5421954
##
## Group:coloniesud
##           1         2         3         4         5         6         7
## 1 0.8279849 0.7335961 0.7335961 0.7335961 0.7335961 0.7335961 0.7335961
## 2           0.8279849 0.7335961 0.7335961 0.7335961 0.7335961 0.7335961
## 3                     0.8279849 0.7335961 0.7335961 0.7335961 0.7335961
## 4                               0.8279849 0.7335961 0.7335961 0.7335961
## 5                                         0.8279849 0.7335961 0.7335961
## 6                                                   0.8279849 0.7335961
## 7                                                             0.8279849
##
##
## Real Parameter p
## Group:colonienord
##           2         3         4         5         6 7         8
## 1 0.9999999 0.7103204 0.4393136 0.1976298 0.5116866 1 0.3497016
## 2           0.7103204 0.4393136 0.1976298 0.5116866 1 0.3497016
## 3                     0.4393136 0.1976298 0.5116866 1 0.3497016
## 4                               0.1976298 0.5116866 1 0.3497016
## 5                                         0.5116866 1 0.3497016
## 6                                                   1 0.3497016
## 7                                                     0.3497016
##
## Group:coloniesud
##           2         3         4        5         6         7        8
## 1 0.8720792 0.7264181 0.5412677 0.790947 0.9126364 0.8445908 0.471142
## 2           0.7264181 0.5412677 0.790947 0.9126364 0.8445908 0.471142
## 3                     0.5412677 0.790947 0.9126364 0.8445908 0.471142
## 4                               0.790947 0.9126364 0.8445908 0.471142
## 5                                        0.9126364 0.8445908 0.471142
## 6                                                  0.8445908 0.471142
## 7                                                            0.471142
```

```r
gros.mod$results$real
```

```
##                        estimate           se           lcl       ucl fixed note
## Phi gnord c1 a0 t1 0.5421954 1.304742e-01  2.971157e-01 0.7684249
## Phi gnord c1 a1 t2 0.6566675 1.044161e-01  4.355444e-01 0.8258105
## Phi gsud c1 a0 t1  0.8279849 6.701740e-02  6.568190e-01 0.9236972
## Phi gsud c1 a1 t2  0.7335961 5.103750e-02  6.227152e-01 0.8212444
## p gnord c1 a1 t2   0.9999999 1.239996e-05 1.018075e-139 1.0000000
## p gnord c1 a2 t3   0.7103204 2.128975e-01  2.439774e-01 0.9490626
## p gnord c1 a3 t4   0.4393136 2.616160e-01  8.901800e-02 0.8626869
## p gnord c1 a4 t5   0.1976298 1.847864e-01  2.447820e-02 0.7074104
## p gnord c1 a5 t6   0.5116866 2.865189e-01  9.968040e-02 0.9084031
## p gnord c1 a6 t7   1.0000000 0.000000e+00  1.000000e+00 1.0000000
## p gnord c1 a7 t8   0.3497016 2.284800e-01  6.981270e-02 0.7939445
## p gsud c1 a1 t2    0.8720792 1.169540e-01  4.662134e-01 0.9815540
## p gsud c1 a2 t3    0.7264181 1.196538e-01  4.492880e-01 0.8962837
## p gsud c1 a3 t4    0.5412677 1.239405e-01  3.072706e-01 0.7583776
## p gsud c1 a4 t5    0.7909470 1.016555e-01  5.313721e-01 0.9266024
## p gsud c1 a5 t6    0.9126364 8.004290e-02  5.935343e-01 0.9867957
## p gsud c1 a6 t7    0.8445908 1.129739e-01  5.014514e-01 0.9670665
## p gsud c1 a7 t8    0.4711420 1.002335e-01  2.882256e-01 0.6621514
```

PIM pour survie et détection dans big model.

