2021 SISCER APC Course, R Notes Set 3

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2021-07-01

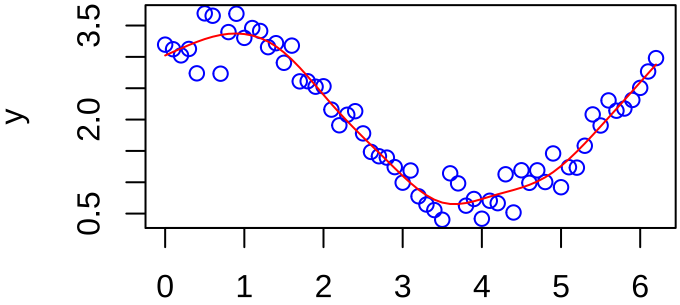
Outline

In these notes we demonstrate the use of

- Splines
- INLA with ANOVA models
- INLA with random walk models

Spline example

```
x <- seq(0, 2 * pi, 0.1)
y <- 2 + sin(x) + cos(x) + rnorm(length(x), 0, 0.2)
par(mfrow = c(1, 1))
plot(y ~ x, col = "blue")
nknot <- 10
knotloc <- seq(0.1, 2 * pi - 0.1, , nknot)
library(Epi)
mod <- lm(y ~ Ns(x, knots = knotloc))
points(x, fitted.values(mod), type = "l", col = "red")
```



Danish male lung cancer incidence data

```
library(Epi)
data(lungDK)
class(lungDK)
## [1] "data.frame"
attach(lungDK)
dftempEpi = data.frame(D = lungDK$D, Y = lungDK$Y,
    A = 37.5 + 5 * ((lungDK$A5 - min(lungDK$A5))/5 +
        1), P = 1945.5 + 5 * (lungDK$P5 - min(lungDK$P5))/5 +
        1)
```

Massaging the data into a convenient form

Sum over the upper and lower triangles in the Lexis diagram to get data in age-period squares, see Carstensen (2007).

```
names(dftempEpi)
## [1] "D" "Y" "A" "P"
head(dftempEpi, 2)
##    D        Y    A      P
## 1 52 336233.8 42.5 1946.5
## 2 28 357812.7 42.5 1946.5
# Sum over upper and lower triangles
dfEpi = aggregate(dftempEpi[, c("D", "Y")], by = list(A = dftemp
    P = dftempEpi$P), sum)
head(dfEpi, 2)
##      A      P   D        Y
## 1 42.5 1946.5  80 694046.5
## 2 47.5 1946.5 135 622256.7
```

Spline models

Rather than a factor model we fit spline smoothers in age and period,

$$E[Y_{ap}] = N_{ap} \exp[f(a) + g(p)].$$

In the `apc.fit` function various constraint options for identifying second differences are available (two of these are described below).

The `model="ns"` refers to natural splines and `npar=5` the number of degrees of freedom in the spline.

Spline models

We illustrate two parameterizations (which give the same fits):

- `ACP`: ML-estimates. Age-effects as rates for the reference cohort. Cohort effects as rate ratios relative to the reference cohort. Period effects constrained to be 0 on average with 0 slope.
- `Ad-C-P`: Age effects are rates for the reference cohort in the Age-drift model (cohort drift). Cohort effects are from the model with cohort alone, using log(fitted values) from the Age-drift model as offset. Period effects are from the model with period alone using log(fitted values) from the cohort model as offset.

In the models below, note that the deviances and the degrees of freedom of the Age-Period model are different from the factor version.

APC spline model: parameterization 1

```
fit2 <- apc.fit(dfEpi, npar = 5, model = "ns", dr.extr = "Holford",
    parm = "ACP", scale = 10^3)
## NOTE: npar is specified as:
## A P C
## 5 5 5
## [1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
##               Model Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
## 1               Age      105 15242.0306       NA        NA            NA
## 2         Age-drift      104  6563.9857        1 8678.0449  0.000000e+00
## 3        Age-Cohort      101  1016.3729        3 5547.6128  0.000000e+00
## 4 Age-Period-Cohort       98   419.2548        3  597.1181 4.247733e-129
## 5        Age-Period      101  2910.5114        3 2491.2565  0.000000e+00
## 6         Age-drift      104  6563.9857        3 3653.4743  0.000000e+00
##   Test dev/df    H0
## 1          NA
## 2   8678.0449 zero drift
## 3   1849.2043 Coh eff|dr.
## 4    199.0394 Per eff|Coh
## 5    830.4188 Coh eff|Per
## 6   1217.8248 Per eff|dr.
## No reference cohort given; reference cohort for age-effects is chosen as
##  the median date of birth for persons  with event:  1914 .
```

