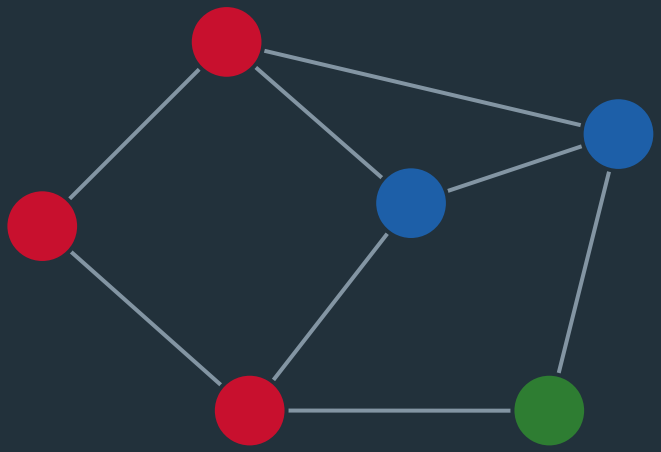
Event models

Course B, Week 11

Chapters 27 and 28 · a lecture and a lab

Statistical models for networks

Instructors are free to use, adapt and share these slides.



Today: a lecture, a break and a lab

0:00	**1 Events and the relational event model**	40 min
0:40	**2 The dynamic network actor model**	40 min
1:20	Break	10 min
1:30	**3 Lab: the relational event model**	40 min
2:10	**4 Lab: the DyNAM**	45 min

One e-mail log, two models

```
. nwrem version
. nwdynam version
```

nwrem 1.5.0 and nwdynam 1.9.0, installed after the core with nwinstall.

```
. nwwebuse manufacturing, nwclear
. keep if t < 14
```

The e-mails of the first two weeks of the data.

Question of the week: given everything that has happened so far, which pair writes next, and who acts before whom they pick?

E-mail log

2,738 e-mails among 129 employees, with the time in days.

Toy log

Ten events among three actors, typed in, for the arithmetic.

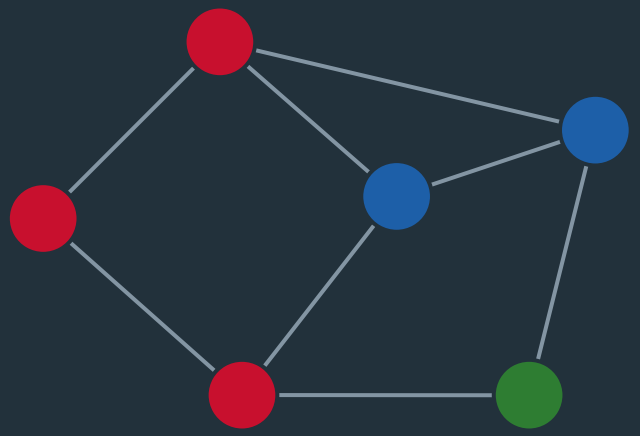
Employee file

Supervisor, level and division, to ask who writes to whom.

1

Events and the relational event model

40 min · Chapter 27: events, the partial likelihood, activity, inertia, recency, a pair covariate



A relational event is a dated act, and the order matters

Event data

Each row is one act: a sender, a receiver and a time. Order and timing are data. There are no waves, and there is no tie that ends unless an event records it.

Panel data

Networks observed at a few moments, as in the SAOM of Chapter 25. What happened between the waves is latent and has to be reconstructed by simulation.

The question changes: not which ties exist, but which pair is most likely to produce the next event, given everything that has happened so far.

The e-mail log is declared as an event network

```
. nwwebuse manufacturing, nwclear
. keep if t < 14
. nwset sender receiver, eventtime(t) name(mail)
. nwsummarize mail
   Nodes: 129
   Arcs: 1162
   Temporal: true
      Temporal type: event
```

2,738

e-mails among 129 employees in the first 14 days

`eventtime()` declares the log as an event network: one row per event, kept in time order. The 1,162 arcs are the pairs that wrote at least once.

