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",111)  # end date picked number greater than longest t
optMarathon$par
```

```
## [1] 0.3962126 0.1205231 3.9097077
```

```
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] 111
```

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

```
##           [,1]
## [1,] 0.3962126
## [2,] 0.1205231
## [3,] 3.9097077
```

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

```
## [1] 0.4505094
```

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

```
##          [,1]
## [1,] 1.137040
## [2,] 0.512247
## [3,] 0.000000
```

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

```
##          [,1]     [,2] [,3]
## [1,] 0.000000 1.292859    0
## [2,] 1.292859 1.164890    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,] 15.19365
## [2,]      Inf
## [3,]      NaN
```

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

```
##               [,1]          [,2]        [,3]
## [1,] -3.146506e+02 -0.0101993302 0.002168279
## [2,] -1.019933e-02 -0.0000157157         NaN
## [3,]  2.168279e-03           NaN         NaN
```

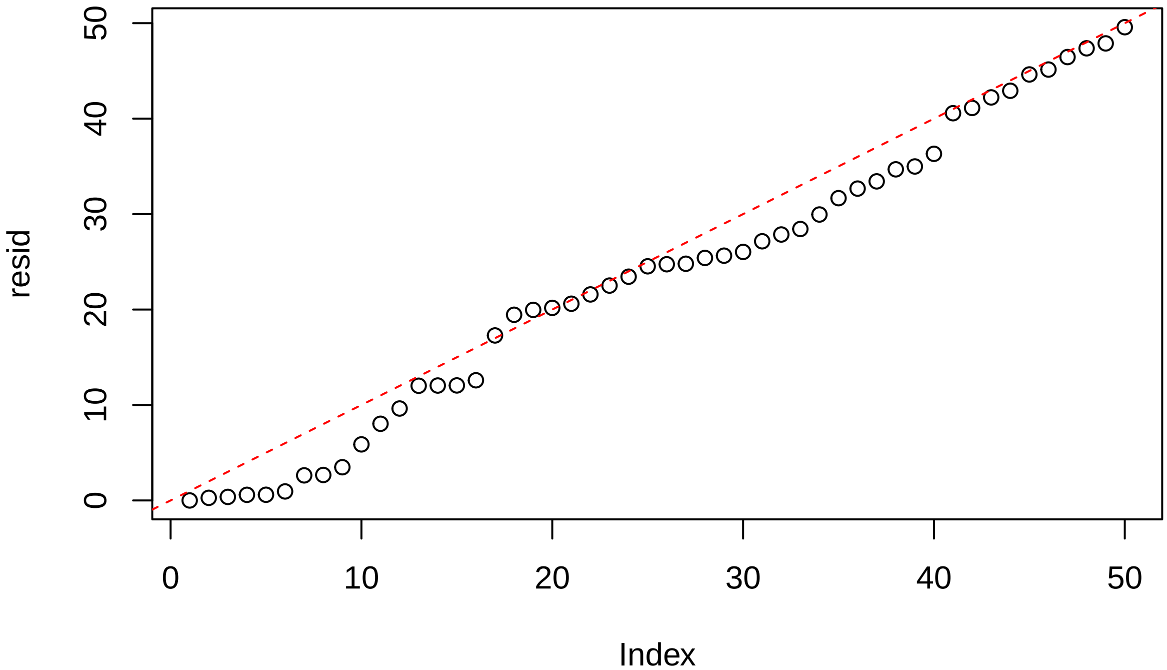
Residuals

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?)

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

