Evidencing preferential attachment in dependency network evolution

Clement Lee

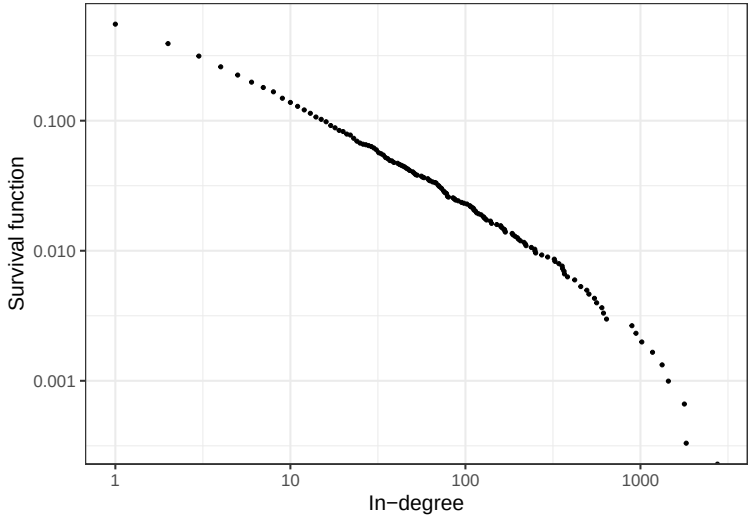
2025-03-21

Preferential attachment (PA)

- Barabási and Albert (1999)
  - Existing nodes: Node i with degree d_i
  - New node brings one new edge, connects to node i with probability $\frac{d_i}{\sum_j d_j}$
- Network grows according to simple rules
- The-rich-get-richer effect
- Borrowing an animation

Power law degree distribution

- Linear on the log-log scale above a certain degree



In the literature

- **Snapshots** seem to follow power law $\rightarrow$ attribute to PA
- Converse not necessarily true
  - E.g. generalised random graphs (Hofstad 2016) lead to similar phenomenon
- Debate on what does it mean to "follow the power law"?
  - Broido and Clauset (2019), Voitalov et al. (2019)
- Extending the linear model
  - $\frac{d_i^\alpha}{\sum_j d_j^\alpha}$, where $\alpha > 0$
  - General: $\frac{g(d_i)}{\sum_j g(d_j)}$

Taking one step back

- **If** we have the data on the evolution:
  - Can we evidence the PA?
  - How do we model the data?
  - What is the precise form of g (the weight function)?

Data: R packages on CRAN

- On 2019-01-29

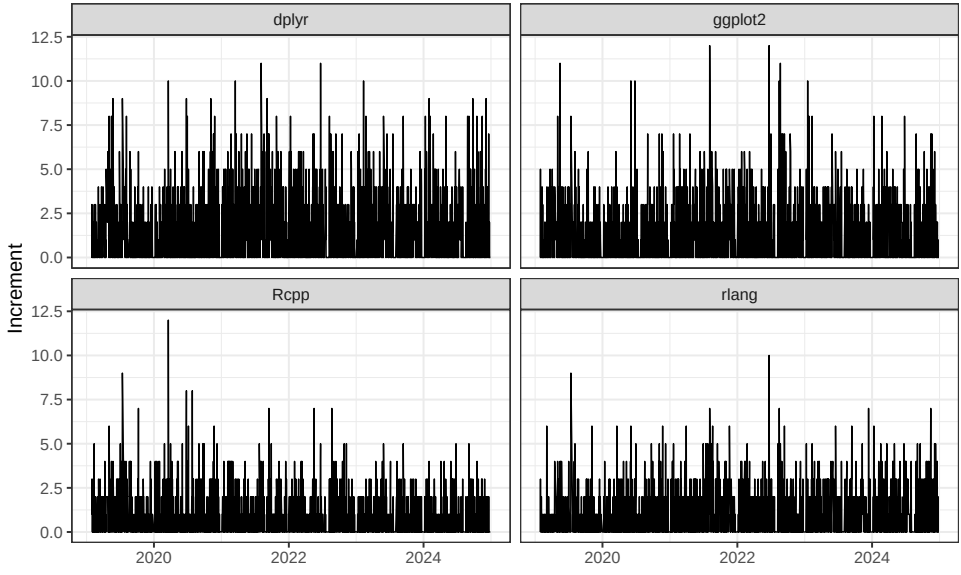
```
## # A tibble: 10 x 4
##    from       to            type     reverse
##    <chr>      <chr>         <chr>    <lgl>
##  1 stabledist alphastable   imports  TRUE
##  2 mlr        aslib         imports  TRUE
##  3 testthat   eurostat      suggests TRUE
##  4 sjlabelled sjPlot        suggests FALSE
##  5 kableExtra magick        suggests FALSE
##  6 preprosim  methods       imports  FALSE
##  7 knitr      MDMR          suggests TRUE
##  8 DEoptim    BMhyb         imports  TRUE
##  9 dotCall64  RColorBrewer  suggests FALSE
## 10 pROC       wevid         imports  TRUE
```

