Understanding Diffusion with **netdiffuseR**

Survival Analysis

Sunbelt 2016 INSNA

George Vega Yon, Stephani Dyal, Timothy Hayes, Thomas Valente

Department of Preventive Medicine
University of Southern California

Newport Beach, CA
March 28, 2016

Setup

We will use the medical innovations data

```r
# Loading the required packages
library(survival)
library(netdiffuseR)

# Loading the data
data("medInnovationsDiffNet")
medInnovationsDiffNet
```

```
## Dynamic network of class -diffnet-
##  # of nodes         : 125 (1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, ...)
##  # of time periods  : 18 (1 - 18)
##  Type               : directed
##  Final prevalence   : 1.00
##  Static attributes  : city, detail, meet, coll, attend, proage, length, ... (58)
##  Dynamic attributes : -
```

Preparing the data

From **netdiffuseR** we will get the following covariates:

- Cohesive exposure: Proportion of ego's adopters at each time period
- Structural equivalence exposure: Same as before but using the structural equivalence graph instead of the baseline network.

```
# Structural equivalence exposure
medInnovationsDiffNet[["seexp"]] <- exposure(
  medInnovationsDiffNet, groupvar="city", alt.graph = "se")
```

```
## Warning in exposure(medInnovationsDiffNet, groupvar = "city", alt.graph =
## "se"): To use alt.graph="se" -valued- has been switched to TRUE.
```

```
# Cohesive exposure
medInnovationsDiffNet[["cohexp"]] <- exposure(medInnovationsDiffNet)
```

Coercing data into a dataframe

```r
# Getting the data for running the cox regression
dat <- diffnet.attrs(medInnovationsDiffNet, as.df = TRUE)
dat <- subset(
  dat,
  subset = !is.na(toa) & per <= toa & per >= date & per < 18,
  select = c(id, per, toa, date, city, proage, proage2, seexp, cohexp))

# Creating the event variable
dat$event <- with(dat, toa==per)

head(dat)
```

```
##       id per toa date city proage proage2      seexp    cohexp event
## 40  1058   1   4    1    1      2       1 0.05297801 0.0000000 FALSE
## 67  2007   1  18    1    2      1       1 0.26751697 0.0000000 FALSE
## 68  2008   1   7    1    2      2       1 0.26929505 0.6666667 FALSE
## 70  2010   1   1    1    2      3       0 0.24794743 0.0000000  TRUE
## 107 3033   1   2    1    3      2       1 0.00000000 0.0000000 FALSE
## 136 1011   2   4    2    1      2       1 0.09889452 0.0000000 FALSE
```

The `survival` package

- In order to work, the `survival` package works with `Surv` objects.
- These store the response/events and the time frame during which these occurred.
- Usually take the following form: `Surv(start, end, event)`.
- In the case of longitudinal diffusion data this should be as follows: `Surv(period, toa, adopted)`, so, just like event analysis, we observe individuals since they are exposed until they adopt (and no further).

The survival package

The Suvreg object

- First, we need to create the Suvreg object using the function with the same name

```r
# Needs a start, stop, event
surv_mi <- with(dat, Surv(date, per, event))
```

```
## Warning in Surv(date, per, event): Stop time must be > start time, NA
## created
```

  Notice the warning as the time frames should be grater than 1.
- Now, let's take a look at the object itself

```r
head(surv_mi, 10)
```

```
##  [1] (NA,1+] (NA,1+] (NA,1+] (NA,1]  (NA,1+] (NA,2+] (NA,2+] (NA,2+]
##  [9] (NA,2+] ( 1,2+]
```

Fitting the model

All cities

```
# Fitting a model
set.seed(1988)
mymodel <- formula(surv_mi ~ factor(city) + proage + proage2 + seexp + cohexp)
out <- coxph(mymodel, data=dat)
```

Fitting the model

All cities (cont.)

Table 1: Fitting Proportional Hazards Regression Model: mymodel

	coef	exp(coef)	se(coef)	z	p
factor(city)2	-2.906	0.05471	0.7609	-3.819	0.0001341
factor(city)3	-1.015	0.3623	0.616	-1.648	0.09932
factor(city)4	-1.876	0.1532	0.7801	-2.405	0.01617
proage	-0.06066	0.9411	0.2329	-0.2605	0.7945
proage2	-0.7402	0.477	1.02	-0.7254	0.4682
seexp	-5.427	0.004395	1.264	-4.292	1.767e-05
cohexp	-0.1089	0.8968	0.652	-0.1671	0.8673

Likelihood ratio test=35.75 on 7 df, p=8.077809e-06 n= 265, number of events= 34 (45 observations deleted due to missingness)

Firring the model

City 1

We also can fit the model by city. Further, we can use the `subset` option of the function and create the `Surv` on the call:

```
# Only in city 1 (which I suspect has diffusion)
out_city1 <- coxph(Surv(date, per, event) ~ proage + proage2 + seexp + cohexp, dat,
      subset = city==1)
```

Table 2: Fitting Proportional Hazards Regression Model: Surv(date, per, event) ~ proage + proage2 + seexp + cohexp

	coef	exp(coef)	se(coef)	z	p
proage	-0.4172	0.6589	0.2714	-1.537	0.1243
proage2	-2.872	0.05658	1.243	-2.311	0.02084
seexp	-8.918	0.0001339	2.09	-4.268	1.972e-05
cohexp	-0.1269	0.8809	0.8882	-0.1428	0.8864

Likelihood ratio test=32.91 on 4 df, p=1.243661e-06 n= 137, number of events= 22 (26 observations deleted due to missingness)

Firring the model

All but city 1

Further, we can fit a model where we include all cities but city 1:

```
# Only in city 1 (which I suspect has diffusion)
out_nocity1 <- coxph(Surv(date, per, event) ~ proage + proage2 + seexp + cohexp, dat,
      subset = city!=1)
```

Table 3: Fitting Proportional Hazards Regression Model: Surv(date, per, event) ~ proage + proage2 + seexp + cohexp

	coef	exp(coef)	se(coef)	z	p
proage	0.2173	1.243	0.5158	0.4214	0.6735
proage2	1.387	4.004	1.502	0.9237	0.3557
seexp	-3.198	0.04085	1.871	-1.709	0.08749
cohexp	-3.111	0.04457	1.394	-2.231	0.02566

Likelihood ratio test=16.03 on 4 df, p=0.002985766 n= 128, number of events= 12 (19 observations deleted due to missingness)

Firring the model

City 1 (cont. 3)

More diganostics can be done as follows:

```
# Diagnostics
fit <- survfit(out)

plot(fit, mark.time=TRUE, lty=1:2,
     xlab="Time period", ylab="Survival")
legend("bottomleft", c("Adopters", "Non-adopters"),
       lty=2:1, bty="n")
```