```
##  [1]  0.0000000  0.2738074  0.3708477  0.5892729  0.5957985  0.9442113
##  [7]  2.6260447  2.6688632  3.4808675  5.8739959  8.0320287  9.6310218
## [13] 12.0148417 12.0379393 12.0484315 12.5882588 17.2862219 19.4507678
## [19] 19.9639760 20.1729511 20.6085922 21.5828104 22.5087735 23.4412637
## [25] 24.5279255 24.7424938 24.7964273 25.4065312 25.6428790 26.0364982
## [31] 27.1507142 27.8614101 28.4357599 29.9557837 31.6676705 32.6685251
## [37] 33.4282170 34.6887476 34.9884094 36.3127391 40.5669006 41.1188652
## [43] 42.2212783 42.9188987 44.6279327 45.1411408 46.4494863 47.3678173
## [49] 47.8810704 49.5769463
```

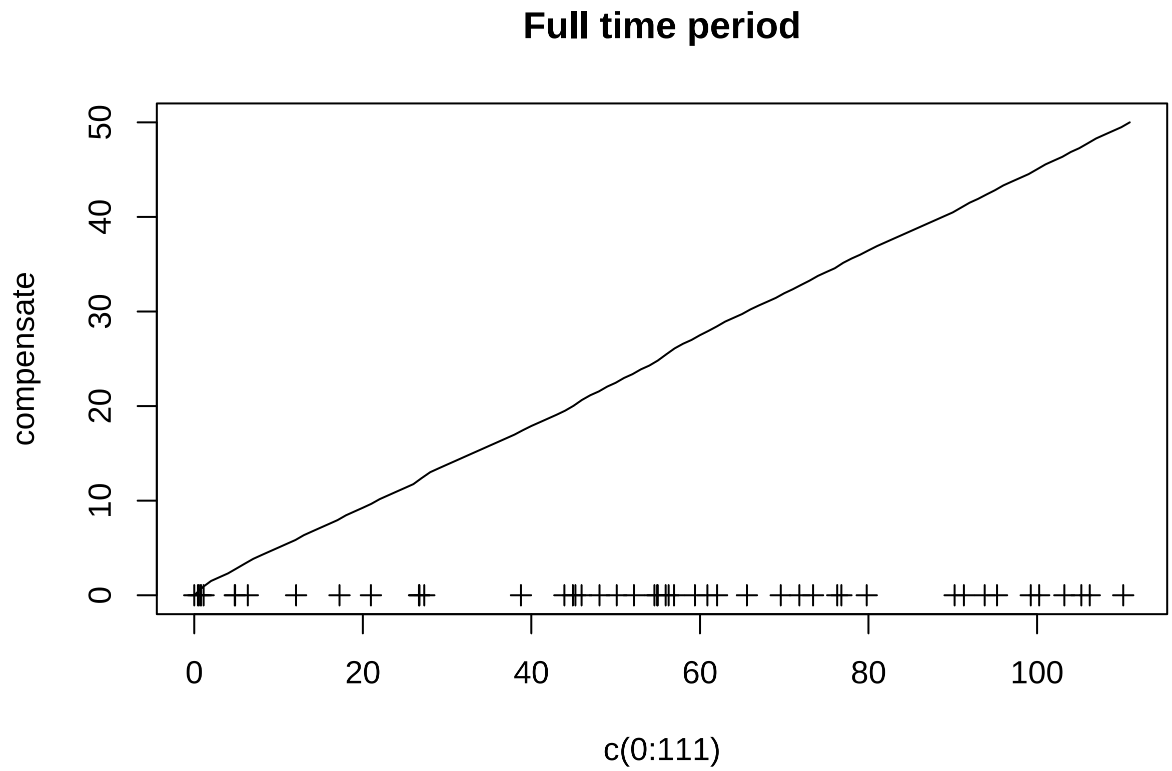
```r
plot(resid)
abline(0, 1, col="red", lty="dashed")
```



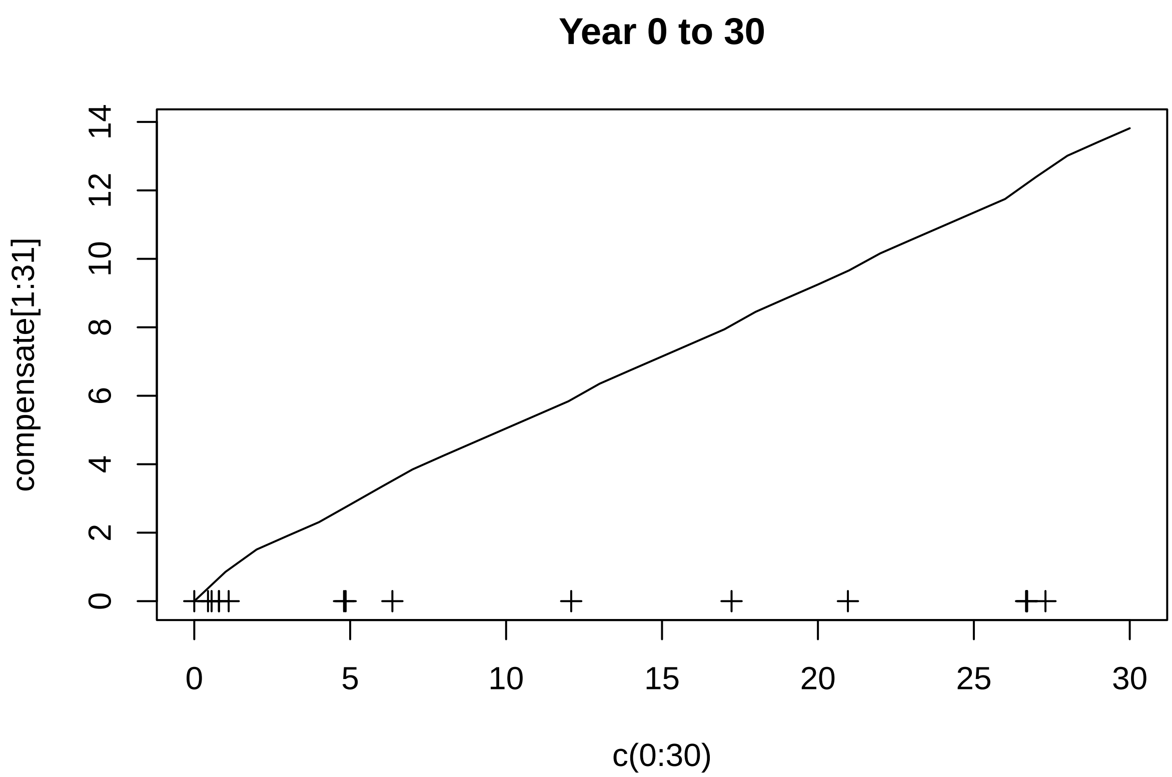
Compensator

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)
```




```r
  ### 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)
```