```r
PIMS(gros.mod,"Phi")
```

```
## group = colonienord
##    1  2  3  4  5  6  7
## 1  1  2  2  2  2  2  2
## 2     1  2  2  2  2  2
## 3        1  2  2  2  2
## 4           1  2  2  2
## 5              1  2  2
## 6                 1  2
## 7                    1
## group = coloniesud
##    1  2  3  4  5  6  7
## 1  3  4  4  4  4  4  4
## 2     3  4  4  4  4  4
## 3        3  4  4  4  4
## 4           3  4  4  4
## 5              3  4  4
## 6                 3  4
## 7                    3
```

```r
PIMS(gros.mod,"p")
```

```
## group = colonienord
##    2  3  4  5  6  7  8
## 1  5  6  7  8  9 10 11
## 2     6  7  8  9 10 11
```

```
## 3        7  8  9 10 11
## 4           8  9 10 11
## 5              9 10 11
## 6                10 11
## 7                   11
## group = coloniesud
##    2  3  4  5  6  7  8
## 1 12 13 14 15 16 17 18
## 2    13 14 15 16 17 18
## 3       14 15 16 17 18
## 4          15 16 17 18
## 5             16 17 18
## 6                17 18
## 7                   18
```

Partie 3 : Hypothèses des modèles de capture-recapture, hétérogénéité et tests d'ajustement

Le but de cet exercice est de se familiariser avec les données de capture-recapture en population ouverte, d'ajuster par maximum de vraisemblance quelques modèles simples, de comparer ces modèles entre eux pour déterminer celui qui fournit la meilleure description des données et de tester la qualité de l'ajustement de ces modèles.

Question 1

On simule 2 jeux de données de capture-recapture avec les paramètres de survie (ϕ) et recapture (p) suivants : * jeu de données G1 : $\phi = 0.8$, $p = 0.8$; * jeu de données G2 : $\phi = 0.8$, $p = 0.2$.

```
simul <- function(nind, nocc, phi, p){
   dat <- matrix(0, nrow = nind, ncol = nocc)
   dat[1:nind, 1] <- 1 # a single cohort
   for (i in 1:nind){
      # processus survie
      for (j in 2:nocc){
         alive.or.dead <- rbinom(1, 1, phi)
         # conditional on being alive at t, alive or dead at t+1
         dat[i, j] <- ifelse(dat[i, j - 1] == 0, 0, alive.or.dead)
      }
      # processus detection
      for (j in 2:nocc){
         detected.or.not <- rbinom(1, 1, p)
         # conditional on being alive at t, detected or not at t
         dat[i, j] <- ifelse(dat[i, j] == 0, 0, detected.or.not)
      }
   }
data.frame(y = dat)
}
```

```
set.seed(2021)
nind <- 500
nocc <- 8
```

```r
G1 <- simul(nind = nind, nocc = nocc, phi = 0.8, p = 0.8)
G2 <- simul(nind = nind, nocc = nocc, phi = 0.8, p = 0.2)
```

A l'aide du package `marked`, ajuster séparément à G1 et G2 le modèle $\Phi(t), p(t)$ appelé aussi le modèle de Cormack-Jolly-Seber (CJS). Que pouvez-vous vous dire sur l'estimation des paramètres ?

```r
G1marked <- data.frame(ch = tidyr::unite(G1, col = "ch", sep = ""),
                       n = rep(1, nrow(G1)))
G2marked <- data.frame(ch = tidyr::unite(G2, col = "ch", sep = ""),
                       n = rep(1, nrow(G2)))
```

Process data

```r
G1.proc <- process.data(G1marked)
G2.proc <- process.data(G2marked)
```

Make design data

```r
G1.ddl <- make.design.data(G1.proc)
G2.ddl <- make.design.data(G2.proc)
```

Look at design data

```r
G1.ddl
```

```
## $Phi
##    par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1          1           1     1      1   0    1          1      0   0    0
## 2          2           2     1      1   1    2          1      0   1    1
## 3          3           3     1      1   2    3          1      0   2    2
## 4          4           4     1      1   3    4          1      0   3    3
## 5          5           5     1      1   4    5          1      0   4    4
## 6          6           6     1      1   5    6          1      0   5    5
## 7          7           7     1      1   6    7          1      0   6    6
## 8          8           8     1      2   0    2          2      1   0    1
## 9          9           9     1      2   1    3          2      1   1    2
## 10        10          10     1      2   2    4          2      1   2    3
## 11        11          11     1      2   3    5          2      1   3    4
## 12        12          12     1      2   4    6          2      1   4    5
## 13        13          13     1      2   5    7          2      1   5    6
## 14        14          14     1      3   0    3          3      2   0    2
## 15        15          15     1      3   1    4          3      2   1    3
## 16        16          16     1      3   2    5          3      2   2    4
## 17        17          17     1      3   3    6          3      2   3    5
## 18        18          18     1      3   4    7          3      2   4    6
## 19        19          19     1      4   0    4          4      3   0    3
## 20        20          20     1      4   1    5          4      3   1    4
## 21        21          21     1      4   2    6          4      3   2    5
## 22        22          22     1      4   3    7          4      3   3    6
## 23        23          23     1      5   0    5          5      4   0    4
## 24        24          24     1      5   1    6          5      4   1    5
```

