Applications of Log-Linear models to Public Health

Elizabeth Colantuoni, Scott Zeger

5/3/2021

I. Objectives

Upon completion of this session, you will be able to do the following:

- Describe how to utilize Poisson log-linear models as tools for prediction
- Implement Poisson log-linear models and parametric bootstrap procedures to estimate excess deaths in Puerto Rico attributable to Hurricane Maria
- Specify and interpret the coefficients for a Poisson log-linear model for discrete time-to-event data
- Understand and explain the proportional and non-proporational risk cases

II. Case Study 1

In this first case study, we are adapting an analysis developed by Dr. Scott Zeger for 140.654 AY 2017-18. The goal of the analysis is to estimate the excess mortality attributable to Hurricane Maria, Sept 20, 2017.

The data available are deaths per month from June 2010 through February 2018 within 18 strata defined by:

- Socioeconomic development: tertiles (seitert 1 = high, seitert 2 = mid, seitert 3 = low)
- Age group: < 40 (agec5 1), 40 - 64 (agec5 2), > 64 (agec5 3)
- sex: sex 1 = male, sex 2 = female

```r
# Load the data
load("./prtest_complete_sz.rdata")
#
d = prtest_complete_sz
# Create a year.month variable or order the monthly data
# Create a t variable to count months 1 to 92, separately for each strata (18)
# Create two "training" set indicators based on calendar time
d = d %>% mutate(
  year.month = year + (month-1)/12,
  t = rep(1:92,18),
  train1 = ifelse(year.month < 2017.2, TRUE, FALSE),
  train2 = ifelse(year.month < 2017.65,TRUE, FALSE)
)
d = d %>% rename(
  sei = seitert,
  age=agec5,
  pop=csi_mig
)
```

A. Displays of the data

We will display the mean number of deaths reported by month and year.

In addition, we make a figure displaying the total deaths per month per 100,000 people in the population.

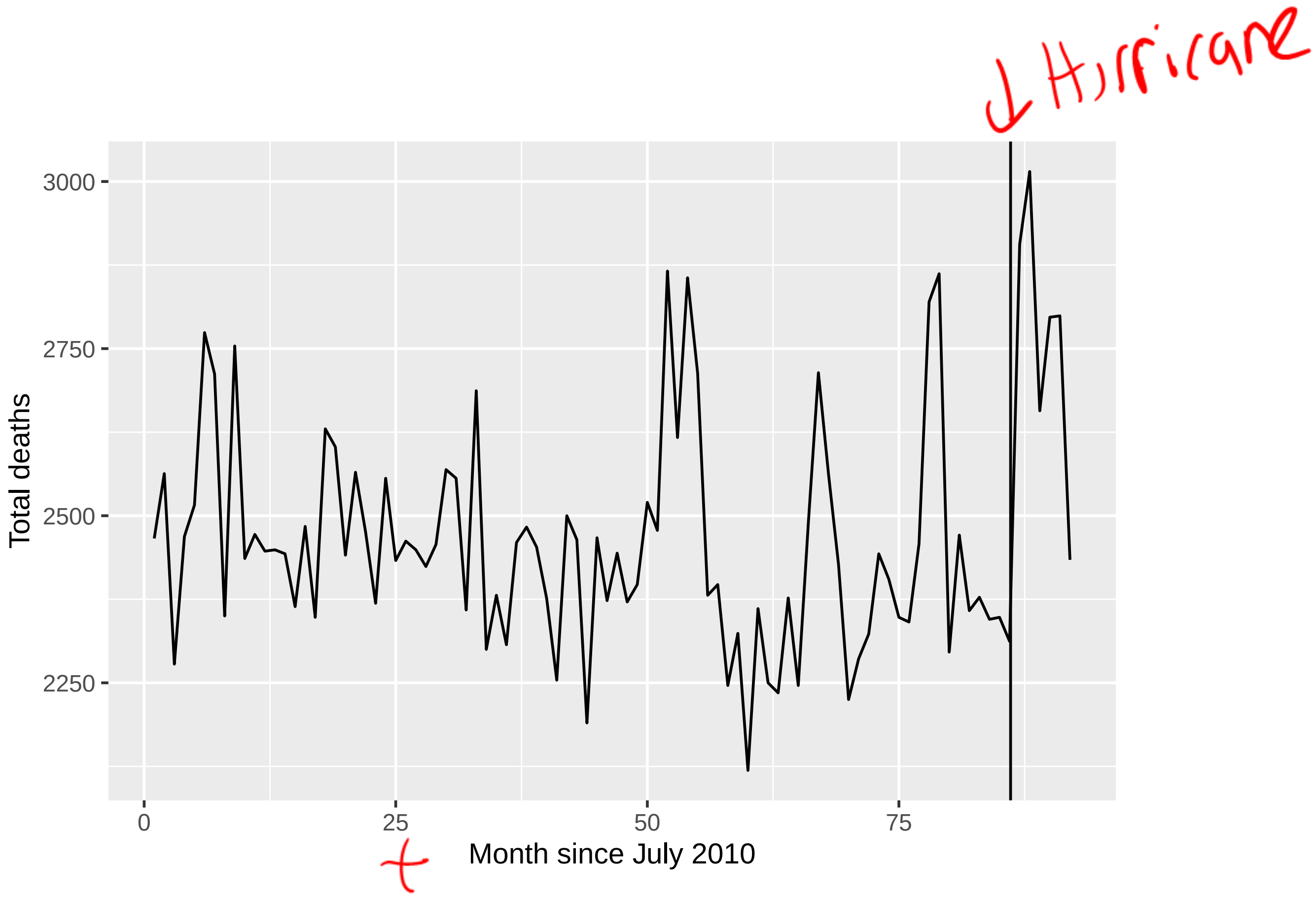
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 12 x 2
##    month mean.month
##    <int>      <dbl>
##  1     1       149.
##  2     2       132.
##  3     3       141.
##  4     4       130.
##  5     5       132.
##  6     6       131.
##  7     7       134.
##  8     8       135.
##  9     9       135.
## 10    10       141.
## 11    11       136.
## 12    12       149.
```

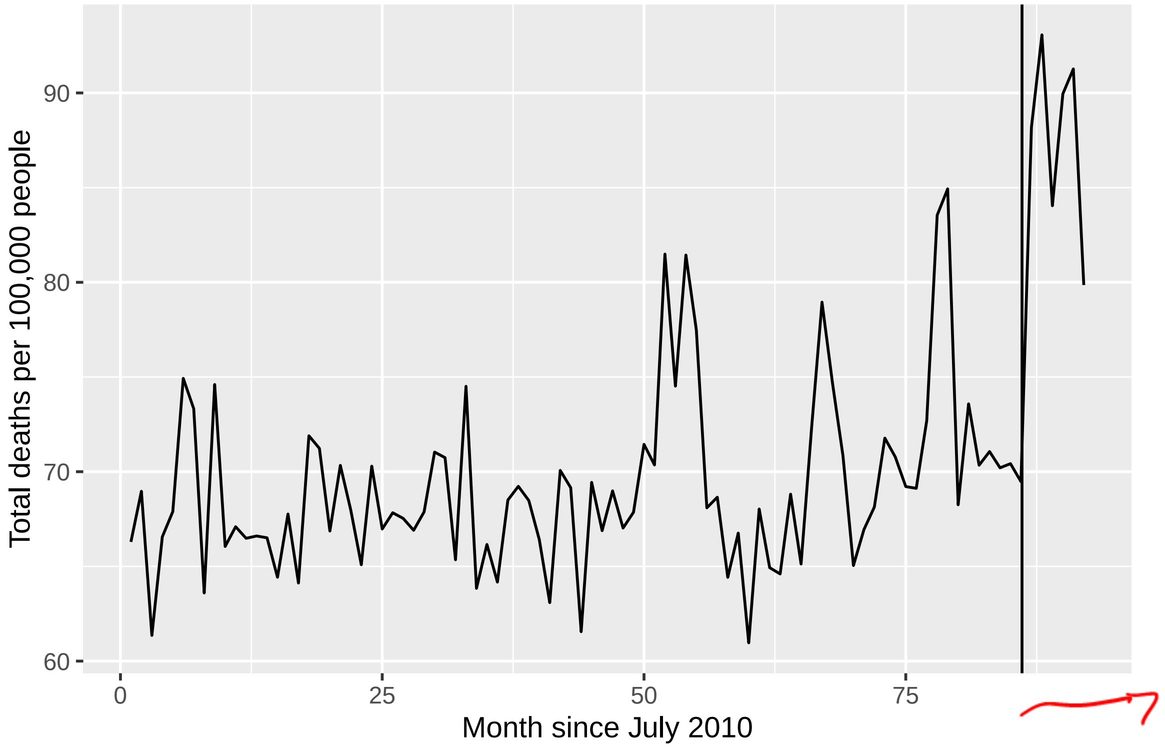
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 9 x 2
##    year mean.year
##   <int>     <dbl>
## 1  2010      140.
## 2  2011      138.
## 3  2012      138.
## 4  2013      135.
## 5  2014      139.
## 6  2015      130.
## 7  2016      136.
## 8  2017      142.
## 9  2018      145.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



