Supplementary material for 'Dealing with many correlated covariates in capture-recapture models' by Gimenez and Barbraud.

Olivier Gimenez

24 mai 2017

## Introduction

We illustrate the principal component capture-recapture (P2CR) method for covariates selection in capture-recapture models using data on survival of Snow petrels in Pointe Géologie Archipelago, Terre Adélie, Antarctica. In total, the dataset consists of 604 female histories from 1973 to 2002. The objective is to investigate the effect of climatic conditions on adult survival.

## Explore climatic covariates

First we explore the covariates sea ice extent in summer (SIE.Su), in autumn and winter (SIE.Au and SIE.Wi), in spring (SIE.Sp), annual southern oscillation index (SOI), air temperature in summer (T.Su), in autumn and winter (T.Au and T.Wi) and in spring (T.Sp).

Let us have a look to the correlations between these covariates:

cov <- read.table('cov-petrel.txt',header=T)  
head(cov)

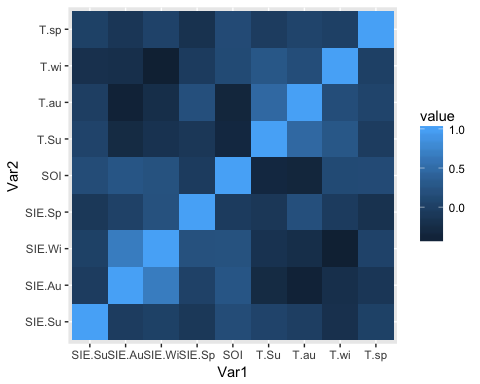
## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi  
## 1 0 341 478 348 0.96 -5.233333 -14.98333 -17.01667  
## 2 189 300 600 341 1.33 -4.150000 -15.08333 -17.85000  
## 3 26 270 337 230 0.06 -5.033333 -16.51667 -16.51667  
## 4 81 256 348 337 -1.14 -4.300000 -13.76667 -15.86667  
## 5 22 207 389 437 -0.29 -4.716667 -14.30000 -15.63333  
## 6 111 215 307 437 -0.26 -5.116667 -15.06667 -16.15000  
## T.sp  
## 1 -6.700000  
## 2 -7.250000  
## 3 -7.683333  
## 4 -7.650000  
## 5 -7.916667  
## 6 -6.766667

round(cor(cov),2)

## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi T.sp  
## SIE.Su 1.00 -0.05 0.01 -0.10 0.15 0.04 -0.02 -0.21 0.01  
## SIE.Au -0.05 1.00 0.67 0.02 0.26 -0.30 -0.43 -0.23 -0.12  
## SIE.Wi 0.01 0.67 1.00 0.21 0.22 -0.18 -0.24 -0.47 0.04  
## SIE.Sp -0.10 0.02 0.21 1.00 -0.06 -0.11 0.19 -0.06 -0.18  
## SOI 0.15 0.26 0.22 -0.06 1.00 -0.34 -0.37 0.13 0.14  
## T.Su 0.04 -0.30 -0.18 -0.11 -0.34 1.00 0.47 0.27 -0.05  
## T.au -0.02 -0.43 -0.24 0.19 -0.37 0.47 1.00 0.17 0.06  
## T.wi -0.21 -0.23 -0.47 -0.06 0.13 0.27 0.17 1.00 0.00  
## T.sp 0.01 -0.12 0.04 -0.18 0.14 -0.05 0.06 0.00 1.00

Visually, with a heatmap:

library(ggplot2)  
library(reshape2)  
qplot(x=Var1, y=Var2, data=melt(cor(cov)), fill=value, geom="tile")



What are the significant correlations?

library(psych)

corr.test(cov)

## Call:corr.test(x = cov)  
## Correlation matrix   
## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi T.sp  
## SIE.Su 1.00 -0.05 0.01 -0.10 0.15 0.04 -0.02 -0.21 0.01  
## SIE.Au -0.05 1.00 0.67 0.02 0.26 -0.30 -0.43 -0.23 -0.12  
## SIE.Wi 0.01 0.67 1.00 0.21 0.22 -0.18 -0.24 -0.47 0.04  
## SIE.Sp -0.10 0.02 0.21 1.00 -0.06 -0.11 0.19 -0.06 -0.18  
## SOI 0.15 0.26 0.22 -0.06 1.00 -0.34 -0.37 0.13 0.14  
## T.Su 0.04 -0.30 -0.18 -0.11 -0.34 1.00 0.47 0.27 -0.05  
## T.au -0.02 -0.43 -0.24 0.19 -0.37 0.47 1.00 0.17 0.06  
## T.wi -0.21 -0.23 -0.47 -0.06 0.13 0.27 0.17 1.00 0.00  
## T.sp 0.01 -0.12 0.04 -0.18 0.14 -0.05 0.06 0.00 1.00  
## Sample Size   
## [1] 29  
## Probability values (Entries above the diagonal are adjusted for multiple tests.)   
## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi T.sp  
## SIE.Su 0.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1  
## SIE.Au 0.79 0.00 0.00 1.00 1.00 1.00 0.64 1.00 1  
## SIE.Wi 0.96 0.00 0.00 1.00 1.00 1.00 1.00 0.37 1  
## SIE.Sp 0.59 0.90 0.28 0.00 1.00 1.00 1.00 1.00 1  
## SOI 0.43 0.17 0.25 0.77 0.00 1.00 1.00 1.00 1  
## T.Su 0.83 0.12 0.35 0.56 0.07 0.00 0.33 1.00 1  
## T.au 0.92 0.02 0.20 0.32 0.05 0.01 0.00 1.00 1  
## T.wi 0.28 0.23 0.01 0.76 0.52 0.15 0.37 0.00 1  
## T.sp 0.97 0.53 0.84 0.34 0.47 0.80 0.77 0.99 0  
##   
## To see confidence intervals of the correlations, print with the short=FALSE option

print(corr.test(cov),short=FALSE)