```
## 25        25          25     1      5   2    7          5      4   2    6
## 26        26          26     1      6   0    6          6      5   0    5
## 27        27          27     1      6   1    7          6      5   1    6
## 28        28          28     1      7   0    7          7      6   0    6
##
## $p
##    par.index model.index group cohort age time occ.cohort Cohort Age Time
## 1          1          29     1      1   1    2          1      0   1    0
## 2          2          30     1      1   2    3          1      0   2    1
## 3          3          31     1      1   3    4          1      0   3    2
## 4          4          32     1      1   4    5          1      0   4    3
## 5          5          33     1      1   5    6          1      0   5    4
## 6          6          34     1      1   6    7          1      0   6    5
## 7          7          35     1      1   7    8          1      0   7    6
## 8          8          36     1      2   1    3          2      1   1    1
## 9          9          37     1      2   2    4          2      1   2    2
## 10        10          38     1      2   3    5          2      1   3    3
## 11        11          39     1      2   4    6          2      1   4    4
## 12        12          40     1      2   5    7          2      1   5    5
## 13        13          41     1      2   6    8          2      1   6    6
## 14        14          42     1      3   1    4          3      2   1    2
## 15        15          43     1      3   2    5          3      2   2    3
## 16        16          44     1      3   3    6          3      2   3    4
## 17        17          45     1      3   4    7          3      2   4    5
## 18        18          46     1      3   5    8          3      2   5    6
## 19        19          47     1      4   1    5          4      3   1    3
## 20        20          48     1      4   2    6          4      3   2    4
## 21        21          49     1      4   3    7          4      3   3    5
## 22        22          50     1      4   4    8          4      3   4    6
## 23        23          51     1      5   1    6          5      4   1    4
## 24        24          52     1      5   2    7          5      4   2    5
## 25        25          53     1      5   3    8          5      4   3    6
## 26        26          54     1      6   1    7          6      5   1    5
## 27        27          55     1      6   2    8          6      5   2    6
## 28        28          56     1      7   1    8          7      6   1    6
##
## $pimtypes
## $pimtypes$Phi
## $pimtypes$Phi$pim.type
## [1] "all"
##
##
## $pimtypes$p
## $pimtypes$p$pim.type
## [1] "all"
```

Outine formulas for each parameter

```
phi <- list(formula=~1)
p <- list(formula=~1)
```

Make model constant survival and detection prob. For G1 first.

```r
cjs.G1 <- mark(G1.proc,
              G1.ddl,
              model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
## -2lnL:  3009.594
## AICc :  3013.601
##
## Beta
##                  estimate        se      lcl      ucl
## Phi:(Intercept) 1.349691 0.0584381 1.235152 1.464229
## p:(Intercept)   1.417211 0.0761598 1.267938 1.566484
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.7940791 0.7940791 0.7940791 0.7940791 0.7940791 0.7940791 0.7940791
## 2           0.7940791 0.7940791 0.7940791 0.7940791 0.7940791 0.7940791
## 3                     0.7940791 0.7940791 0.7940791 0.7940791 0.7940791
## 4                               0.7940791 0.7940791 0.7940791 0.7940791
## 5                                         0.7940791 0.7940791 0.7940791
## 6                                                   0.7940791 0.7940791
## 7                                                             0.7940791
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.8049008 0.8049008 0.8049008 0.8049008 0.8049008 0.8049008 0.8049008
## 2           0.8049008 0.8049008 0.8049008 0.8049008 0.8049008 0.8049008
## 3                     0.8049008 0.8049008 0.8049008 0.8049008 0.8049008
## 4                               0.8049008 0.8049008 0.8049008 0.8049008
## 5                                         0.8049008 0.8049008 0.8049008
## 6                                                   0.8049008 0.8049008
## 7                                                             0.8049008
```

```r
cjs.G1$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.7940791 0.0095556 0.7747191 0.8121787
## p g1 c1 a1 t2   0.8049008 0.0119598 0.7803896 0.8272818
```

Then for G2.

