HawkesModel

Fitting Hawkes model to marathon record data

Produce estimates for model, exponential Hawkes process, using MLE: - baseline intensity, - reproduction mean
- exponential fertility function rate

Note: originally data was in days from the first world record which is set as time 0. The model was fit through day 40,300 (the last record at day 40,231). Rescaling helps with visualization - several options, one put in years by dividing by 365.

Three parameter estimates are first output.

NOTE: the results change if one doesn't set a seed…rescaling seems to help variability in that.

```
record_table_mod<-read_rds("record_table_mod.rds")

days_between = as.numeric(diff(record_table_mod$Date_ymd))
daysfromstart <- cumsum(days_between)
daysfromstart <- c(0,daysfromstart)  ### get data in terms of days from first record (time 0)

### Rescale

  # max time is 400 (about 402 days per unit)
#daysfromstart_mod <- daysfromstart/max(daysfromstart)*100
#days_between_mod = diff(daysfromstart_mod)
  # units of year
daysfromstart_mod2 <- daysfromstart/365
days_between_mod2 = diff(daysfromstart_mod2)

set.seed(13)
optMarathon<-mle(daysfromstart_mod2,"Exponential",110.5)  # end date picked number greater than longest
optMarathon$par
```

```
## [1] 0.3964426 0.1247670 3.6463273
```

```
summary(optMarathon)
```

```
##        Length Class            Mode
## par     3     -none-           numeric
## model   1     Rcpp_Exponential S4
## events 50     -none-           numeric
## end     1     -none-           numeric
## opt    20     nloptr           list
```

```r
optMarathon$events
```

```
##  [1]   0.0000000   0.4410959   0.5561644   0.7890411   0.7945205   1.1041096
##  [7]   4.8027397   4.8547945   6.3534247  12.0876712  17.2301370  20.9616438
## [13]  26.6739726  26.7013699  26.7095890  27.2958904  38.7616438  43.9205479
## [19]  44.9178082  45.2273973  45.9534247  48.0849315  50.1178082  52.1671233
## [25]  54.6054795  54.9287671  54.9863014  55.9260274  56.2821918  56.9232877
## [31]  59.4000000  60.8904110  62.0383562  65.5671233  69.5835616  71.8054795
## [37]  73.4191781  76.2958904  76.7917808  79.7863014  90.2191781  91.3123288
## [43]  93.7863014  95.2438356  99.2520548 100.2493151 103.2410959 105.2547945
## [49] 106.2520548 110.2219178
```

```r
optMarathon$end
```

```
## [1] 110.5
```

```r
optMarathon$model$param
```

```
##           [,1]
## [1,] 0.3964426
## [2,] 0.1247670
## [3,] 3.6463273
```

```r
optMarathon$model$mean()  # expected value
```

```
## [1] 0.4529567
```

```r
optMarathon$model$dmean()  # Jacobian matrix of expected value
```

```
##          [,1]
## [1,] 1.142553
## [2,] 0.517527
## [3,] 0.000000
```

```r
optMarathon$model$ddmean()  # Hessian matrix of expected value
```

```
##          [,1]     [,2] [,3]
## [1,] 0.000000 1.305427    0
## [2,] 1.305427 1.182604    0
## [3,] 0.000000 0.000000    0
```

```r
optMarathon$model$loglik(daysfromstart, optMarathon$end)  #  log-likelihood
```

```
## [1] Inf
```

```r
optMarathon$model$dloglik(daysfromstart, optMarathon$end)  # Jacobian matrix of log-lik
```

```
##          [,1]
## [1,] 15.61963
## [2,]      Inf
## [3,]      NaN
```

```
optMarathon$model$ddloglik(daysfromstart, optMarathon$end) # hessian matrix of log-lik
```

```
##               [,1]          [,2]        [,3]
## [1,] -3.142830e+02 -1.617551e-02 0.003534246
## [2,] -1.617551e-02 -3.914255e-05         NaN
## [3,]  3.534246e-03           NaN         NaN
```