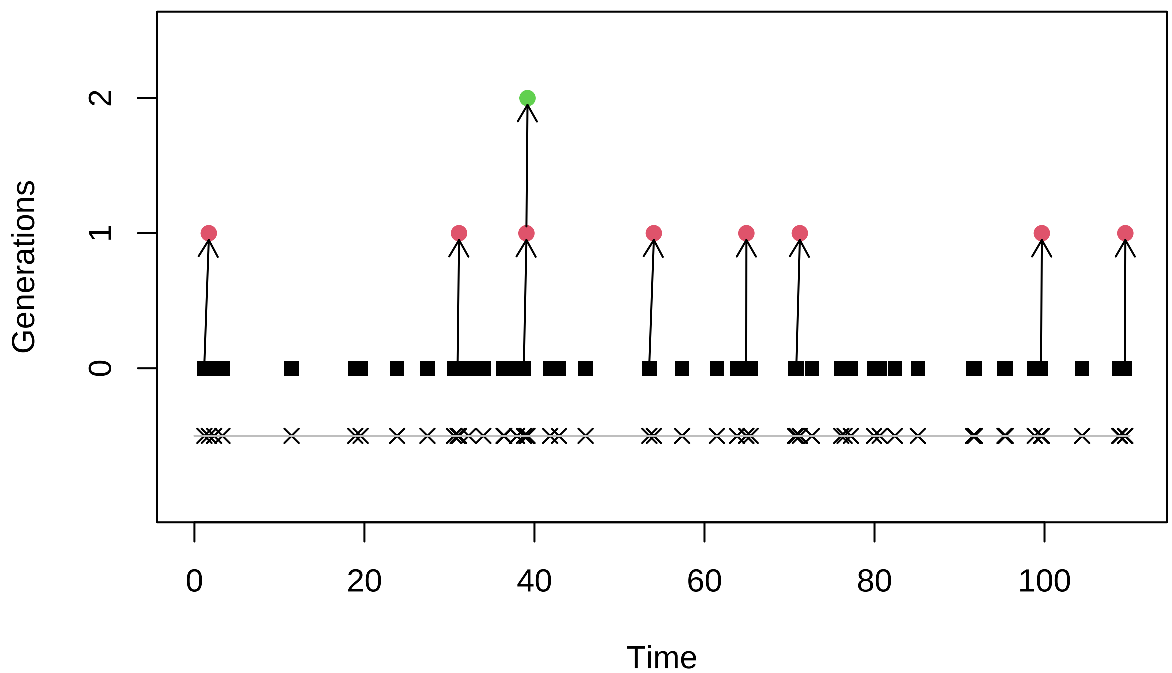


Simulation using Hawkes model with MLE parameter estimates from data

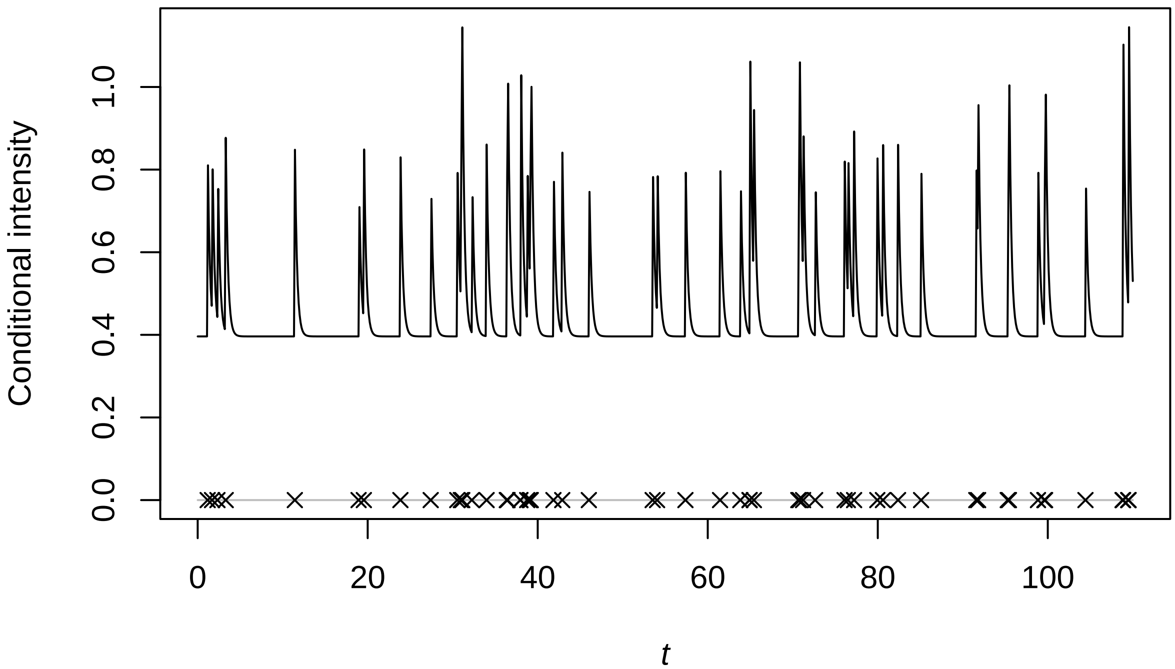
Simulation for the same number of years used in the estimation. Last plot is the residuals - we see since from the model we simulated generally seem to follow the y=x line.