```
## `summarise()` ungrouping output (override with `.groups` argument)
```



B. Model specification, fitting and summary

We fit several overdispersed log-linear (Poisson) regression models to the first 81 months (train1, July 2010 to Feb 2017) or 86 months (train2: July 2010 to August 2017).

Each model includes an offset for population size at each month that includes migration.

We will build up the model sequentially by adding components:

1. season: a (sin, cos) pair of predictors at the annual frequencies (2 degrees of freedom)
2. season; a (sin, cos) pair of predictors at the semi-annual frequencies (2 degrees of freedom)
3. trend: a natural spline of time (2 degrees of freedom)
4. stratum: an (age, sex, and SES) stratum-specific intercepts (log rates of mortality; 17 degrees of freedom);

A fifth model will be considered that interacts the long run time trend and the seasonal effects.

The parameters in the model are log relative risks of death.

```
#
#  create a data frame for the training data set
#
d.train <- d %>% filter(train1==TRUE)
#
# set the boundary and internal knots for the intended
# degrees of freedom for the longer-term time trend
#
df.trend = 2
Boundary.knots.trend = c(min(d.train$t),max(d.train$t))
knots.trend = quantile(d.train$t,p=(1:(df.trend))/(df.trend+1))
#
# specify formula fj, fit model to produce mj and
# obtain predicted values in prj for j=1,..,5
#
f1 = deaths~cos(2*pi*t/12) + sin(2*pi*t/12)
m1 = glm(data=d.train,
         formula=f1,
         family=quasipoisson(),offset=log(pop),x=TRUE)
pr1=as.data.frame(predict(m1,type="response",newdata=d,se.fit=TRUE))
colnames(pr1)=c("fit1","se.fit1","scale.fit1")

f2 = deaths~cos(2*pi*t/12) + sin(2*pi*t/12)+cos(2*pi*t/6) + sin(2*pi*t/6)
m2 = glm(data=d.train,
         formula=f2,
         family=quasipoisson(),offset=log(pop),x=TRUE)
pr2=as.data.frame(predict(m2,type="response",newdata=d,se.fit=TRUE))
colnames(pr2)=c("fit2","se.fit2","scale.fit2")

f3 = deaths~ns(t,Boundary.knots=Boundary.knots.trend,knots = knots.trend)+
  cos(2*pi*t/12) + sin(2*pi*t/12)+cos(2*pi*t/6) + sin(2*pi*t/6)
m3 = glm(data=d.train,
         formula=f3,
         family=quasipoisson(),offset=log(pop),x=TRUE)
pr3=as.data.frame(predict(m3,type="response",newdata=d,se.fit=TRUE))
colnames(pr3)=c("fit3","se.fit3","scale.fit3")
```

```r
f4 = deaths~factor(age)*factor(sex)*factor(sei) +
ns(t,Boundary.knots=Boundary.knots.trend,knots = knots.trend)+
  cos(2*pi*t/12) + sin(2*pi*t/12)+cos(2*pi*t/6) + sin(2*pi*t/6)
m4 = glm(data=d.train,
         formula=f4,
         family=quasipoisson(),offset=log(pop),x=TRUE)
pr4=as.data.frame(predict(m4,type="response",newdata=d,se.fit=TRUE))
colnames(pr4)=c("fit4","se.fit4","scale.fit4")