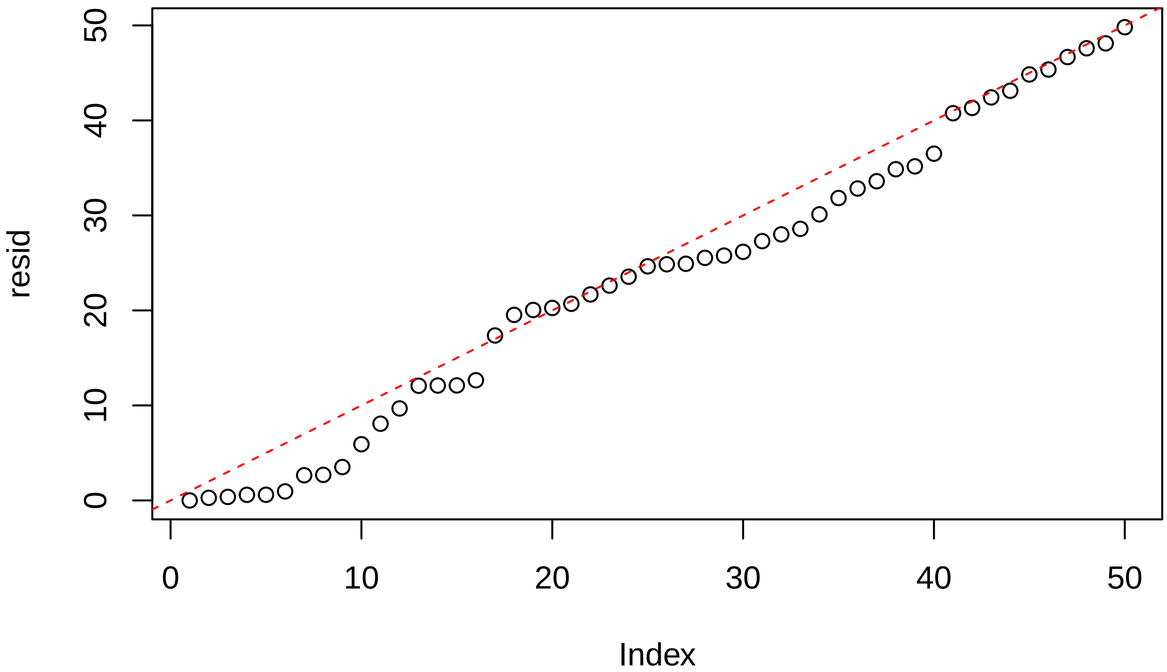
Residuals

From the help: "Outputs the residuals (values of the compensator at the times of arrival) of a Hawkes process. Useful function for diagnosis through the random time change theorem: the residuals should follow a unit rate Poisson process" Based on the example in the help I assume should follow the y=x line… we see divergence here suggesting an issue (in what direction?)

```
resid<-residuals(daysfromstart_mod2, fun = optMarathon$par[1], repr = optMarathon$par[2],
                 family = "exp", rate = optMarathon$par[3])
resid
```

```
##  [1]  0.0000000  0.2746564  0.3715890  0.5916302  0.5981594  0.9513823
##  [7]  2.6526125  2.6948194  3.5159392  5.9149714  8.0784311  9.6825264
## [13] 12.0719043 12.0946276 12.1049033 12.6508528 17.3630407 19.5330173
## [19] 20.0498534 20.2592292 20.7014652 21.6829670 22.6136352 23.5508425
## [25] 24.6423317 24.8568918 24.9105772 25.5318177 25.7698100 26.1695075
## [31] 27.2916865 28.0067883 28.5852882 30.1109147 31.8279694 32.8335616
## [37] 33.5977580 34.8633204 35.1642265 36.4966030 40.7574098 41.3132310
## [43] 42.4210883 43.1230856 44.8374954 45.3543315 46.6684531 47.5914577
## [49] 48.1083724 49.8102519
```