At every event: the pair that fired against all pairs at risk

$$P(\text{pair that fired}) = \frac{\exp(\theta \cdot s \text{ of that pair})}{\text{sum over pairs at risk of } \exp(\theta \cdot s)}$$

s are statistics of the pair computed from the history strictly before the event: past events of this pair, activity of the sender, popularity of the receiver, a covariate.

Like a choice probability: the realized pair is compared with its contemporaneous alternatives, and the product over all events is the likelihood.

6

ordered pairs at risk in the toy log: 3 × 2

16,512

ordered pairs at risk at every e-mail: 129 × 128

No intercept, and blind to speed

It says

Only the order of events enters. A pair with more history than the others gets more weight in the denominator, whatever the clock says.

It does not say

How long a wait was. A burst of messages within an hour and the same messages spread over a month give the same estimates.

No intercept

A term that is the same for every pair cancels from the numerator and the denominator, so none can be estimated. Do not look for one.

Real time

The rate sub-model of the DyNAM with an intercept uses the waiting times (block 2).

A ten-event toy log: the arithmetic is small enough to check

```
. nwset sender receiver, eventtime(t) name(chat)
. nwrem chat, nodsnd nidrec
Events: 10                  Log likelihood:  -15.9033

        chat | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
      nodsnd |   1.268585   1.354195     0.94   0.349
      nidrec |  -5.567766   3.511139    -1.59   0.113
```

−15.90

log likelihood of the fit; with all coefficients at zero it is −10 ln 6 = −17.92

Ten events among three actors carry little information: the standard errors are as large as the estimates.

- **The ten events, t = 1 to 10**

  1→2, 1→3, 2→1, 1→2, 3→2, 2→3, 1→3, 3→1, 2→1, 1→3

Busy senders keep writing, popular receivers keep receiving

```
. nwrem mail, nodsnd nidrec
Events: 2738                  Log likelihood: -2.53e+04

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
      nodsnd |   36.99531   1.478789    25.02   0.000
      nidrec |   35.10519   .6882604    51.01   0.000
```

Both positive and large. Neither is stable: both shift when other effects enter.

nodsnd

the share of all events so far that this sender sent

nidrec

the share of all events so far that this receiver received

Inertia: people write to those they wrote to before

```
. nwrem mail, nodsnd nidrec frpsndsnd
Events: 2738                  Log likelihood: -2.44e+04

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
      nodsnd |   39.51321    1.54486    25.58   0.000
      nidrec |   26.72635   .7611982    35.11   0.000
   frpsndsnd |   4.406264   .0747082    58.98   0.000
```

× 1.55

in the odds of a pair per 10 percentage points more of the sender's earlier mail to that receiver

`frpsndsnd` is the fraction of the sender's earlier events that went to this receiver: exp(4.406 × 0.1) = 1.55.

Frequency or recency? Recency fits better

```
. nwrem mail, nodsnd nidrec rsndsnd
Events: 2738                Log likelihood: -2.36e+04
      nodsnd |    30.2814   1.465531    20.66   0.000
      nidrec |   20.73686   .7949586    26.09   0.000
     rsndsnd |   3.592788   .0498983    72.00   0.000
. estimates stats freq rec
       Model |          N   ll(null)  ll(model)      df        AIC
        freq |      2,738  -26591.03  -24364.72       3   48735.45
         rec |      2,738  -26591.03  -23599.47       3   47204.93
```

1,531

AIC points in favor of recency, with the same number of parameters (3 and 3)

`rsndsnd`: how recently the sender wrote to this receiver, as a rank: the last receiver ranks first. Recent contact predicts the next e-mail better than frequent contact.

Both in one model: recency stays, frequency falls to a fifth

Model	nodsnd	nidrec	frpsndsnd	rsndsnd	log lik.	AIC
baseline	36.995	35.105			−25,292.5	50,589.1
frequency	39.513	26.726	4.406		−24,364.7	48,735.5
recency	30.281	20.737		3.593	−23,599.5	47,204.9
both	32.813	20.026	.894	3.279	−23,570.8	47,149.7

Next to recency, frequency keeps only a fifth of its size: 4.41 alone, 0.89 together (z = 7.8). Both stay positive and significant. The degree coefficients move from column to column.

