Actuarial Science with R

2. life insurance & mortality tables

Arthur Charpentier

joint work with **Christophe Dutang** & **Vincent Goulet**
and **Rob Hyndman**'s `demography` package

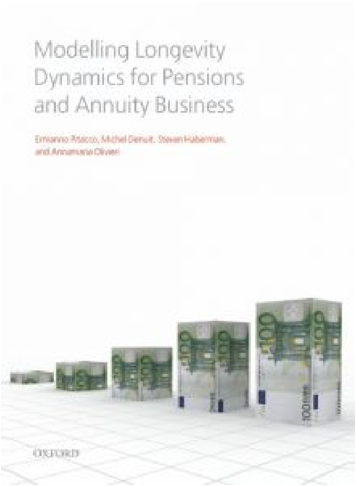
UQÀM Université du Québec à Montréal
Quantact

Meielisalp 2012 Conference, June

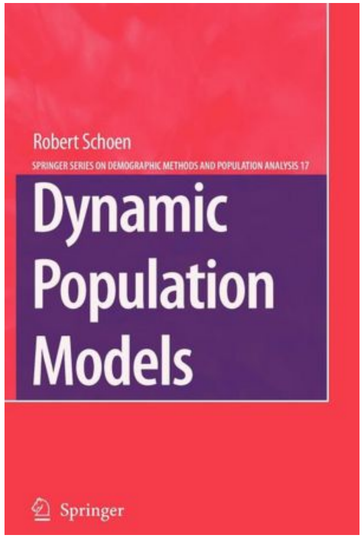
6th R/Rmetrics Meielisalp Workshop & Summer School
on Computational Finance and Financial Engineering

Some (standard) references

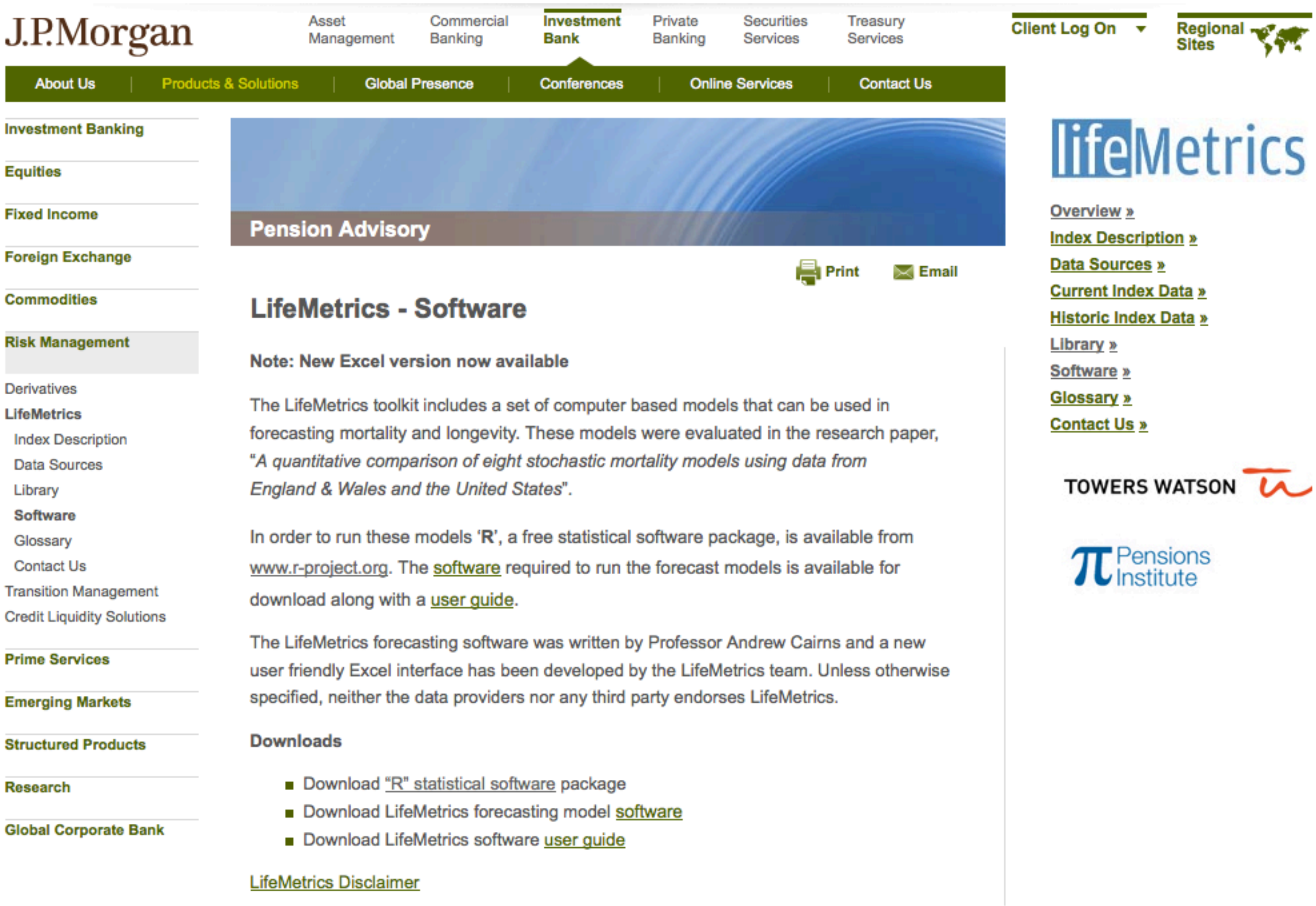
Pitacco, E., Denuit, M., Haberman, S.
& Olivieri, A. (2008) *Modeling Longevity Dynamics for Pensions and Annuity Business*
Oxford University Press



Schoen, R. (2007)
Dynamic Population Models
Springer Verlag

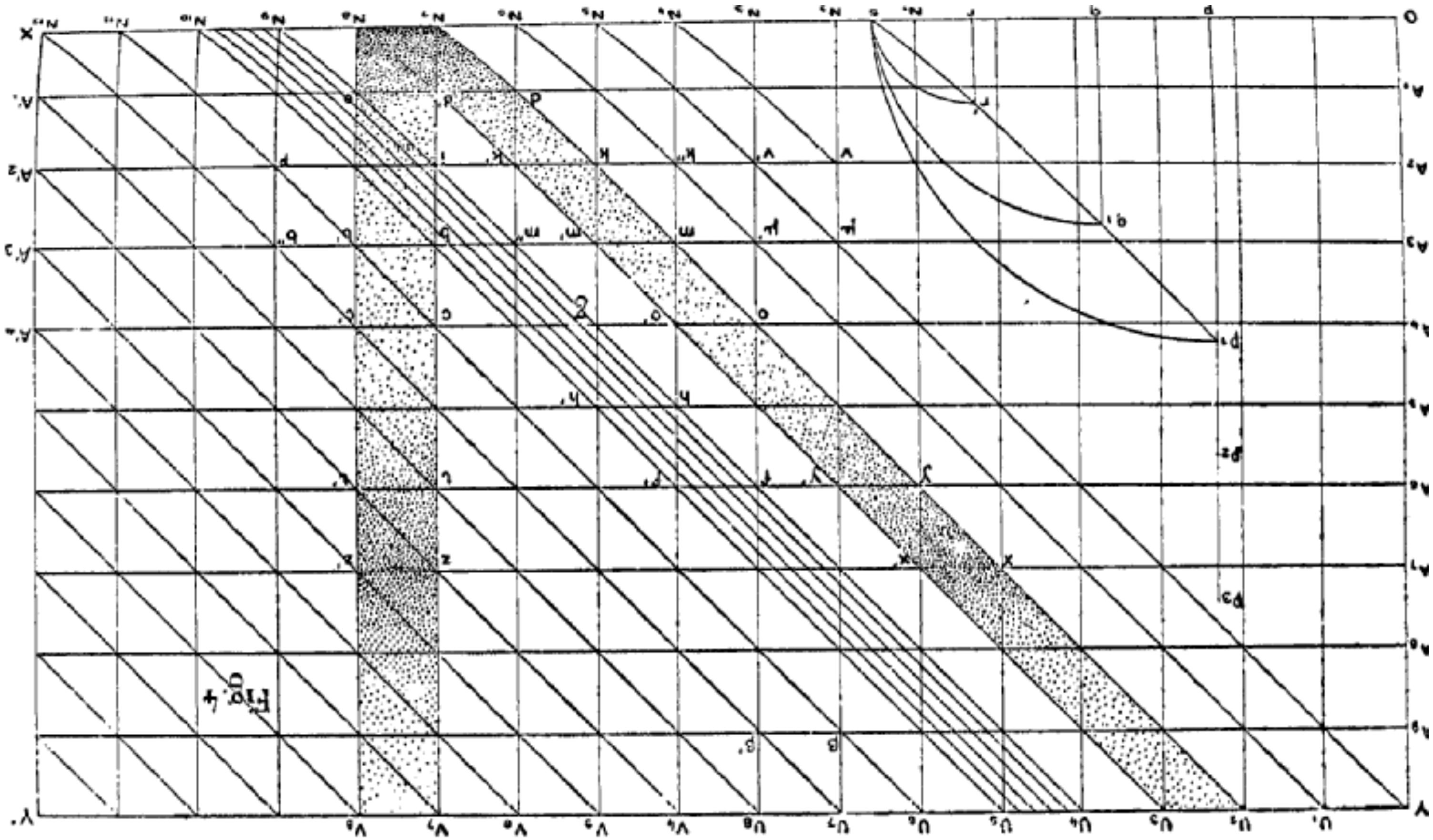


A possible motivation ?



Lexis diagram, age and time

From Lexis (1880), idea of visualizing lifetime, age x, time t and year of birth y



Lexis diagram, age and time

Idea : Life tables L_x should depend on time, $L_{x,t}$.

Let $D_{x,t}$ denote the number of deaths of people aged x, during year t, data frame `DEATH` and let $E_{x,t}$ denote the exposure, of age x, during year t, data frame `EXPOSURE`, from `http://www.mortality.org/`

The Human Mortality Database

John R. Wilmoth, *Director*	University of California, Berkeley
Vladimir Shkolnikov, *Co-Director*	Max Planck Institute for Demographic Research

The Human Mortality Database (HMD) was created to provide detailed mortality and population data to researchers, students, journalists, policy analysts, and others interested in the history of human longevity. The project began as an outgrowth of earlier projects in the Department of Demography at the University of California, Berkeley, USA, and at the Max Planck Institute for Demographic Research in Rostock, Germany (see history). It is the work of two teams of researchers in the USA and Germany (see research teams), with the help of financial backers and scientific collaborators from around the world (see acknowledgements).

Lexis diagram, age and time

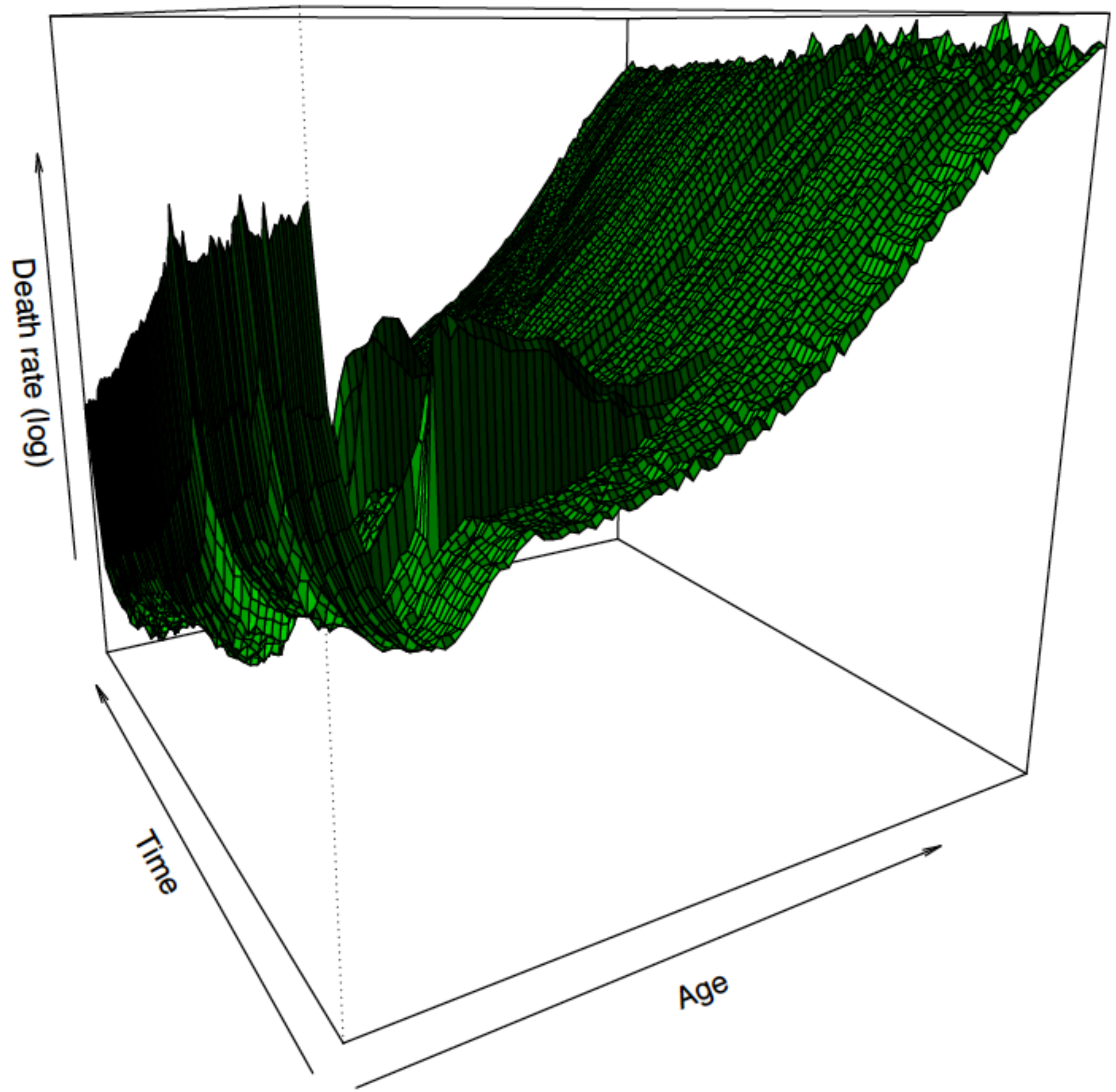
Remark : be carefull of age `110+`

```
> DEATH$Age=as.numeric(as.character(DEATH$Age))
> DEATH$Age[is.na(DEATH$Age)]=110
> EXPOSURE$Age=as.numeric(as.character(EXPOSURE$Age))
> EXPOSURE$Age[is.na(EXPOSURE$Age)]=110
```