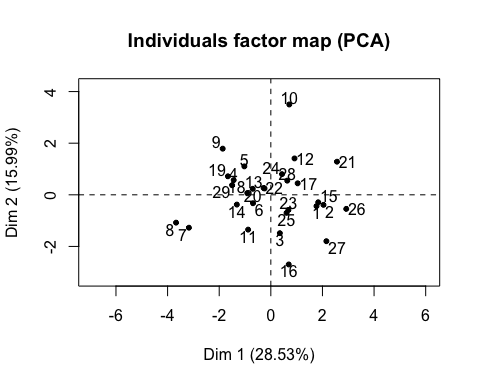
## Call:corr.test(x = cov)  
## Correlation matrix   
## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi T.sp  
## SIE.Su 1.00 -0.05 0.01 -0.10 0.15 0.04 -0.02 -0.21 0.01  
## SIE.Au -0.05 1.00 0.67 0.02 0.26 -0.30 -0.43 -0.23 -0.12  
## SIE.Wi 0.01 0.67 1.00 0.21 0.22 -0.18 -0.24 -0.47 0.04  
## SIE.Sp -0.10 0.02 0.21 1.00 -0.06 -0.11 0.19 -0.06 -0.18  
## SOI 0.15 0.26 0.22 -0.06 1.00 -0.34 -0.37 0.13 0.14  
## T.Su 0.04 -0.30 -0.18 -0.11 -0.34 1.00 0.47 0.27 -0.05  
## T.au -0.02 -0.43 -0.24 0.19 -0.37 0.47 1.00 0.17 0.06  
## T.wi -0.21 -0.23 -0.47 -0.06 0.13 0.27 0.17 1.00 0.00  
## T.sp 0.01 -0.12 0.04 -0.18 0.14 -0.05 0.06 0.00 1.00  
## Sample Size   
## [1] 29  
## Probability values (Entries above the diagonal are adjusted for multiple tests.)   
## SIE.Su SIE.Au SIE.Wi SIE.Sp SOI T.Su T.au T.wi T.sp  
## SIE.Su 0.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1  
## SIE.Au 0.79 0.00 0.00 1.00 1.00 1.00 0.64 1.00 1  
## SIE.Wi 0.96 0.00 0.00 1.00 1.00 1.00 1.00 0.37 1  
## SIE.Sp 0.59 0.90 0.28 0.00 1.00 1.00 1.00 1.00 1  
## SOI 0.43 0.17 0.25 0.77 0.00 1.00 1.00 1.00 1  
## T.Su 0.83 0.12 0.35 0.56 0.07 0.00 0.33 1.00 1  
## T.au 0.92 0.02 0.20 0.32 0.05 0.01 0.00 1.00 1  
## T.wi 0.28 0.23 0.01 0.76 0.52 0.15 0.37 0.00 1  
## T.sp 0.97 0.53 0.84 0.34 0.47 0.80 0.77 0.99 0  
##   
## To see confidence intervals of the correlations, print with the short=FALSE option  
##   
## Confidence intervals based upon normal theory. To get bootstrapped values, try cor.ci  
## lower r upper p  
## SIE.Su-SIE.A -0.41 -0.05 0.32 0.79  
## SIE.Su-SIE.W -0.36 0.01 0.38 0.96  
## SIE.Su-SIE.Sp -0.45 -0.10 0.27 0.59  
## SIE.Su-SOI -0.23 0.15 0.49 0.43  
## SIE.Su-T.Su -0.33 0.04 0.40 0.83  
## SIE.Su-T.au -0.38 -0.02 0.35 0.92  
## SIE.Su-T.wi -0.53 -0.21 0.17 0.28  
## SIE.Su-T.sp -0.36 0.01 0.37 0.97  
## SIE.A-SIE.W 0.40 0.67 0.83 0.00  
## SIE.A-SIE.Sp -0.35 0.02 0.39 0.90  
## SIE.A-SOI -0.12 0.26 0.57 0.17  
## SIE.A-T.Su -0.60 -0.30 0.08 0.12  
## SIE.A-T.au -0.69 -0.43 -0.08 0.02  
## SIE.A-T.wi -0.55 -0.23 0.15 0.23  
## SIE.A-T.sp -0.47 -0.12 0.26 0.53  
## SIE.W-SIE.Sp -0.17 0.21 0.53 0.28  
## SIE.W-SOI -0.16 0.22 0.54 0.25  
## SIE.W-T.Su -0.51 -0.18 0.20 0.35  
## SIE.W-T.au -0.56 -0.24 0.13 0.20  
## SIE.W-T.wi -0.71 -0.47 -0.12 0.01  
## SIE.W-T.sp -0.33 0.04 0.40 0.84  
## SIE.Sp-SOI -0.41 -0.06 0.32 0.77  
## SIE.Sp-T.Su -0.46 -0.11 0.26 0.56  
## SIE.Sp-T.au -0.19 0.19 0.52 0.32  
## SIE.Sp-T.wi -0.42 -0.06 0.31 0.76  
## SIE.Sp-T.sp -0.52 -0.18 0.20 0.34  
## SOI-T.Su -0.63 -0.34 0.03 0.07  
## SOI-T.au -0.65 -0.37 0.00 0.05  
## SOI-T.wi -0.25 0.13 0.47 0.52  
## SOI-T.sp -0.24 0.14 0.48 0.47  
## T.Su-T.au 0.13 0.47 0.72 0.01  
## T.Su-T.wi -0.10 0.27 0.58 0.15  
## T.Su-T.sp -0.41 -0.05 0.32 0.80  
## T.au-T.wi -0.21 0.17 0.51 0.37  
## T.au-T.sp -0.32 0.06 0.41 0.77  
## T.wi-T.sp -0.37 0.00 0.37 0.99

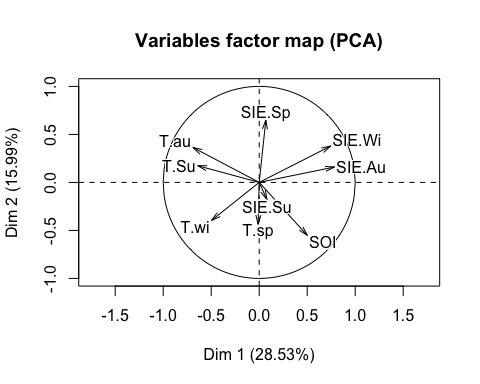
Seems like sea ice extent in autumn and winter are positively correlated, while sea ice extent in autumn and temperature in autumn are negatively correlated.

## PCA on covariates

Let's perform a PCA on this set of covariates:

library(FactoMineR)  
res.pca = PCA(cov,scale.unit=T,graph=T,ncp=9)





Find the covariates associated to each principal component:

dimdesc(res.pca,axes = 1:9)

## $Dim.1  
## $Dim.1$quanti  
## correlation p.value  
## SIE.Au 0.7846343 4.703772e-07  
## SIE.Wi 0.7444495 3.650675e-06  
## SOI 0.5012694 5.604172e-03  
## T.wi -0.4966878 6.129402e-03  
## T.Su -0.6409107 1.798385e-04  
## T.au -0.6912833 3.291814e-05  
##   
##   
## $Dim.2  
## $Dim.2$quanti  
## correlation p.value  
## SIE.Sp 0.6505897 0.000132906  
## SIE.Wi 0.3787178 0.042773054  
## T.wi -0.3960426 0.033438570  
## T.sp -0.4367718 0.017836263  
## SOI -0.5519112 0.001910053  
##   
##   
## $Dim.3  
## $Dim.3$quanti  
## correlation p.value  
## SIE.Su 0.7413696 4.205825e-06  
## T.wi -0.5808892 9.527347e-04  
##   
##   
## $Dim.4  
## $Dim.4$quanti  
## correlation p.value  
## T.sp 0.8228262 4.300265e-08  
## SIE.Su -0.3946619 3.411627e-02  
##   
##   
## $Dim.5  
## $Dim.5$quanti  
## correlation p.value  
## SIE.Sp 0.6329789 0.0002286769  
## SOI 0.4046400 0.0294590693  
## SIE.Su 0.4020185 0.0306294870  
##   
##   
## $Dim.6  
## $Dim.6$quanti  
## correlation p.value  
## T.Su 0.5642562 0.001431237  
## T.wi 0.3861381 0.038548955  
##   
##   
## $Dim.8  
## NULL

#plot(res.pca)

Percentage of variance explained:

res.pca$eig[,3]

## [1] 28.53227 44.52647 57.72122 68.86126 79.32825 89.27306 93.70949  
## [8] 97.95676 100.00000

The loadings:

res.pca$var$cor

## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5  
## SIE.Su 0.08057735 -0.1786150 0.7413696 -0.39466187 0.40201850  
## SIE.Au 0.78463426 0.1601660 -0.1358873 -0.02360527 -0.28475060  
## SIE.Wi 0.74444946 0.3787178 0.1390330 0.26085759 -0.10566944  
## SIE.Sp 0.06940453 0.6505897 -0.2874635 0.14838490 0.63297890  
## SOI 0.50126942 -0.5519112 -0.1726551 0.02166076 0.40463998  
## T.Su -0.64091072 0.1741091 0.1976988 -0.02932482 -0.29134793  
## T.au -0.69128327 0.3642634 0.1570772 0.27691731 0.17674456  
## T.wi -0.49668781 -0.3960426 -0.5808892 -0.03417841 0.08316850  
## T.sp -0.01138968 -0.4367718 0.2940550 0.82282616 0.02694411  
## Dim.6 Dim.7 Dim.8 Dim.9  
## SIE.Su 0.20439145 0.16833601 0.15205046 -0.03696014  
## SIE.Au 0.32726859 0.07959270 0.31757770 0.21046404  
## SIE.Wi 0.33851943 -0.03721296 -0.08556715 -0.28376578  
## SIE.Sp 0.00660304 0.22829056 -0.08806289 0.08176243  
## SOI 0.34355549 -0.29450675 -0.19246270 0.09112650  
## T.Su 0.56425618 0.10752351 -0.30114518 0.11600140  
## T.au 0.20978102 -0.36666032 0.27977904 0.01097100  
## T.wi 0.38613805 0.19899778 0.18999768 -0.16165470  
## T.sp -0.04458262 0.19690112 0.03311646 0.05485501