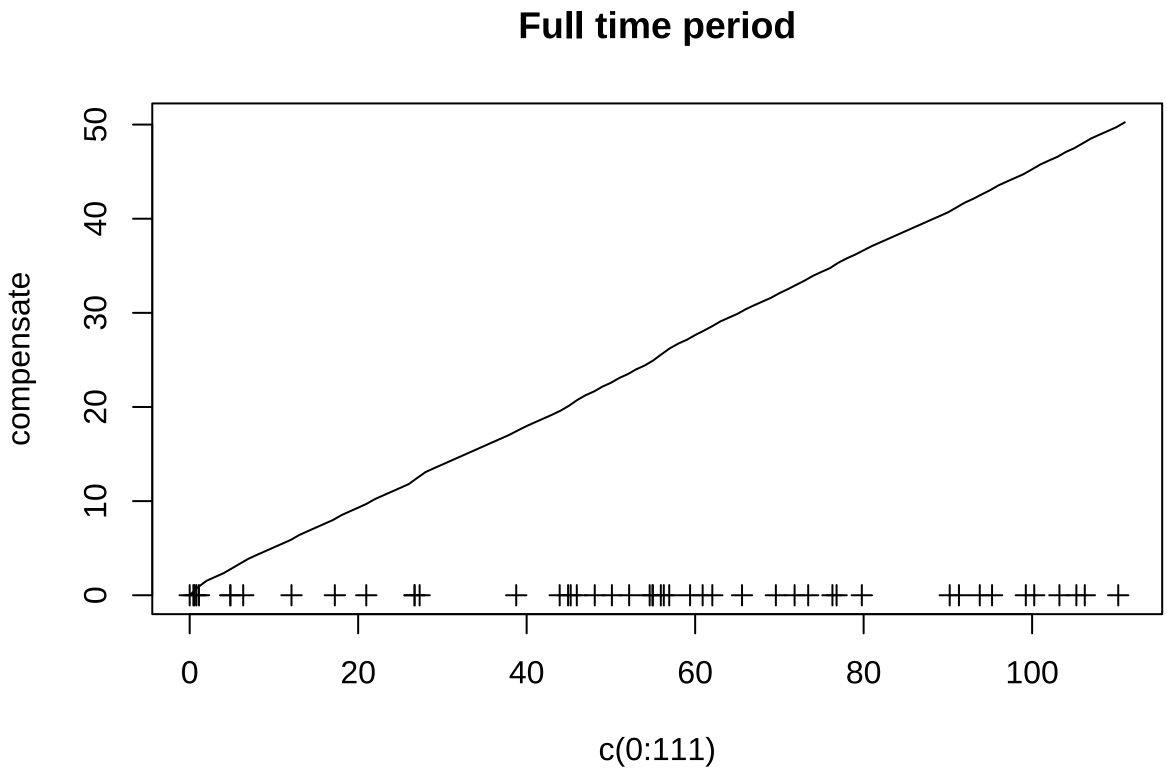
```
plot(resid)
abline(0, 1, col="red", lty="dashed")
```