Consider force of mortality function

$$\mu_{x,t} = \frac{D_{x,t}}{E_{x,t}}$$

```
> MU=DEATH[,3:5]/EXPOSURE[,3:5]
> MUT=matrix(MU[,3],length(AGE),length(ANNEE))
> persp(AGE[1:100],ANNEE,log(MUT[1:100,]),
+  theta=-30,col="light green",shade=TRUE)
```


Death rate (log)
Time
Age

Tables, per year t and Survival lifetimes

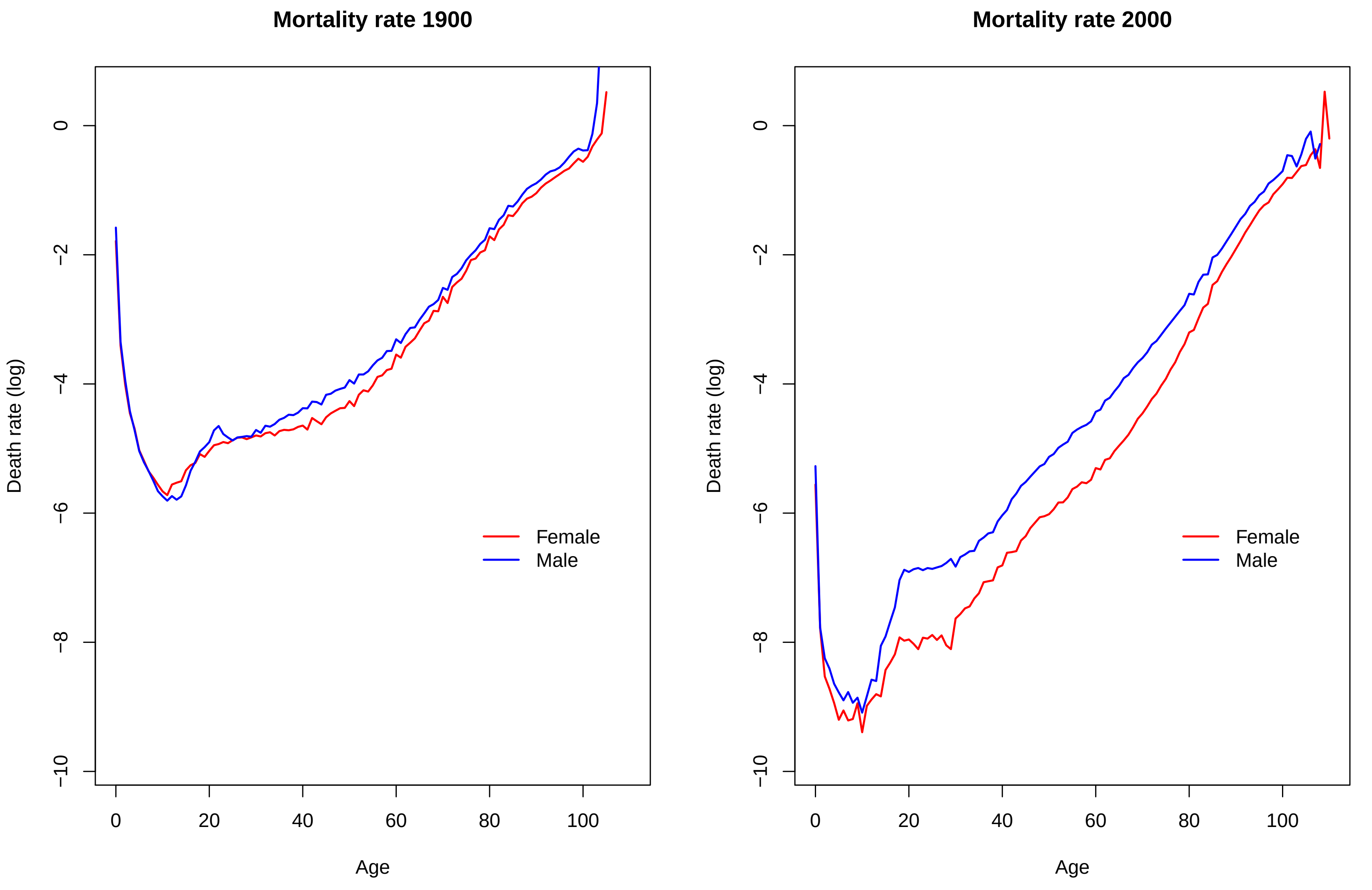
Let us study deaths occurred in `year=1900` or `2000`, $x \mapsto \log \mu_{x,t}$ when $t = 1900$ or 2000.

```
> D=DEATH[DEATH$Year==year,]; E=EXPOSURE[EXPOSURE$Year==year,]
> MU = D[,3:5]/E[,3:5]
> plot(0:110,log(MU[,1]),type="l",col="red"); lines(0:110,log(MU[,2]),col="blue")
```

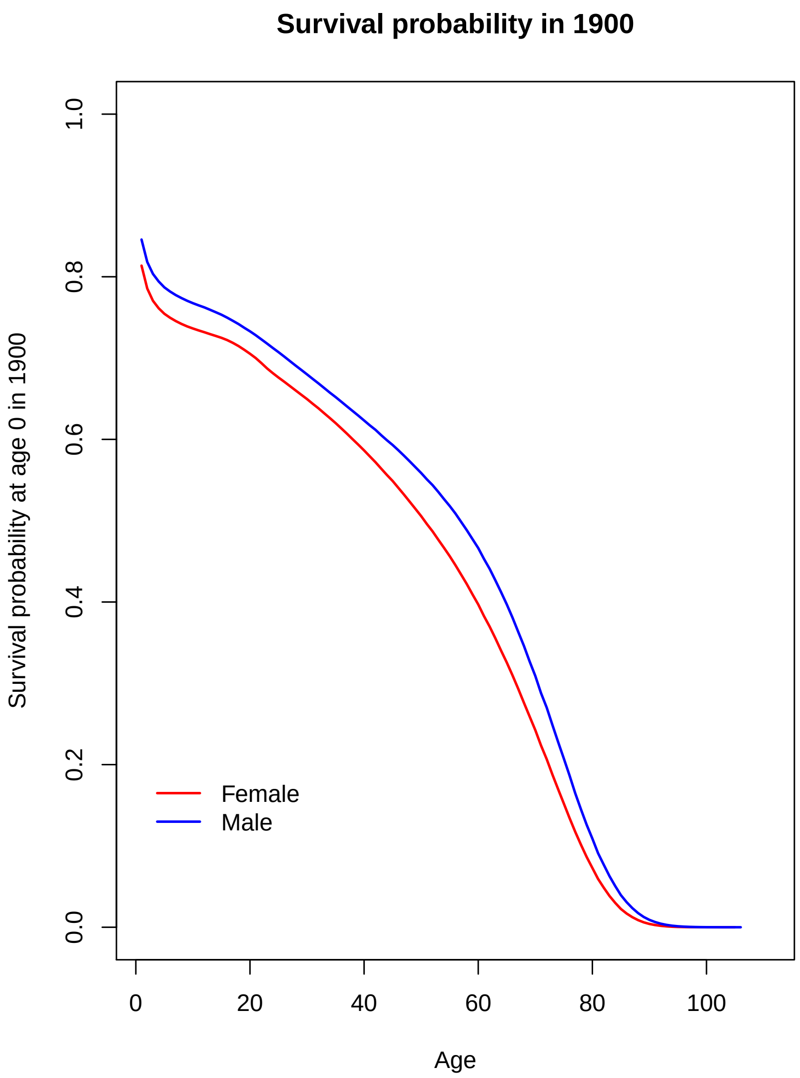
Evolution of $x \mapsto L_{x,t} = \exp\left(-\int_0^x \mu_{h,t} dh\right)$ when $t = 1900$ or 2000,

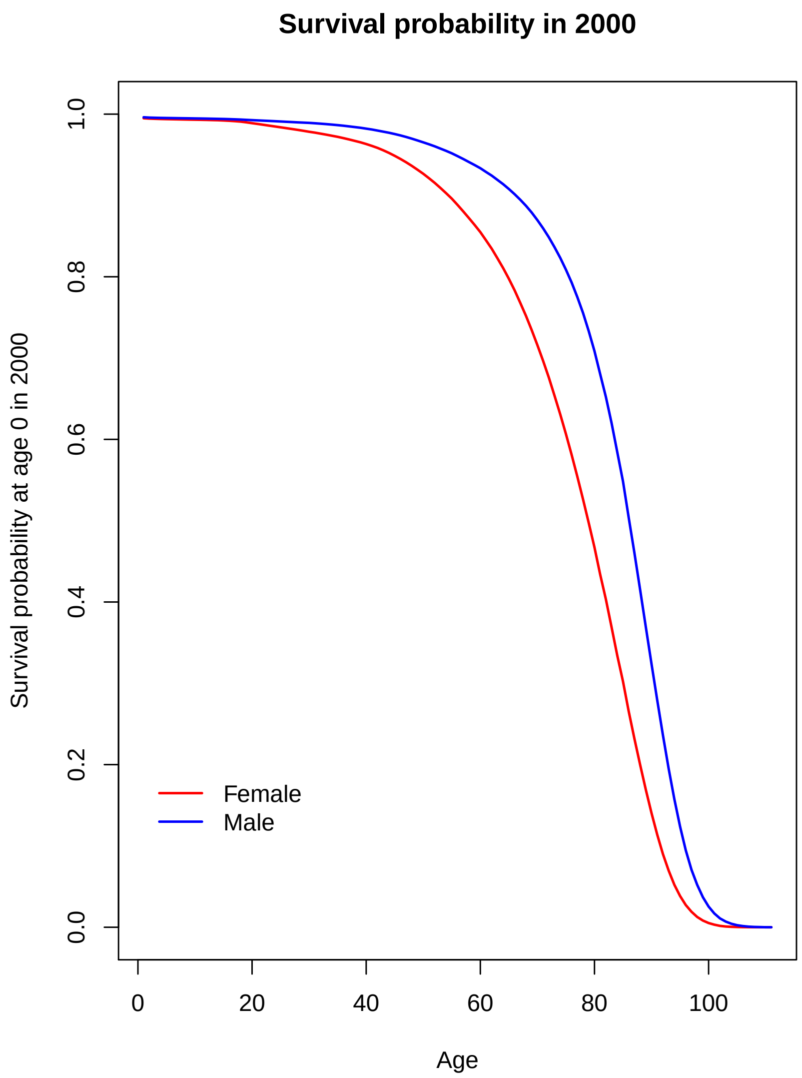
```
> PH=PF=matrix(NA,111,111)
> for(x in 0:110){
+ PH[x+1,1:(111-x)]=exp(-cumsum(MU[(x+1):111,2]))
+ PF[x+1,1:(111-x)]=exp(-cumsum(MU[(x+1):111,1]))}
> x=0; plot(1:111,PH[x+1,],ylim=c(0,1),type="l",col="blue")
> lines(1:111,PF[x+1,],col="red")
```

Tables, per year t and Survival lifetimes

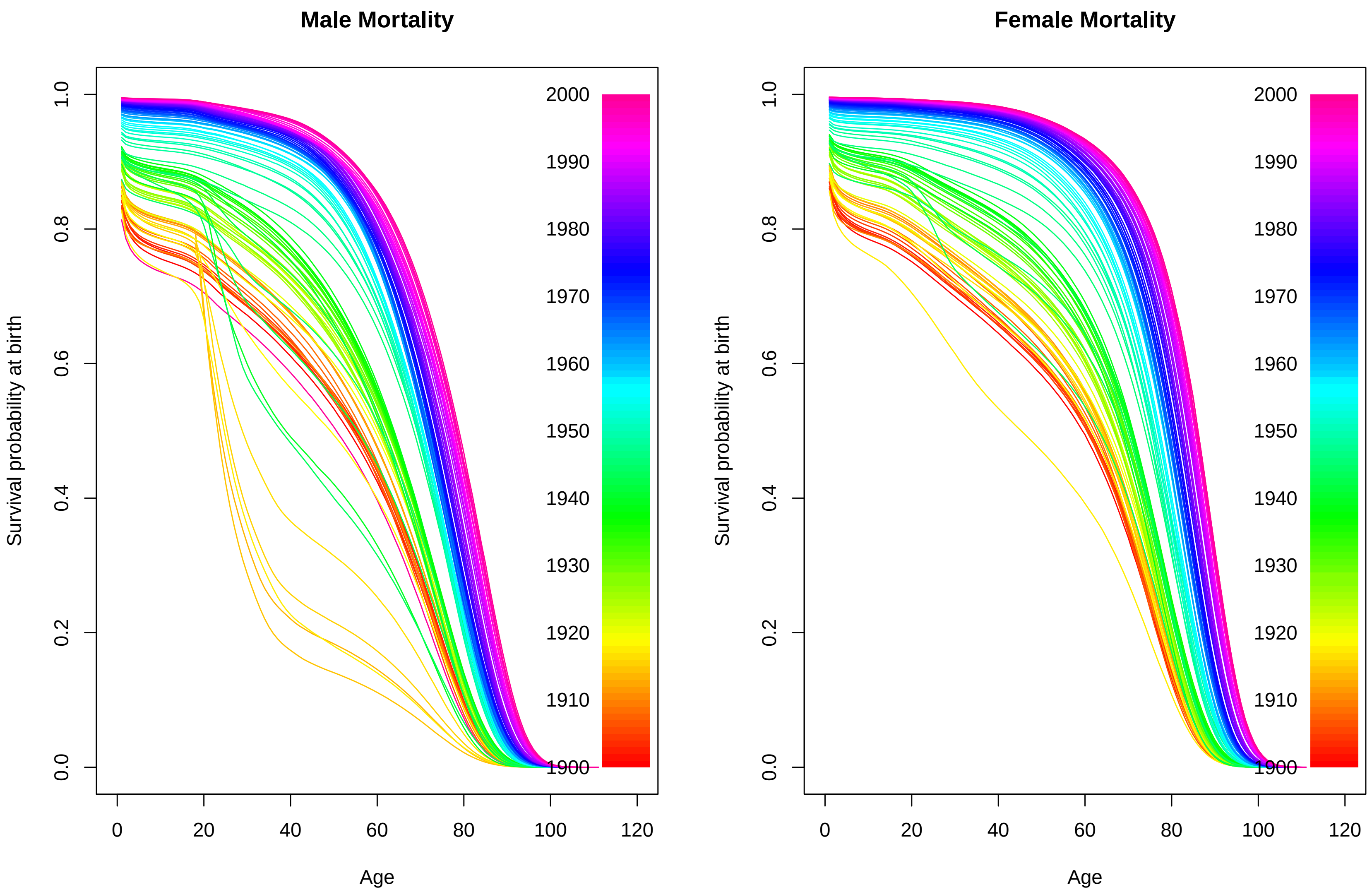


Tables, per year t and Survival lifetimes





Survival lifetimes and rectangularization’