Re-project each covariate on each principal component:

pcs = res.pca$ind$coord  
round(pcs,2)

## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6 Dim.7 Dim.8 Dim.9  
## 1 1.77 -0.44 -0.78 1.45 -0.08 0.12 -0.97 -0.13 0.85  
## 2 2.04 -0.39 2.70 -0.78 1.49 1.99 -0.44 -0.63 -0.21  
## 3 0.35 -1.49 -0.46 -1.19 -1.33 -1.04 -0.78 -0.45 0.16  
## 4 -1.44 0.57 0.48 -0.86 -0.07 -0.20 -0.01 0.59 -0.01  
## 5 -1.03 1.10 -1.26 -0.40 1.18 -0.56 -0.37 -0.76 -0.33  
## 6 -0.69 -0.33 0.59 0.08 2.17 -0.99 1.00 -0.10 0.37  
## 7 -3.17 -1.28 -0.08 0.21 0.39 0.83 0.91 -0.79 -0.37  
## 8 -3.67 -1.08 -1.54 0.46 1.39 0.75 -0.45 0.80 0.20  
## 9 -1.87 1.79 0.56 2.40 0.67 -1.92 -0.76 -0.42 -0.51  
## 10 0.72 3.50 0.01 -1.05 0.25 0.99 -0.45 0.57 0.30  
## 11 -0.88 -1.35 1.83 -1.38 0.28 0.08 0.51 -0.52 0.43  
## 12 0.92 1.41 -1.34 0.60 0.12 0.57 0.57 -1.13 0.63  
## 13 -0.24 0.25 -0.19 0.61 -2.19 1.61 -0.49 -1.09 -0.63  
## 14 -1.32 -0.38 1.73 0.04 -1.37 -0.51 0.10 0.65 -0.05  
## 15 1.84 -0.29 0.57 0.29 1.08 0.43 -0.21 0.27 0.32  
## 16 0.69 -2.70 -0.83 -0.01 -0.08 -1.06 0.02 0.14 0.20  
## 17 1.04 0.45 -0.63 0.50 0.31 0.52 0.94 0.61 0.12  
## 18 -0.91 0.06 1.18 1.48 -1.31 1.58 0.28 1.16 0.03  
## 19 -1.66 0.72 -0.74 -2.07 -1.05 -0.19 -0.27 0.38 0.33  
## 20 -0.70 0.23 -0.57 -0.99 -1.25 -0.31 0.93 -0.87 -0.01  
## 21 2.56 1.28 -0.60 -0.26 -0.47 -1.17 1.54 0.61 -0.63  
## 22 -0.28 0.27 0.13 -1.18 0.92 -0.31 -0.72 0.59 -1.06  
## 23 0.68 -0.59 -1.73 1.23 0.13 1.37 0.41 -0.03 -0.33  
## 24 0.43 0.80 1.79 1.19 -0.77 -1.32 0.27 -0.05 0.56  
## 25 0.61 -0.70 1.29 0.93 -0.25 -0.66 -0.18 -0.29 -0.39  
## 26 2.92 -0.55 -0.47 -0.47 0.15 -0.83 -0.75 -0.23 -0.15  
## 27 2.15 -1.80 -0.93 0.19 -0.11 0.10 -0.05 0.95 -0.38  
## 28 0.64 0.55 -0.39 -0.99 0.36 0.24 0.05 -0.17 0.05  
## 29 -1.50 0.37 -0.33 -0.03 -0.52 -0.12 -0.61 0.33 0.52

## Model fitting

We're gonna fit various capture-recapture models to the petrel data. We use [RMark](http://www.phidot.org/software/mark/docs/book/pdf/app_3.pdf) because everything can be done in R, and it's cool for reproducible research. But other pieces of software could be used too, like e.g. E-SURGE.

Before fitting capture-recapture models to the data, we check whether the standard Cormack-Jolly-Seber model is fitting the data well. We use the R package R2ucare.

library(R2ucare)  
geese = read\_inp("females\_petrel.inp")  
petrel.ch = geese$encounter\_histories  
freq = geese$sample\_size  
test3sr(petrel.ch, freq)

## $test3sr  
## stat df p\_val sign\_test   
## 29.095 27.000 0.356 0.903   
##   
## $details  
## component stat p\_val signed\_test test\_perf  
## 1 2 0.001 0.975 -0.032 Chi-square  
## 2 3 0.249 0.618 -0.499 Fisher  
## 3 4 0.213 0.644 0.462 Chi-square  
## 4 5 0 1 0 Fisher  
## 5 6 4.174 0.041 -2.043 Chi-square  
## 6 7 0 1 0 Fisher  
## 7 8 0 1 0 Fisher  
## 8 9 0 0 0 None  
## 9 10 1.13 0.288 -1.063 Chi-square  
## 10 11 0 1 0 Fisher  
## 11 12 1.766 0.184 -1.329 Fisher  
## 12 13 1.19 0.275 1.091 Fisher  
## 13 14 0 1 0 Fisher  
## 14 15 1.224 0.269 1.106 Fisher  
## 15 16 2.696 0.101 -1.642 Chi-square  
## 16 17 0 1 0 Fisher  
## 17 18 0 1 0 Fisher  
## 18 19 3.695 0.055 1.922 Fisher  
## 19 20 0 1 0 Fisher  
## 20 21 1.885 0.17 1.373 Fisher  
## 21 22 0.296 0.586 0.544 Fisher  
## 22 23 0 0.984 0 Chi-square  
## 23 24 0 1 0 Fisher  
## 24 25 6.514 0.011 2.552 Fisher  
## 25 26 0.749 0.387 0.865 Chi-square  
## 26 27 0.102 0.749 -0.319 Chi-square  
## 27 28 0 1 0 Fisher  
## 28 29 3.211 0.073 1.792 Fisher

test3sm(petrel.ch, freq)

## $test3sm  
## stat df p\_val   
## 39.260 31.000 0.147   
##   
## $details  
## component stat df p\_val test\_perf  
## 1 2 0.756 1 0.384 Chi-square  
## 2 3 4.883 1 0.027 Chi-square  
## 3 4 0.172 2 0.918 Chi-square  
## 4 5 0 1 1 Fisher  
## 5 6 1.022 1 0.312 Chi-square  
## 6 7 0.748 1 0.387 Chi-square  
## 7 8 0 1 1 Fisher  
## 8 9 0.294 1 0.588 Fisher  
## 9 10 0.939 1 0.333 Chi-square  
## 10 11 2.88 3 0.411 Chi-square  
## 11 12 1.709 1 0.191 Chi-square  
## 12 13 0.19 1 0.663 Chi-square  
## 13 14 0 1 1 Fisher  
## 14 15 5.705 1 0.017 Fisher  
## 15 16 14.009 2 0.001 Chi-square  
## 16 17 0.309 1 0.578 Chi-square  
## 17 18 0.305 1 0.581 Chi-square  
## 18 19 0 1 1 Fisher  
## 19 20 1.337 1 0.248 Chi-square  
## 20 21 0.547 1 0.46 Chi-square  
## 21 22 0 1 1 Fisher  
## 22 23 0 1 1 Fisher  
## 23 24 1.867 1 0.172 Chi-square  
## 24 25 0.657 1 0.417 Fisher  
## 25 26 0.456 1 0.5 Chi-square  
## 26 27 0.212 1 0.645 Chi-square  
## 27 28 0.263 1 0.608 Fisher  
## 28 29 0 0 0 None

test2ct(petrel.ch, freq)