Daily increments

- From 2019-01-29 to 2019-01-30

```
## # A tibble: 10 x 5
##    from        to         type       reverse add
##    <chr>       <chr>      <chr>      <lgl>   <lgl>
##  1 pmxTools    stats      imports    FALSE   FALSE
##  2 cliqueMS    MSnbase    imports    FALSE   TRUE
##  3 mRchmadness shiny      imports    FALSE   FALSE
##  4 pmxTools    xpose      imports    FALSE   FALSE
##  5 taxize      vcr        suggests   FALSE   TRUE
##  6 xpose       vpc        imports    FALSE   FALSE
##  7 maptools    GpGp       suggests   TRUE    TRUE
##  8 glmmsr      rmarkdown  suggests   FALSE   FALSE
##  9 xpose       rlang      imports    FALSE   FALSE
## 10 glmmsr      Rcpp       linking to FALSE   FALSE
```

Aggregating once

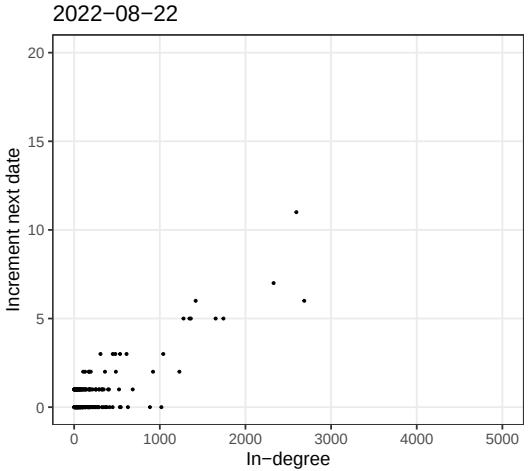


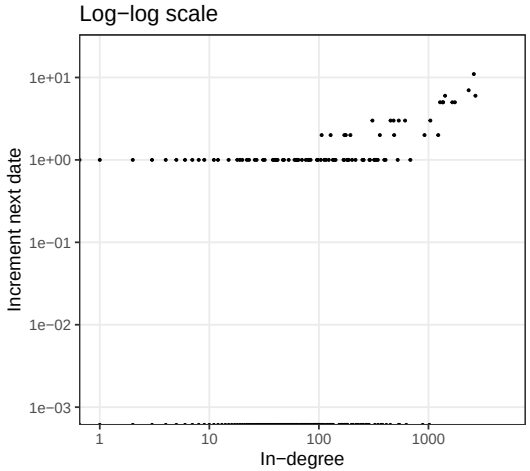
Aggregating twice

```
## # A tibble: 355,785 x 4
##    `previous date` indegree increment count
##    <date>              <dbl>     <dbl> <dbl>
##  1 2019-01-29              0         0  9315
##  2 2019-01-29              0         1     1
##  3 2019-01-29              1         0  1306
##  4 2019-01-29              2         0   459
##  5 2019-01-29              3         0   226
##  6 2019-01-29              4         0   161
##  7 2019-01-29              5         0   103
##  8 2019-01-29              6         0    75
##  9 2019-01-29              7         0    49
## 10 2019-01-29              8         0    40
## # i 355,775 more rows
```