Life table and transversality

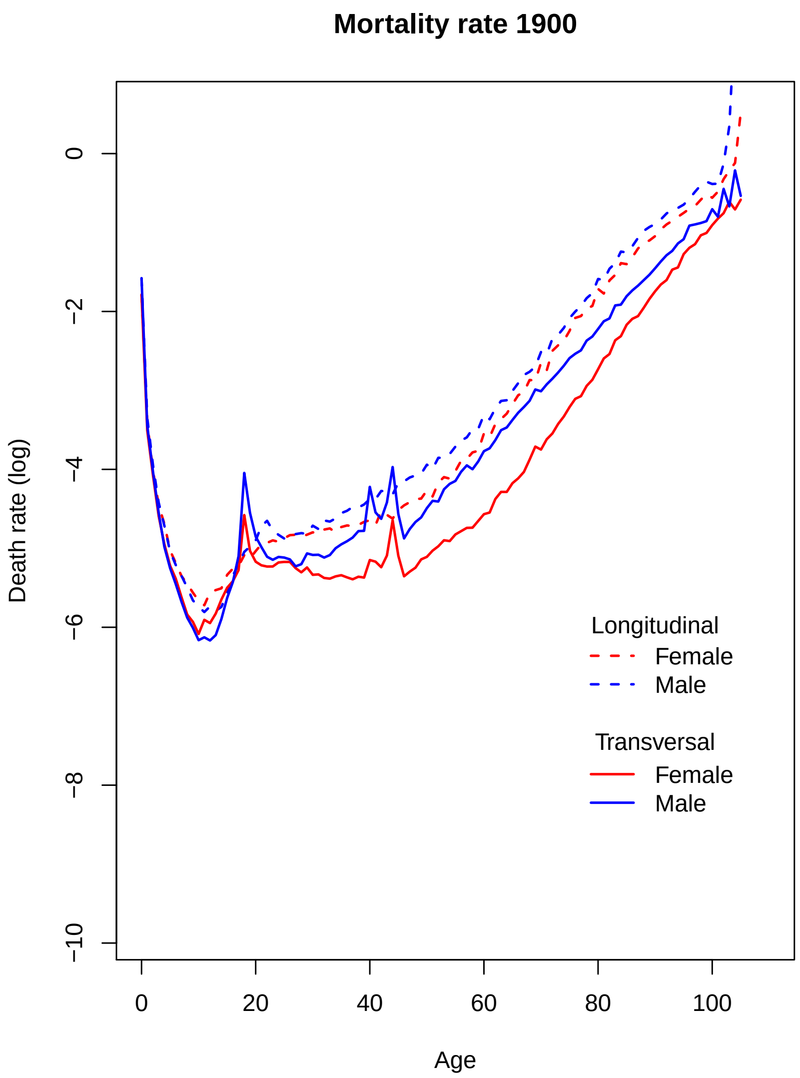
```
> XV<-unique(Deces$Age); YV<-unique(Deces$Year)
> DTF<-t(matrix(Deces[,3],length(XV),length(YV)))
>  ownames(DTF)=YV;colnames(DTF)=XV
> t(DTF)[1:13,1:10]
    1899  1900  1901  1902  1903  1904  1905  1906  1907  1908
0  64039 61635 56421 53321 52573 54947 50720 53734 47255 46997
1  12119 11293 10293 10616 10251 10514  9340 10262 10104  9517
2   6983  6091  5853  5734  5673  5494  5028  5232  4477  4094
3   4329  3953  3748  3654  3382  3283  3294  3262  2912  2721
4   3220  3063  2936  2710  2500  2360  2381  2505  2213  2078
5   2284  2149  2172  2020  1932  1770  1788  1782  1789  1751
6   1834  1836  1761  1651  1664  1433  1448  1517  1428  1328
7   1475  1534  1493  1420  1353  1228  1259  1250  1204  1108
8   1353  1358  1255  1229  1251  1169  1132  1134  1083   961
9   1175  1225  1154  1008  1089   981  1027  1025   957   885
10  1174  1114  1063   984   977   882   955   937   942   812
11  1162  1055  1038  1020   945   954   931   936   880   851
12  1100  1254  1076  1034  1023  1009  1041  1026   954   908
13  1251  1283  1190  1126  1108  1093  1111  1054  1103   940
```

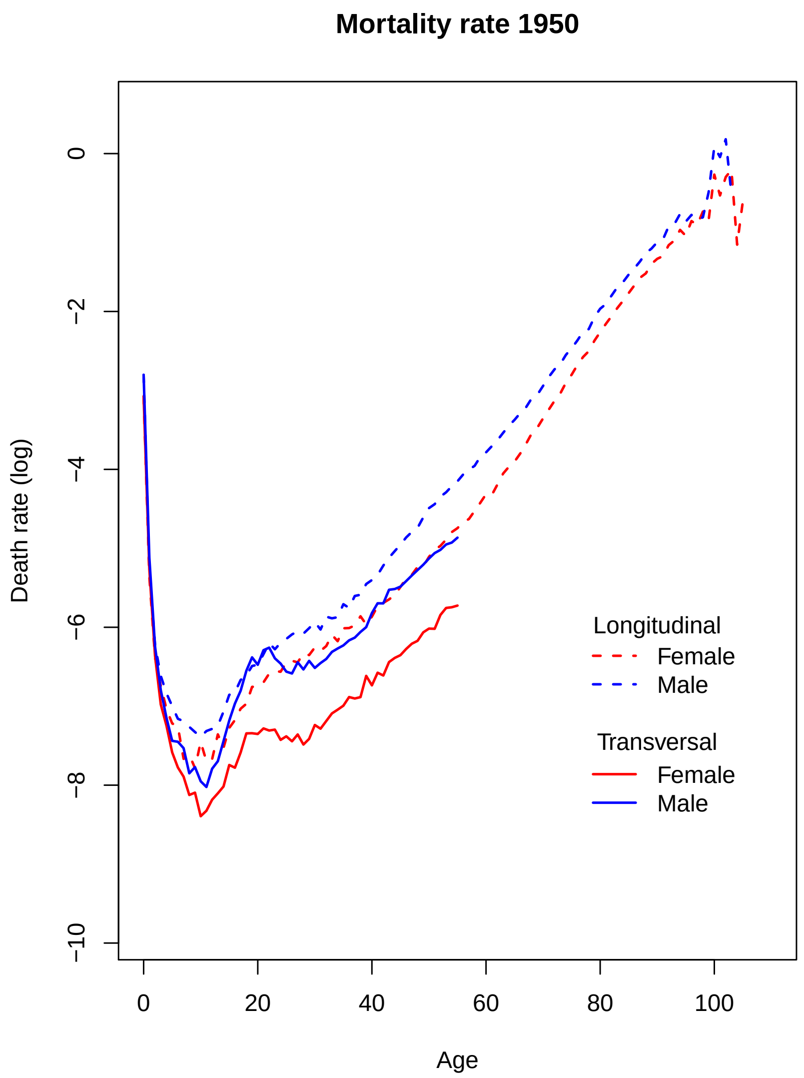
Life table and transversality

It might be interesting to follow a cohort, per year of birth $x - t$,

```
> Nannee <- max(DEATH$Year)
> naissance <- 1950
> taille <- Nannee - naissance
> Vage <- seq(0,length=taille+1)
> Vnaissance <- seq(naissance,length=taille+1)
> Cagreg <- DEATH$Year*1000+DEATH$Age
> Vagreg <- Vnaissance*1000+Vage
> indice <- Cagreg%in%Vagreg
> DEATH[indice,]
      Year Age   Female     Male    Total
5662  1950   0 18943.05 25912.38 44855.43
5774  1951   1  2078.41  2500.70  4579.11
5886  1952   2   693.20   810.32  1503.52
5998  1953   3   375.08   467.12   842.20
6110  1954   4   287.04   329.09   616.13
6222  1955   5   205.03   246.07   451.10
6334  1956   6   170.00   244.00   414.00
```

Life table and transversality





Lee & Carter (1992) model

Assume here (as in the original model)

$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \varepsilon_{x,t},$$

with some i.i.d. noise $\varepsilon_{x,t}$. Identification assumptions are usually

$$\sum_{x=x_m}^{x_M} \beta_x = 1 \text{ and } \sum_{t=t_m}^{t_M} \kappa_t = 0.$$

Then sets of parameters $\boldsymbol{\alpha} = (\alpha_x)$, $\boldsymbol{\beta} = (\beta_x)$ and $\boldsymbol{\kappa} = (\kappa_t)$, are obtained solving

$$\left(\hat{\alpha}_x, \hat{\beta}_x, \kappa_t\right) = \arg\min \sum_{x,t} \left(\ln \mu_{xt} - \alpha_x - \beta_x \cdot k_t\right)^2 .$$

Using demography package

Package demography can be used to fit a Lee-Carter model

```
> library(forecast)
> library(demography)
> YEAR=unique(DEATH$Year);nC=length(YEAR)
> AGE =unique(DEATH$Age);nL=length(AGE)
> MUF =matrix(DEATH$Female/EXPOSURE$Female,nL,nC)
> MUH =matrix(DEATH$Male/EXPOSURE$Male,nL,nC)
> POPF=matrix(EXPOSURE$Female,nL,nC)
> POPH=matrix(EXPOSURE$Male,nL,nC)
```

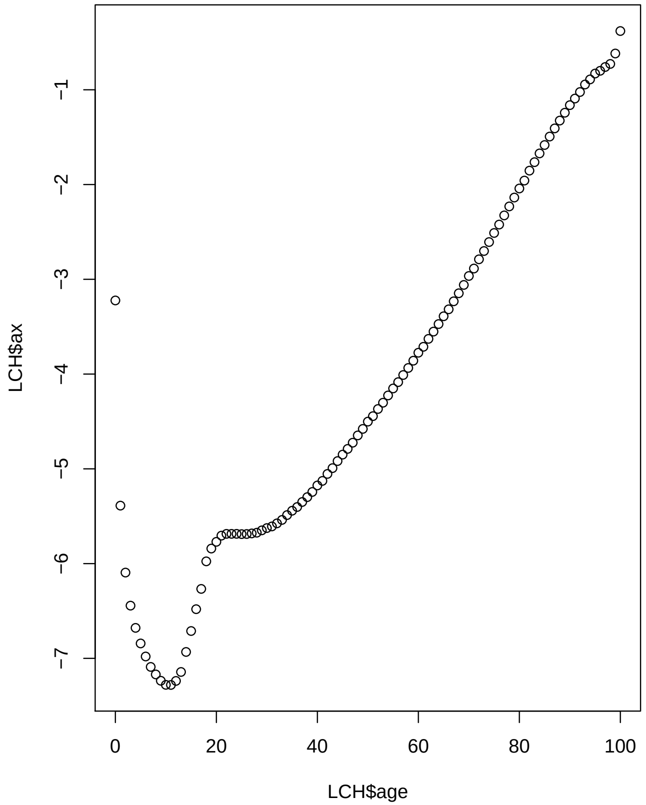
Then wo use the demogdata format

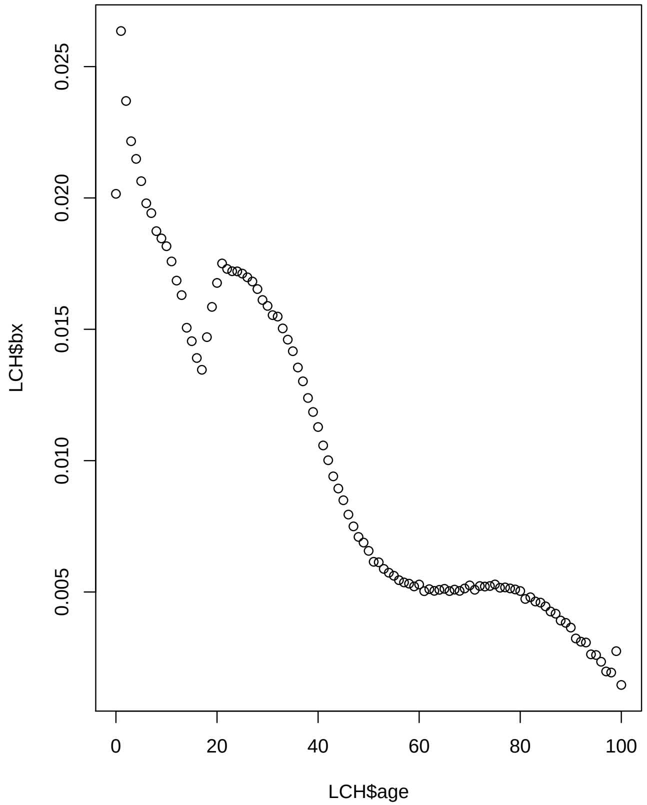
```
> BASEH <- demogdata(data=MUH, pop=POPH, ages=AGE, years=YEAR, type="mortality",
+ label="France", name="Hommes", lambda=1)
> BASEF <- demogdata(data=MUF, pop=POPF,ages=AGE, years=YEAR, type="mortality",
+ label="France", name="Femmes", lambda=1)
```

Estimation of $\boldsymbol{\alpha} = (\alpha_x)$ and $\boldsymbol{\beta} = (\beta_x)$

The code is simply `LCH <- lca(BASEH)`

```
> plot(LCH$age,LCH$ax,col="blue"); plot(LCH$age,LCH$bx,col="blue")
```