```r
cjs.G2 <- mark(G2.proc,
              G2.ddl,
              model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
## -2lnL:  2091.359
## AICc :  2095.374
##
## Beta
##                   estimate        se       lcl       ucl
## Phi:(Intercept)  1.487792 0.1111557  1.269927  1.705657
## p:(Intercept)   -1.398919 0.0940821 -1.583320 -1.214518
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.8157466 0.8157466 0.8157466 0.8157466 0.8157466 0.8157466 0.8157466
## 2           0.8157466 0.8157466 0.8157466 0.8157466 0.8157466 0.8157466
## 3                     0.8157466 0.8157466 0.8157466 0.8157466 0.8157466
## 4                               0.8157466 0.8157466 0.8157466 0.8157466
## 5                                         0.8157466 0.8157466 0.8157466
## 6                                                   0.8157466 0.8157466
## 7                                                             0.8157466
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.1979877 0.1979877 0.1979877 0.1979877 0.1979877 0.1979877 0.1979877
## 2           0.1979877 0.1979877 0.1979877 0.1979877 0.1979877 0.1979877
## 3                     0.1979877 0.1979877 0.1979877 0.1979877 0.1979877
## 4                               0.1979877 0.1979877 0.1979877 0.1979877
## 5                                         0.1979877 0.1979877 0.1979877
## 6                                                   0.1979877 0.1979877
## 7                                                             0.1979877
```

```
cjs.G2$results$real
```

```
##                  estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.8157466 0.0167072 0.7807302 0.8462721
## p g1 c1 a1 t2   0.1979877 0.0149392 0.1703258 0.2289026
```

Question 2

a) Grouper les jeux de données G1 et G2 pour obtenir le jeu de données G1+G2.

```
G1plusG2 <- rbind(G1, G2)
```

b) Ajuster le modèle CJS à G1+G2. Que remarquez-vous concernant l'estimation des paramètres ?

```r
G1G2marked <- data.frame(ch = tidyr::unite(G1plusG2, col = "ch", sep = ""),
                        n = rep(1, nrow(G1plusG2)))
G1G2.proc <- process.data(G1G2marked)
G1G2.ddl <- make.design.data(G1G2.proc)
cjs.G1G2 <- mark(G1G2.proc,
                G1G2.ddl,
                model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
## -2lnL:  5825.357
## AICc :  5829.362
##
## Beta
##                   estimate        se       lcl       ucl
## Phi:(Intercept) 1.1639804 0.0436060 1.0785126 1.2494483
## p:(Intercept)   0.3450607 0.0495103 0.2480206 0.4421009
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.7620552 0.7620552 0.7620552 0.7620552 0.7620552 0.7620552 0.7620552
## 2           0.7620552 0.7620552 0.7620552 0.7620552 0.7620552 0.7620552
## 3                     0.7620552 0.7620552 0.7620552 0.7620552 0.7620552
## 4                               0.7620552 0.7620552 0.7620552 0.7620552
## 5                                         0.7620552 0.7620552 0.7620552
## 6                                                   0.7620552 0.7620552
## 7                                                             0.7620552
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.5854193 0.5854193 0.5854193 0.5854193 0.5854193 0.5854193 0.5854193
## 2           0.5854193 0.5854193 0.5854193 0.5854193 0.5854193 0.5854193
## 3                     0.5854193 0.5854193 0.5854193 0.5854193 0.5854193
## 4                               0.5854193 0.5854193 0.5854193 0.5854193
## 5                                         0.5854193 0.5854193 0.5854193
## 6                                                   0.5854193 0.5854193
## 7                                                             0.5854193
```

```r
cjs.G1G2$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.7620552 0.0079070 0.7462124 0.7772043
## p g1 c1 a1 t2   0.5854193 0.0120163 0.5616892 0.6087595
```

Modèle avec survie qui dépend du temps.