Scatter plot for a single day

- More than 1 new edge





Tweaking the model

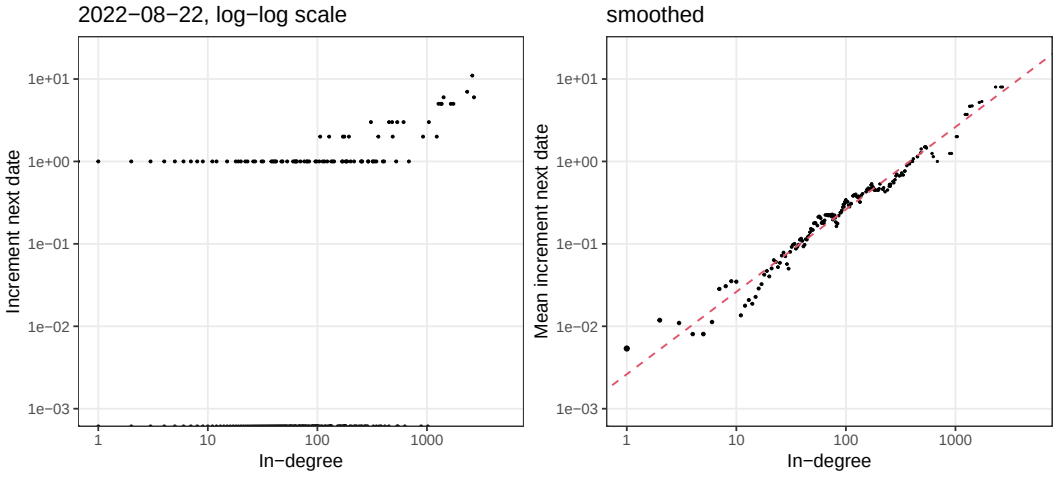
Y_i: number of new edges / increments for node i

- 1 new edge, $(Y_1, Y_2, \ldots) \sim \text{Multinomial}\left(\frac{d_1^\alpha}{\sum_j d_j^\alpha}, \frac{d_2^\alpha}{\sum_j d_j^\alpha}, \ldots\right)$
- m new edges, $(Y_1, Y_2, \ldots) \sim \text{Multinomial}\left(\frac{d_1^\alpha}{\sum_j d_j^\alpha}, \frac{d_2^\alpha}{\sum_j d_j^\alpha}, \ldots\right)$
- $M \sim \text{Po}(m)$ new edges, $(Y_1, Y_2, \ldots)|M = m \sim \text{Multinomial}\left(\frac{d_1^\alpha}{\sum_j d_j^\alpha}, \frac{d_2^\alpha}{\sum_j d_j^\alpha}, \ldots\right)$
  - $(Y_1, Y_2, \ldots) \sim$ Independent Poissons with means $\left(\frac{m\ d_1^\alpha}{\sum_j d_j^\alpha}, \frac{m\ d_2^\alpha}{\sum_j d_j^\alpha}, \ldots\right)$

Taking expectation

- $E(Y_i) = \frac{m\, d_i^{\alpha}}{\sum_j d_j^{\alpha}}$
- $\log E(Y_i) = \alpha \log d_i + \log m - \log(\sum_j d_j^{\alpha})$
- $\log y_i \approx \alpha \log d_i + c$
- y_i observed increment in (in-)degree

Smoothed scatter plot



Animation

Tail behaviour poorly captured when $g(x) = x^{\alpha}$

Theoretically

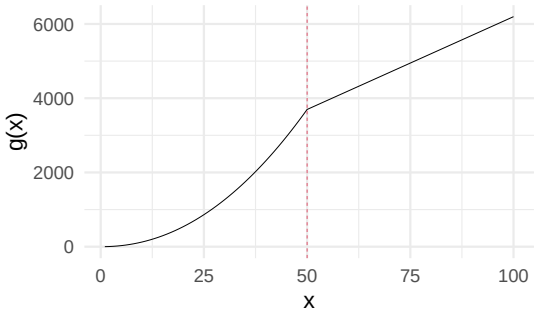
- $\alpha < 1$ (sublinear): Weibull tail (long-tailed but light-tailed)
- $\alpha > 1$ (superlinear): Degeneracy