Estimation and projection of $\boldsymbol{\kappa} = (\kappa_t)$'s

Use `library(forcast)` to predict future κ_t's, e.g. using exponential smoothing

```
> library(forecast)
> Y <- LCH$kt
> (ETS <- ets(Y))
ETS(A,N,N)

Call:
 ets(y = Y)

  Smoothing parameters:
    alpha = 0.8923

  Initial states:
    l = 71.5007

  sigma:  12.3592

     AIC     AICc      BIC
```

```
1042.074 1042.190 1047.420

> plot(forecast(ETS,h=100))
```

But as in Lee & Carter original model, it is possible to fit an ARMA(1,1) model, on the differentiate series $(\Delta\kappa_t)$

$$\Delta\kappa_t = \phi\Delta\kappa_{t-1} + \delta + u_t - \theta u_{t-1}$$

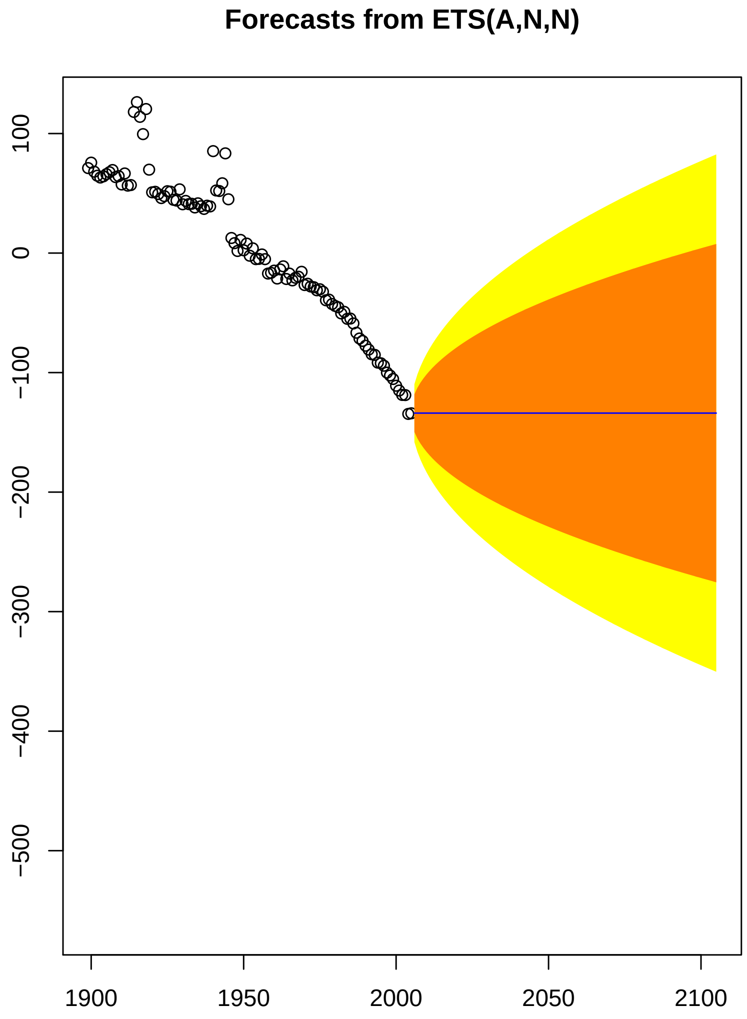
It is also possible to consider a linear tendency

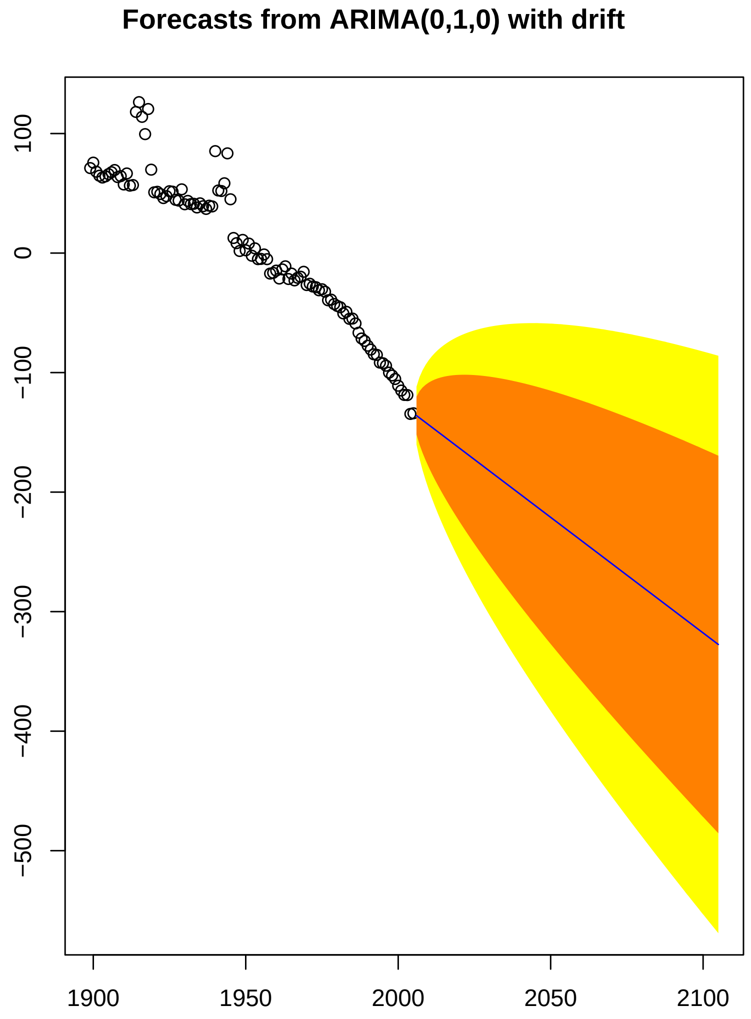
$$\kappa_t = \alpha + \beta t + \phi\kappa_{t-1} + u_t - \theta u_{t-1}.$$

```
> (ARIMA <- auto.arima(Y,allowdrift=TRUE))
Series: Y
ARIMA(0,1,0) with drift

Call: auto.arima(x = Y, allowdrift = TRUE)

Coefficients:
```

```
        drift
      -1.9346
s.e.   1.1972

sigma^2 estimated as 151.9:  log likelihood = -416.64
AIC = 837.29   AICc = 837.41   BIC = 842.62

> plot(forecast(ARIMA,h=100))
```

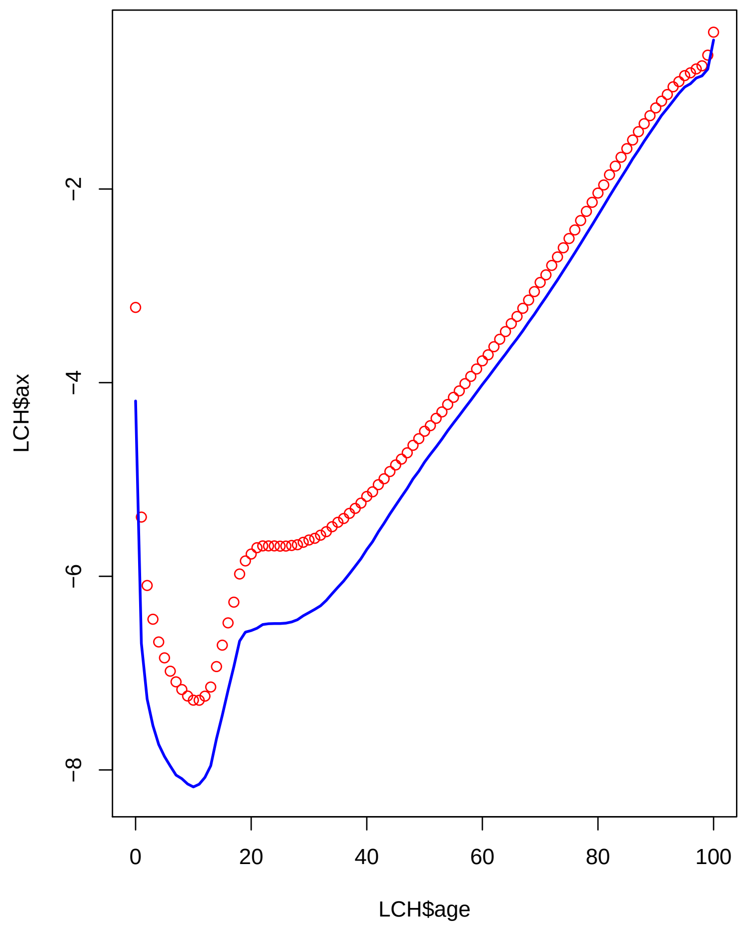
Projection of $\widehat{\kappa}_t$'s

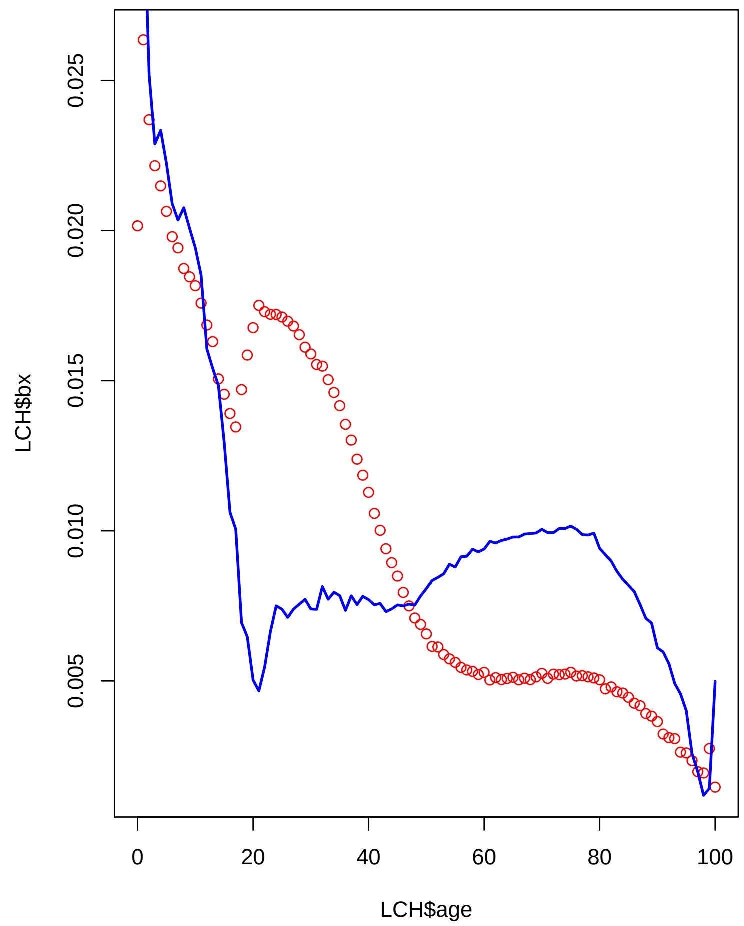




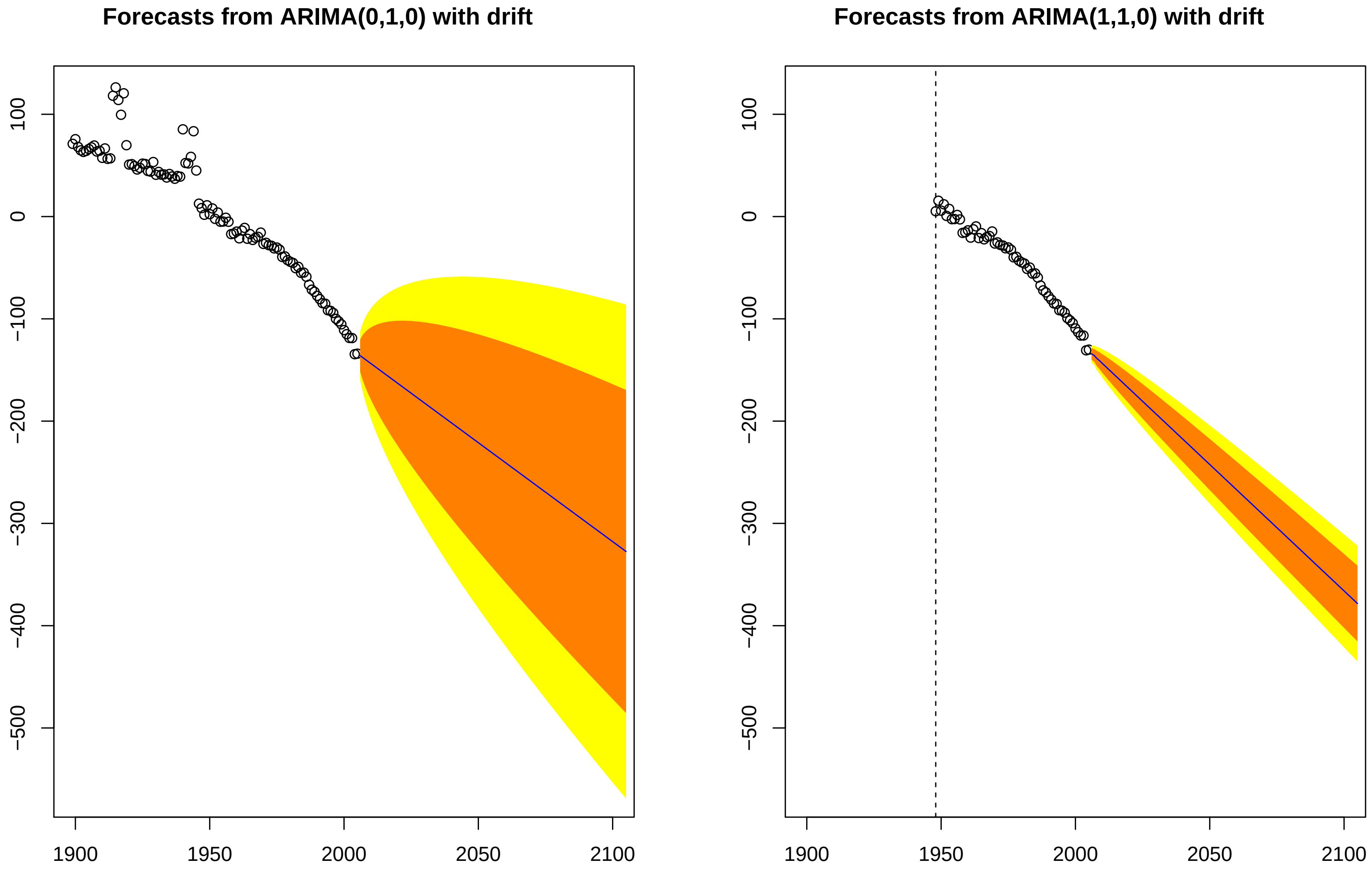
Shouldn't we start modeling after 1945 ?