```r
phi.time <- list(formula=~time)
cjs.G1G2 <- mark(G1G2.proc,
                G1G2.ddl,
                model.parameters = list(Phi = phi.time, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~time)p(~1)
##
## Npar :  8
## -2lnL:  5792.723
## AICc :  5808.782
##
## Beta
##                   estimate        se        lcl       ucl
## Phi:(Intercept) 0.7598154 0.0909051  0.5816415 0.9379894
## Phi:time2       0.3979977 0.1933475  0.0190366 0.7769588
## Phi:time3       0.7072567 0.2137486  0.2883094 1.1262040
## Phi:time4       0.6334194 0.2291571  0.1842715 1.0825673
## Phi:time5       0.4558457 0.2336744 -0.0021562 0.9138477
## Phi:time6       0.8182725 0.3428356  0.1463147 1.4902303
## Phi:time7       1.0221797 0.5473045 -0.0505372 2.0948966
## p:(Intercept)   0.3870597 0.0505148  0.2880506 0.4860688
##
##
## Real Parameter Phi
##
##           1         2        3         4         5         6         7
## 1 0.6813137 0.7609351 0.812612 0.8011082 0.7712991 0.8289336 0.8559431
## 2           0.7609351 0.812612 0.8011082 0.7712991 0.8289336 0.8559431
## 3                     0.812612 0.8011082 0.7712991 0.8289336 0.8559431
## 4                              0.8011082 0.7712991 0.8289336 0.8559431
## 5                                        0.7712991 0.8289336 0.8559431
## 6                                                  0.8289336 0.8559431
## 7                                                            0.8559431
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.5955747 0.5955747 0.5955747 0.5955747 0.5955747 0.5955747 0.5955747
## 2           0.5955747 0.5955747 0.5955747 0.5955747 0.5955747 0.5955747
## 3                     0.5955747 0.5955747 0.5955747 0.5955747 0.5955747
## 4                               0.5955747 0.5955747 0.5955747 0.5955747
## 5                                         0.5955747 0.5955747 0.5955747
## 6                                                   0.5955747 0.5955747
## 7                                                             0.5955747
```

```r
cjs.G1G2$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.6813137 0.0197378 0.6414450 0.7186933
```

```
## Phi g1 c1 a1 t2 0.7609351 0.0259835 0.7063779 0.8081090
## Phi g1 c1 a2 t3 0.8126120 0.0294917 0.7479047 0.8637363
## Phi g1 c1 a3 t4 0.8011082 0.0335599 0.7271891 0.8588852
## Phi g1 c1 a4 t5 0.7712991 0.0379757 0.6886256 0.8372110
## Phi g1 c1 a5 t6 0.8289336 0.0470412 0.7166457 0.9027616
## Phi g1 c1 a6 t7 0.8559431 0.0668939 0.6723155 0.9450759
## p g1 c1 a1 t2   0.5955747 0.0121673 0.5715188 0.6191799
```

Question 3

A l'aide du package `R2ucare`, tester la qualité de l'ajustement du modèle CJS aux données G1, G2 et G1+G2. Quelles sont vos conclusions ?

G1

```
overall_CJS(G1, rep(1,nrow(G1)))
```

```
##                             chi2 degree_of_freedom p_value
## Gof test for CJS model: 3.327                 9    0.95
```

G2

```
overall_CJS(G2, rep(1,nrow(G2)))
```

```
##                              chi2 degree_of_freedom p_value
## Gof test for CJS model: 15.041                14   0.375
```

G1G2

```
overall_CJS(G1plusG2, rep(1,nrow(G1plusG2)))
```

```
##                               chi2 degree_of_freedom p_value
## Gof test for CJS model: 150.342                15       0
```

Question 4

Il peut y avoir des animaux en transit sur la zone d'étude.

a) Pour créer artificiellement une telle situation, rajouter 50 individus en transit (i.e. possédant une histoire avec un seul événement de capture) à chaque date dans G1. Voir la fin du fichier G1transit.inp

```
G1transit <- as.matrix(G1)
ntransients <- 50
for (j in 1:nocc){
   zeros <- matrix(0, nrow = ntransients, ncol = nocc)
   zeros[, j] <- 1
   G1transit <- rbind(G1transit, zeros)
}
G1transit <- data.frame(y = G1transit)
```