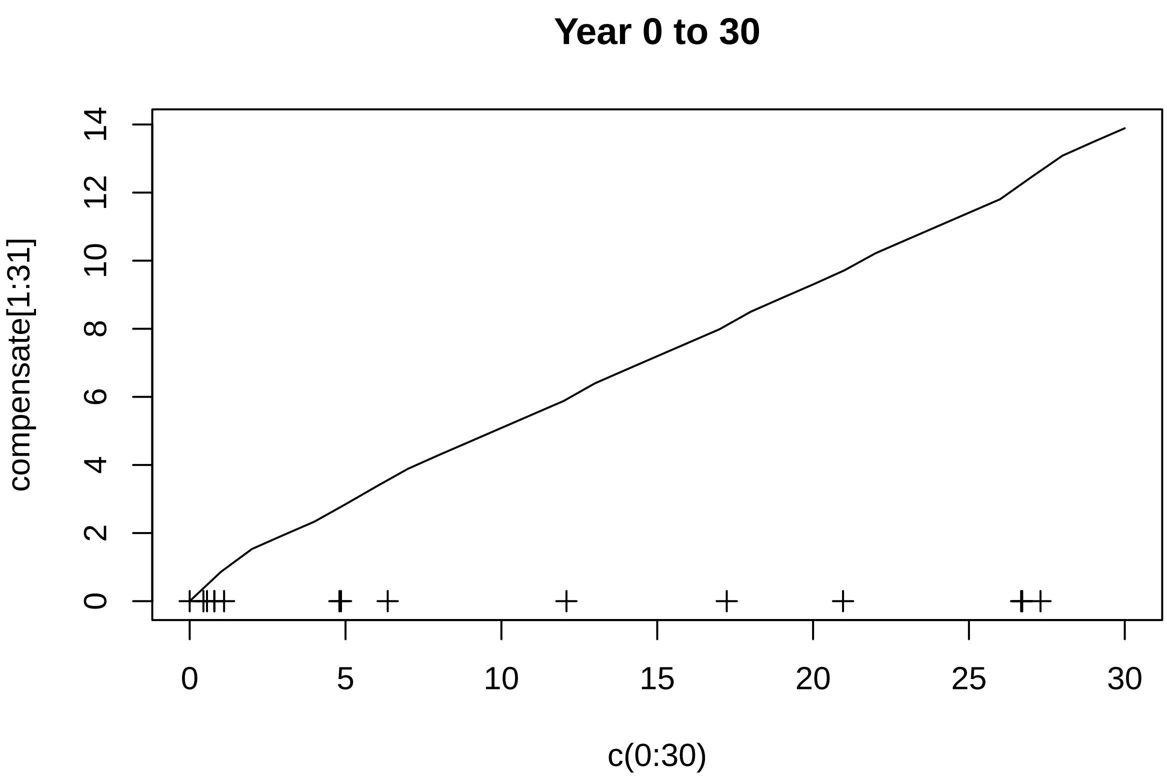
Compensator

From the help: the compensator (integrated intensity) of a Hawkes process…kind of cool, can see how the events (world records) up the intensity. Event times are the plus symbols plotted below the intensity function line.

```
compensate<-compensator(daysfromstart_mod2, t=0:111, fun = optMarathon$par[1],
                        repr = optMarathon$par[2], family = "exp", rate = optMarathon$par[3])
plot(c(0:111),compensate,main="Full time period", type = "l")
points(daysfromstart_mod2,rep(0,length(daysfromstart_mod2)), pch=3)
```




```
### zoom in...
plot(c(0:30),compensate[1:31],main="Year 0 to 30", type = "l" )
points(daysfromstart_mod2[0:16],rep(0,length(daysfromstart_mod2[0:16])), pch=3)
```