Empirically

- Subtle tail behaviour (Lee, Eastoe, and Farrell 2024) in real data
- Heavy-tailed, but not as heavy as implied by when $\alpha = 1$

Recent findings

- $g(x) = \begin{cases} x^{\alpha}, & x \leq \gamma, \\ \gamma^{\alpha} + \beta(x - \gamma), & x \geq \gamma \end{cases}$
- Sub/super-linear up to threshold γ, then linear above
- Flexible heavy-tail behaviour



Actual modelling

Regress increment on (in-)degree

- $Y_i \sim \text{Po}\left(\frac{m\, g(d_i)}{\sum_j g(d_j)}\right)$

Choice of $g(x)$

- Power function: $g(x) = x^\alpha + \delta$
- δ: "zero appeal"
- Piecewise function: $g(x) = \begin{cases} x^\alpha + \delta, & x \leq \gamma, \\ \gamma^\alpha + \beta(x - \gamma) + \delta, & x \geq \gamma \end{cases}$

Infer (α, δ) (and (β, γ) if necessary)

Back to CRAN packages

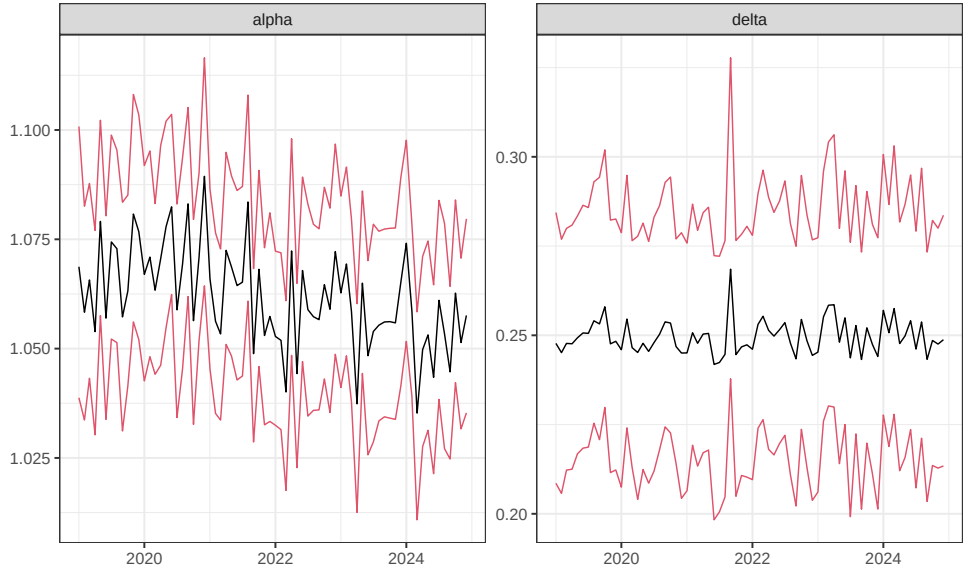
rstan: R Interface to Stan

User-facing R functions are provided to parse, compile, test, estimate, and analyze Stan models by accessing the header-only Stan library provided by the 'StanHeaders' package. The Stan project develops a probabilistic programming language that implements full Bayesian statistical inference via Markov Chain Monte Carlo, rough Bayesian inference via 'variational' approximation, and (optionally penalized) maximum likelihood estimation via optimization. In all three cases, automatic differentiation is used to quickly and accurately evaluate gradients without burdening the user with the need to derive the partial derivatives.