f5 = deaths~factor(age)*factor(sex)*factor(sei) +
ns(t,Boundary.knots=Boundary.knots.trend,knots = knots.trend)*
  (cos(2*pi*t/12) + sin(2*pi*t/12)+cos(2*pi*t/6) + sin(2*pi*t/6))
m5 = glm(data=d.train,
         formula=f5,
         family=quasipoisson(),offset=log(pop),x=TRUE)
pr5=as.data.frame(predict(m5,type="response",newdata=d,se.fit=TRUE))
colnames(pr5)=c("fit5","se.fit5","scale.fit5")
#
# obtain summary statistics
# (deviance, residual.df, model.df,
# over-dispersion estimates (Mean squared Pearson residuals))
# for each of the models
#
d.pr = cbind(d,pr1,pr2,pr3,pr4,pr5)
dev.all=c(deviance(m1),deviance(m2),deviance(m3),deviance(m4),deviance(m5))
df.resid.all=c(df.residual(m1),df.residual(m2),df.residual(m3),
               df.residual(m4),df.residual(m5))
df.model.all = m1$df.null - df.resid.all
phi.all=dev.all/df.resid.all
summary.stats=data.frame(dev=dev.all,df.m=round(df.model.all,0),
                         df.r=round(df.resid.all,0),odp=phi.all)
rownames(summary.stats) = c("Model 1","Model 2","Model 3","Model 4","Model 5")
print(summary.stats)
```

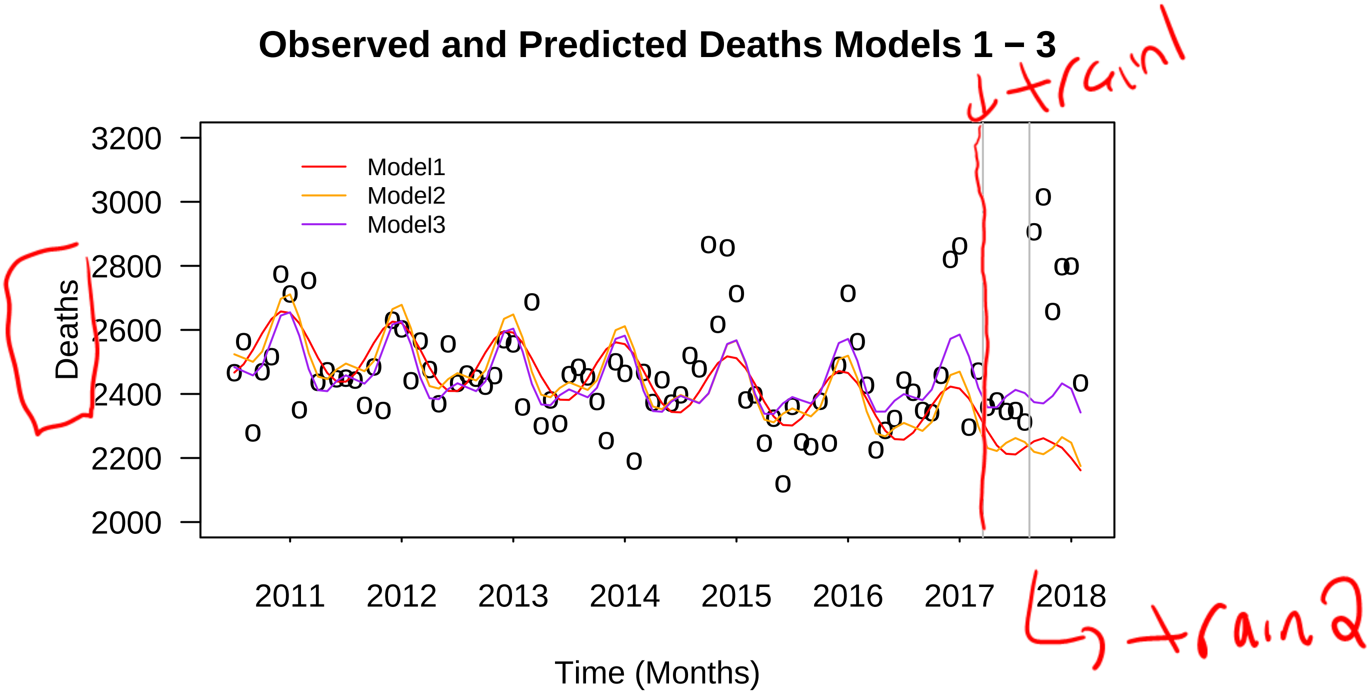
```
##                dev df.m df.r        odp
## Model 1 353722.851    2 1455 243.108488
## Model 2 353670.041    4 1453 243.406773
## Model 3 353583.991    7 1450 243.851028
## Model 4   2128.786   24 1433   1.485545
## Model 5   2010.022   36 1421   1.414512
```

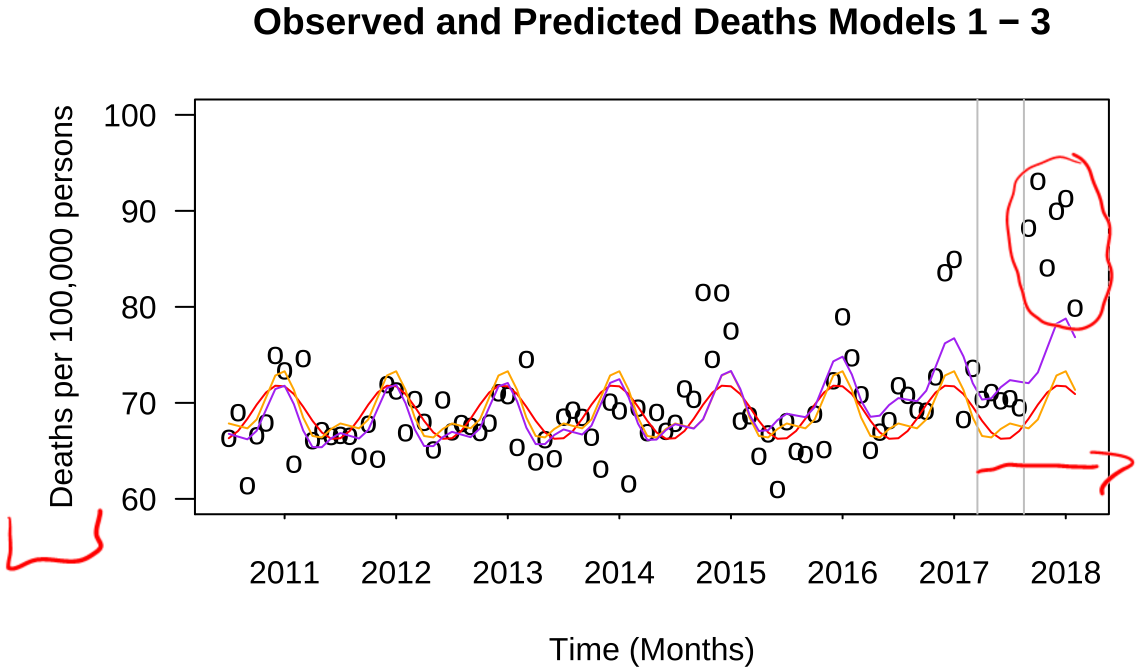
```r
#
```

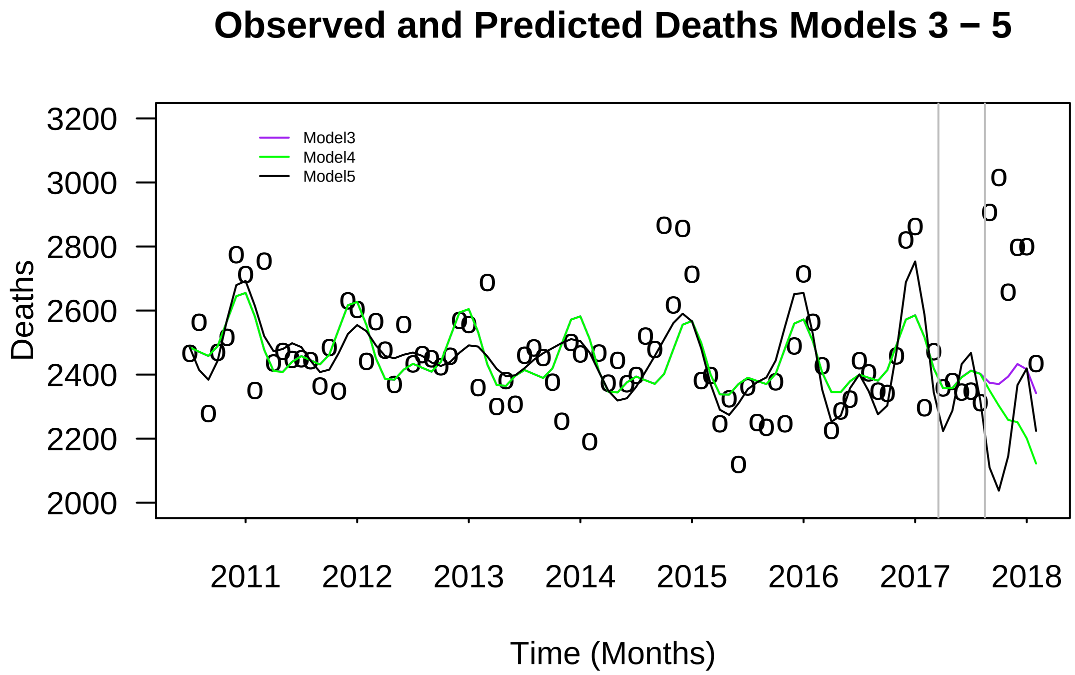
C. Model fit visualization

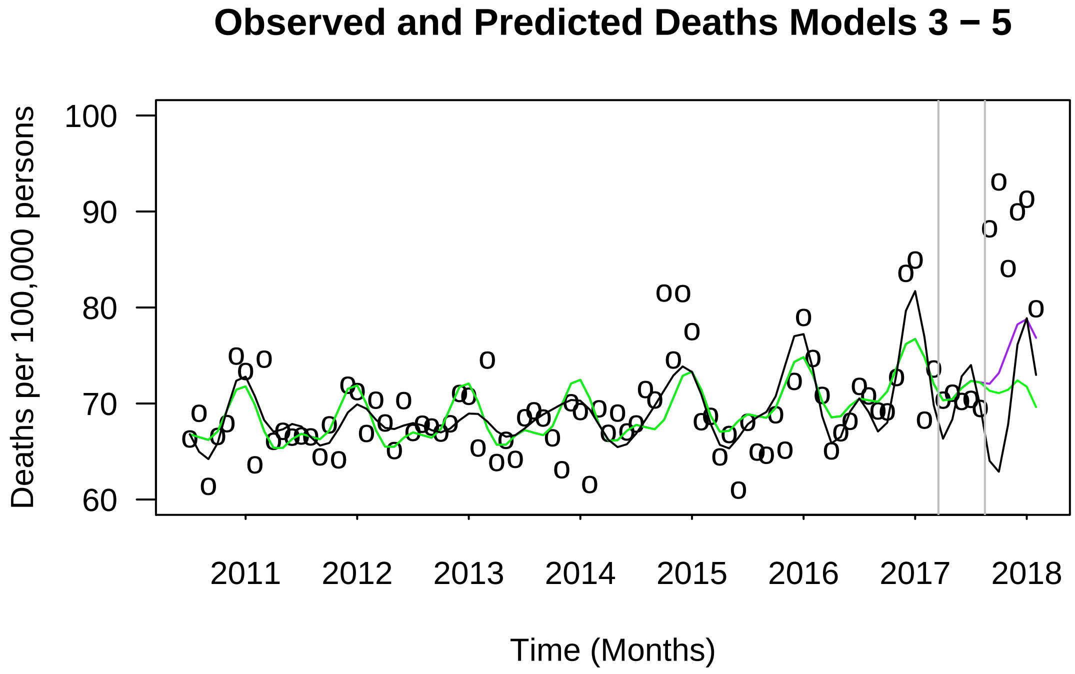
Display observed total deaths by month and predicted totals using first 81 months from Models 1-5.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```






Observed and Predicted Deaths Models 3 − 5
Model3
Model4
Model5
Deaths
3200
3000
2800
2600
2400
2200
2000
2011
2012
2013
2014
2015
2016
2017
2018
Time (Months)


Observed and Predicted Deaths Models 3 − 5
Deaths per 100,000 persons
100
90
80
70
60
2011
2012
2013
2014
2015
2016
2017
2018
Time (Months)

D. Change the duration of the training data

In the analysis above, we utilized data from July 2010 to Feb 2017 (81 months) to predict total numbers of deaths from Mar 2017 through Feb 2018.

Now, we will utilize 86 months of data (July 2010 to August 2017), i.e. the data leading up to the month when Hurricane Maria struck, to predict total number of deaths from Sept 2017 through Feb 2018.

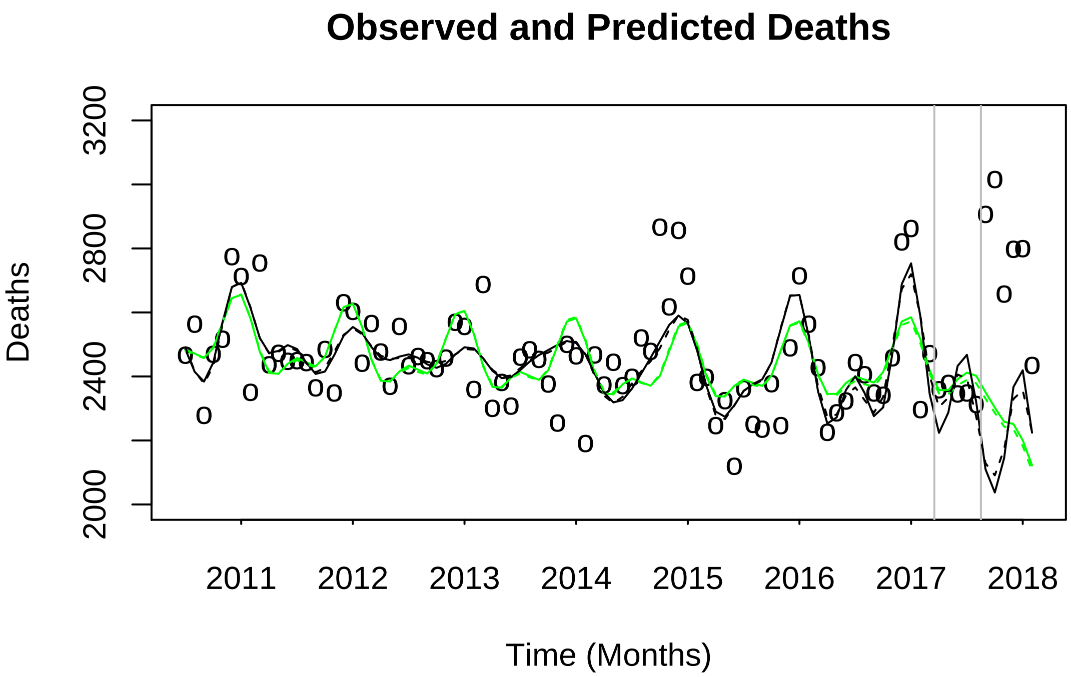
We focus on fitting on Model 4 and 5.

The table below summarizes the fit of Model 4 and 5 based on the two training sets (Train 1 and 2).