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

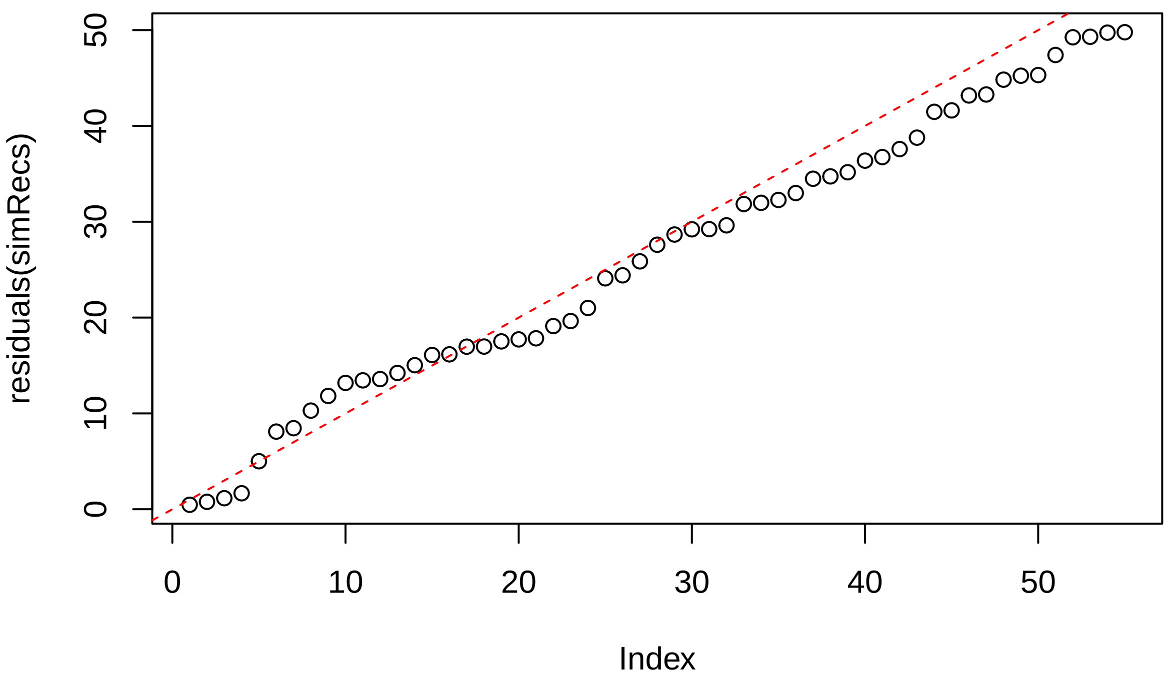
```
set.seed(777)
simRecs <- hawkes(110, fun = optMarathon$par[1], repr = optMarathon$par[2],
                  family = "exp", rate = optMarathon$par[3])
plot(simRecs, intensity = FALSE)
```