APC spline model: parameterization 2

```
fit3 <- apc.fit(dfEpi, npar = 5, model = "ns", dr.extr = "Holford",
    parm = "Ad-C-P", scale = 10^3)
## NOTE: npar is specified as:
## A P C
## 5 5 5
## [1] "Sequential modelling Poisson with log(Y) offset : ( AD-C-P ):\n"
##               Model Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
## 1               Age      105 15242.0306       NA        NA            NA
## 2         Age-drift      104  6563.9857        1 8678.0449  0.000000e+00
## 3        Age-Cohort      101  1016.3729        3 5547.6128  0.000000e+00
## 4 Age-Period-Cohort       98   419.2548        3  597.1181 4.247733e-129
## 5        Age-Period      101  2910.5114        3 2491.2565  0.000000e+00
## 6         Age-drift      104  6563.9857        3 3653.4743  0.000000e+00
##   Test dev/df     H0
## 1          NA
## 2   8678.0449 zero drift
## 3   1849.2043 Coh eff|dr.
## 4    199.0394 Per eff|Coh
## 5    830.4188 Coh eff|Per
## 6   1217.8248 Per eff|dr.
```

Spline parameterization 1: age, cohort, period curves

```r
apc.plot(fit2)
```

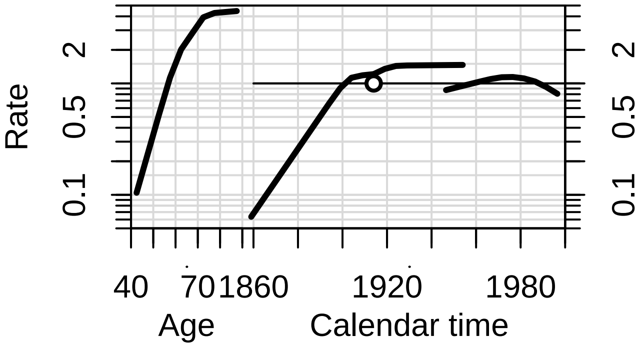


Figure 1: Age-period-cohort estimates under the first “ACP“ constraint.

```
## cp.offset    RR.fac
##      1765         1
```

Spline parameterization 2: age, cohort, period curves

```r
apc.plot(fit3)
```

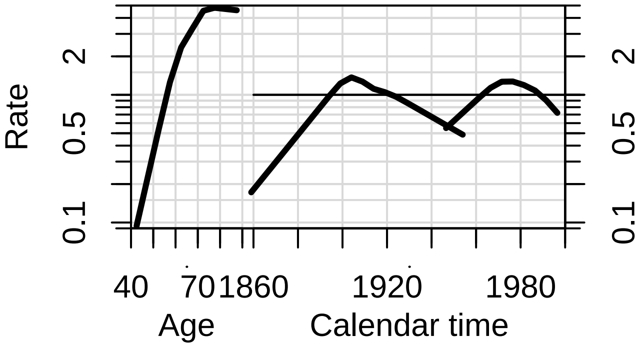


Figure 2: Age-period-cohort estimates under the first "Ad C-P" constraint.

```
## cp.offset    RR.fac
##       1765         1
```

Different fitted values since different spline smoothers

```
fit4 <- apc.fit(dfEpi, npar = 3, model = "bs", dr.extr = "Holford",
    parm = "ACP", scale = 10^3)
## NOTE: npar is specified as:
## A P C
## 3 3 3
## [1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
##               Model Mod. df.  Mod. dev. Test df. Test dev.       Pr(>Chi)
## 1               Age      106 15107.7170       NA        NA             NA
## 2         Age-drift      105  6423.8836        1 8683.8334  0.000000e+00
## 3        Age-Cohort      103  1137.7086        2 5286.1751  0.000000e+00
## 4 Age-Period-Cohort      101   477.7064        2  660.0021 4.812363e-144
## 5        Age-Period      103  2757.4175        2 2279.7110  0.000000e+00
## 6         Age-drift      105  6423.8836        2 3666.4662  0.000000e+00
##   Test dev/df    H0
## 1          NA
## 2   8683.8334 zero drift
## 3   2643.0875 Coh eff|dr.
## 4    330.0011 Per eff|Coh
## 5   1139.8555 Coh eff|Per
## 6   1833.2331 Per eff|dr.
## No reference cohort given;  reference cohort for age-effects is chosen as
##  the median date of birth for persons  with event:  1914 .
```