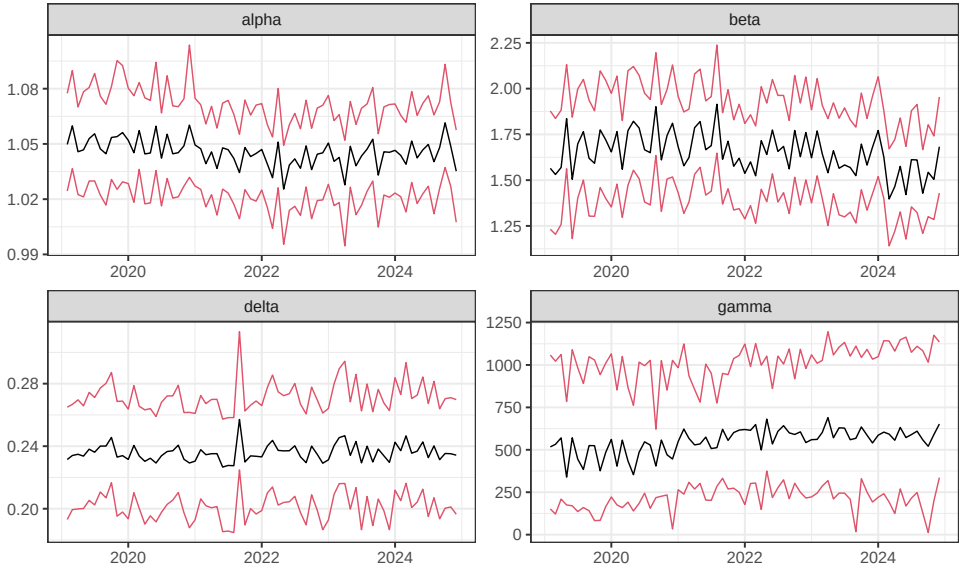
Version:	2.32.7
Depends:	R (≥ 3.4.0), StanHeaders (≥ 2.32.0)
Imports:	methods, stats4, inline (≥ 0.3.19), gridExtra (≥ 2.3), Rcpp (≥ 1.0.7), RcppParallel (≥ 5.1.4), loo (≥ 2.4.1), pkgbuild (≥ 1.2.0), QuickJSR, ggplot2 (≥ 3.3.5)
LinkingTo:	Rcpp (≥ 1.0.7), RcppEigen (≥ 0.3.4.0.0), BH (≥ 1.75.0-0), StanHeaders (≥ 2.32.0), RcppParallel (≥ 5.1.4)
Suggests:	testthat (≥ 3.0.4), parallel, KernSmooth, shinystan, bayesplot, rmarkdown, rstantools, rstudioapi, Matrix, knitr, coda, V8

- Imports only; new edges brought by new packages only
- Increments of all days in each month bundled
  - $\neq$ Comparing snapshots at beginning and end of month
- Hierarchical model across the months – $(\{\alpha_t\}, \{\beta_t\}, \{d_{0t}\}, \{\delta_t\})$
- Also a single fit as a benchmark – $(\alpha, \beta, \gamma, \delta)$