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




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



Comparing Hawkes model to constant rate model

The estimated constant rate is 0.44 using the year scaled data. The baseline parameter rate from the Hawkes model is lower; depends slightly on the seed but around 0.396. This makes sense - the conditional "excite" portion of the model will increase the rate at times so the baseline is set lower. The other two parameters determine how much the rate jumps…max will be at T1-t = 0 or at (in example for seed I used) around

0.47...so an immediate doubling of the rate. Decay is fairly rapid though (see intensity from simulation) as the parameter for the exponential decay is large (3.9).

```
expfit=fitdistr(days_between_mod2,"exponential")
exprate<-expfit$estimate
exprate
```

```
##      rate
## 0.4445577
```

```
optMarathon$par
```

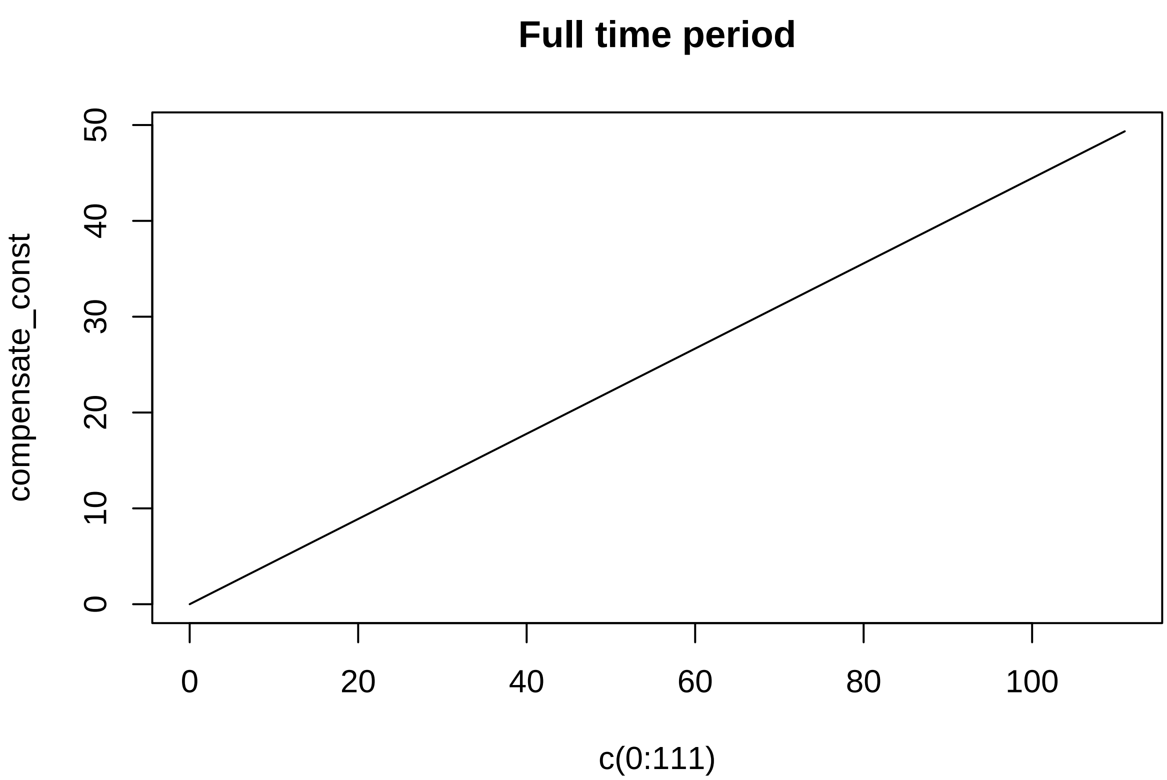
```
## [1] 0.3962126 0.1205231 3.9097077
```

```
optMarathon$par[2]*optMarathon$par[3]
```