## $test2ct  
## stat df p\_val sign\_test   
## 103.115 27.000 0.000 -8.441   
##   
## $details  
## component dof stat p\_val signed\_test test\_perf  
## 1 2 1 0.013 0.908 0.114 Chi-square  
## 2 3 1 8.1 0.004 -2.846 Fisher  
## 3 4 1 2.599 0.107 -1.612 Chi-square  
## 4 5 1 1.207 0.272 -1.099 Chi-square  
## 5 6 1 1.162 0.281 -1.078 Chi-square  
## 6 7 1 0.499 0.48 -0.706 Chi-square  
## 7 8 1 0.958 0.328 -0.979 Chi-square  
## 8 9 1 0.977 0.323 -0.988 Chi-square  
## 9 10 1 6.397 0.011 -2.529 Chi-square  
## 10 11 1 2.674 0.102 -1.635 Chi-square  
## 11 12 1 8.56 0.003 -2.926 Chi-square  
## 12 13 1 0.056 0.814 -0.237 Chi-square  
## 13 14 1 0.015 0.903 0.122 Chi-square  
## 14 15 1 5.736 0.017 -2.395 Chi-square  
## 15 16 1 5.291 0.021 -2.3 Chi-square  
## 16 17 1 2.057 0.152 -1.434 Chi-square  
## 17 18 1 10.988 0.001 -3.315 Chi-square  
## 18 19 1 7.809 0.005 -2.794 Chi-square  
## 19 20 1 0.149 0.699 -0.386 Chi-square  
## 20 21 1 5.228 0.022 -2.286 Chi-square  
## 21 22 1 9.259 0.002 -3.043 Chi-square  
## 22 23 1 3.826 0.05 -1.956 Chi-square  
## 23 24 1 9.147 0.002 -3.024 Chi-square  
## 24 25 1 0 1 0 Chi-square  
## 25 26 1 6.442 0.011 -2.538 Chi-square  
## 26 27 1 0 0.984 0 Chi-square  
## 27 28 1 3.966 0.046 -1.991 Chi-square

test2cl(petrel.ch, freq)

## $test2cl  
## stat df p\_val   
## 49.741 42.000 0.192   
##   
## $details  
## component dof stat p\_val test\_perf  
## 1 2 1 0 1 Fisher  
## 2 3 1 1.077 0.299 Fisher  
## 3 4 1 1.42 0.233 Chi-square  
## 4 5 1 0.033 0.855 Chi-square  
## 5 6 3 0.246 0.97 Chi-square  
## 6 7 3 0.955 0.812 Chi-square  
## 7 8 2 0.906 0.636 Chi-square  
## 8 9 1 0.101 0.75 Chi-square  
## 9 10 1 0.808 0.369 Chi-square  
## 10 11 3 8.064 0.045 Chi-square  
## 11 12 2 0.545 0.761 Chi-square  
## 12 13 2 0.973 0.615 Chi-square  
## 13 14 1 1.709 0.191 Chi-square  
## 14 15 2 1.416 0.493 Chi-square  
## 15 16 3 7.218 0.065 Chi-square  
## 16 17 3 9.25 0.026 Chi-square  
## 17 18 2 3.995 0.136 Chi-square  
## 18 19 2 4.387 0.112 Chi-square  
## 19 20 1 0.402 0.526 Chi-square  
## 20 21 1 0.545 0.46 Chi-square  
## 21 22 1 0.683 0.408 Chi-square  
## 22 23 1 1.155 0.283 Chi-square  
## 23 24 1 2.093 0.148 Fisher  
## 24 25 1 0.229 0.633 Chi-square  
## 25 26 1 1.319 0.251 Chi-square  
## 26 27 1 0.212 0.645 Chi-square

overall\_CJS(petrel.ch, freq)

## chi2 degree\_of\_freedom p\_value  
## Gof test for CJS model: 221.211 127 0

It sounds like there is a strong trap-dependence effect. Let's deal with it and create an individual time-varying covariate for trap-dependence (see appendix C of the Gentle introduction to Mark):

# let's read in the data:  
library(RMark)

## This is RMark 2.2.0

petrel=convert.inp("females\_petrel")  
petrel.ch <- unlist(strsplit(petrel$ch, ""))  
nocc <- nchar(petrel$ch[1])  
petrel.td <- matrix(as.numeric(petrel.ch), ncol = nocc, byrow = TRUE)  
petrel.td <- petrel.td[, 1:(nocc - 1)]  
petrel.td <- as.data.frame(petrel.td)  
begin.time <- 1974  
names(petrel.td) <- paste('td', (begin.time + 1):(begin.time + nocc - 1), sep = "")  
#head(petrel.td) # dim 430 x 29  
dim(petrel.td)

## [1] 430 29

petrel <- cbind(petrel, petrel.td)  
#head(petrel)

Now process the data:

petrel.processed=process.data(petrel, model="CJS", begin.time=1974)

Create the default design matrix:

design.p=list(time.varying=c('td')) #td  
design.parameters <- list(p=design.p)  
petrel.ddl <- make.design.data(petrel.processed,parameters=design.parameters)

Standardize the covariates:

# standardize  
moy = apply(cov,2,mean)  
prec = apply(cov,2,sd)  
moymat = matrix(rep(moy,nrow(cov)),ncol=ncol(cov),byrow=T)  
precmat = matrix(rep(prec,nrow(cov)),ncol=ncol(cov),byrow=T)  
covstar = (cov - moymat)/precmat  
#apply(covstar,2,mean)  
#apply(covstar,2,sd)  
cov = covstar

Add raw covariates to the design matrix:

petrel.ddl$Phi$x1=0  
petrel.ddl$Phi$x2=0  
petrel.ddl$Phi$x3=0  
petrel.ddl$Phi$x4=0  
petrel.ddl$Phi$x5=0  
petrel.ddl$Phi$x6=0  
petrel.ddl$Phi$x7=0  
petrel.ddl$Phi$x8=0  
petrel.ddl$Phi$x9=0  
ind=1  
for (i in 1974:2002){  
 petrel.ddl$Phi$x1[petrel.ddl$Phi$time==i]=cov[ind,1]  
 petrel.ddl$Phi$x2[petrel.ddl$Phi$time==i]=cov[ind,2]  
 petrel.ddl$Phi$x3[petrel.ddl$Phi$time==i]=cov[ind,3]  
 petrel.ddl$Phi$x4[petrel.ddl$Phi$time==i]=cov[ind,4]  
 petrel.ddl$Phi$x5[petrel.ddl$Phi$time==i]=cov[ind,5]  
 petrel.ddl$Phi$x6[petrel.ddl$Phi$time==i]=cov[ind,6]  
 petrel.ddl$Phi$x7[petrel.ddl$Phi$time==i]=cov[ind,7]  
 petrel.ddl$Phi$x8[petrel.ddl$Phi$time==i]=cov[ind,8]  
 petrel.ddl$Phi$x9[petrel.ddl$Phi$time==i]=cov[ind,9]  
 ind=ind+1  
}

Specify the effects on survival and detection probabilities:

# for survival probabilities  
Phidot=list(formula=~1) # constant  
Phitime=list(formula=~time) # time  
PhiCov=list(formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9) # all covariates  
# Define range of models for detection probabilities  
pdot=list(formula=~td) # constant, with trap-dependence  
ptime=list(formula=~time+td) # additive effect of time and trap-dependence (no interaction because of severe identifiability issues Gimenez et al. 2003)

Fit models:

# phi,p  
phip = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phidot,p=pdot),output = FALSE,delete=T)  
# phit,p  
phitp = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phitime,p=pdot),output = FALSE,delete=T)  
# phi,pt  
phipt = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phidot,p=ptime),output = FALSE,delete=T)  
# phit,pt  
phitpt = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phitime,p=ptime),output = FALSE,delete=T)

Compare models

collect.models()

## model npar AICc DeltaAICc weight Deviance  
## 4 Phi(~time)p(~time + td) 59 6535.213 0.00000 1.000000e+00 6414.843  
## 2 Phi(~1)p(~time + td) 31 6580.016 44.80257 1.867433e-10 6517.358  
## 3 Phi(~time)p(~td) 31 7035.204 499.99017 0.000000e+00 6972.546  
## 1 Phi(~1)p(~td) 3 7141.081 605.86732 0.000000e+00 7135.073

Clearly, there is time variation in the detection process. Also, it's worth investigating further time variation in survival.

Now, let's fit a model with all covariates:

phixpt = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=PhiCov,p=ptime),output = FALSE,delete=T)

And have a look to the parameter estimates:

phixpt$results$beta

## estimate se lcl ucl  
## Phi:(Intercept) 3.0215027 0.1266500 2.7732686 3.2697368  
## Phi:x1 -0.0214576 0.1238148 -0.2641346 0.2212193  
## Phi:x2 0.5045058 0.2438022 0.0266534 0.9823581  
## Phi:x3 -0.5050225 0.2135759 -0.9236313 -0.0864136  
## Phi:x4 -0.1875442 0.1633865 -0.5077816 0.1326933  
## Phi:x5 -0.3384226 0.1285907 -0.5904604 -0.0863848  
## Phi:x6 0.0366735 0.1194053 -0.1973609 0.2707079  
## Phi:x7 0.4200931 0.2218857 -0.0148029 0.8549890  
## Phi:x8 -0.5388508 0.1450647 -0.8231777 -0.2545240  
## Phi:x9 -0.1858322 0.1744092 -0.5276742 0.1560098  
## p:(Intercept) -0.2762133 0.3518067 -0.9657543 0.4133278  
## p:time1976 1.4053121 0.5015022 0.4223678 2.3882563  
## p:time1977 1.1939267 0.4353984 0.3405459 2.0473075  
## p:time1978 -1.3620787 0.3974020 -2.1409866 -0.5831708  
## p:time1979 0.8851171 0.4070317 0.0873350 1.6828992  
## p:time1980 -1.3972306 0.4109121 -2.2026184 -0.5918429  
## p:time1981 -0.8393120 0.4093691 -1.6416754 -0.0369486  
## p:time1982 -1.4862563 0.4314896 -2.3319760 -0.6405367  
## p:time1983 1.0600671 0.4128135 0.2509526 1.8691817  
## p:time1984 0.4538590 0.3939755 -0.3183330 1.2260510  
## p:time1985 -1.0118973 0.3845834 -1.7656808 -0.2581137  
## p:time1986 0.0048467 0.3855301 -0.7507922 0.7604857  
## p:time1987 -1.0459573 0.3890010 -1.8083993 -0.2835153  
## p:time1988 0.5125888 0.3873746 -0.2466655 1.2718431  
## p:time1989 1.1524762 0.4016411 0.3652597 1.9396928  
## p:time1990 -0.3716563 0.3782383 -1.1130033 0.3696907  
## p:time1991 0.6520759 0.3844498 -0.1014458 1.4055976  
## p:time1992 0.6619091 0.3826238 -0.0880335 1.4118518  
## p:time1993 0.6817374 0.3839835 -0.0708703 1.4343450  
## p:time1994 0.0532153 0.3753604 -0.6824912 0.7889217  
## p:time1995 0.7756711 0.3815839 0.0277667 1.5235755  
## p:time1996 1.6026013 0.4091297 0.8007071 2.4044955  
## p:time1997 1.4893316 0.4098044 0.6861150 2.2925482  
## p:time1998 0.4982784 0.3783386 -0.2432652 1.2398220  
## p:time1999 1.4753445 0.4081701 0.6753310 2.2753580  
## p:time2000 0.5118000 0.3840496 -0.2409372 1.2645371  
## p:time2001 -0.1193068 0.3820731 -0.8681701 0.6295565  
## p:time2002 -0.0079716 0.3829367 -0.7585277 0.7425844  
## p:time2003 0.2303389 0.3863065 -0.5268217 0.9874996  
## p:td 0.7281680 0.0827993 0.5658814 0.8904547

The covariates are in that order: SIE.Su (x1), SIE.Au (x2), SIE.Wi (x3), SIE.Sp (x4), SOI (x5), T.Su (x6), T.au (x7), T.wi (x8) and T.sp (x9). Remember, from our preliminary exploration step above, we know that covariates 2 and 3 are highly positively correlated. However by inspecting the estimates here, these covariates seem to have an opposite effect on survival!

## Standard forward stepwise covariate selection approach

Following a referee's suggestion, we perform here a forward covariate selection. Bearing in mind that we found strong correlation among covariates (see above), we do not recommend performing this analysis without first dealing with the multicollinearity issue.

In the first step of the analysis, we consider each covariate separately:

Phix1=list(formula=~x1)  
Phix2=list(formula=~x2)  
Phix3=list(formula=~x3)  
Phix4=list(formula=~x4)  
Phix5=list(formula=~x5)  
Phix6=list(formula=~x6)  
Phix7=list(formula=~x7)  
Phix8=list(formula=~x8)  
Phix9=list(formula=~x9)  
phix1 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix1,p=ptime),output = FALSE,delete=T)  
phix2 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix2,p=ptime),output = FALSE,delete=T)  
phix3 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix3,p=ptime),output = FALSE,delete=T)  
phix4 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix4,p=ptime),output = FALSE,delete=T)  
phix5 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix5,p=ptime),output = FALSE,delete=T)  
phix6 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix6,p=ptime),output = FALSE,delete=T)  
phix7 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix7,p=ptime),output = FALSE,delete=T)  
phix8 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix8,p=ptime),output = FALSE,delete=T)  
phix9 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix9,p=ptime),output = FALSE,delete=T)

We now use ANODEV to test the significance of these covariates:

# get info on model with time-dependent survival  
devtime = phitpt$results$lnl  
npartime = phitpt$results$npar  
  
# get info on model with constant survival  
devct = phipt$results$lnl  
nparct = phipt$results$npar  
  
# test each covariate:  
stat = rep(NA,9)  
df1 = rep(NA,9)  
df2 = rep(NA,9)  
for (i in 1:9){  
 name = paste('phix',i,sep="")  
 devco = get(name)$results$lnl  
 nparco = get(name)$results$npar  
 num = (devct - devco)/(nparco-nparct)  
 den = (devco - devtime)/(npartime-nparco)   
 stat[i] <- num/den  
 df1[i] <- nparco-nparct  
 df2[i] <- npartime-nparco  
}  
# calculate p-value  
pval = 1-pf(stat,df1,df2)  
stat

## [1] 1.43111265 0.01122451 0.75434906 1.85305569 7.96593098 0.56817195  
## [7] 0.11635867 7.75091055 1.77150769

df1

## [1] 1 1 1 1 1 1 1 1 1

df2

## [1] 27 27 27 27 27 27 27 27 27

pval

## [1] 0.241982884 0.916408601 0.392757731 0.184681802 0.008838179 0.457509768  
## [7] 0.735658189 0.009686743 0.194326441