```r
dim(G1transit)
```

```
## [1] 900   8
```

```r
head(G1transit)
```

```
##   y.y.1 y.y.2 y.y.3 y.y.4 y.y.5 y.y.6 y.y.7 y.y.8
## 1     1     1     0     0     1     0     1     1
## 2     1     0     0     0     0     0     0     0
## 3     1     0     0     0     0     0     0     0
## 4     1     0     0     0     0     0     0     0
## 5     1     1     0     0     0     0     0     0
## 6     1     1     1     0     0     0     0     0
```

```r
tail(G1transit)
```

```
##     y.y.1 y.y.2 y.y.3 y.y.4 y.y.5 y.y.6 y.y.7 y.y.8
## 895     0     0     0     0     0     0     0     1
## 896     0     0     0     0     0     0     0     1
## 897     0     0     0     0     0     0     0     1
## 898     0     0     0     0     0     0     0     1
## 899     0     0     0     0     0     0     0     1
## 900     0     0     0     0     0     0     0     1
```

b) faire tourner le modèle CJS à ces nouvelles données avec `RMark`. Quelles sont vos conclusions concernant les estimations ?

```r
G1transitmarked <- data.frame(ch = tidyr::unite(G1transit, col = "ch", sep = ""),
                              n = rep(1, nrow(G1transit)))
```

Process data

```r
G1transit.proc <- process.data(G1transitmarked)
```

Make design data

```r
G1transit.ddl <- make.design.data(G1transit.proc)
```

Fit CJS model.

```r
cjs.G1transit <- mark(G1transit.proc,
                      G1transit.ddl,
                      model.parameters = list(Phi = phi, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~1)p(~1)
##
## Npar :  2
```

```
## -2lnL:  3793.282
## AICc :  3797.288
##
## Beta
##                   estimate        se       lcl       ucl
## Phi:(Intercept) 0.7871844 0.0479482 0.6932058 0.8811629
## p:(Intercept)   1.1885539 0.0770665 1.0375036 1.3396043
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.6872264 0.6872264 0.6872264 0.6872264 0.6872264 0.6872264 0.6872264
## 2           0.6872264 0.6872264 0.6872264 0.6872264 0.6872264 0.6872264
## 3                     0.6872264 0.6872264 0.6872264 0.6872264 0.6872264
## 4                               0.6872264 0.6872264 0.6872264 0.6872264
## 5                                         0.6872264 0.6872264 0.6872264
## 6                                                   0.6872264 0.6872264
## 7                                                             0.6872264
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.7664823 0.7664823 0.7664823 0.7664823 0.7664823 0.7664823 0.7664823
## 2           0.7664823 0.7664823 0.7664823 0.7664823 0.7664823 0.7664823
## 3                     0.7664823 0.7664823 0.7664823 0.7664823 0.7664823
## 4                               0.7664823 0.7664823 0.7664823 0.7664823
## 5                                         0.7664823 0.7664823 0.7664823
## 6                                                   0.7664823 0.7664823
## 7                                                             0.7664823
```

```
cjs.G1transit$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.6872264 0.0103063 0.6666797 0.7070632
## p g1 c1 a1 t2   0.7664823 0.0137939 0.7383680 0.7924249
```

Idem avec survie qui dépend du temps.

```
cjs.G1transit <- mark(G1transit.proc,
                      G1transit.ddl,
                      model.parameters = list(Phi = phi.time, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~time)p(~1)
##
## Npar :  8
## -2lnL:  3776.066
## AICc :  3792.138
##
```