Chapter 27 listings; log likelihood and AIC from estimates stats, 2,738 events, 129 actors.

Same supervisor, with and without history

Model with covevent(samesup)	coefficient	z	odds ratio
nodsnd nidrec	1.463	27.8	× 4.3
+ frpsndsnd	1.135	20.8	× 3.1
+ rsndsnd (instead)	1.014	18.8	× 2.8

A shared supervisor is a narrow tie: 330 of the 8,256 pairs (4.0 percent). The effect falls once history is in the model and most of it stays.

It says
With recency in the model, a pair with the same supervisor has about 2.8 times the odds of the next e-mail, other things equal.

It does not say
That the reporting line causes the e-mails: it carries information beyond the sender's earlier e-mails to this partner.

What to take away from the relational event model

Compare pairs

Each event is a comparison of the pair that fired with all pairs at risk, 129 × 128 for the e-mail log. No intercept; only the order enters.

Frequency and recency

Different claims about how past contact shapes the next event. Here recency fits better (AIC 47,205 against 48,735), and frequency keeps a fifth of its size next to it.

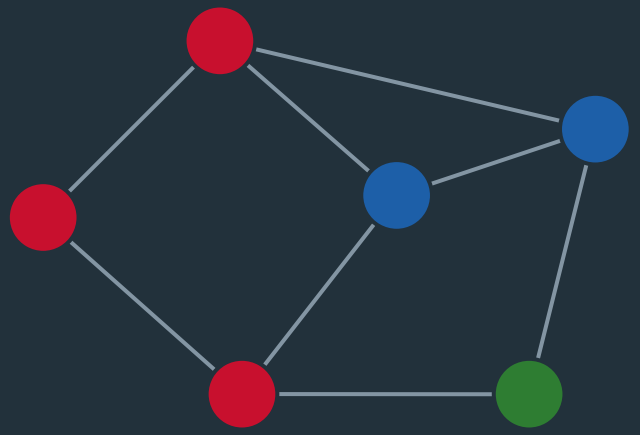
Covariates with history

Read a covariate with and without the history effects, and say which question each version answers: same supervisor 1.46, then 1.14 and 1.01.

2

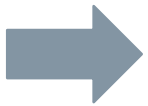
The dynamic network actor model

40 min · Chapter 28: who acts and whom they choose, the two sub-models, same branch, the rate with an intercept, four models



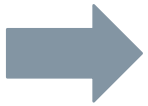
Two decisions, not one

Relational event model
One step: which ordered pair fires next? Compared among 16,512 pairs.



one comparison among pairs

DyNAM, step 1: rate
Which actor acts next? Compared among 129 actors; statistics of the actor.



DyNAM, step 2: choice
Whom does the actor pick? Compared among the 128 others; statistics of the pair.

Why split

A very active sender produces many events with many partners. A pair-level model must explain all of them through the pairs; the DyNAM puts activity in the rate and partner choice in the choice sub-model.

The choice sub-model: whom the sender picks

```
. nwset sender receiver, eventtime(t) directed name(mail)
. nwdynam mail
Dynamic Network Actor Model - choice sub-model (MLE)
Events: 2738              Log likelihood: -9.96e+03

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
     inertia |   2.036706   .0636816    31.98   0.000
       recip |   1.883335   .0609398    30.90   0.000
       indeg |    .019011   .0011356    16.74   0.000
```

× 7.7

in the weight of a candidate the sender wrote to before, against an otherwise identical one: exp(2.037)

`recip` works the same way for candidates who wrote to the sender before: exp(1.883) = 6.6.

Reading a choice coefficient: a comparison of two candidates

It says

Given that an employee is about to write, candidates already written to before are far more likely to be chosen, net of reciprocity and in-degree (z = 32.0).

It does not say

That a repeat e-mail is exp(2.04) = 7.7 times as likely in general. The number compares two candidates within one sender's choice; it is not a ratio of probabilities across the log.