It seems like SOI () and temperature in winter () have a significant effect. In step 2 of the analysis, we keep these two covariates and test the significance of the other covariates:

Phix0=list(formula=~x5+x8) # constant model in the current ANODEV  
Phix1=list(formula=~x5+x8+x1)  
Phix2=list(formula=~x5+x8+x2)  
Phix3=list(formula=~x5+x8+x3)  
Phix4=list(formula=~x5+x8+x4)  
Phix5=list(formula=~x5+x8+x6)  
Phix6=list(formula=~x5+x8+x7)  
Phix7=list(formula=~x5+x8+x9)  
phix58 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix1,p=ptime),output = FALSE,delete=T)  
phix581 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix1,p=ptime),output = FALSE,delete=T)  
phix582 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix2,p=ptime),output = FALSE,delete=T)  
phix583 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix3,p=ptime),output = FALSE,delete=T)  
phix584 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix4,p=ptime),output = FALSE,delete=T)  
phix585 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix5,p=ptime),output = FALSE,delete=T)  
phix586 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix6,p=ptime),output = FALSE,delete=T)  
phix587 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phix7,p=ptime),output = FALSE,delete=T)

Then, we perform ANODEV tests to assess the significance of the remaining covariates in presence of both and .

stat = rep(NA,7)  
df1 = rep(NA,7)  
df2 = rep(NA,7)  
for (i in 1:7){  
 name = 'phix58'  
 devct = get(name)$results$lnl  
 nparct = get(name)$results$npar  
 namex = paste('phix58',i,sep="")  
 devco = get(namex)$results$lnl  
 nparco = get(namex)$results$npar  
 df1[i] <- 1 # we add a single covariate  
 df2[i] <- npartime-nparco  
 num = (devct - devco)/df1[i]  
 den = (devco - devtime)/df2[i]   
 if (devct == devco) stat[i] <- 0 # it happens for covariate x1  
 if (devct != devco) stat[i] <- num/den  
}  
  
pval = 1-pf(stat,df1,df2)  
stat

## [1] 0.0000000 0.1697841 2.2834398 0.8267684 -0.3533329 0.3438980  
## [7] -0.3574468

df1

## [1] 1 1 1 1 1 1 1

df2

## [1] 25 25 25 25 25 25 25

pval

## [1] 1.0000000 0.6838155 0.1432996 0.3718918 1.0000000 0.5628460 1.0000000

There is no more significant covariates.

## P2CR analysis

In this section, we show how to perform a P2CR analysis. First, we amend the design matrix we built before, and add the coordinates of the raw covariates on the principal components:

petrel.ddl$Phi$pc1=0  
petrel.ddl$Phi$pc2=0  
petrel.ddl$Phi$pc3=0  
petrel.ddl$Phi$pc4=0  
petrel.ddl$Phi$pc5=0  
petrel.ddl$Phi$pc6=0  
petrel.ddl$Phi$pc7=0  
petrel.ddl$Phi$pc8=0  
petrel.ddl$Phi$pc9=0  
ind=1  
for (i in 1974:2002){  
 petrel.ddl$Phi$pc1[petrel.ddl$Phi$time==i]=pcs[ind,1]  
 petrel.ddl$Phi$pc2[petrel.ddl$Phi$time==i]=pcs[ind,2]  
 petrel.ddl$Phi$pc3[petrel.ddl$Phi$time==i]=pcs[ind,3]  
 petrel.ddl$Phi$pc4[petrel.ddl$Phi$time==i]=pcs[ind,4]  
 petrel.ddl$Phi$pc5[petrel.ddl$Phi$time==i]=pcs[ind,5]  
 petrel.ddl$Phi$pc6[petrel.ddl$Phi$time==i]=pcs[ind,6]  
 petrel.ddl$Phi$pc7[petrel.ddl$Phi$time==i]=pcs[ind,7]  
 petrel.ddl$Phi$pc8[petrel.ddl$Phi$time==i]=pcs[ind,8]  
 petrel.ddl$Phi$pc9[petrel.ddl$Phi$time==i]=pcs[ind,9]  
 ind=ind+1  
}

In the first step of the P2CR analysis, we consider each PC separately:

Phipc1=list(formula=~pc1)  
Phipc2=list(formula=~pc2)  
Phipc3=list(formula=~pc3)  
Phipc4=list(formula=~pc4)  
Phipc5=list(formula=~pc5)  
Phipc6=list(formula=~pc6)  
Phipc7=list(formula=~pc7)  
Phipc8=list(formula=~pc8)  
Phipc9=list(formula=~pc9)  
phipc1 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc1,p=ptime),output = FALSE,delete=T)  
phipc2 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc2,p=ptime),output = FALSE,delete=T)  
phipc3 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc3,p=ptime),output = FALSE,delete=T)  
phipc4 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc4,p=ptime),output = FALSE,delete=T)  
phipc5 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc5,p=ptime),output = FALSE,delete=T)  
phipc6 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc6,p=ptime),output = FALSE,delete=T)  
phipc7 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc7,p=ptime),output = FALSE,delete=T)  
phipc8 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc8,p=ptime),output = FALSE,delete=T)  
phipc9 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc9,p=ptime),output = FALSE,delete=T)

We now use ANODEV to to test the significance of these PCs:

# get info on model with time-dependent survival  
devtime = phitpt$results$lnl  
npartime = phitpt$results$npar  
  
# get info on model with constant survival  
devct = phipt$results$lnl  
nparct = phipt$results$npar  
  
# test each PC:  
stat = rep(NA,9)  
df1 = rep(NA,9)  
df2 = rep(NA,9)  
for (i in 1:9){  
 name = paste('phipc',i,sep="")  
 devco = get(name)$results$lnl  
 nparco = get(name)$results$npar  
 num = (devct - devco)/(nparco-nparct)  
 den = (devco - devtime)/(npartime-nparco)   
 stat[i] <- num/den  
 df1[i] <- nparco-nparct  
 df2[i] <- npartime-nparco  
}  
# calculate p-value  
pval = 1-pf(stat,df1,df2)  
stat

## [1] 0.4561618 2.0348053 7.3439558 3.0089184 3.0594652 2.4351876 0.1359247  
## [8] 0.3111153 0.7808127

df1

## [1] 1 1 1 1 1 1 1 1 1

df2

## [1] 27 27 27 27 27 27 27 27 27

pval

## [1] 0.50516694 0.16519557 0.01154684 0.09421181 0.09162924 0.13028569  
## [7] 0.71524166 0.58159232 0.38469346

We can reject the null hypothesis that PC3 has no effect on survival.