Different fitted values since different spline smoothers

```
fit5 <- apc.fit(dfEpi, npar = 8, model = "bs", dr.extr = "Holford",
    parm = "ACP", scale = 10^5)
## NOTE: npar is specified as:
## A P C
## 8 8 8
## [1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
##               Model Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
## 1               Age      101 15103.6974       NA        NA            NA
## 2         Age-drift      100  6418.0122        1 8685.6852  0.000000e+00
## 3        Age-Cohort       93   865.9163        7 5552.0959  0.000000e+00
## 4 Age-Period-Cohort       86   248.7888        7  617.1275 4.986257e-129
## 5        Age-Period       93  2727.3948        7 2478.6060  0.000000e+00
## 6         Age-drift      100  6418.0122        7 3690.6173  0.000000e+00
##   Test dev/df    H0
## 1          NA
## 2  8685.68524 zero drift
## 3   793.15655 Coh eff|dr.
## 4    88.16107 Per eff|Coh
## 5   354.08657 Coh eff|Per
## 6   527.23105 Per eff|dr.
## No reference cohort given;  reference cohort for age-effects is chosen as
##  the median date of birth for persons  with event:  1914 .
```

Age-Period-Cohort models

```r
apc.plot(fit4)
```

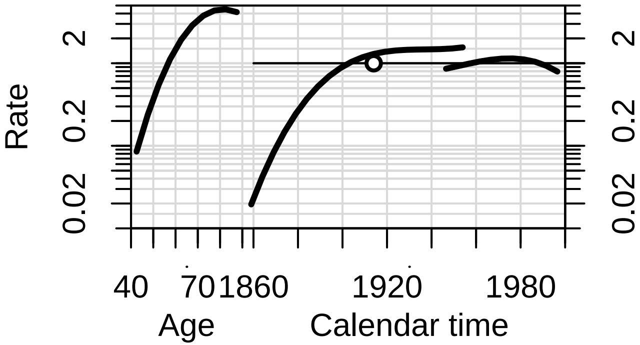


Figure 3: Age-period-cohort estimates under the first "Ad C-P" constraint.

```
## cp.offset    RR.fac
##      1765         1
```

Age-Period-Cohort models

```
apc.plot(fit5)
```

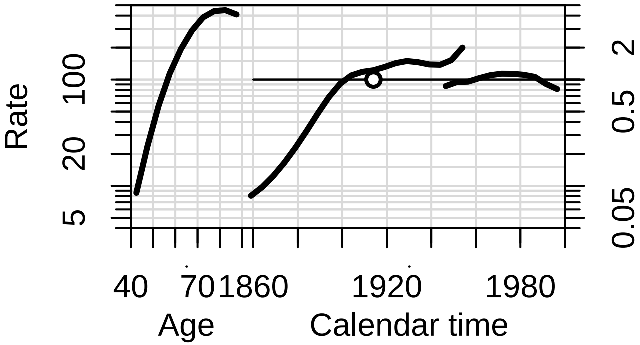


Figure 4: Age-period-cohort estimates under the first “Ad C-P“ constraint.

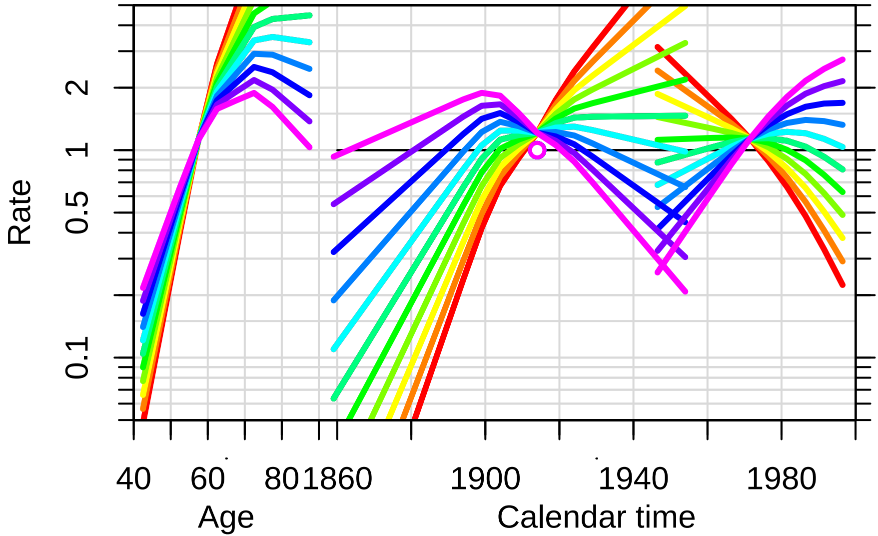
```
## cp.offset    RR.fac
##       1765       100
```