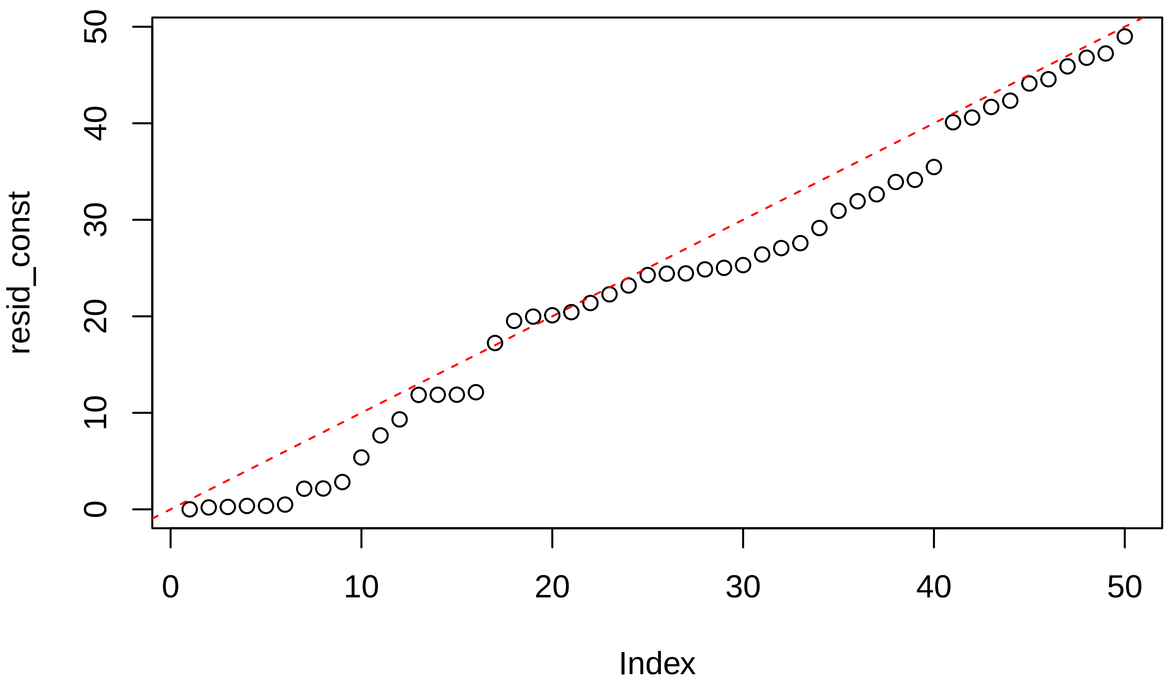
```
## [1] 0.4712102
```

CONSTANT MODEL compensator and residual plot are shown below. Note the features of Hawkesbow package make it easy to do this...simply set the baseline rate to the rate estimated for the Poisson process model and set the other two rates to 0.

```
compensate_const<-compensator(daysfromstart_mod2, t=0:111, fun = exprate, repr = 0,
                        family = "exp", rate = 0)
plot(c(0:111),compensate_const,main="Full time period", type = "l")
```