Starting in 1948, `LCH0 <- lca(BASEH,years=1948:2005)`



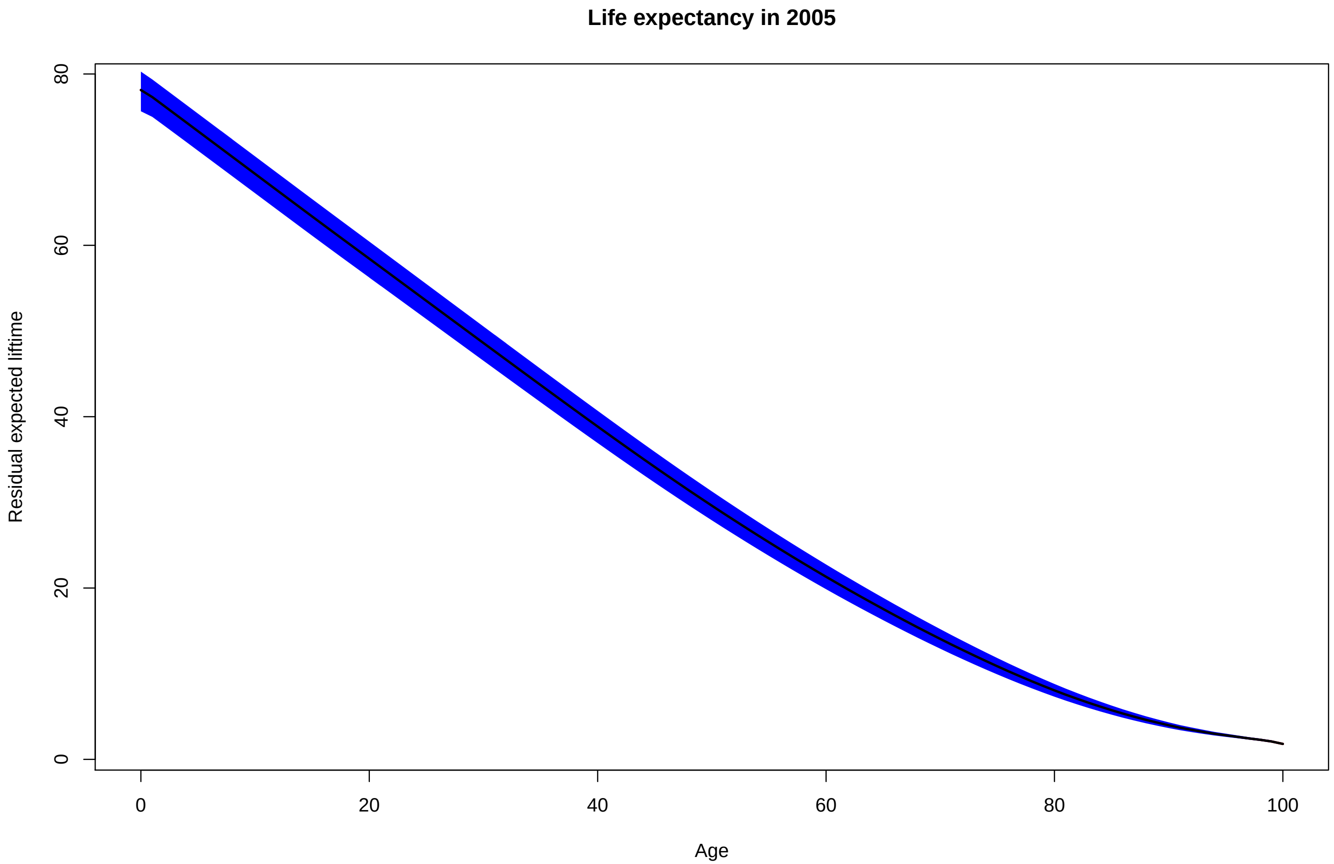


Projection of $\widehat{\kappa}_t$'s



Projection of life expectancy, born in 2005

```
> LCHT=lifetable(LCHf); plot(0:100,LCHT$ex[,5],type="l",col="red")
> LCHTu=lifetable(LCHf,"upper"); lines(0:100,LCHTu$ex[,5],lty=2)
> LCHTl=lifetable(LCHf,"lower"); lines(0:100,LCHTl$ex[,5],lty=2)
```



Residuals in Lee & Carter model

Recall that

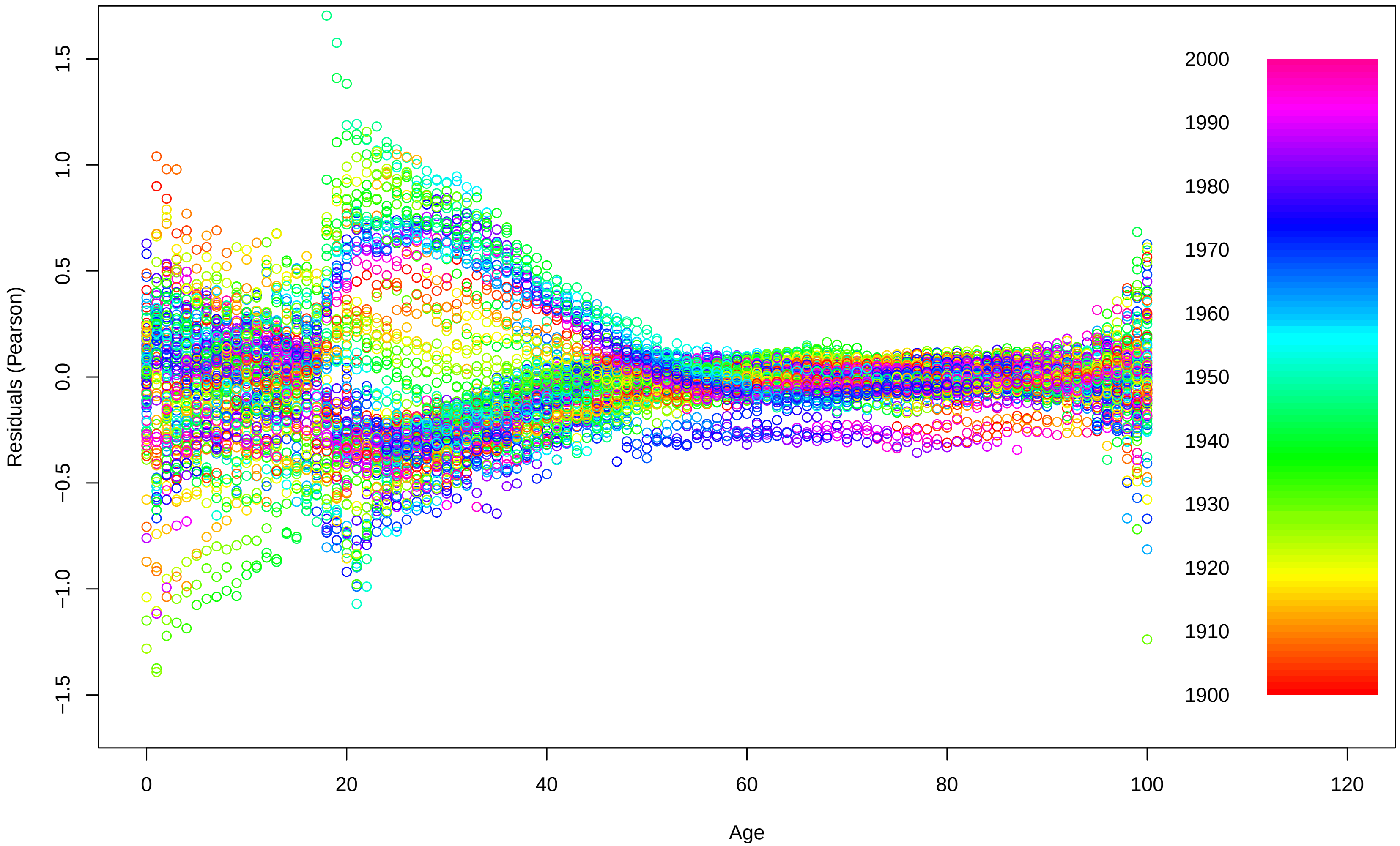
$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \varepsilon_{x,t}$$

Let $\widehat{\varepsilon}_{x,t}$ denote pseudo-residuals, obtained from estimation

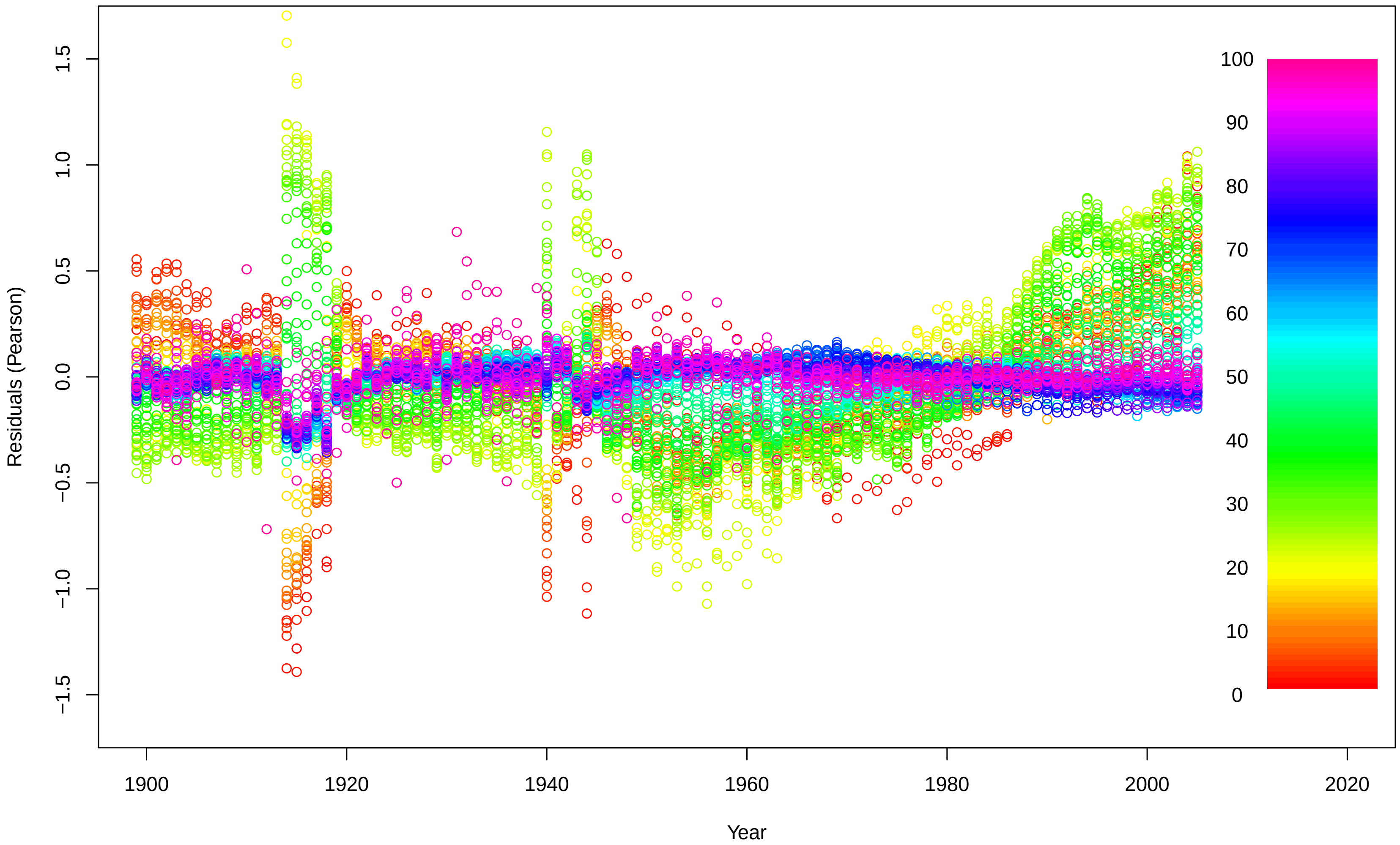
$$\widehat{\varepsilon}_{x,t} = \log \mu_{x,t} - \left(\widehat{\alpha}_x + \widehat{\beta}_x \cdot \widehat{\kappa}_t\right).$$

```
> RES=residuals(LCH,"pearson")
> colr=function(k) rainbow(110)[k*100]
> couleur=Vectorize(colr)(seq(.01,1,by=.01))
> plot(rep(RES$y,length(RES$x)),(RES$z),col=couleur[rep(RES$x,
+ each=length(RES$y))-RES$x[1]+1])
> plot(rep(RES$x,each=length(RES$y)),t(RES$z),col=couleur[rep(RES$y,length(RES$x))+1])
```

Residuals in Lee & Carter model



Residuals in Lee & Carter model



LifeMetrics Functions

LifeMetrics is based on R functions that can be downloaded from JPMorgan's website, that can be uploaded using `source("fitModels.r")`.

Standard functions are based on two matrices `etx` (for the exposure) and `dtx` for death counts, respectively at dates `t` and ages `x`.

Recall that, with discrete notation,

$$m(x,t) = \frac{\#\text{ deaths during calendar year } t \text{ aged } x \text{ last birthday}}{\text{average population during calendar year } t \text{ aged } x \text{ last birthday}}$$

Note that not only the Lee-Carter model is implemented, but several models,

Lee & Carter (1992), $\log m(x,t) = \beta_x^{(1)} + \beta_x^{(2)}\kappa_t^{(2)}$,

Renshaw & Haberman (2006), $\log m(x,t) = \beta_x^{(1)} + \beta_x^{(2)}\kappa_t^{(2)} + \beta_x^{(3)}\gamma_{t-x}^{(3)}$,

Currie (2006), $\log m(x,t) = \beta_x^{(1)} + \kappa_t^{(2)} + \gamma_{t-x}^{(3)}$,

Cairns, Blake & Dowd (2006), $\text{logit}(1-e^{-m(x,t)}) = \kappa_t^{(1)} + (x-\alpha)\kappa_t^{(2)}$,

Cairns *et al.* (2007), $\text{logit}(1-e^{-m(x,t)}) = \kappa_t^{(1)} + (x-\alpha)\kappa_t^{(2)} + \gamma_{t-x}^{(3)}$.