Reciprocity

recip 1.883: exp(1.883) = 6.6 for a candidate who wrote to the sender before.

In-degree

indeg .019 is per colleague: ten more earlier senders make a candidate exp(0.19) ≈ 1.2 times as likely a pick.

The rate sub-model: who writes next

```
. nwdynam mail, submodel(rate)
Dynamic Network Actor Model - rate sub-model, no intercept (MLE)
Events: 2738            Log likelihood: -1.27e+04

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
       indeg |  -.0020857    .002177    -0.96   0.338
      outdeg |   .1307426   .0043335    30.17   0.000
```

× 1.14

in the weight of an employee as the next sender for each further colleague already written to: exp(0.131)

It says

Employees who have already written to more colleagues are more likely to write next (z = 30.2). Mail received does not matter (p = 0.34).

It does not say

How long the actor waits. Without an intercept the rate model uses only the order of events.

indeg: the actor in one model, the candidate in the other

Effect	What it measures	Compared among	Estimate
`nidrec, nwrem`	share of all events so far received by the receiver	16,512 ordered pairs	26.7
`indeg, choice`	number of distinct senders who wrote to the candidate	the 128 other employees	.019
`indeg, rate`	number of distinct senders who wrote to the acting employee	129 candidate senders	−.002
`outdeg, rate`	number of distinct receivers the acting employee wrote to	129 candidate senders	.131

Same sign, different scale. A share compared among pairs and a count compared among candidates give coefficients that cannot be ranked by size. Compare signs, significance and the substantive reading.

Same branch, net of structure

```
. nwdynam mail, inertia recip indeg same(branch)
Events: 2738              Log likelihood: -9.91e+03

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
     inertia |   1.987294    .063794    31.15   0.000
       recip |   1.842186   .0609646    30.22   0.000
       indeg |   .0194229   .0011295    17.20   0.000
        same |   .4944068   .0467183    10.58   0.000
```

× 1.6

in the weight of a candidate from the sender's own branch: exp(0.494)

same(branch) is 1 when sender and candidate belong to the same branch of the company: 12 branches, 72 of the 129 employees in the largest. The effect stays after inertia, reciprocity and in-degree, as covevent(samesup) did in Chapter 27.

When real time matters: the rate with an intercept

```
. nwdynam mail, submodel(rate) intercept indeg outdeg
Dynamic Network Actor Model - rate sub-model, with intercept (MLE)
Events: 2738            Log likelihood: -435.5168

        mail | Coefficient  Std. err.      z    P>|z|
-------------+----------------------------------------
   Intercept |   .1240626   .0317477     3.91   0.000
       indeg |  -.0018054   .0022274    -0.81   0.418
      outdeg |   .0889101    .003791    23.45   0.000
```

21 h

expected wait for an employee without history: exp(−0.124) = 0.88 days

Each employee is a Poisson clock with a rate of exp(intercept + effects) e-mails per day. The assumption: a constant pace, which nights and weekends violate. The log likelihood is not comparable with the order-only models.

Four models, four accounts of the process

Model	Data	What is modeled	What it cannot see
ERGM	One network	The network as one state; each tie depends on others through configurations	Order and timing; how the network came about
SAOM	Two or more panel waves	Actors changing one tie at a time between waves, from rate and objective functions	The single changes between waves
REM	Time-stamped log of events	Each event as one pair chosen among all pairs at risk, given the history	Ties that end; whether an effect works through who acts or whom they choose
DyNAM	Time-stamped log of events	Each event as an actor deciding to act (rate) and then choosing a receiver (choice)	Ties that end, unless recorded as events

Table 28.1 of Chapter 28, shortened. QAP regression treats dependence between ties as a nuisance for the standard errors, not as something to model.

What can go wrong

Reading indeg the same in both

Under submodel(rate) they describe the acting actor, under submodel(choice) the candidate receiver.

Order or waiting times?

Without intercept the rate sub-model is order-only however carefully the data were time-stamped; the choice sub-model never has an intercept.

Declaring the log undirected

nwdynam stops for the rate and choice sub-models; only choice_coordination needs an undirected network.