In step 2 of the P2CR, we keep PC3 and test the significance of the other PCs:

Phipc1=list(formula=~pc1+pc3)  
Phipc2=list(formula=~pc2+pc3)  
Phipc3=list(formula=~pc4+pc3)  
Phipc4=list(formula=~pc5+pc3)  
Phipc5=list(formula=~pc6+pc3)  
Phipc6=list(formula=~pc7+pc3)  
Phipc7=list(formula=~pc8+pc3)  
Phipc8=list(formula=~pc9+pc3)  
phipc11 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc1,p=ptime),output = FALSE,delete=T)  
phipc21 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc2,p=ptime),output = FALSE,delete=T)  
phipc31 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc3,p=ptime),output = FALSE,delete=T)  
phipc41 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc4,p=ptime),output = FALSE,delete=T)  
phipc51 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc5,p=ptime),output = FALSE,delete=T)  
phipc61 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc6,p=ptime),output = FALSE,delete=T)  
phipc71 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc7,p=ptime),output = FALSE,delete=T)  
phipc81 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc8,p=ptime),output = FALSE,delete=T)

stat = rep(NA,8)  
df1 = rep(NA,8)  
df2 = rep(NA,8)  
for (i in 1:8){  
 name = paste('phipc',3,sep="")  
 devct = get(name)$results$lnl  
 nparct = get(name)$results$npar  
 namex = paste('phipc',paste(i,'1',sep=""),sep="")  
 devco = get(namex)$results$lnl  
 nparco = get(namex)$results$npar  
 num = (devct - devco)/(nparco-nparct)  
 den = (devco - devtime)/(npartime-nparco)   
 stat[i] <- num/den  
 df1[i] <- nparco-nparct  
 df2[i] <- npartime-nparco  
  
}  
  
pval = 1-pf(stat,df1,df2)  
stat

## [1] 0.115032061 2.935263243 4.629627302 2.517956493 3.502807470 0.275952354  
## [7] 0.006486012 0.723243675

df1

## [1] 1 1 1 1 1 1 1 1

df2

## [1] 26 26 26 26 26 26 26 26

pval

## [1] 0.73721058 0.09856598 0.04088835 0.12464492 0.07255722 0.60381802  
## [7] 0.93642787 0.40284611

Now PC4 is significant according the ANODEV (remember that PC3 was removed from the list).

In step 3 of the P2CR analysis, we reiterate the process, that is we test the significance of the other PCs in presence of PC3 and PC4:

Phipc1=list(formula=~pc1+pc3+pc4)  
Phipc2=list(formula=~pc2+pc3+pc4)  
Phipc3=list(formula=~pc5+pc3+pc4)  
Phipc4=list(formula=~pc6+pc3+pc4)  
Phipc5=list(formula=~pc7+pc3+pc4)  
Phipc6=list(formula=~pc8+pc3+pc4)  
Phipc7=list(formula=~pc9+pc3+pc4)  
phipc12 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc1,p=ptime),output = FALSE,delete=T)  
phipc22 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc2,p=ptime),output = FALSE,delete=T)  
phipc32 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc3,p=ptime),output = FALSE,delete=T)  
phipc42 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc4,p=ptime),output = FALSE,delete=T)  
phipc52 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc5,p=ptime),output = FALSE,delete=T)  
phipc62 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc6,p=ptime),output = FALSE,delete=T)  
phipc72 = mark(petrel.processed,petrel.ddl,model.parameters=list(Phi=Phipc7,p=ptime),output = FALSE,delete=T)

What does the ANODEV tell us?

stat = rep(NA,7)  
df1 = rep(NA,7)  
df2 = rep(NA,7)  
for (i in 1:7){  
 name = paste('phipc',31,sep="")  
 devct = get(name)$results$lnl  
 nparct = get(name)$results$npar  
 namex = paste('phipc',paste(i,'2',sep=""),sep="")  
 devco = get(namex)$results$lnl  
 nparco = get(namex)$results$npar  
 num = (devct - devco)/(nparco-nparct)  
 den = (devco - devtime)/(npartime-nparco)   
 stat[i] <- num/den  
 df1[i] <- nparco-nparct  
 df2[i] <- npartime-nparco  
  
}  
  
pval = 1-pf(stat,df1,df2)  
stat

## [1] 0.074403780 1.878326793 1.383423294 0.547418815 0.235266864 0.002265893  
## [7] 1.282105461

df1

## [1] 1 1 1 1 1 1 1

df2

## [1] 25 25 25 25 25 25 25

pval

## [1] 0.7872701 0.1827058 0.2505979 0.4662660 0.6318690 0.9624122 0.2682518

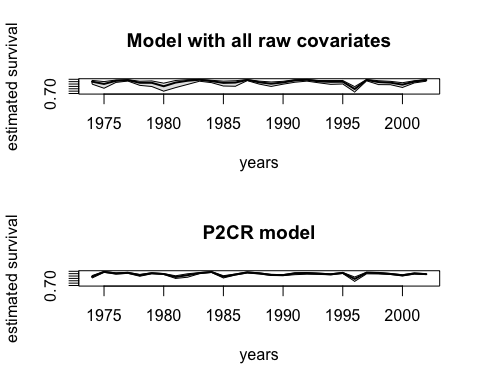
No more significant PC, the algorithm stops here.

## Post-process results

We will make two plots, one with time-varying survival estimates, and another oneto illustrate the relationship between survival and the selected PCs.

First, a figure displaying the time variation in survival according to a model with all raw covariates and the PC2R model:

#phit\_mle <- phitpt$results$real[1:29,]  
phicov\_mle <- phixpt$results$real[1:29,]  
phipca\_mle <- phipc31$results$real[1:29,]  
# Make a 6x6 inch image at 300dpi  
#ppi <- 300  
#png("time\_survival\_allcov.png", width=6\*ppi, height=6\*ppi, res=ppi)  
par(mfrow=c(2,1))  
plot(1974:2002,phicov\_mle[,1],lwd=2,col='black',type='n',ylim=c(0.7,1),xlab='years',ylab='estimated survival',main='Model with all raw covariates')  
polygon(x=c(1974:2002, rev(1974:2002)),y=c(phicov\_mle[,3], rev(phicov\_mle[,4])),col='grey90')  
lines(1974:2002,phicov\_mle[,1],lwd=2,col='black')  
#dev.off()  
#png("time\_survival\_p2cr.png", width=6\*ppi, height=6\*ppi, res=ppi)  
plot(1974:2002,phipca\_mle[,1],lwd=2,col='black',type='n',ylim=c(0.7,1),xlab='years',ylab='estimated survival',main='P2CR model')  
polygon(x=c(1974:2002, rev(1974:2002)),y=c(phipca\_mle[,3], rev(phipca\_mle[,4])),col='grey90')  
lines(1974:2002,phipca\_mle[,1],lwd=2,col='black')



#dev.off()

Second, a figure displaying the relationship between survival and the PCs selected by the P2CR analysis.

Get the coefficient estimates for each PC and the intercept:

phipc31$results$beta[1:3,]

## estimate se lcl ucl  
## Phi:(Intercept) 2.9065529 0.0930352 2.7242038 3.0889019  
## Phi:pc4 -0.3179566 0.0937601 -0.5017265 -0.1341868  
## Phi:pc3 0.4987768 0.1117004 0.2798440 0.7177096

Get estimates of recapture probabilities:

# recapture given no recapture before  
lp1=phipc31$results$beta$estimate[4] + phipc31$results$beta$estimate[5:32]  
p1 = 1/(1+exp(-lp1))  
p1

## [1] 0.6849745 0.6950768 0.1598928 0.6532744 0.1491627 0.2186211 0.1380853  
## [8] 0.6799227 0.5559287 0.2101187 0.4365715 0.2081895 0.5650883 0.7013976  
## [15] 0.3322994 0.5843023 0.5899872 0.6135580 0.4454382 0.6270110 0.7892106  
## [22] 0.7614156 0.5662430 0.7626804 0.5415123 0.3768221 0.4066764 0.4896956

# recapture given recapture before  
lp2=phipc31$results$beta$estimate[4]+phipc31$results$beta$estimate[5:32]+phipc31$results$beta$estimate[33]  
p2 = 1/(1+exp(-lp2))  
p2

## [1] 0.8225428 0.8293326 0.2886225 0.8006571 0.2720511 0.3736061 0.2545782  
## [8] 0.8191144 0.7274251 0.3618676 0.6228950 0.3591787 0.7347352 0.8335369  
## [15] 0.5147804 0.7497731 0.7541473 0.7719290 0.6313058 0.7818291 0.8886590  
## [22] 0.8718481 0.7356502 0.8726254 0.7157292 0.5631322 0.5936851 0.6716636