```
##                           dev df.m df.r      odp
## Train1: Model 4 2128.786   24 1433 1.485545
## Train2: Model 4 2242.335   24 1523 1.472315
## Train1: Model 5 2010.022   36 1421 1.414512
## Train2: Model 5 2130.855   36 1511 1.410228
```

The figure below compares the fitted/predicted total deaths based on the two training datasets and Models 4 and 5.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



1. Decomposition of the model results

One question you may have after reviewing the results above is: why did we see the expected number of monthly deaths decrease over time in the model that only included sine and cosine functions?

The answer is that the total population size is decreasing over time.

To illustrate this I will ignore the information about the strata (seitert, agec and sex variables) for now, since Models 1, 2, and 3 do not depend on these variables.

Let Y_i be the reported number of deaths in Puerto Rico for month i, $i = 1, ..., 81$ (the duration of follow-up in the first training data set).

Then recall that our model is:

$$Log(E(Y_i|sine_i, cosine_i)) = Log(N_i\lambda_i) = Log(N_i) + \beta_0 + \beta_1 sine(2 \times \pi \times i/12) + \beta_2 cosine(2 \times \pi \times i/12)$$

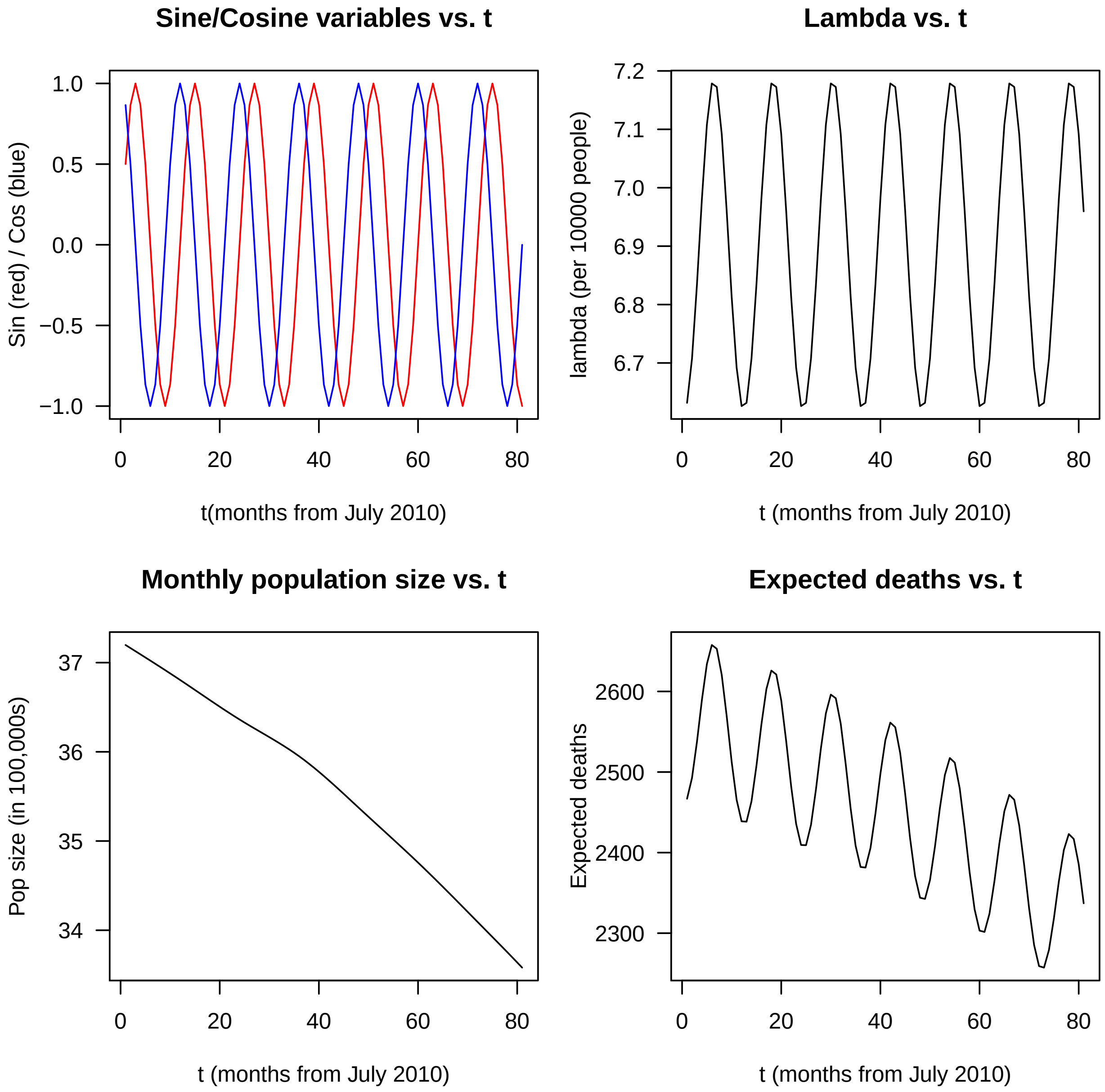
So the components that go into predicting/estimating the monthly deaths include the estimate of λ_i and N_i.

In the code below, I do the following:

1. Fit the model above and print the estimates of β_0, β_1 and β_2.
2. Make a 4 panel figure that displays:

- Values of the sine and cosine variables as a function of t (calendar month, expressed as number of months since July 2010). Here you will see the sine and cosine variables with annual frequency.
- Estimates of λ_i as a function of t. For plotting purposes, instead of plotting λ_i = risk of death per person, I plotted the expected deaths per 10,000 persons. Here you will see that the estimates of λ_i are the same for each month (e.g. January) regardless of year. This is what we expect from the model fit.
- The observed population size (N_i) as a function of t. NOTE: you see that the population size is decreasing over t.
- The estimated total number of deaths given the population size observed in each month! Even though λ_i is fixed, the same for a given month (e.g. λ_i is the same for all Januaries, etc). Since the population size is decreasing, the expected number of deaths decreases as a function of t, i.e. the final figure is plotting $N_i\lambda_i$.

```
##
## Call:
## glm(formula = f1, family = quasipoisson(), data = d.train, offset = log(pop),
##     x = TRUE)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -24.163  -13.376   -4.363   16.124   32.648
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -7.279280   0.041025 -177.435   <2e-16 ***
## cos         -0.040037   0.058356   -0.686    0.493
## sin         -0.009054   0.057567   -0.157    0.875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 333.2951)
##
##     Null deviance: 353890  on 1457  degrees of freedom
## Residual deviance: 353723  on 1455  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```



E. Estimate Hurricane Maria effect

To estimate the hurricane effect, we calculate the difference between the observed deaths and the predicted (estimated mean) deaths for the six months from Sept, 2017 thru Feb, 2018.