Looking for an intercept in nwrem

The likelihood compares pairs at the same instant, so none can be estimated.

What to take away from the dynamic network actor model

Split the decision

Rate: who acts next, among 129 actors. Choice: whom the actor picks, among 128. Different statistics, different claims about the group.

Same name, two meanings

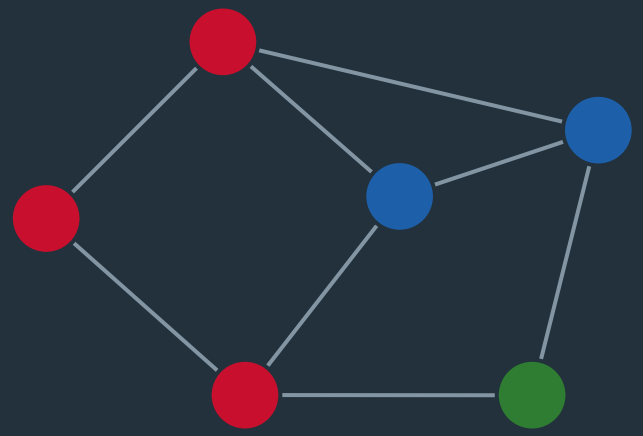
indeg and outdeg describe the actor in the rate and the candidate in the choice sub-model. Do not rank coefficients across models by size.

Order or real time

Without intercept the rate model uses the order of events only. With it, every actor has a clock: 21 hours for an employee without history.

Break

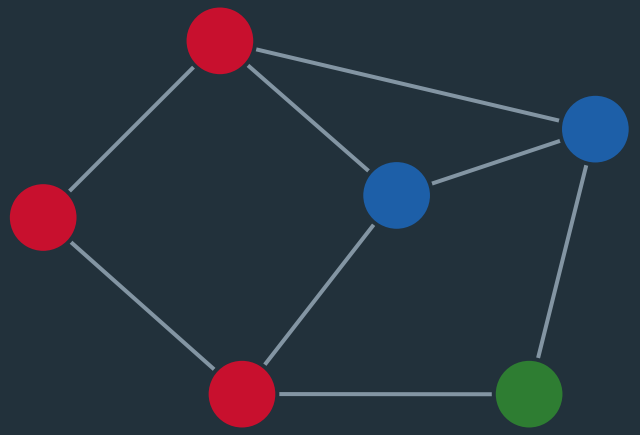
10 minutes



3

Lab: the relational event model

40 min · Run along: nwset, nwrem, estimates stats, covevent(); Ch. 27, Ex. 2 and 4



Run along 1: baseline, frequency, recency

```
. nwwebuse manufacturing, nwclear
. keep if t < 14
. nwset sender receiver, eventtime(t) name(mail)
. nwrem mail, nodsnd nidrec
. quietly nwrem mail, nodsnd nidrec frpsndsnd
. estimates store freq
. quietly nwrem mail, nodsnd nidrec rsndsnd
. estimates store rec
. estimates stats freq rec
```

Predict first. Which of the two inertia effects will have the lower AIC? Then press return.

A mismatch usually means a missing keep if t < 14 or an earlier network still in memory (nwclear).

Checkpoint	Value
actors; events	129; 2738
baseline nodsnd	36.995
baseline nidrec	35.105
ll(null)	−26591.03
AIC, frequency	48735.45
AIC, recency	47204.93

Run along 2: the same-supervisor network and covevent()

```
. nwload mail, xvars
. rename _nwnode label
. merge 1:1 label using "$nwwebpath/manufacturing_nodes.dta", ///
>         keep(master match) nogen
. nwexpand supervisor, name(samesup)
. nwsummarize samesup
. nwrem mail, nodsnd nidrec covevent(samesup)
. nwrem mail, nodsnd nidrec rsndsnd covevent(samesup)
```

Checkpoint	Value
matched employees	129
edges of samesup	330
density	.04
covevent_samesup alone	1.463
with rsndsnd	1.014
rsndsnd, same fit	3.432

Predict first. Will the supervisor coefficient grow or shrink when recency enters? By roughly how much?

covevent() needs a network over the actors, one row per actor: nwload with xvars, not the event rows.