```
## Beta
##                    estimate        se        lcl        ucl
## Phi:(Intercept)  1.1097474 0.1223381  0.8699648  1.3495301
## Phi:time2       -0.3581909 0.1908886 -0.7323326  0.0159507
## Phi:time3       -0.2686701 0.1890295 -0.6391680  0.1018278
## Phi:time4       -0.4461152 0.1934614 -0.8252996 -0.0669308
## Phi:time5       -0.4810596 0.2040143 -0.8809277 -0.0811916
## Phi:time6       -0.3850179 0.2250947 -0.8262034  0.0561677
## Phi:time7       -0.8490590 0.2307664 -1.3013612 -0.3967568
## p:(Intercept)    1.1866979 0.0786387  1.0325661  1.3408297
##
##
## Real Parameter Phi
##
##          1         2         3         4         5         6         7
## 1 0.752082 0.6795178 0.6986921 0.6600758 0.6521919 0.6736477 0.5648055
## 2          0.6795178 0.6986921 0.6600758 0.6521919 0.6736477 0.5648055
## 3                    0.6986921 0.6600758 0.6521919 0.6736477 0.5648055
## 4                              0.6600758 0.6521919 0.6736477 0.5648055
## 5                                        0.6521919 0.6736477 0.5648055
## 6                                                  0.6736477 0.5648055
## 7                                                            0.5648055
##
##
## Real Parameter p
##
##         2       3       4       5       6       7       8
## 1 0.76615 0.76615 0.76615 0.76615 0.76615 0.76615 0.76615
## 2         0.76615 0.76615 0.76615 0.76615 0.76615 0.76615
## 3                 0.76615 0.76615 0.76615 0.76615 0.76615
## 4                         0.76615 0.76615 0.76615 0.76615
## 5                                 0.76615 0.76615 0.76615
## 6                                         0.76615 0.76615
## 7                                                 0.76615
```

```r
cjs.G1transit$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.7520820 0.0228105 0.7047384 0.7940528
## Phi g1 c1 a1 t2 0.6795178 0.0270283 0.6244072 0.7300381
## Phi g1 c1 a2 t3 0.6986921 0.0305397 0.6356994 0.7549906
## Phi g1 c1 a3 t4 0.6600758 0.0337817 0.5911056 0.7228668
## Phi g1 c1 a4 t5 0.6521919 0.0371981 0.5762203 0.7211351
## Phi g1 c1 a5 t6 0.6736477 0.0419754 0.5867402 0.7500640
## Phi g1 c1 a6 t7 0.5648055 0.0489543 0.4676276 0.6572466
## p g1 c1 a1 t2   0.7661500 0.0140892 0.7374131 0.7926263
```

c) Tester l'ajustement du modèle CJS à ces mêmes données avec `R2ucare`. Interpréter en particulier la composante 3.SR du test.

```r
overall_CJS(G1transit, rep(1,nrow(G1transit)))
```

```
##                               chi2 degree_of_freedom p_value
## Gof test for CJS model: 543.606                15       0
```

```r
test2ct(G1transit, rep(1,nrow(G1transit)))
```

```
## $test2ct
##      stat        df     p_val sign_test
##     1.135     5.000     0.951     0.600
##
## $details
##   component dof  stat p_val signed_test  test_perf
## 1         2   1 0.112 0.737      -0.335 Chi-square
## 2         3   1 0.003 0.953       0.055 Chi-square
## 3         4   1 0.721 0.396       0.849 Chi-square
## 4         5   1 0.139 0.709       0.373 Chi-square
## 5         6   1  0.16  0.69         0.4     Fisher
```

```r
test3sr(G1transit, rep(1,nrow(G1transit)))
```

```
## $test3sr
##      stat        df     p_val sign_test
##   540.279     6.000     0.000    23.140
##
## $details
##   component    stat p_val signed_test  test_perf
## 1         2  96.827     0        9.84 Chi-square
## 2         3 103.329     0      10.165 Chi-square
## 3         4  88.333     0       9.399 Chi-square
## 4         5   94.62     0       9.727 Chi-square
## 5         6 100.743     0      10.037 Chi-square
## 6         7  56.427     0       7.512 Chi-square
```

d) faire tourner un modèle à 2 classes d'âge sur la survie $\phi(a2 * t)$ avec `RMark`. Vos conclusions ?

```r
G1transit.ddl <- make.design.data(G1transit.proc)
# create 0, 1+ age variable
G1transit.ddl <- add.design.data(G1transit.proc,
                                 G1transit.ddl, # add 2 age-class structure to design matrix
                                 "Phi",
                                 type = "age",
                                 bins = c(0, 1, nocc - 1),
                                 name = "ageclass",
                                 right = FALSE)
```

Parameter specification.

```r
phi.age <- list(formula=~ageclass) # age effect on survival
```

Fit CJS model.