This difference is a non-linear function of the models' regression coefficients. The approach is therefore to use **parametric bootstrapping** to estimate the joint mean and covariance matrix of the estimated means for the 6 months and for various linear combinations of them (totals for first 2, 4 and 6 months).

Here is the process:

1. We approximate the joint distribution of the regression coefficients by a Gaussian distribution with mean equal to the maximum likelihood estimate and variance equal to its asymptotic covariance matrix, including the over-dispersion.

2. We take a draw/simulation from this multivariate Gaussian distribution
3. We sum the exponentials of the simulated log linear predictors to get the expected total deaths absent the hurricane.
4. Repeat Steps 2 and 3 many times to obtain a distribution for the expected total deaths absent the hurricane and compute confidence intervals from the empirical distribution of the simulated values.

The confidence intervals for a given model condition on it being the "correct model" (which of course does not exist since a model is just a tool to predict the counterfactual of the mortality absent Hurricane Maria). By comparing across the 2 models, we get a sense of the importance of the choice of model in influencing the model-specific causal estimates.

```
# Create the dataset we need for prediction;
# i.e. take the data for the time post hurricane

d.6 = d %>% filter(train2==FALSE)

# Create the model matrices we need for prediction
# for Models 4 and 5
m4.matrix= model.matrix(f4,data=d)
m4.matrix.6 = m4.matrix[d$train2==FALSE,]
m5.matrix= model.matrix(f5,data=d)
m5.matrix.6 = m5.matrix[d$train2==FALSE,]

# Name the coefficients and var/cov of coefficients
# for Models 4 and 5
mean4=m42$coefficients
var4=vcov(m42)
mean5=m52$coefficients
var5=vcov(m52)
#
#
#
# generate B simulated values from the joint distribution
# of the 6 predicted total deaths from each model
#
set.seed(09202017)
B=500
#
# Generate multivariate Gaussian samples for Models 4 and 5
#
bs.coefs4 = mvrnorm(B, mean4, var4)
bs.coefs5 = mvrnorm(B, mean5, var5)
#
results = NULL
#
# Run B bootstrap replications
#
for ( b in 1:B) {
  #
  # generate the linear predictors from
  # Models 4 and 5 for the 6 month period during the Hurricane
  #
  bs.lp4 = m4.matrix.6 %*% t(bs.coefs4)[,b]
  bs.lp5 = m5.matrix.6 %*% t(bs.coefs5)[,b]
  #
  # Create the predicted numbers of deaths from the linear predictors
  #
  d.6$pr4 = exp(bs.lp4)*d.6$pop
  d.6$pr5 = exp(bs.lp5)*d.6$pop
  d.6$b = b
  #
  temp = d.6[,c("b","deaths","pop","pr4","pr5","sex","sei","age","t")]
  if(b==1) {results=temp}
  else {results = bind_rows(results,temp) }
}
#
# calculate 6, 4, and 2 month totals of predicted values
#
results.s6 = results %>% group_by(b,sei,sex,age) %>%
  summarise(t = mean(t)+6, deaths=sum(deaths), pop= mean(pop), pr4= sum(pr4), pr5= sum(pr5))
```

```
## `summarise()` regrouping output by 'b', 'sei', 'sex' (override with `.groups` argument)
```

```
results.s4 = results %>% filter(.,t<91) %>%
  group_by(b, sei, sex, age) %>%
  summarise(t = mean(t)+6, deaths=sum(deaths), pop = mean(pop),pr4= sum(pr4),pr5= sum(pr5))
```

```
## `summarise()` regrouping output by 'b', 'sei', 'sex' (override with `.groups` argument)
```

```
results.s2 = results %>% filter(.,t<89) %>%
  group_by(b, sei, sex, age) %>%
  summarise(t= mean(t)+6, deaths=sum(deaths), pop = mean(pop),pr4 = sum(pr4),pr5= sum(pr5))
```

```
## `summarise()` regrouping output by 'b', 'sei', 'sex' (override with `.groups` argument)
```

```
results.all = bind_rows(results,results.s6,results.s4,results.s2)
#
# calculate for each model:
# (1) ratio of the observed to expected numbers of deaths
#     expressed as percentage above or below expected;
# (2) difference between observed and expected
#
results.ext = results.all %>% mutate(
  rr4=100*(deaths/pr4-1),rr5=100*(deaths/pr5-1),
  diff4 = deaths - pr4,diff5 = deaths - pr5
)

#  create table1 summary without stratum
#
results.tot = results.ext %>% group_by(t,b) %>% summarise(deaths= sum(deaths), pop= sum(pop),
              pr4 = sum(pr4), pr5 = sum(pr5),
              rr4 = mean(rr4), rr5 = mean(rr5),
              diff4 = sum(diff4), diff5 = sum(diff5)
        )
```

```
## `summarise()` regrouping output by 't' (override with `.groups` argument)
```

```
#
table1 = results.tot %>% group_by(t) %>% summarise(
          deaths = mean(deaths), pop = mean(pop),
          pr4 = mean(pr4), mean.diff4 = mean(diff4),
          cil.diff4 = quantile(diff4,0.025),
          ciu.diff4 = quantile(diff4,p=0.975),
          mean.rr4 = mean(rr4),
          cil.rr4 = quantile(rr4,p=0.025),
          ciu.rr4 = quantile(rr4,p=0.975),
          pr5 = mean(pr5), mean.diff5 = mean(diff5),
          cil.diff5 = quantile(diff5,0.025),
          ciu.diff5 = quantile(diff5,p=0.975),
          mean.rr5 = mean(rr5),
          cil.rr5 = quantile(rr5,p=0.025),
          ciu.rr5 = quantile(rr5,p=0.975)
          )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
table1 = table1[,-1]
#
colnames(table1)= c("Deaths", "Population",
                    "Predicted Deaths-Model 4", "Excess Deaths-Model 4",
                    "CI-Lower","CI-Upper","%Change-Model 4",
                    "CI-Lower","CI-Upper",
                    "Predicted Deaths-Model 5", "Excess Deaths-Model 5",
                    "CI-Lower","CI-Upper","%Change-Model 5",
                    "CI-Lower","CI-Upper")
rownames(table1)=c(month.abb[c(9:12,1:2)],"Total:2","Total:4","Total:6")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
table1 = t(round(table1,0))
#
write.csv(d.pr,file=paste("./predicted.migrate.total",".csv",sep=""))
#
write.csv(table1,file=paste("./results.total",".csv",sep=""))
#
#  results by sei
#
#  create table1 summary with sei stratum
#
results2.tot = results.ext %>% group_by(sei,t,b) %>%
  summarise(deaths= sum(deaths), pop= sum(pop), pr4 = sum(pr4), pr5 = sum(pr5),
          rr4 = mean(rr4), rr5 = mean(rr5),
          diff4 = sum(diff4), diff5 = sum(diff5)
        )
```

```
## `summarise()` regrouping output by 'sei', 't' (override with `.groups` argument)
```

```
#
table2 = results2.tot %>% group_by(sei,t) %>% summarise(
          deaths = mean(deaths), pop = mean(pop),
          pr4 = mean(pr4), mean.diff4 = mean(diff4),
          cil.diff4 = quantile(diff4,0.025),
          ciu.diff4 = quantile(diff4,p=0.975),
          mean.rr4 = mean(rr4),
          cil.rr4 = quantile(rr4,p=0.025),
          ciu.rr4 = quantile(rr4,p=0.975),
          pr5 = mean(pr5), mean.diff5 = mean(diff5),
          cil.diff5 = quantile(diff5,0.025),
          ciu.diff5 = quantile(diff5,p=0.975),
          mean.rr5 = mean(rr5),
          cil.rr5 = quantile(rr5,p=0.025),
          ciu.rr5 = quantile(rr5,p=0.975)
          )
```

```
## `summarise()` regrouping output by 'sei' (override with `.groups` argument)
```

```
table2[,-2] = round(table2[,-2],0)
table2=t(table2)
cnames = rep(c(month.abb[c(9:12,1:2)],"Total:2","Total:4","Total:6"),
             length(unique(d.6$sei)))

table2 = as.data.frame(table2,row.names =
                         c("SEI", "Time","Obs Deaths", "Population",
                           "Deaths-Model 4", "Excess Deaths-Model 4",
                           "ED.CIL-4","ED.CIU-4","%Change-Model 4",
                           "%C.CIL-4","%C.CIU-4",
                           "Deaths-Model 5", "Excess Deaths-Model 5",
                           "ED.CIL-5","ED.CIU-5","%Change-Model 5",
                           "%C.CIL-5","%C.CIU-5")
)
colnames(table2) = cnames
#
write.csv(table2,file=paste("./results.sei.strata",".csv",sep=""))
#
#
#  create table1 summary with age_sex strata
#
results3.tot = results.ext %>% group_by(age,sex,t,b) %>%
  summarise(deaths= sum(deaths), pop= sum(pop),
            pr4 = sum(pr4), pr5 = sum(pr5), rr4 = mean(rr4),
            rr5 = mean(rr5), diff4 = sum(diff4), diff5 = sum(diff5)
        )
```

```
## `summarise()` regrouping output by 'age', 'sex', 't' (override with `.groups` argument)
```

```
#
table3 = results3.tot %>% group_by(age,sex,t) %>%
  summarise(
          deaths = mean(deaths), pop = mean(pop),
          pr4 = mean(pr4), mean.diff4 = mean(diff4),
          cil.diff4 = quantile(diff4,0.025),
          ciu.diff4 = quantile(diff4,p=0.975),
          mean.rr4 = mean(rr4),
          cil.rr4 = quantile(rr4,p=0.025),
          ciu.rr4 = quantile(rr4,p=0.975),
          pr5 = mean(pr5), mean.diff5 = mean(diff5),
          cil.diff5 = quantile(diff5,0.025),
          ciu.diff5 = quantile(diff5,p=0.975),
          mean.rr5 = mean(rr5),
          cil.rr5 = quantile(rr5,p=0.025),
          ciu.rr5 = quantile(rr5,p=0.975)
          )
```

```
## `summarise()` regrouping output by 'age', 'sex' (override with `.groups` argument)
```

```
table3[,-3] = round(table3[,-3],0)
table3=t(table3)
cnames = rep(c(month.abb[c(9:12,1:2)],
               "Total:2","Total:4","Total:6"),
             length(unique(d.6$age_sex)))

table3 = as.data.frame(table3,
                       row.names = c("Age","Sex","Time",
                       "Obs Deaths", "Population",
                       "Deaths-Model 4",
                       "Excess Deaths-Model 4",
                       "ED.CIL-4","ED.CIU-4",
                       "%Change-Model 4","%C.CIL-4",
                       "%C.CIU-4",
                       "Deaths-Model 5",
                       "Excess Deaths-Model 5",
                       "ED.CIL-5","ED.CIU-5",
                       "%Change-Model 5","%C.CIL-5","%C.CIU-5")
)
colnames(table3) = cnames
#
write.csv(table3,file=paste("./results.age_sex.strata",".csv",sep=""))
```