Power function



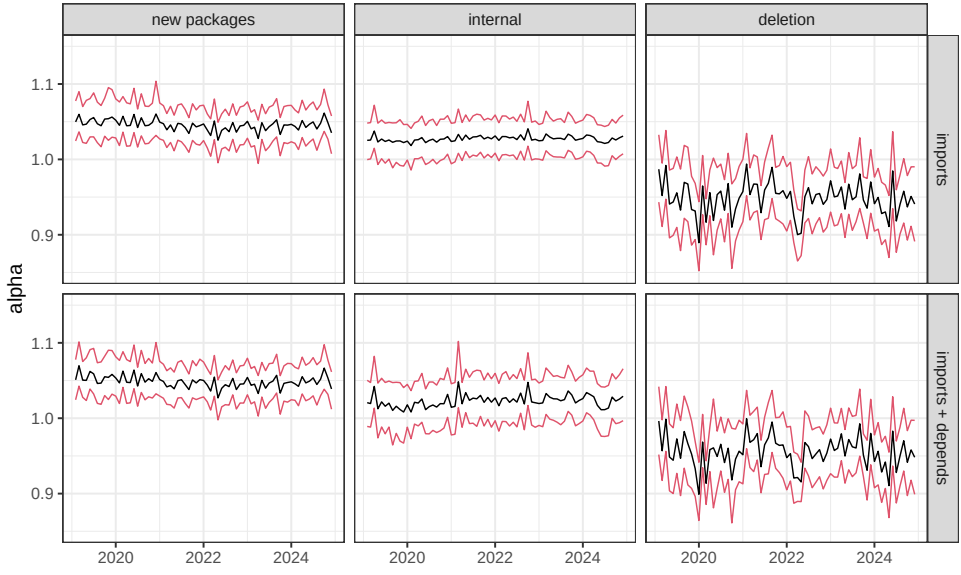
Piecewise function



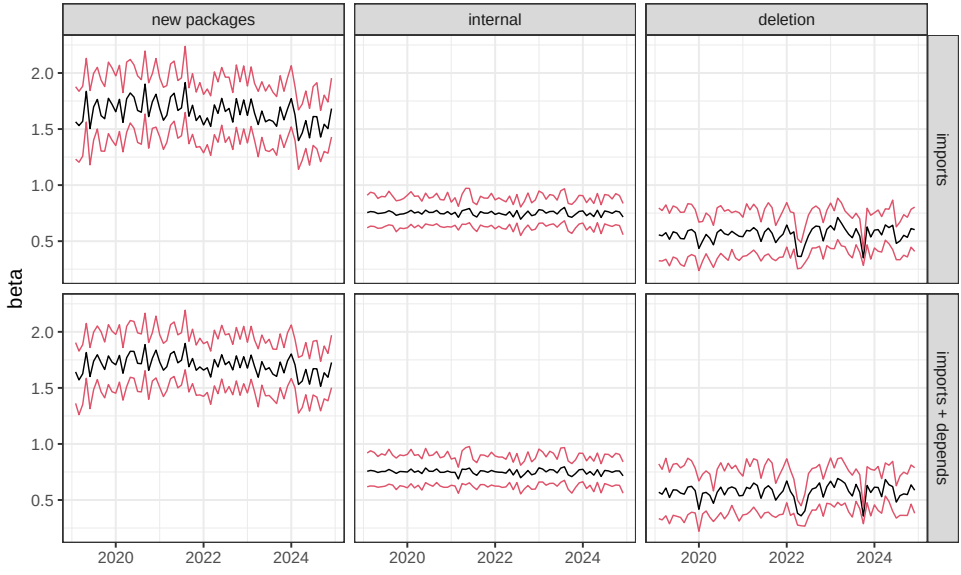
Extending

- Previously: imports, new edges by new packages only
- $$\left\{ \begin{array}{l} \text{imports} \\ \text{imports} + \text{depends} \end{array} \right\} \times \left\{ \begin{array}{l} \text{new packages} \\ \text{internal} \\ \text{new packages} + \text{internal} \\ \text{deletion} \end{array} \right\}$$
- Reporting fit with piecewise function here