LifeMetrics Functions

For Lee & Carter model,

```
> res <- fit701(x, y, etx, dtx, wa)
```

where `wa` is a (possible) weight function. Here, assume that `wa=1`.

Remark : we have to remove very old ages,

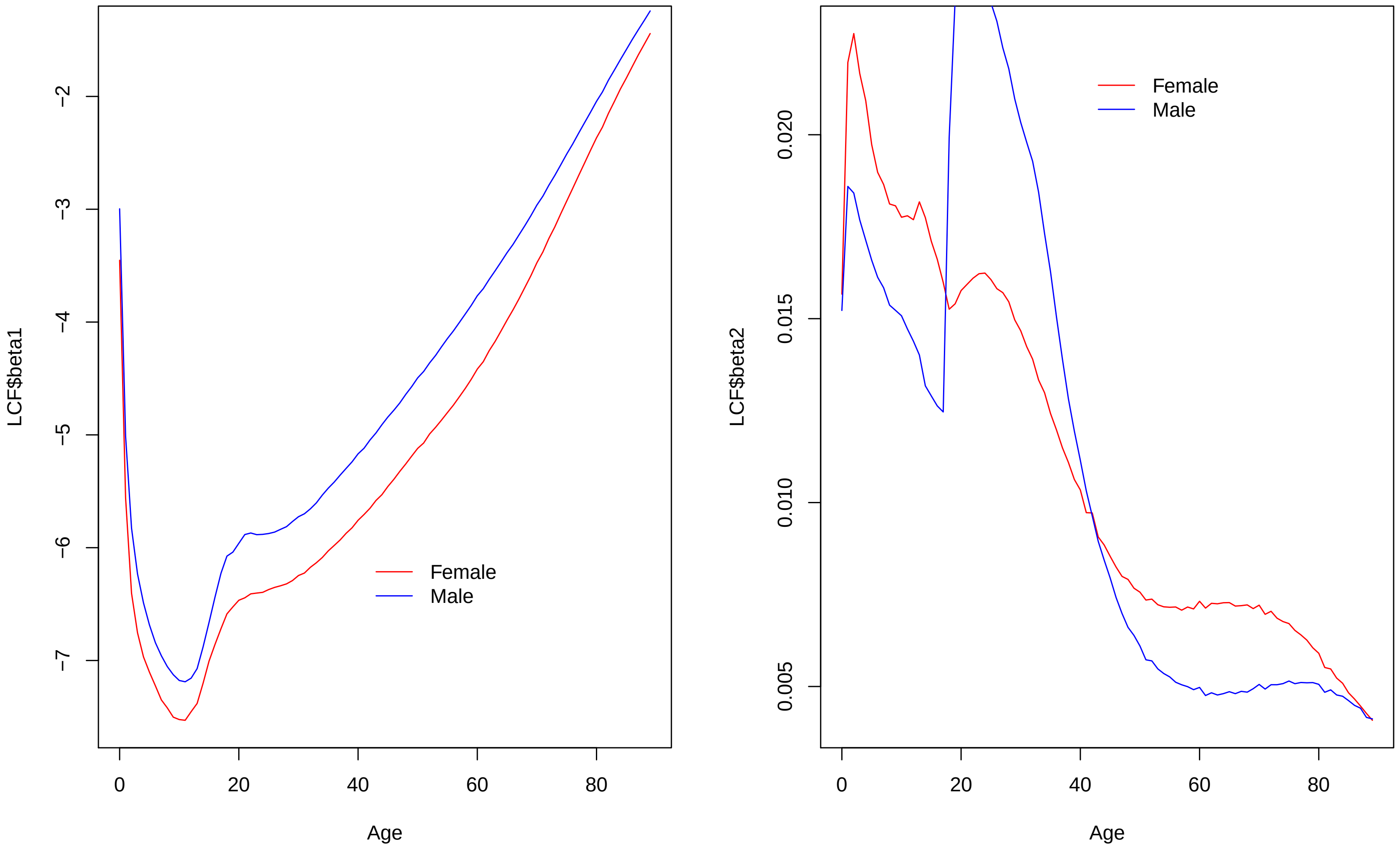
```
> DEATH <- DEATH[DEATH$Age<90,]
> EXPOSURE <- EXPOSURE[EXPOSURE$Age<90,]
> XV <- unique(DEATH$Age)
> YV <- unique(DEATH$Year)
> ETF <- t(matrix(EXPOSURE[,3],length(XV),length(YV)))
> DTF <- t(matrix(DEATH[,3],length(XV),length(YV)))
> ETH <- t(matrix(EXPOSURE[,4],length(XV),length(YV)))
> DTH <- t(matrix(DEATH[,4],length(XV),length(YV)))
> WA <- matrix(1,length(YV),length(XV))
> LCF  <-  fit701(xv=XV,yv=YV,etx=ETF,dtx=DTF,wa=WA)
> LCH  <-  fit701(xv=XV,yv=YV,etx=ETH,dtx=DTH,wa=WA)
```

LifeMetrics Functions

The output is the following, `LC$kappa1`, `LC$beta1`, ... `LC$ll` for the maximum log-likelihood estimators of different parameters, `LC$mtx` is an array with crude death rates, and `LC$mhat` with fitted death rates. `LC$cy` is the vector of cohort years of birth (corresponding to `LC$gamma3`).

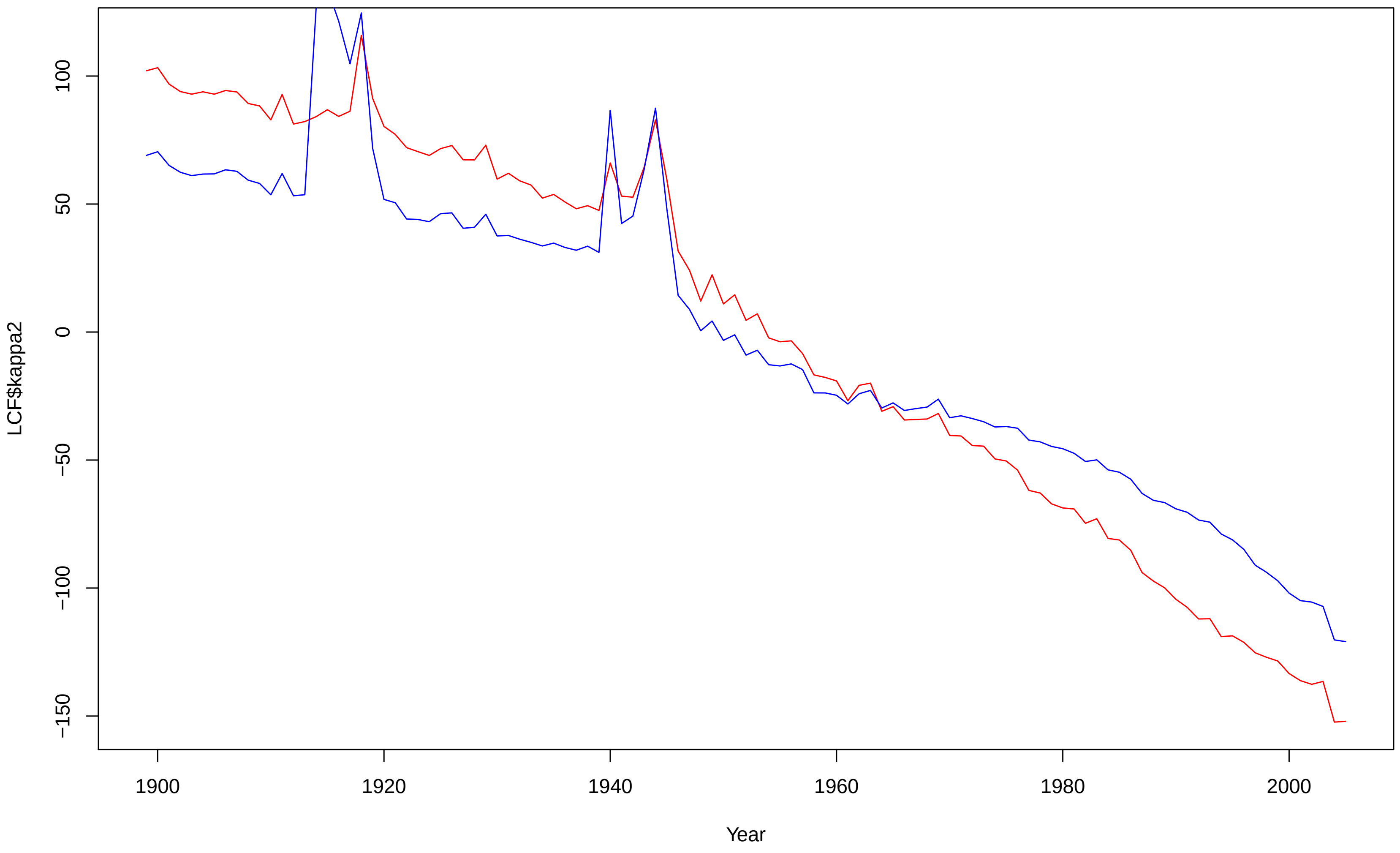
It it then possible to plot one of the cofficients against either `LC$x` or `LC$y`.

```
> plot(LCF$x,LCF$beta1,type="l",col="red")
> lines(LCH$x,LCH$beta1,col="blue",lty=2)
> plot(LCF$x,LCF$beta2,type="l",col="red")
> lines(LCH$x,LCH$beta2,col="blue",lty=2)
```



LCF$beta1
−2
−3
−4
−5
−6
−7
0
20
40
60
80
Age
Female
Male
LCF$beta2
0.005
0.010
0.015
0.020
0
20
40
60
80
Age
Female
Male

```
> plot(LCF$y,LCF$kappa2,type="l",col="red")
> lines(LCH$y,LCH$kappa2,col="blue")
```



Using the gnm package

(Much) more generally, it is possible to use the gnm package, to run a regression. Assume here that

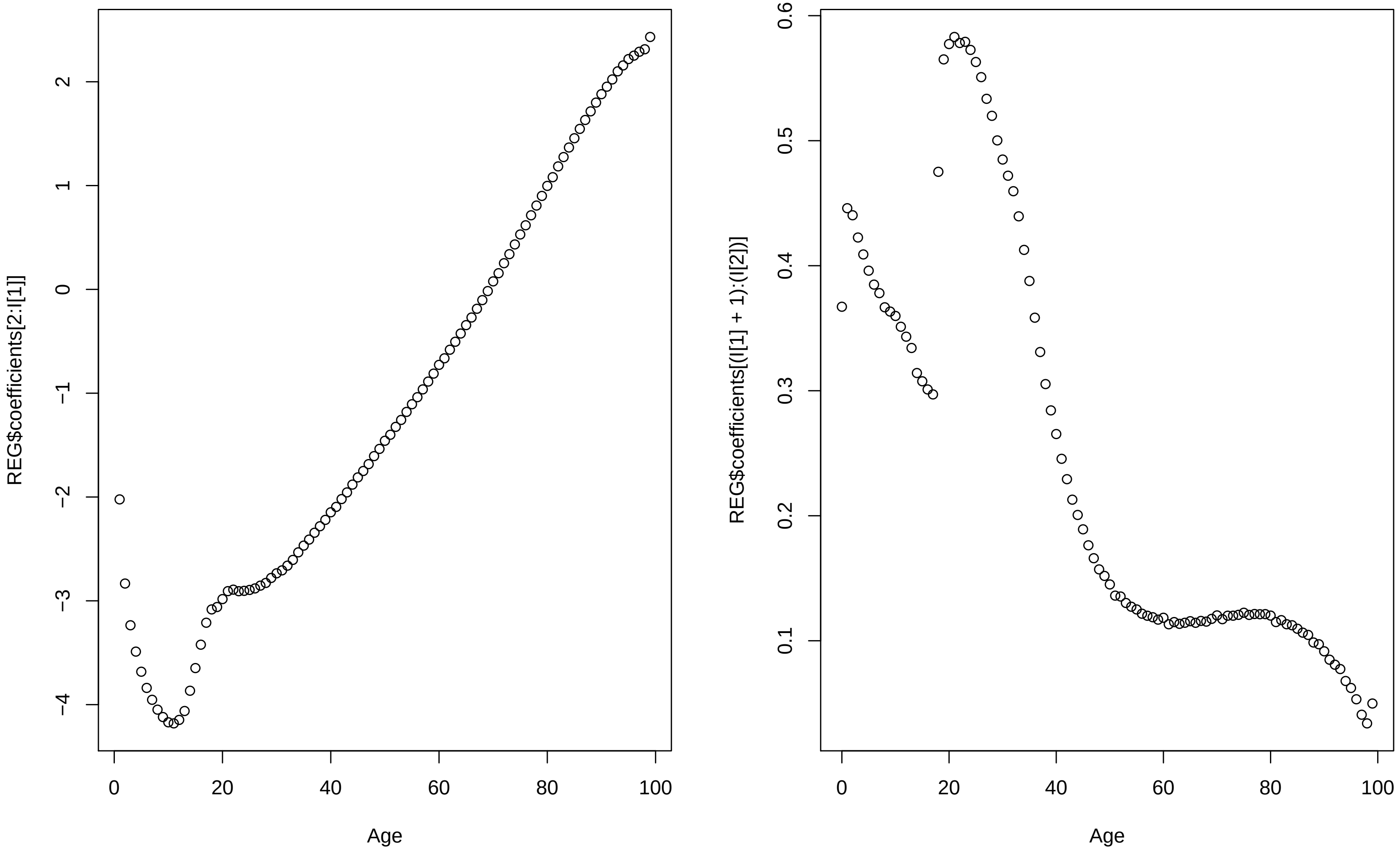
$$D_{x,t} \sim \mathcal{P}(\lambda_{x,t}) \text{ where } \lambda_{x,t} = E_{x,t} \exp\left(\alpha_x + \beta_x \cdot \kappa_t\right)$$