Examine the results

```
table1
```

```
##                             [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
## Deaths                      2906    3015    2657    2797    2799    2434
## Population               3295345 3239635 3161423 3109546 3066826 3047975
## Predicted Deaths-Model 4    2333    2289    2245    2237    2185    2107
## Excess Deaths-Model 4        573     726     412     560     614     327
## CI-Lower                     519     670     352     495     552     268
## CI-Upper                     626     783     473     622     672     383
## %Change-Model 4               15      25      16      24      21       9
## CI-Lower                      12      21      13      20      18       6
## CI-Upper                      18      28      20      27      24      12
## Predicted Deaths-Model 5    2133    2097    2177    2330    2355    2226
## Excess Deaths-Model 5        773     918     480     467     444     208
## CI-Lower                     678     806     360     332     296      84
## CI-Upper                     874    1038     600     601     572     341
## %Change-Model 5               26      36      20      19      12       3
## CI-Lower                      20      29      14      12       6      -2
## CI-Upper                      32      44      27      26      19      10
##                             [,7]    [,8]    [,9]
## Deaths                      5921   11375   16608
## Population               3267490 3201487 3153458
## Predicted Deaths-Model 4    4622    9105   13397
## Excess Deaths-Model 4       1299    2270    3211
## CI-Lower                    1185    2045    2865
## CI-Upper                    1406    2488    3527
## %Change-Model 4               20      20      18
## CI-Lower                      17      17      15
## CI-Upper                      23      23      21
## Predicted Deaths-Model 5    4230    8738   13319
## Excess Deaths-Model 5       1691    2637    3289
## CI-Lower                    1489    2291    2837
## CI-Upper                    1897    3005    3740
## %Change-Model 5               31      25      19
## CI-Lower                      25      20      15
## CI-Upper                      38      30      23
```

Examine the results by Socioeconomic development level

```
table2[,c(9,18,27)]
```

```
##                         Total:6 Total:6.1 Total:6.2
## SEI                         1.0       2.0       3.0
## Time                       95.5      95.5      95.5
## Obs Deaths               2983.0    4940.0    8685.0
## Population             561133.0  974456.0 1617869.0
## Deaths-Model 4           2098.0    4041.0    7258.0
## Excess Deaths-Model 4     885.0     899.0    1427.0
## ED.CIL-4                  825.0     794.0    1238.0
## ED.CIU-4                  939.0     999.0    1605.0
## %Change-Model 4            28.0      14.0      12.0
## %C.CIL-4                   24.0      11.0       9.0
## %C.CIU-4                   31.0      17.0      15.0
## Deaths-Model 5           2075.0    4014.0    7230.0
## Excess Deaths-Model 5     908.0     926.0    1455.0
## ED.CIL-5                  837.0     787.0    1207.0
## ED.CIU-5                  982.0    1061.0    1694.0
## %Change-Model 5            29.0      15.0      13.0
## %C.CIL-5                   24.0      11.0       9.0
## %C.CIU-5                   33.0      19.0      17.0
```

III. Case study 2

Here we will introduce some key concepts in survival analysis and demonstrate the log-linear models can be used to analyze survival data.

A survival outcome describes when an event of interest occurs, i.e. a cancer patient may experience a recurrence of cancer 6 months after surgery to remove an initial tumor. The data we get to observe for a patient is: the cancer reoccurred AND that happened at 6 months.

A. Data

The data contains information about *time to death* for inpatients hospitalized for a severe mental disorder. Survival time from hospitalization is in years.

In most studies measuring survival time of patients, we don't get to follow patients long enough to see the when the event occurs for all patients.

Patients for whom we can not follow long enough are "censored". Censoring can occur for several reasons: the study period is over; i.e. administrative censoring (you only had the budget to follow persons for so long) or because the patient drops-out of the study. Another type of censoring can be that the patient experiences another event (e.g. death) that precludes you from being able to observe other events of interest.

In the data, "censor" is 1 if censored; 0 if the patient died; "age" of hospitalization for mental disorder is in years; "male" is 1 for males and 0 for females.

One question is whether survival is different for men than for women with and without control for age. A listing of the data is:

```
d = read.table("./survival.csv",sep=",",header=T)
d$event = 1 - d$censor
d
```

```
##    survive censor age male event
## 1        1      0  58    0     1
## 2        1      0  51    0     1
## 3        2      0  55    0     1
## 4       11      0  48    0     1
## 5       14      0  47    0     1
## 6       22      0  28    0     1
## 7       24      0  45    0     1
## 8       26      0  43    0     1
## 9       31      1  31    0     0
## 10      32      0  25    0     1
## 11      35      1  35    0     0
## 12      35      1  33    0     0
## 13      36      1  25    0     0
## 14      37      1  30    0     0
## 15      40      0  36    0     1
## 16      22      0  41    1     1
## 17      25      0  36    1     1
## 18      28      0  19    1     1
## 19      30      1  35    1     0
## 20      30      1  21    1     0
## 21      31      1  30    1     0
## 22      33      1  25    1     0
## 23      33      1  24    1     0
## 24      34      1  29    1     0
## 25      35      0  32    1     1
## 26      39      1  32    1     0
```

B. Binned survival data

To analyze the survival data using a log-linear model, the first step is to create "bins" or intervals of time and to determine the person-years and number of deaths in each interval, separately for men and for women or for other strata.

We focus only on sex to illustrate the application of log-linear models.

We will bin the data by 10-year increments.

```r
library(survival)
library(dplyr)
library(biostat3)

## Cut the survival data into bins of 10 years
Cutoff <- tcut(rep(0, length(d$survive)),
               breaks=c(-1,10,20,30,40),
               labels=c("0-10","11-20","21-30","31-40"))

# We provided argument scale=1 to report patient-years
py <- pyears(Surv(survive, event) ~ Cutoff + male, data = d,
             scale = 1,data.frame = TRUE)

# Using the binned data, create the event rate per bin
binned = py$data
binned$rate = round(binned$event/binned$pyears,3)
```

```
binned$midp = c(5,15,25,35,5,15,25,35)

# Display the data
binned
```

```
##   Cutoff male pyears  n event  rate midp
## 1   0-10    0    124 15     3 0.024    5
## 2  11-20    0    105 12     2 0.019   15
## 3  21-30    0     82 10     3 0.037   25
## 4  31-40    0     36  7     2 0.056   35
## 5   0-10    1    110 11     0 0.000    5
## 6  11-20    1    110 11     0 0.000   15
## 7  21-30    1     95 11     3 0.032   25
## 8  31-40    1     25  6     1 0.040   35
```

C. Model definition

Now, we assume a model for the incidence rate or "hazard" of an event in each interval.