The identifiability problem illustrated

Linear drifts are added on in usch a way to give the same overall fit, but each of the age-period-cohort curves are changed, dramatically over the whole range.

```
fp <- apc.plot(fit2)
apc.lines(fit2, frame.par = fp, drift = 1.01, col = "red")
for (i in 1:11) apc.lines(fit2, frame.par = fp, drift = 1 +
    (i - 6)/100, col = rainbow(12)[i])
```

This plot beautifully shows the unidentifiability!


Rate
0.1
0.5
1
2
40
60
80
1860
1900
1940
1980
Age
Calendar time

INLA ANOVA

We demonstrate how frequentist GLM and Bayes INLA approaches give virtually identical under the default relatively flat priors.

```
# install.packages('INLA',
# repos=c(getOption('repos'),
# INLA='https://inla.r-inla-download.org/R/testing'),
# dep=TRUE)
library(INLA)
glmMA <- glm(D ~ as.factor(A) + offset(log(Y)), data = dfEpi,
    family = "poisson")
inlaMA <- inla(D ~ as.factor(A), data = dfEpi, offset = log(Y),
    family = "poisson")
```

INLA ANOVA

```
GLMpt <- coef(glmMA)
INLApt <- inlaMA$summary.fixed[4]
GLMse <- sqrt(diag(vcov(glmMA)))
INLAsd <- inlaMA$summary.fixed[2]
cbind(GLMpt, INLApt, GLMse, INLAsd)
##                        GLMpt   0.5quant      GLMse         sd
## (Intercept)       -9.0172259 -9.0169981 0.03094922 0.03096924
## as.factor(A)47.5   0.9504258  0.9502499 0.03673205 0.03676239
## as.factor(A)52.5   1.7840788  1.7838372 0.03382212 0.03385230
## as.factor(A)57.5   2.4288055  2.4284488 0.03264480 0.03267468
## as.factor(A)62.5   2.8937852  2.8932298 0.03215539 0.03218517
## as.factor(A)67.5   3.1962247  3.1954416 0.03200143 0.03203123
## as.factor(A)72.5   3.3749712  3.3740709 0.03208295 0.03211290
## as.factor(A)77.5   3.3787580  3.3779555 0.03260569 0.03263607
## as.factor(A)82.5   3.2636171  3.2630904 0.03422012 0.03425121
## as.factor(A)87.5   3.0227998  3.0225402 0.04022849 0.04025733
```

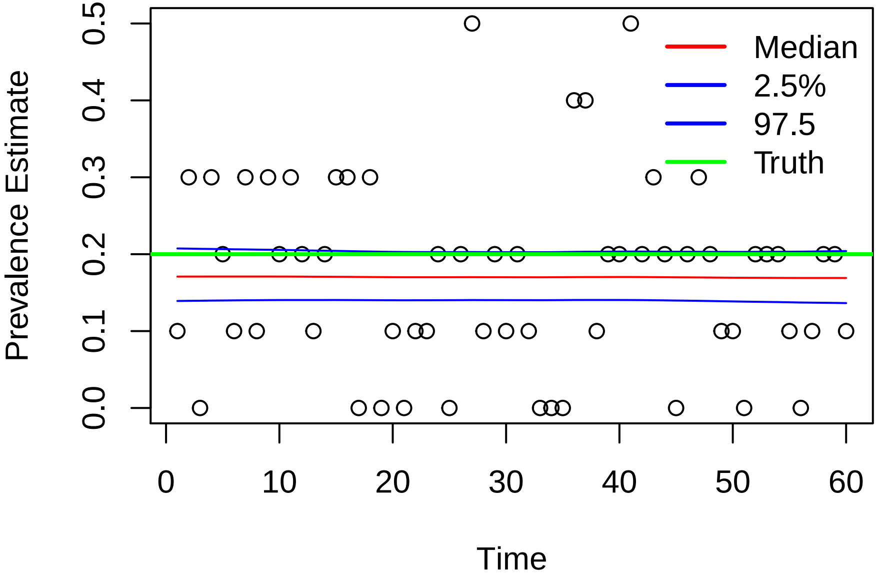
Random Walk Models

```r
n1 <- 10
p <- 0.2
time <- seq(1, 60)
# Simulate data
y1 <- rbinom(length(time), n1, p)
inladf1 <- data.frame(y1 = y1, time = time)
# Define model
formula1 = y1 ~ f(time, model = "rw1")
fit1 <- inla(formula1, data = inladf1, family = "binomial",
    Ntrials = n1, control.predictor = list(compute = TRUE))
formula2 = y1 ~ f(time, model = "rw2")
fit2 <- inla(formula2, data = inladf1, family = "binomial",
    Ntrials = n1, control.predictor = list(compute = TRUE))
```