Hands-on (20 min)

Ch. 27, Ex. 2

The e-mail log has 129 actors and 2,738 events. (a) How many ordered pairs are in the risk set at each event? (b) If all coefficients were zero, so that every pair were equally likely, what would the log likelihood of the 2,738 events be? (c) Compare it with the log likelihood of the baseline model with nodsnd and nidrec (-2.53×10^{4}).

Hint Each of the 129 employees can write to any of the 128 others. With equal weights every event has the same probability, one over the size of the risk set, so the log likelihood is 2,738 times the logarithm of that probability; display computes it.

Ch. 27, Ex. 4

Fit nwrem mail, nodsnd nidrec rsndsnd rrecsnd. rrecsnd, recency of receipt, is the reciprocal rank of the receiver among the people who wrote to the sender, ordered by how recently they did. Report the coefficients of rsndsnd and rrecsnd and the log likelihood, next to the fit with rsndsnd alone (3.59; -2.36×10^{4}). Does writing again to one's latest partner still matter once answering the latest writer is in the model, and by how much does its coefficient shrink?

Hint display e(ll) after each fit prints the log likelihood with all its digits. Divide the rsndsnd coefficient of the joint fit by its value in the fit with rsndsnd alone.

Chapter 27, Ex. 2 and 4: the answers

Ex. 2	Result
(a) pairs at risk	129 × 128 = 16,512
(b) log likelihood at zero	−2,738 ln 16,512 = −26,591.03
(c) baseline, nodsnd nidrec	−25,292.54: about 1,300 better

Two coefficients alone raise the log likelihood by about 1,300 over equal weights. The null value is the ll(null) of estimates stats.

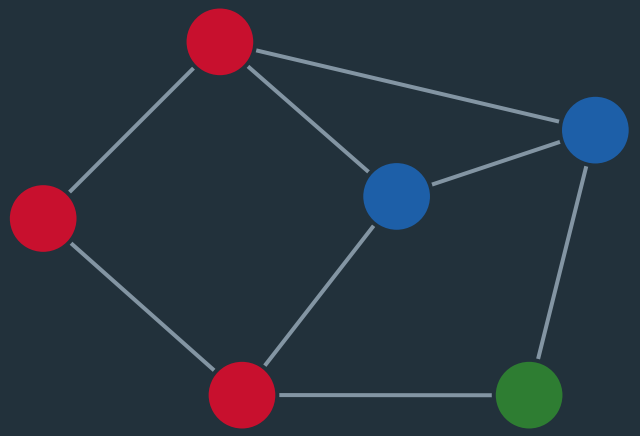
Ex. 4	rsndsnd alone	joint fit
rsndsnd	3.593	1.637 (z = 23.5)
rrecsnd		3.316 (z = 53.2)
log likelihood	−23,599.47	−22,363.94

Still positive, at 46 percent of its size: 1.637 / 3.593. About half of what looked like repeated writing was answering the latest writer.

4

Lab: the DyNAM

45 min · Run along: nwdynam, submodel(rate), same(), intercept; Ch. 28, Ex. 3 and 8



Run along 3: choice and rate sub-models

```
. nwwebuse manufacturing, nwclear
. keep if t < 14
. nwset sender receiver, eventtime(t) directed name(mail)
. nwdynam mail
. nwdynam mail, submodel(rate)
```

Predict first. Which of in-degree and out-degree will matter for who writes next?

Without directed, nwset can declare an undirected network and nwdynam stops. The toy log of Chapter 27 is one: nwclear, then declare it with directed.