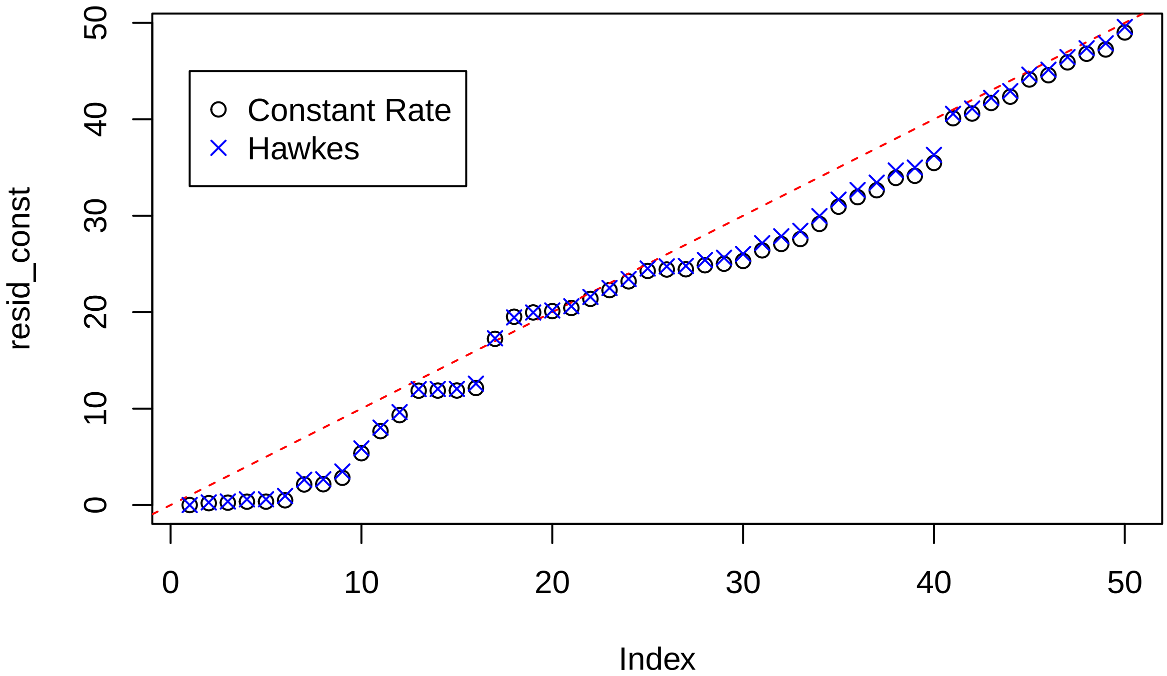


```
resid_const<-residuals(daysfromstart_mod2, fun = exprate, repr = 0,
                       family = "exp", rate = 0)
plot(resid_const)
abline(0, 1, col="red", lty="dashed")
```



Residual comparison is below. We see Hawkes model generally better but maybe not by a "significant" amount?

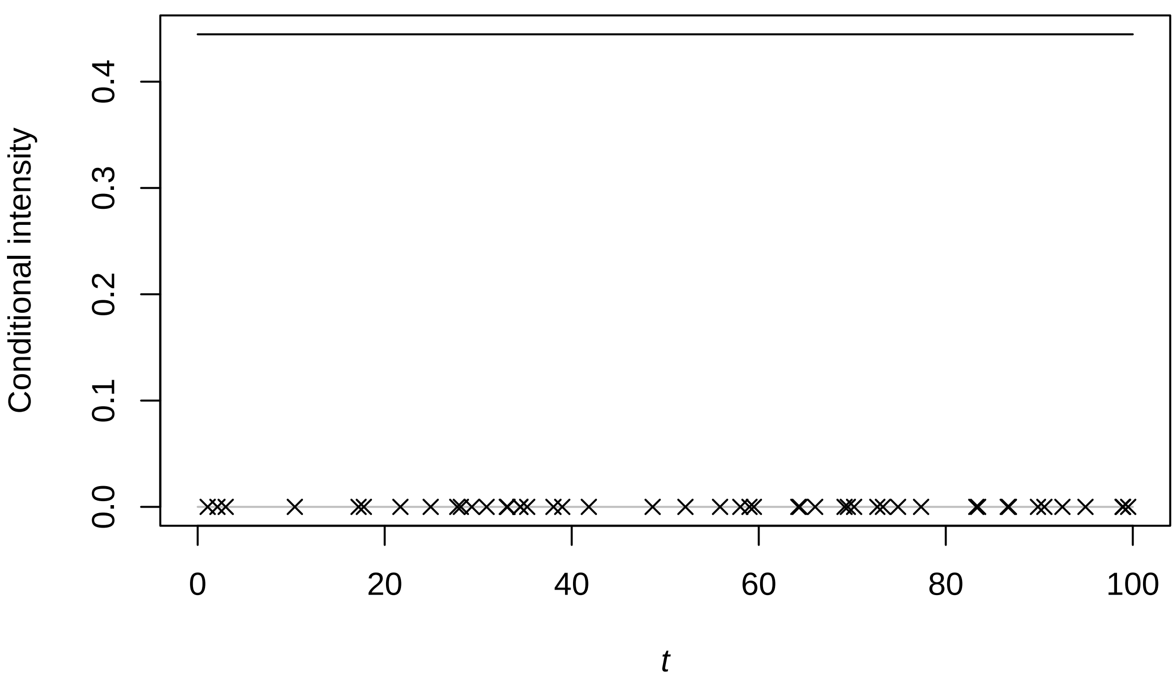
```
plot(resid_const)
abline(0, 1, col="red", lty="dashed")
points(resid, col ="blue", pch = 4)
legend(1, 45, legend=c("Constant Rate", "Hawkes"),
       col=c("black", "blue"), pch=c(1,4))
```