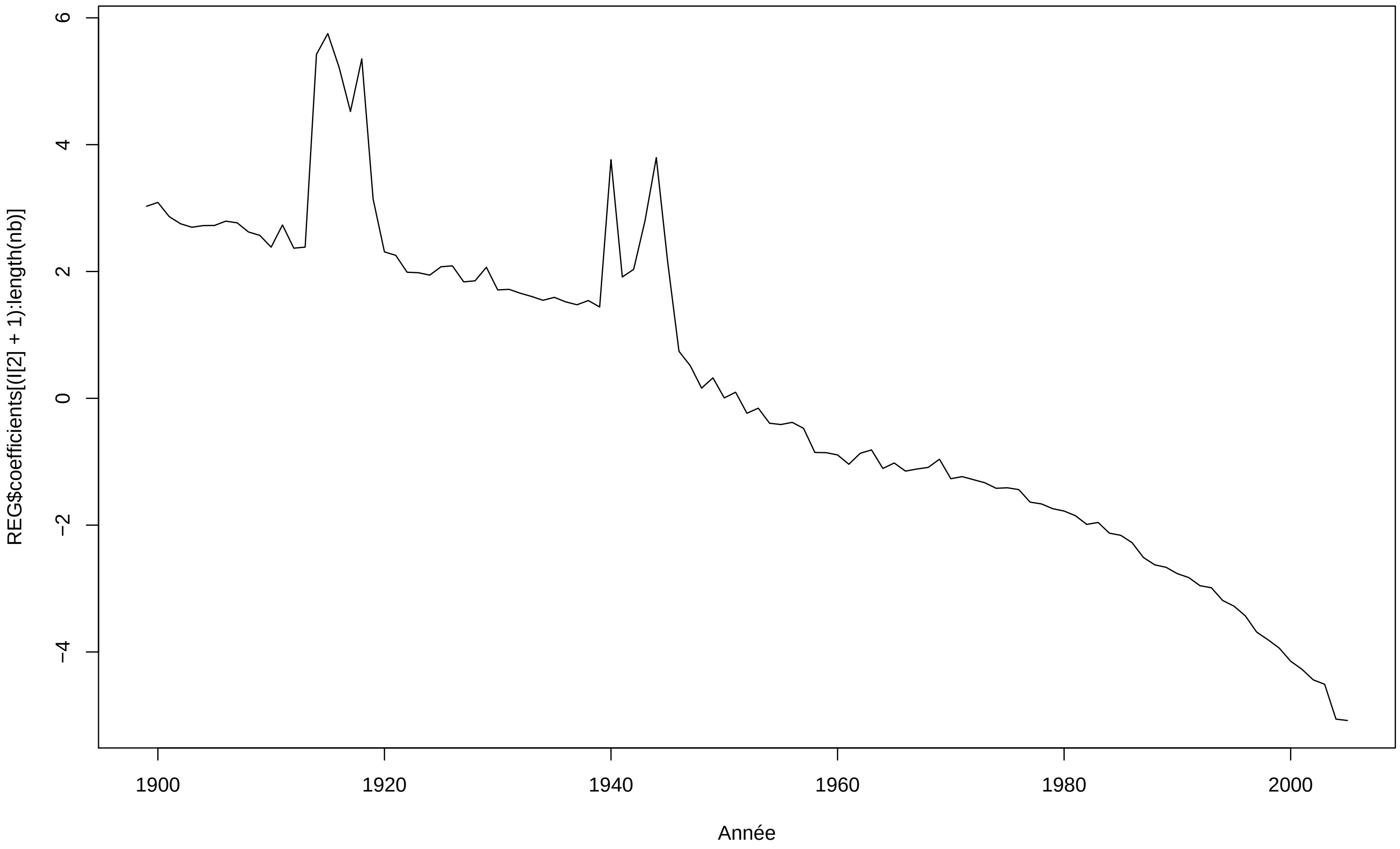
which is a generalized nonlinear regression model.

```
> library(gnm)
> Y=DEATH$Male
> E=EXPOSURE$Male
> Age= DEATH$Age
> Year=DEATH$Year
> I=(DEATH$Age<100)
> base=data.frame(Y=Y[I],E=E[I],Age=Age[I],Year=Year[I])
> REG=gnm(Y~factor(Age)+Mult((factor(Age)),factor(Year)),
  data=base,offset=log(E),family=quasipoisson)
Initialising
Running start-up iterations..
Running main iterations.......................
Done
```

Using the gnm package

```
> names(REG$coefficients[c(1:5,85:90)])
 [1] "(Intercept)"   "factor(Age)1"  "factor(Age)2"  "factor(Age)3"
 [5] "factor(Age)4"  "factor(Age)84" "factor(Age)85" "factor(Age)86"
 [9] "factor(Age)87" "factor(Age)88" "factor(Age)89"
> names(REG$coefficients[c(91:94,178:180)])
[1] "Mult(., factor(Year)).factor(Age)0"  "Mult(., factor(Year)).factor(Age)1"
[3] "Mult(., factor(Year)).factor(Age)2"  "Mult(., factor(Year)).factor(Age)3"
[5] "Mult(., factor(Year)).factor(Age)87" "Mult(., factor(Year)).factor(Age)88"
[7] "Mult(., factor(Year)).factor(Age)89"
> nomvar <- names(REG$coefficients)
> nb3 <- substr(nomvar,nchar(nomvar)-3,nchar(nomvar))
> nb2 <- substr(nomvar,nchar(nomvar)-1,nchar(nomvar))
> nb1 <- substr(nomvar,nchar(nomvar),nchar(nomvar))
> nb <- nb3
> nb[substr(nb,1,1)=="g"]<- nb1[substr(nb,1,1)=="g"]
> nb[substr(nb,1,1)=="e"]<- nb2[substr(nb,1,1)=="e"]
> nb <- as.numeric(nb)
> I <- which(abs(diff(nb))>1)
```


REG$coefficients[2:I[1]]
Age
REG$coefficients[(I[1] + 1):(I[2])]
Age

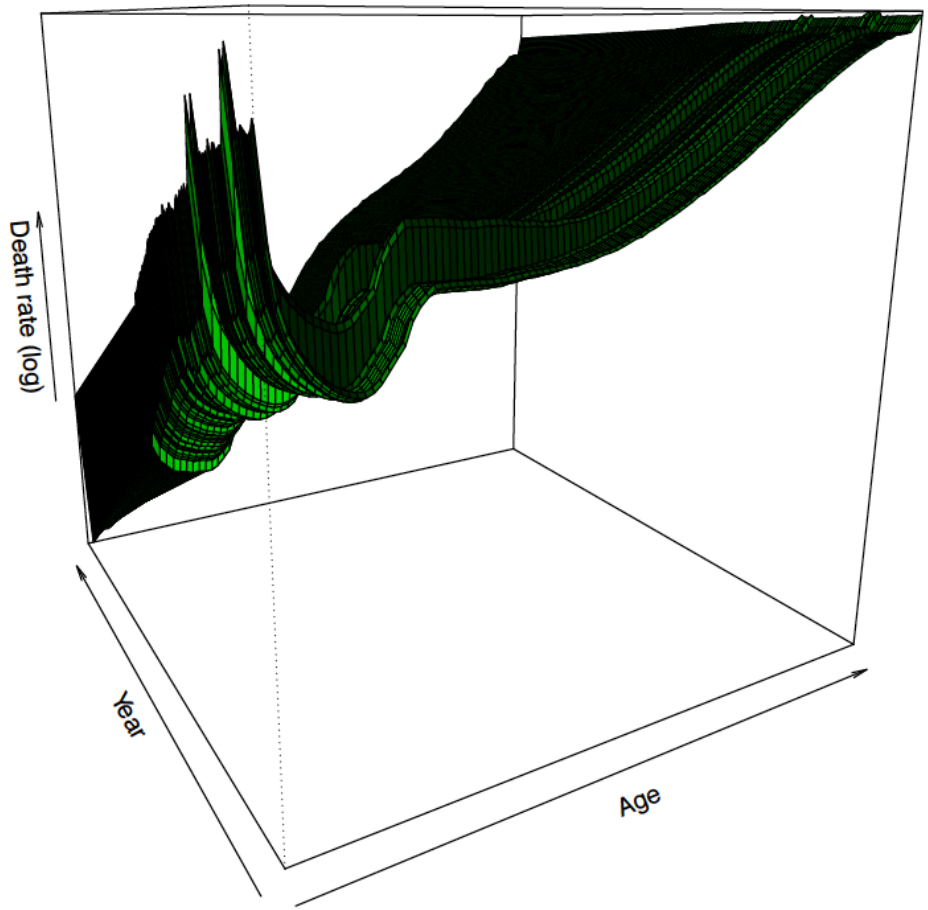


REG$coefficients[(I[2] + 1):length(nb)]
6
4
2
0
−2
−4
1900
1920
1940
1960
1980
2000
Année

Using Lee & Carter projections

Using estimators of α_x's, β_x's, as well as projection of κ_t's, it is possible to obtain projection of any actuarial quantities, based on projections of $\mu_{x,t}$'s. E.g.

```
> A <- LCH$ax; B <- LCH$bx
> K1 <- LCH$kt; K2 <- LCH$kt[99]+LCHf$kt.f$mean; K <- c(K1,K2)
> MU <- matrix(NA,length(A),length(K))
> for(i in 1:length(A)){ for(j in 1:length(K)){
+ MU[i,j] <- exp(A[i]+B[i]*K[j]) }}
```



It is then possible to extrapolate $k \mapsto {}_k p_x$'s

```
> t=2000
> x=40
> s=seq(0,99-x-1)
```

```
> MUd=MU[x+1+s,t+s-1898]
> (Pxt=cumprod(exp(-diag(MUd))))
 [1] 0.99838440 0.99663098 0.99469369 0.99248602 0.99030804 0.98782725
 [7] 0.98417947 0.98017722 0.97575106 0.97098896 0.96576107 0.96006617
[13] 0.95402111 0.94749333 0.94045500 0.93291535 0.92484762 0.91622709
[19] 0.90707101 0.89726011 0.88690981 0.87577047 0.86405282 0.85159220
[25] 0.83850049 0.82472277 0.81011757 0.79478797 0.77847592 0.76144457
[31] 0.74364218 0.72457570 0.70474824 0.68387491 0.66193090 0.63903821
[37] 0.61469237 0.58924560 0.56257772 0.53478172 0.50577349 0.47480005
[43] 0.44324965 0.41055038 0.37750446 0.34390607 0.30973747 0.27613617
[49] 0.24253289 0.21038508 0.17960626 0.14970800 0.12276231 0.09902686
[55] 0.07742879 0.05959964 0.04495042 0.03281240 0.02366992
```

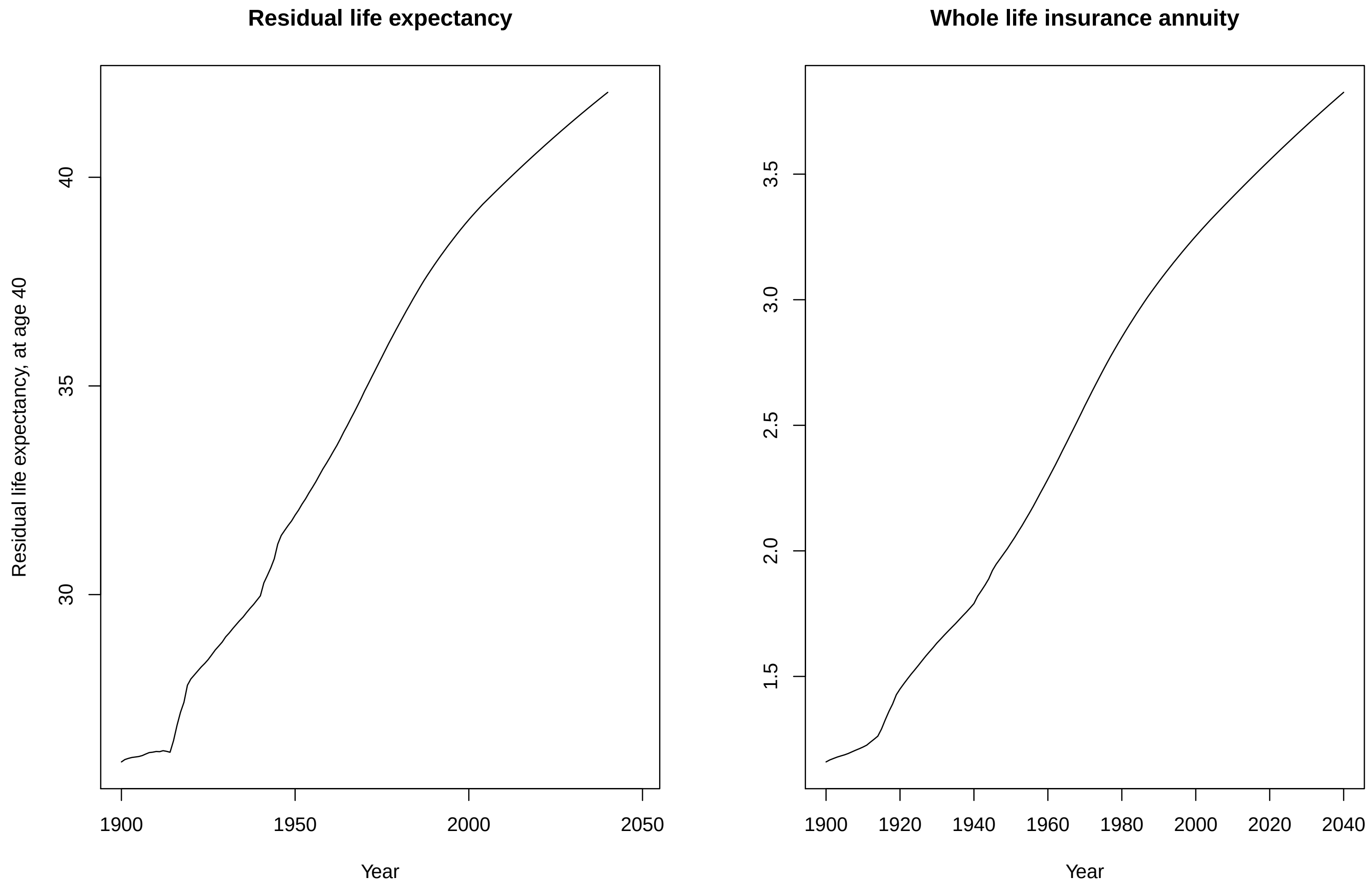
and then we can derive projections of several actuarial (or demography) quantities. E.g. remaining lifetimes

```
> x=40
> E=rep(NA,150)
> for(t in 1900:2040){
+ s=seq(0,90-x-1)
+ MUd=MU[x+1+s,t+s-1898]
```

```
+ Pxt=cumprod(exp(-diag(MUd)))
+ ext=sum(Pxt)
+ E[t-1899]=ext}
> plot(1900:2049,E)
```

or expected present value of deferred whole life annuities, purchased at age 40, deferred of 30 years

```
> r=.035: m=70
> VV=rep(NA,141)
> for(t in 1900:2040){
+ s=seq(0,90-x-1)
+ MUd=MU[x+1+s,t+s-1898]
+ Pxt=cumprod(exp(-diag(MUd)))
+ h=seq(0,30)
+ V=1/(1+r)^(m-x+h)*Pxt[m-x+h]
+ VV[t-1899]=sum(V,na.rm=TRUE)}
> plot(1900:2040,VV)
```