The incidence is the risk per unit time of the event occurring among those that enter the interval.

The term hazard is usually reserved for the limit of the incidence rate as the interval width goes to zero. We can get a crude estimate by the number of events in the interval divided by the person-time experienced in the interval. For example, we estimate the incidence rate to be $1/25 = 0.04$ events per year for 31-40 year old men.

But this is a crude estimate based upon few events. We want to smooth these rates using a log-linear model.

We assume the incidence rate λ_i satisfies a log-linear regression

$$\lambda_i = exp(X_i^{\prime}\beta)$$

In this example, we will consider two X variables: time represented by the mid-point of each interval and sex.

The number of events in an interval is assumed to be a Poisson variable since this count is a sum of independent random events assuming each person lives and dies independently of the others (unless of course they are in the same hospital cared for by a gruesome nurse who is systematically "doing-in" patients, but more about that later). The expected number of events in an interval is the rate of events multiplied by the person-time for which this rate is experienced. Hence, we have

$$E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + X_i^{\prime}\beta)$$

Here, the term $log(PT_i)$ is called an "offset" because it is added to the linear predictor without needing a regression coefficient. You can think of an offset as a predictor variable whose coeficient is known to be 1.

1. Model A

First, we will estimate the overall rate of death; i.e. fit a log-linear model with only an intercept.

$$\text{Model A: } E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + \beta_0)$$

```
fitA = glm(event~1,offset=log(pyears),data=binned,family="poisson")
summary(fitA)
```

```
##
## Call:
## glm(formula = event ~ 1, family = "poisson", data = binned, offset = log(pyears))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1174  -0.6018   0.4477   0.7612   1.2161
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.8933     0.2673  -14.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 12.259  on 7  degrees of freedom
## Residual deviance: 12.259  on 7  degrees of freedom
## AIC: 30.462
##
## Number of Fisher Scoring iterations: 5
```

```
fitA$fitted[1]
```

```
##        1
## 2.526929
```

```
lincom(fitA,"(Intercept)",eform=TRUE)
```

```
##               Estimate     2.5 %     97.5 %    Chisq   Pr(>Chisq)
## (Intercept) 0.02037846 0.0120692 0.03440838 212.2069 4.533119e-48
```

The average squared Pearson residual is 1.26 rather than 1.0. This suggests some over-dispersion. Where would over-dispersion arise from in this model?

To inflate the standard errors, we can use the quasi-poisson family that assumes the variance of each response is proportional to the mean, rather than equal to it. The proportionality constant is the mean squared Pearson residual, phihat. This increases all of the standard errors by sqrt(phihat).

```
fitAq = glm(event~1,offset=log(pyears),data=binned,family="quasipoisson")
summary(fitAq)
```

```
##
## Call:
## glm(formula = event ~ 1, family = "quasipoisson", data = binned,
##     offset = log(pyears))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1174  -0.6018   0.4477   0.7612   1.2161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.893      0.301  -12.93 3.84e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.26872)
##
##     Null deviance: 12.259  on 7  degrees of freedom
## Residual deviance: 12.259  on 7  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
lincom(fitAq,"(Intercept)",eform=TRUE)
```

```
##               Estimate      2.5 %     97.5 %    Chisq   Pr(>Chisq)
## (Intercept) 0.02037846 0.01129611 0.03676323 167.2606 2.934282e-38
```

What about the *fitted.values* from the model?

We know that to get the expected deaths per bin of time, we would compute:

$$exp(\hat{\beta}_0)PT_i$$

This is what is being computed in the *fitted.values*.

Confirmation below.

```
binned$expected = exp(fitAq$coeff[1])*binned$pyears
cbind(binned,fitAq$fitted.values)
```

```
##   Cutoff male pyears  n event  rate midp  expected fitAq$fitted.values
## 1   0-10    0    124 15     3 0.024    5 2.5269287           2.5269287
## 2  11-20    0    105 12     2 0.019   15 2.1397380           2.1397380
## 3  21-30    0     82 10     3 0.037   25 1.6710335           1.6710335
## 4  31-40    0     36  7     2 0.056   35 0.7336245           0.7336245
## 5   0-10    1    110 11     0 0.000    5 2.2416303           2.2416303
## 6  11-20    1    110 11     0 0.000   15 2.2416303           2.2416303
## 7  21-30    1     95 11     3 0.032   25 1.9359534           1.9359534
## 8  31-40    1     25  6     1 0.040   35 0.5094614           0.5094614
```

2. Model B

In the next model, we estimate the relative risk of death comparing men to women.

$$\text{Model B: } E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + \beta_0 + \beta_1 male_i)$$

```
fitB = glm(event~1+male,offset=log(pyears),data=binned,family="quasipoisson")
summary(fitB)
```

```
##
## Call:
## glm(formula = event ~ 1 + male, family = "quasipoisson", data = binned,
##     offset = log(pyears))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.60880  -0.87391   0.04272   0.88211   1.46956
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.5467     0.3864  -9.180 9.42e-05 ***
## male         -0.8959     0.7228  -1.239    0.261
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.492863)
##
##     Null deviance: 12.2590  on 7  degrees of freedom
## Residual deviance:  9.7233  on 6  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
fitB$fitted[binned$male==1][1]
```

```
##        5
## 1.294118
```

```
fitB$fitted[binned$male==0][1]
```

```
##        1
## 3.573487
```

```
lincom(fitB,c("(Intercept)","(Intercept)+male","male"),eform=TRUE)
```

```
##                    Estimate   2.5 %       97.5 %     Chisq    Pr(>Chisq)
## (Intercept)        0.02881844 0.01351409 0.06145458 84.26334 4.330714e-20
## (Intercept)+male 0.01176471 0.003552796 0.03895757 52.88403 3.538347e-13
## male               0.4082353  0.09899778 1.683432   1.536181 0.2151872
```

We estimate that the risk for a man is 0.41 as great as for a woman (that is 59% less). However, the analysis also shows that the uncertainty in this relative rate is substantial. In fact, the hypothesis that the risk is the same for men and women is reasonably consistent with the observations.

3. Proportional hazards models

A priori, we would expect the hazard of death to depend on how long one has been in the hospital since we do not live forever. Models C and D estimate the relative risk of death for men as compared to women, controlling for a time-varying baseline hazard.

NOTE: In survival analysis, we refer to the "baseline hazard" as the hazard function when setting exposure variables $X_i = 0$.

$$E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + f(time_i) + X_i^{'}\beta)$$

In the above, when we set $X_i = 0$ then we are describing the "baseline hazard" which is some function of $time_i$.

We will consider two models for the "baseline hazard": a linear function of the midpoint of each time interval and a step function representing each time interval.

$$\text{Model C: } E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + \beta_0 + \beta_1 midp_i + \beta_2 male_i)$$

Model D: $E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + \beta_0 + \beta_1 I(midp_i = 15) + \beta_2 I(midp_i = 25) + \beta_3 I(midp_i = 35) + \beta_4 male_i)$