# get min/max for p1 with SEs  
ind.min = which.min(p1) # index min p1  
ind.max = which.max(p1) # index max p1  
varlp1 = phipc31$results$beta$se[4]^2 + phipc31$results$beta$se[5:32]^2 # var of p1 on logit scale  
lp1mi = lp1[ind.min]  
varlp1mi = varlp1[ind.min]  
library(msm)  
sep1mi = deltamethod(~ 1/(1+exp(-x1)), lp1mi, varlp1mi)  
min(p1)

## [1] 0.1380853

sep1mi

## [1] 0.06614093

lp1ma = lp1[ind.max]  
varlp1ma = varlp1[ind.max]  
sep1ma = deltamethod(~ 1/(1+exp(-x1)), lp1ma, varlp1ma)  
max(p1)

## [1] 0.7892106

sep1ma

## [1] 0.08957705

# get min/max for p2 with SEs  
ind.min = which.min(p2) # index min p2  
ind.max = which.max(p2) # index max p2  
varlp2 = phipc31$results$beta$se[4]^2 + phipc31$results$beta$se[5:32]^2 + phipc31$results$beta$estimate[33]^2# var of p2 on logit scale  
lp2mi = lp2[ind.min]  
varlp2mi = varlp2[ind.min]  
sep2mi = deltamethod(~ 1/(1+exp(-x1)), lp2mi, varlp2mi)  
min(p2)

## [1] 0.2545782

sep2mi

## [1] 0.1781993

lp2ma = lp2[ind.max]  
varlp2ma = varlp2[ind.max]  
sep2ma = deltamethod(~ 1/(1+exp(-x1)), lp2ma, varlp2ma)  
max(p2)

## [1] 0.888659

sep2ma

## [1] 0.09191182

Get confidence intervals using the delta-method:

library(msm)  
PC3 = pcs[,3]  
PC4 = pcs[,4]  
phi\_SE3 = matrix(0, nrow = 29, ncol = 1)  
estmean3 <- c(2.9065503,0.4987728)  
estvar3 <- diag(c(0.0930351,0.1117004)^2)  
phi\_SE4 = matrix(0, nrow = 29, ncol = 1)  
estmean4 <- c(2.9065503,-0.3179579)  
estvar4 <- diag(c(0.0930351,0.0937603)^2)  
for (i in 1:29){  
 temp3 <- PC3[i]  
 temp4 <- PC4[i]  
 phi\_SE3[i,] <- deltamethod(~ x1+x2\*temp3, estmean3, estvar3)  
 phi\_SE4[i,] <- deltamethod(~ x1+x2\*temp4, estmean4, estvar4)  
}  
  
ilogitphi3 <- estmean3[1] + estmean3[2] \* PC3  
ilogitphi3lb <- ilogitphi3 - 1.96 \* as.vector(phi\_SE3)  
ilogitphi3ub <- ilogitphi3 + 1.96 \* as.vector(phi\_SE3)  
phi3lb <- 1/(1+exp(-(ilogitphi3lb)))  
phi3ub <- 1/(1+exp(-(ilogitphi3ub)))  
phi3 <- 1/(1+exp(-(ilogitphi3)))  
  
ilogitphi4 <- estmean4[1] + estmean4[2] \* PC4  
ilogitphi4lb <- ilogitphi4 - 1.96 \* as.vector(phi\_SE4)  
ilogitphi4ub <- ilogitphi4 + 1.96 \* as.vector(phi\_SE4)  
phi4lb <- 1/(1+exp(-(ilogitphi4lb)))  
phi4ub <- 1/(1+exp(-(ilogitphi4ub)))  
phi4 <- 1/(1+exp(-(ilogitphi4)))

Before plotting the survival as a function of the PC values, we need to find out about the raw covariates that were used to build these PCs:

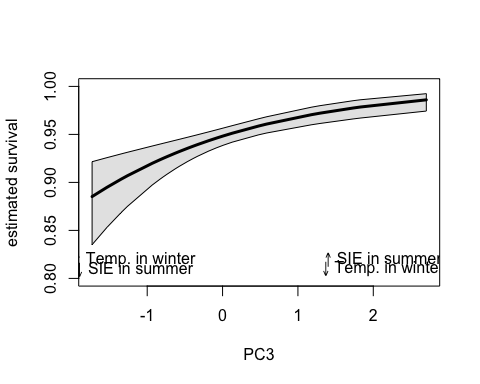
dimdesc(res.pca,axes = c(3:4))

## $Dim.3  
## $Dim.3$quanti  
## correlation p.value  
## SIE.Su 0.7413696 4.205825e-06  
## T.wi -0.5808892 9.527347e-04  
##   
##   
## $Dim.4  
## $Dim.4$quanti  
## correlation p.value  
## T.sp 0.8228262 4.300265e-08  
## SIE.Su -0.3946619 3.411627e-02

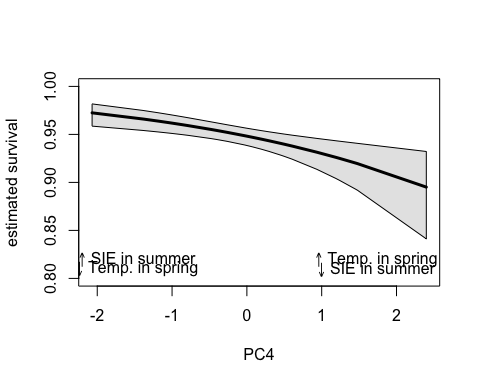
High (resp. low) values of PC3 mean high (resp. low) values of SIE in summer and low (resp. high) values of temperature in winter. High (resp. low) values of PC4 mean high (resp. low) values of temperature in spring and low (resp. high) values of SIE in summer.

Now we can plot the survival - PC relationships, and add the interpretation of the PCs:

# Make a 6x6 inch image at 300dpi  
#ppi <- 300  
#png("pc3\_survival.png", width=6\*ppi, height=6\*ppi, res=ppi)  
ord<-order(PC3)  
plot(PC3[ord],phi3[ord],lwd=3,col='black',type='n',xlab='PC3',ylab='estimated survival',main='',ylim=c(0.8,1))  
polygon(x=c(PC3[ord], rev(PC3[ord])),y=c(phi3lb[ord], rev(phi3ub[ord])),col='grey90')  
lines(PC3[ord],phi3[ord],lwd=3,col='black')  
text(-1.2,0.82,expression('' %up% 'Temp. in winter'),cex=1)  
text(-1.2,0.81,expression('' %down% 'SIE in summer'),cex=1)  
text(2.1,0.82,expression('' %up% 'SIE in summer'),cex=1)  
text(2.1,0.81,expression('' %down% 'Temp. in winter'),cex=1)



ord<-order(PC4)  
#dev.off()  
#png("pc4\_survival.png", width=6\*ppi, height=6\*ppi, res=ppi)  
plot(PC4[ord],phi4[ord],lwd=3,col='black',type='n',xlab='PC4',ylab='estimated survival',main='',ylim=c(0.8,1))  
polygon(x=c(PC4[ord], rev(PC4[ord])),y=c(phi4lb[ord], rev(phi4ub[ord])),col='grey90')  
lines(PC4[ord],phi4[ord],lwd=3,col='black')  
text(-1.5,0.82,expression('' %up% 'SIE in summer'),cex=1)  
text(-1.5,0.81,expression('' %down% 'Temp. in spring'),cex=1)  
text(1.7,0.82,expression('' %up% 'Temp. in spring'),cex=1)  
text(1.7,0.81,expression('' %down% 'SIE in summer'),cex=1)



#dev.off()