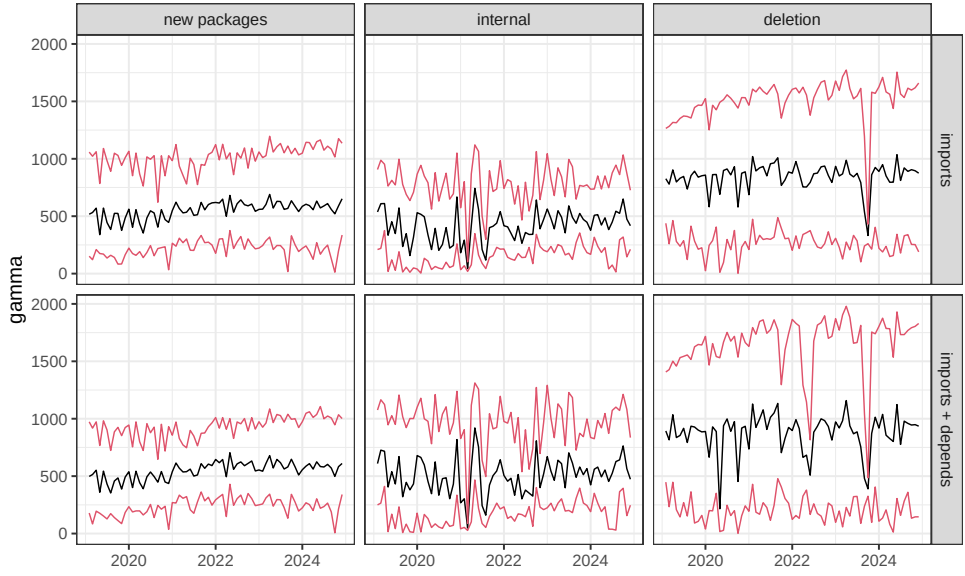
α, piecewise function



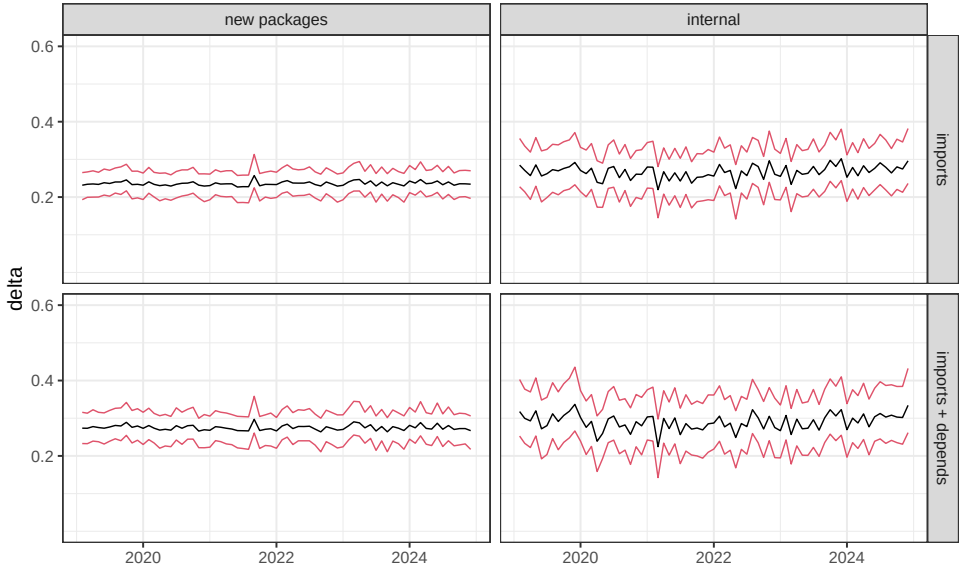
β, piecewise function



γ, piecewise function



δ, piecewise function



Overall

- Parameters stable over the observation period
- Hierarchical model generally in line with single fit
- Adding edges: Partial superlinear preferential attachment (up to γ)
- Deleting edges: Sublinear preferential detachment without zero-appeal (δ)
- Not much difference between imports and imports + depends

Summary

- Poisson regression model for increments of network evolution
- Piecewise weight function of (in-)degree
  - accommodates flexible heavy tail (of the degree distribution)
  - reveals partial super-/sub-linear preferential attachment
- Hierarchical model applied to CRAN package dependencies

Next step

- Model selection between power & piecewise functions

Bibliography

Barabási, Albert-László, and Réka Albert. 1999. "Emergence of Scaling in Random Networks." *Science* 286 (5439): 509–12. https://doi.org/10.1126/science.286.5439.509.

Broido, A. D., and A. Clauset. 2019. "Scale-Free Networks Are Rare." *Nature Communications* 10 (1017). https://doi.org/10.1038/s41467-019-08746-5.

Hofstad, Remco van der. 2016. "Generalized Random Graphs." In *Random Graphs and Complex Networks*, 183–215. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. https://doi.org/10.1017/9781316779422.009.

Lee, Clement, Emma F Eastoe, and Aiden Farrell. 2024. "Degree Distributions in Networks: Beyond the Power Law." *Statistica Neerlandica*, 1–17. https://doi.org/10.1111/stan.12355.

Voitalov, Ivan, Pim van der Hoorn, Remco van der Hofstad, and Dmitri Krioukov. 2019. "Scale-Free Networks Well Done." *Phys. Rev. Res.* 1 (3): 033034. https://doi.org/10.1103/PhysRevResearch.1.033034.