Appendix. Example R Code for Estimating Cohort Deviations

```
# Age and period main effects.
# wtdata is a weighted survey design object; acc denotes age categories; pcc denotes
# period categories.

library(survey)
modelresult = svyglm(inlfc ~  acc * pcc, wtdata2, family =
quasibinomial)
r6 = modelresult$coefficients
r6se = summary(modelresult)$coef[,"Std. Error"]
r6p = summary(modelresult)$coef[,"Pr(>|t|)"]


# Compute transformation matrix T.
# A is the number of age categories; P is the number of time periods.

T = array(rep(0, A*P*(A-1)*(P-1)), dim=c(A*P, (A-1)*(P-1)))

ind1 = A*1:(P-1)
ind2 = (A*(P-1)+1):(A*P-1)
ind3 = A*P

ind = c(ind1,ind2,ind3)

newind = 1:(A*P)
newind = newind[-ind]

T[newind,]  = diag((A-1)*(P-1))
T[ind1,]    = -diag(P-1)[,rep(1:(P-1),each=A-1)]
T[ind2,]    = -diag(A-1)[,rep(1:(A-1),P-1)]
T[ind3,]    = rep(1,(A-1)*(P-1))

# Compute "full" interaction estimates.
# covn is the level of the covariate – 1. For example, for an educational attainment variable with
# three levels, covn = 3-1 = 2.

iatemp = vcov(modelresult)[(covn+A+P): length(r6), (covn+A+P):
length(r6)]
iavcov = T%*%iatemp%*%t(T)
df = modelresult$df.residual

iaesti = as.vector(T%*%r6[(covn+A+P): length(r6)])
iase   = sqrt(diag(iavcov))
iap    = pt(-abs(iaesti/iase), df)*2
```

```
cindex = array(rep(0, A*P), dim = c(A, P))
for (j in 1:P){
  cindex [,j] = seq((A+j-1),j, -1)
}
```

############ Step 3.1 inter-cohort average deviations ############

C is the number of cohorts. For birth cohorts, C = A+P-1.

```
cint   = rep(NA, C)
cintse = rep(NA, C)
cintt  = rep(NA, C)
cintp  = rep(NA, C)

for (k in 1:C){
  O = sum(cindex == k)
  k1 = rep(1/O, O)
  k2 = rep(0, A*P)
  k2[cindex == k] = k1

  contresti = k2%*%iaesti                # point estimates in Step 3.1
  contrse = sqrt(t(k2)%*%iavcov%*%k2)    # standard errors in Step 3.1
  t = contresti/contrse
  if (t > 0){
    p = 2*pt(t, df, lower.tail=F)
  } else {
    p = 2*pt(t, df, lower.tail=T)
  }

  cint[k]   = contresti
  cintse[k] = contrse
  cintt[k]  = t
  cintp[k]  = p
}
```

############ Step 3.2 intra-cohort life-course changes ############

```
cslope   = rep(NA, C)
cslopese = rep(NA, C)
cslopet  = rep(NA, C)
cslopep  = rep(NA, C)

poly = 1
for (k in (poly+1):(C-poly)){
  o = sum(cindex == k)
```

```
  k1 = contr.poly(o)
  k2 = rep(0, A*P)
  k2[cindex == k] = k1[,poly]

  contresti = k2%*%iaesti                # point estimates in Step 3.2
  contrse = sqrt(t(k2)%*%iavcov%*%k2) # standard errors in Step 3.2
  t = contresti/contrse
  if (t > 0){
    p = 2*pt(t, df, lower.tail=F)
  } else {
    p = 2*pt(t, df, lower.tail=T)
  }

  cslope[k]   = contresti
  cslopese[k] = contrse
  cslopet[k]  = t
  cslopep[k]  = p
}
```