```r
cjsage.G1transit <- mark(G1transit.proc,
                         G1transit.ddl,
                         model.parameters = list(Phi = phi.age, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~ageclass)p(~1)
##
## Npar :  3
## -2lnL:  3604.772
## AICc :  3610.784
##
## Beta
##                       estimate        se        lcl      ucl
## Phi:(Intercept)    -0.1000537 0.0738121 -0.2447253 0.044618
## Phi:ageclass[1,7]   1.4656701 0.1046905  1.2604768 1.670864
## p:(Intercept)       1.3783584 0.0769186  1.2275979 1.529119
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.4750074 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710
## 2           0.4750074 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710
## 3                     0.4750074 0.7966710 0.7966710 0.7966710 0.7966710
## 4                               0.4750074 0.7966710 0.7966710 0.7966710
## 5                                         0.4750074 0.7966710 0.7966710
## 6                                                   0.4750074 0.7966710
## 7                                                             0.4750074
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 2           0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 3                     0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 4                               0.7987272 0.7987272 0.7987272 0.7987272
## 5                                         0.7987272 0.7987272 0.7987272
## 6                                                   0.7987272 0.7987272
## 7                                                             0.7987272
```

```
cjsage.G1transit$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
## Phi g1 c1 a0 t1 0.4750074 0.0184069 0.4391222 0.5111526
## Phi g1 c1 a1 t2 0.7966710 0.0113847 0.7734445 0.8180764
## p g1 c1 a1 t2   0.7987272 0.0123656 0.7733979 0.8218774
```

Autre façon.

```
G1transit.ddl <- make.design.data(G1transit.proc)
#max age 4
G1transit.ddl$Phi$max.age <- as.factor((G1transit.ddl$Phi$Age < 1) * G1transit.ddl$Phi$Age + (G1transit
phi.max.age <- list(formula=~max.age)
cjsaget.G1transit <- mark(G1transit.proc,
                     G1transit.ddl,
                     model.parameters = list(Phi = phi.max.age, p = p))
```

```
##
## Output summary for CJS model
## Name : Phi(~max.age)p(~1)
##
## Npar :  3
## -2lnL:  3604.772
## AICc :  3610.784
##
## Beta
##                     estimate        se        lcl      ucl
## Phi:(Intercept) -0.1000537 0.0738121 -0.2447253 0.044618
## Phi:max.age1     1.4656701 0.1046905  1.2604767 1.670863
## p:(Intercept)    1.3783584 0.0769186  1.2275979 1.529119
##
##
## Real Parameter Phi
##
##           1         2         3         4         5         6         7
## 1 0.4750074 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710
## 2           0.4750074 0.7966710 0.7966710 0.7966710 0.7966710 0.7966710
## 3                     0.4750074 0.7966710 0.7966710 0.7966710 0.7966710
## 4                               0.4750074 0.7966710 0.7966710 0.7966710
## 5                                         0.4750074 0.7966710 0.7966710
## 6                                                   0.4750074 0.7966710
## 7                                                             0.4750074
##
##
## Real Parameter p
##
##           2         3         4         5         6         7         8
## 1 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 2           0.7987272 0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 3                     0.7987272 0.7987272 0.7987272 0.7987272 0.7987272
## 4                               0.7987272 0.7987272 0.7987272 0.7987272
## 5                                         0.7987272 0.7987272 0.7987272
## 6                                                   0.7987272 0.7987272
## 7                                                             0.7987272
```

```
PIMS(cjsaget.G1transit,"Phi")
```

```
## group = Group 1
##    1  2  3  4  5  6  7
## 1  1  2  2  2  2  2  2
## 2     1  2  2  2  2  2
## 3        1  2  2  2  2
## 4           1  2  2  2
## 5              1  2  2
## 6                 1  2
## 7                    1
```

```
cjsaget.G1transit$results$real
```

```
##                   estimate        se       lcl       ucl fixed note
```

```
## Phi g1 c1 a0 t1 0.4750074 0.0184069 0.4391222 0.5111526
## Phi g1 c1 a1 t2 0.7966710 0.0113847 0.7734445 0.8180764
## p g1 c1 a1 t2   0.7987272 0.0123656 0.7733979 0.8218774
```

Supprime fichiers créés en cours de route.

```
cleanup(ask = FALSE)
```