Checkpoint	Value
choice: inertia	2.037
choice: recip	1.883
choice: indeg	.0190
choice log likelihood	−9.96e+03
rate: outdeg (z)	.1307 (30.2)
rate: indeg (p)	−.0021 (.338)

Run along 4: same(branch) and the rate with an intercept

```
. nwload mail, xvars
. rename _nwnode label
. merge 1:1 label using "$nwwebpath/manufacturing_nodes.dta", ///
>         keep(master match) nogen
. rename label _nwnode
. encode division, generate(branch)
. nwdynam mail, inertia recip indeg same(branch)
. nwdynam mail, submodel(rate) intercept indeg outdeg
```

Checkpoint	Value
same(branch)	.494 (z = 10.58)
log likelihood	−9.91e+03
Intercept	.1241 (z = 3.91)
outdeg, with intercept	.0889
log likelihood, intercept	−435.5168

Predict first. Can the log likelihood −435.5 of the model with an intercept be compared with −1.27e+04 of the order-only rate model?

Hands-on (20 min)

Ch. 28, Ex. 3

Using the rate sub-model with intercept, indeg and outdeg (intercept 0.1241, outdeg 0.0889), compute the expected waiting time until the next email for an employee who has so far written to 20 different colleagues and not yet been written to (in-degree 0), and compare it with the 21 hours of an employee who has written to none. Express the result in hours as well as in days.

Hint The hazard is exp(0.1241 + 0.0889 × out-degree) emails per day, and the expected waiting time is its reciprocal, exp{−(0.1241 + 0.0889 × out-degree)} days; multiply by 24 for hours.

Ch. 28, Ex. 8

For each question, choose among QAP regression, ERGM, SAOM, REM and DyNAM, and give the data the model needs and one reason: (a) do friendships observed in three survey waves become more reciprocated as they change? (b) In a log of time-stamped emails in which a few employees send most messages, to whom do employees write, once they write? (c) Is a tie pattern observed once more common than chance and the network's own structure would produce?

Hint Ask first whether the data are one network, a few waves or a stream of events, and then whether the question is about the ties, the change in them, who acts, or whom the actor chooses.

Chapter 28, Ex. 3 and 8: the answers

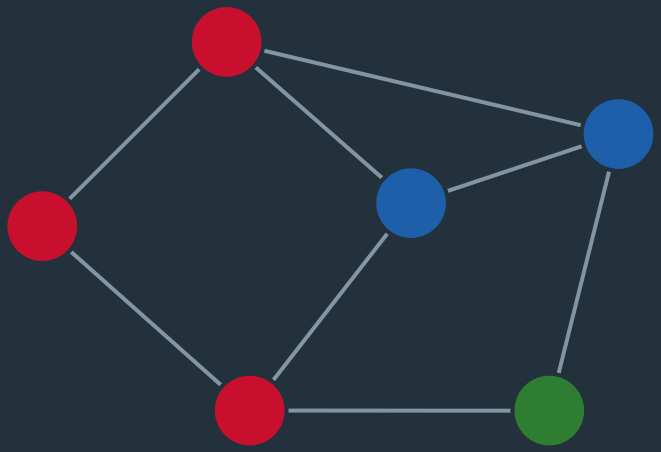
Ex. 3: out-degree	hazard per day	expected waiting time
0 (no e-mail written yet)	exp(0.124) = 1.13	0.88 days = 21.2 hours
20, in-degree 0	exp(1.902) = 6.70	0.149 days = 3.58 hours

The more someone has written, the sooner the next e-mail. In-degree 0 adds nothing; the expected wait is the reciprocal of the hazard.

Ex. 8	Model	Data and one reason
(a) reciprocated friendships, three waves	SAOM	panel waves; ties change between waves, and reciprocity is an effect of the model
(b) to whom they write, once they write	DyNAM, choice	an event log; the rate carries who is busy, the choice carries whom they write to
(c) a pattern beyond chance and structure	ERGM	one network; configurations model the structure that chance would not produce

One defensible answer; accept others that name the data the model needs and a reason.

Before next week



Week 12: read Chapters 29 and 30, A complete network analysis and Reproducible and responsible network research

Left over: Ch. 27 Ex. 1, 3 and 5 to 8; Ch. 28 Ex. 1, 2 and 4 to 7

Manuals: nwcommands.org/nwrem/manual · nwcommands.org/nwdynam/manual

Do-files and teaching material: nwcommands.org/book

Instructors are free to use, adapt and share these slides.