NOTE: Both of these models are examples of proportional hazards models because we assume that the risk for a man equals the risk for a women times a constant that is the same at all periods. Said another way, the ratio of risks for men versus women is constant over time. The two models C and D differ in how they control for period.

```
fitC = glm(event~1+male+midp,offset=log(pyears),data=binned,family="quasipoisson")
summary(fitC)
```

```
##
## Call:
## glm(formula = event ~ 1 + male + midp, family = "quasipoisson",
##     data = binned, offset = log(pyears))
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
##  0.8931  -0.3142  -0.1774  -0.3262  -1.1181  -1.4641   0.9973   0.3164
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.55525    0.60351  -7.548 0.000647 ***
## male        -0.88461    0.54687  -1.618 0.166674
## midp         0.05391    0.02444   2.206 0.078504 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.8538321)
##
##     Null deviance: 12.2590  on 7  degrees of freedom
## Residual deviance:  5.5227  on 5  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
lincom(fitC,"male",eform=TRUE)
```

```
##       Estimate    2.5 %   97.5 %    Chisq Pr(>Chisq)
## male 0.4128736 0.141358 1.205907 2.616609  0.1057502
```

```r
tapply(fitC$fitted,list(binned$male,binned$midp),mean)
```

```
##           5       15       25        35
## 0 1.7067693 2.477927 3.317869 2.4974342
## 1 0.6251193 1.071788 1.587034 0.7160587
```

```r
fitD = glm(event~1+male+as.factor(midp),offset=log(pyears),data=binned,family="quasipoisson")
summary(fitD)
```

```
##
## Call:
## glm(formula = event ~ 1 + male + as.factor(midp), family = "quasipoisson",
##     data = binned, offset = log(pyears))
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
##  0.5112   0.4772  -0.5552  -0.2247  -1.2653  -1.0965   0.7093   0.3817
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4.0321     0.6024  -6.694   0.0068 **
## male                 -0.8909     0.5975  -1.491   0.2327
## as.factor(midp)15  -0.2863     0.9187  -0.312   0.7757
## as.factor(midp)25   1.0282     0.7122   1.444   0.2445
## as.factor(midp)35   1.2965     0.8219   1.577   0.2128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.012253)
##
##     Null deviance: 12.259  on 7  degrees of freedom
## Residual deviance:  4.300  on 3  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
lincom(fitD,"male",eform=TRUE)
```

```
##       Estimate     2.5 %   97.5 %    Chisq Pr(>Chisq)
## male 0.4102721 0.1271973 1.323323 2.223383  0.1359349
```

```r
tapply(fitD$fitted,list(binned$male,binned$midp),mean)
```

```
##           5        15       25        35
## 0 2.1994924 1.3987884 4.066927 2.3347917
## 1 0.8005076 0.6012116 1.933073 0.6652083
```

Interpret the effect of “male”:

4. Non-proportional hazards

In the final model, we look for evidence that the relative rate for men as compared to women changes over the duration of follow-up, that is, we look for evidence that the proportional hazards assumption is inadequate for our data. In this final model, we choose to center the midpoint variable at 20 years duration so that the male coefficient has a more reasonable interpretation.

$$\text{Model E: } E(Y_i) = \lambda_i PT_i = exp(log(PT_i) + \beta_0 + \beta_1(midp_i - 20) + \beta_2 male_i + \beta_3(midp_i - 20)male_i)$$

```r
binned$midc = binned$midp - 20
fitE = glm(event~1+male*midc,offset=log(pyears),data=binned,family="quasipoisson")
summary(fitE)
```

```
##
## Call:
## glm(formula = event ~ 1 + male * midc, family = "quasipoisson",
##     data = binned, offset = log(pyears))
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
##  0.32725  -0.51317   0.02144   0.18269  -0.49523  -0.98551   0.93239  -0.60373
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.46957    0.24520 -14.150 0.000145 ***
## male        -1.26510    0.58774  -2.152 0.097711 .
## midc         0.02981    0.02344   1.272 0.272360
## male:midc    0.10782    0.05454   1.977 0.119211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.595734)
##
##     Null deviance: 12.2590  on 7  degrees of freedom
## Residual deviance:  2.8546  on 4  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

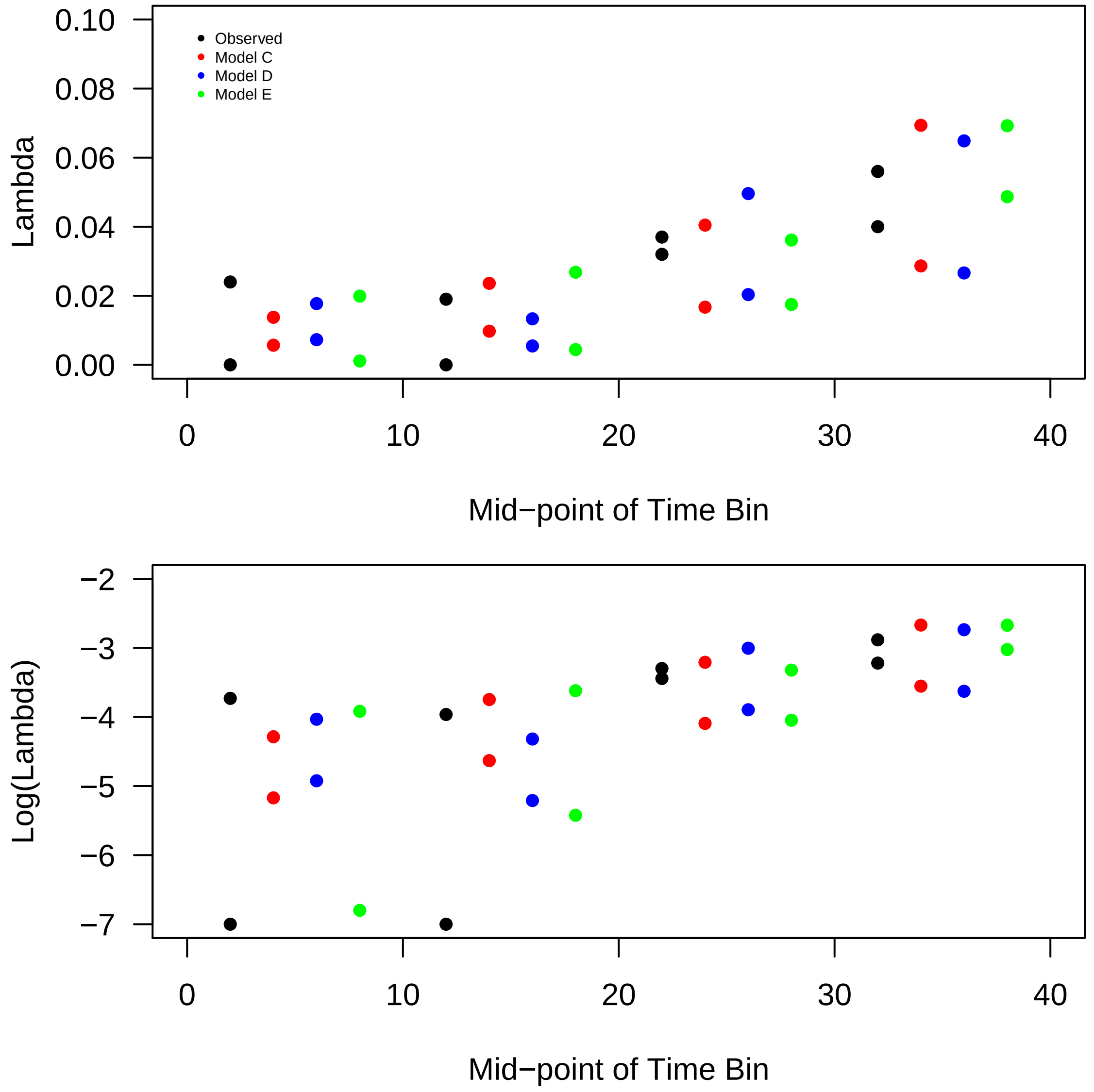
```r
lincom(fitE,c("male-15*male:midc","male-5*male:midc","male+5*male:midc","male+15*male:midc"),eform=TRUE
```

```
##                   Estimate   2.5 %       97.5 %    Chisq     Pr(>Chisq)
## male-15*male:midc 0.05600358 0.004905048 0.639423  5.38189   0.02034682
## male-5*male:midc  0.1646096  0.03624414  0.7476055 5.460166  0.0194548
## male+5*male:midc  0.4838319  0.1839557   1.272553  2.165207  0.1411656
## male+15*male:midc 1.422112   0.3629106   5.572732  0.2553863 0.6133077
```

```r
tapply(fitE$fitted,list(binned$male,binned$midp),mean)
```

```
##           5        15       25       35
## 0 2.4683054 2.8160342 2.963015 1.752645
## 1 0.1226269 0.4856199 1.660880 1.730874
```

What can you conclude about the proportional hazards assumption?



5. Summary

Here are some main points from the analysis

1. We have used Poisson regression models for the number of deaths by decade of year in the hospital to estimate the relative risk of death for men as compared to women.
2. The overall rate of death in this small group of 26 people is estimated to be 2% per year (95% CI: 1.1, 3.8% per year).
3. The rate for men is estimated to be 0.41 times that for women (95% CI: 0.13, 1.3). The evidence is not sufficiently strong to conclude men and women have different rates.
4. There is also a suggestion that the relative risk increases toward 1.0 with increasing duration of follow-up.

Some additional questions for you to consider:

- This analysis has considered sex and 10-year increment of hospitalization as important variables in understanding risk of death over time.

- What other key variable is missing?
- How would you incorporate this variable into the analysis?