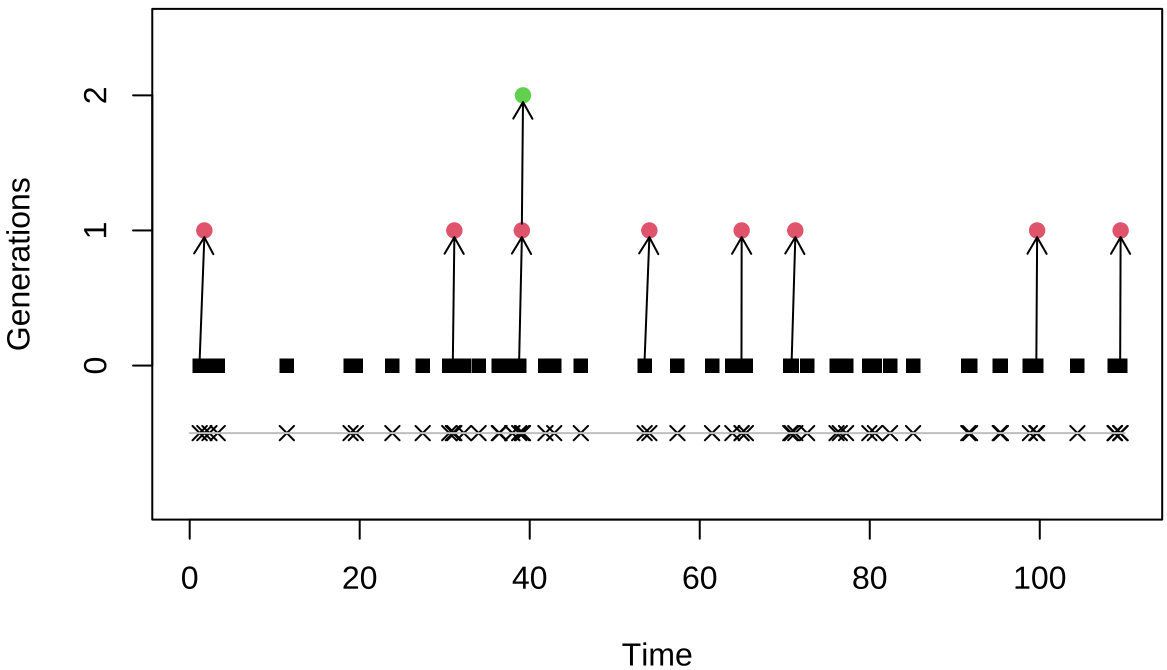


Simulation using Hawkes model with MLE parameter estimates from data

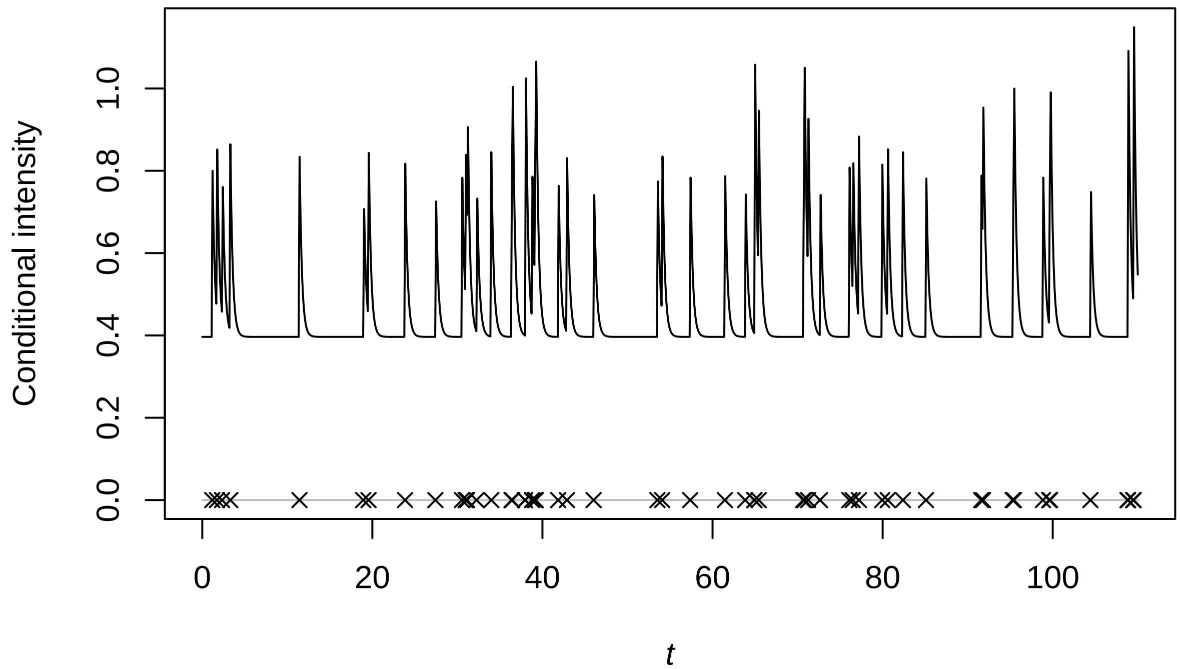
Simulation for the same number of years used in the estimation. Last plot is the residuals - we see since from the model we simulated generally seem to follow the y=x line.

Intensity plots with the rescaling are much easier to see. Different seeds different results, some have second “child”.

```
set.seed(777)
simRecs <- hawkes(110, fun = optMarathon$par[1], repr = optMarathon$par[2],
                  family = "exp", rate = optMarathon$par[3])
plot(simRecs, intensity = FALSE)
```




```
plot(simRecs, intensity = TRUE)
```




```
plot(residuals(simRecs))
abline(0, 1, col="red", lty="dashed")
```