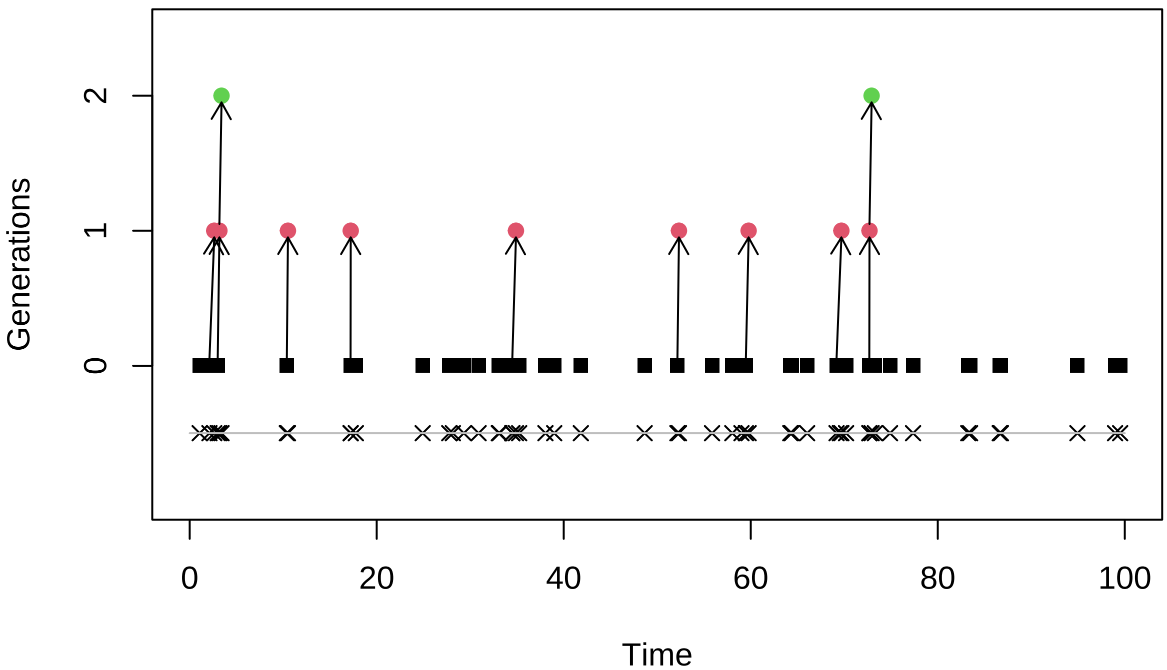
Simulations from both models

An example of simulations based on the two models - 100 years here. We can see the intensity roughly doubling in the Hawkes process.

```
set.seed(777)
simRecs_const <- hawkes(100, fun = exprate, repr = 0, family = "exp", rate = 0)
#plot(simRecs_const, intensity = FALSE)
plot(simRecs_const, intensity = TRUE)
```



```
set.seed(777)
simRecs_Hawkes <- hawkes(100, fun = optMarathon$par[1], repr = optMarathon$par[2], family = "exp", rate
plot(simRecs_Hawkes, intensity = FALSE)
```




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