RW1 Fit

```
plot(y1/n1 ~ time, ylab = "Prevalence Estimate", xlab = "Time")
lines(fit1$summary.fitted.values$`0.5quant` ~ time,
    col = "red")
lines(fit1$summary.fitted.values$`0.025quant` ~ time,
    col = "blue")
lines(fit1$summary.fitted.values$`0.975quant` ~ time,
    col = "blue")
legend("topright", legend = c("Median", "2.5%", "97.5",
    "Truth"), lty = 1, lwd = 2, col = c("red", "blue",
    "blue", "green"), bty = "n")
abline(h = p, col = "green", lwd = 2)
```

RW1 Fit

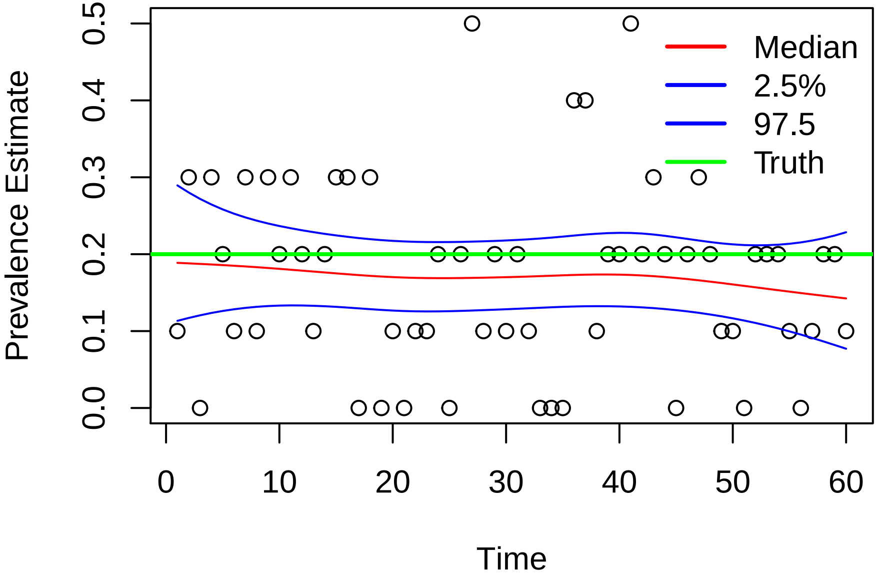


Median
2.5%
97.5
Truth
Prevalence Estimate
Time
0.0
0.1
0.2
0.3
0.4
0.5
0
10
20
30
40
50
60

RW2 Fit

```
plot(y1/n1 ~ time, ylab = "Prevalence Estimate", xlab = "Time")
lines(fit2$summary.fitted.values$`0.5quant` ~ time,
    col = "red")
lines(fit2$summary.fitted.values$`0.025quant` ~ time,
    col = "blue")
lines(fit2$summary.fitted.values$`0.975quant` ~ time,
    col = "blue")
legend("topright", legend = c("Median", "2.5%", "97.5",
    "Truth"), lty = 1, lwd = 2, col = c("red", "blue",
    "blue", "green"), bty = "n")
abline(h = p, col = "green", lwd = 2)
```

RW2 Fit



Prevalence Estimate
0.0
0.1
0.2
0.3
0.4
0.5
0
10
20
30
40
50
60
Time
Median
2.5%
97.5
Truth