Residual life expectancy
Residual life expectancy, at age 40
30
35
40
1900
1950
2000
2050
Year
Whole life insurance annuity
1.5
2.0
2.5
3.0
3.5
1900
1920
1940
1960
1980
2000
2020
2040
Year

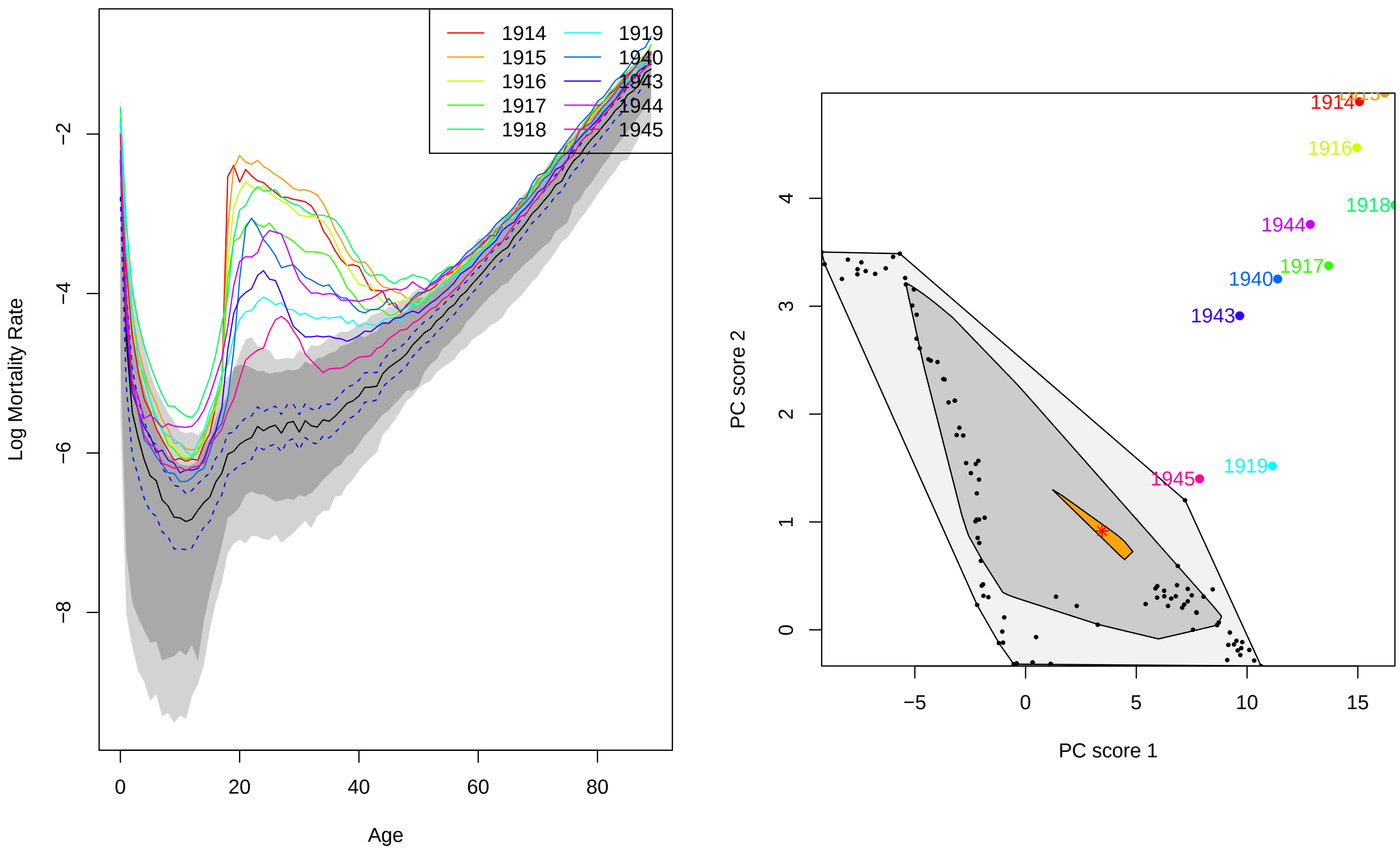
Mortality rates as functional time series

It is possible to consider functional time series using `rainbow` package

```
> library(rainbow)
> rownames(MUH)=AGE
> colnames(MUH)=YEAR
> rownames(MUF)=AGE
> colnames(MUF)=YEAR
> MUH=MUH[1:90,]
> MUF=MUF[1:90,]
> MUHF=fts(x = AGE[1:90], y = log(MUH), xname = "Age",yname = "Log Mortality Rate")
> MUFF=fts(x = AGE[1:90], y = log(MUF), xname = "Age",yname = "Log Mortality Rate")
> fboxplot(data = MUHF, plot.type = "functional", type = "bag")
```

Using principal components, it is possible to detect outliers

```
> fboxplot(data = MUHF, plot.type = "bivariate", type = "bag")
```


Log Mortality Rate
Age
1914
1915
1916
1917
1918
1919
1940
1943
1944
1945
PC score 1
PC score 2

Cohort effect and Lee & Carter model

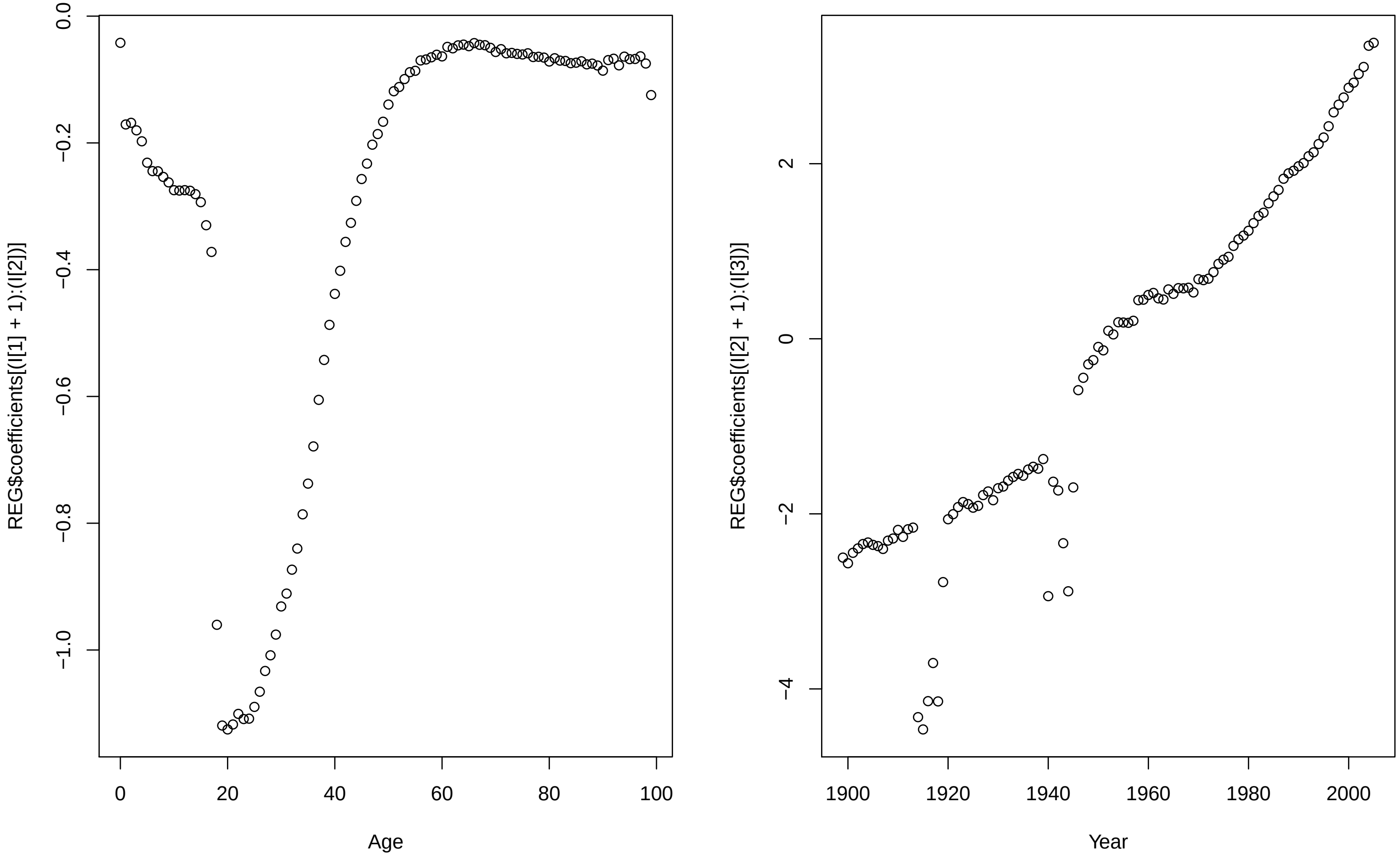
A natural idea is to include (on top of the age x and the year t) a cohort factor, based on the year of birth, $t - x$

$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \gamma_x \cdot \delta_{t-x} + \eta_{x,t},$$

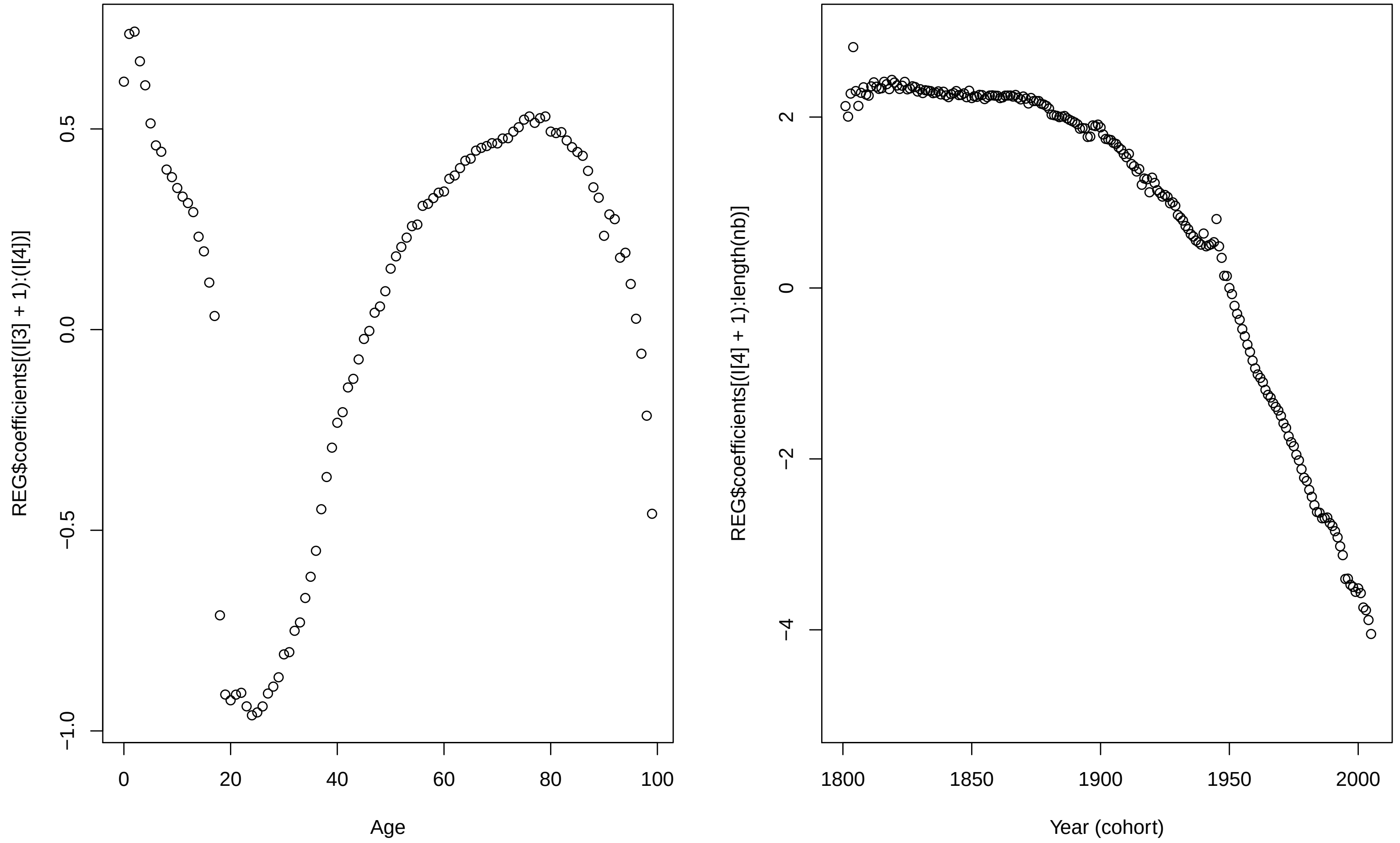
as in Renshaw & Haberman (2006).

Using gnm function, it is possible to estimate that model, assuming again that a log-Poisson model for death counts is valid,

```
> D=as.vector(BASEB)
> E=as.vector(BASEC)
> A=rep(AGE,each=length(ANNEE))
> Y=rep(ANNEE,length(AGE))
> C=Y-A
> base=data.frame(D,E,A,Y,C,a=as.factor(A),
+ y=as.factor(Y),c=as.factor(C))
> LCC=gnm(D~a+Mult(a,y)+Mult(a,c),offset=log(E), family=poisson,data=base)
```



REG$coefficients[(I[1] + 1):(I[2])]
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0
20
40
60
80
100
Age
REG$coefficients[(I[2] + 1):(I[3])]
2
0
−2
−4
1900
1920
1940
1960
1980
2000
Year



REG$coefficients[(I[3] + 1):(I[4])]
0.5
0.0
−0.5
−1.0
0
20
40
60
80
100
Age
REG$coefficients[(I[4] + 1):length(nb)]
2
0
−2
−4
1800
1850
1900
1950
2000
Year (cohort)