ERPM — Avancement Wrapper Et Nouveaux Effets

Jérémie Chichignoud (CUB'iTECH)

26 novembre 2025

Table des matières

État d'avancement : wrapper `erpm` et statistiques/effets

Objectifs du travail

La demande initiale est de doter le package ERPM d'un `wrapper` capable d'estimer des modèles statistiques définis sur des **partitions**, en réutilisant le moteur d'estimation de `ergm`. L'enjeu est double :

- fournir une interface simple `erpm(partition ~ effets)` ;
- rendre compatibles les statistiques définies sur les groupes avec l'infrastructure d'ERGM (changestats C, contraintes, propositions MH).

Le travail inclut la définition complète du wrapper, la création des effets ERPM (absents d'ERGM), la traduction automatique vers les effets ERGM existants, la génération des réseaux bipartis, les tests, et la documentation.

—

Contexte

ERGM n'agit que sur des **réseaux**. Les statistiques d'ERPM sont définies sur des **partitions** d'acteurs en groupes. Pour combiner les deux, une partition est transformée en un **réseau biparti** acteur–groupe, ce qui permet à `ergm()` d'être appliqué tel quel, avec des change-stats C dédiées lorsque l'effet n'existe pas.

Le wrapper gère :

- la conversion partition → biparti ;
- le choix automatique des contraintes (`b1part`) et propositions ;
- la traduction des effets ERPM vers ERGM ou vers du code C ;
- l'appel final à `ergm()` avec un contrôle unifié.

L'objectif final est une API simple, stable et modulaire pour enrichir ERGM d'effets orientés "groupes".

—

Travail réalisé

1. Wrapper `erpm()`

Le wrapper est maintenant complet et robuste. Il assure :

- normalisation de l'entrée (partition, réseau biparti, formules, covariables, dyadic attributes) ;
- reconstruction propre du réseau biparti (labels acteurs/groupes, tailles, alignement des matrices) ;
- ajout automatique des contraintes et propositions adaptées au biparti ;
- traduction systématique des effets ERPM :
  - effet direct → effet ERGM existant ;
  - effet partiel → ERGM + pré/post-traitement ;
  - effet spécifique → appel à `InitErgmTerm()` + changestat C.
- pipeline d'estimation cohérent (contrôles par défaut, MPLE/MCMLE/CD unifiés, logs).

Le wrapper gère aussi les matrices dyadiques externes via un constructeur biparti interne réutilisable.

2. Implémentation des effets ERPM

Les effets suivants sont complètement implémentés, testés et intégrés :

- **squared_sizes** + init R + changestat C un-toggle + selftests et MWE.
- **groups** (via `b2degrange`).
- **cliques(k)** normalisé ou non ; prise en charge de $k \geq 1$.
- **cliques_GW(λ)** pondération géométrique.
- **cov_ingroup**.
- **log_factorial_sizes**.
- **cov_diff**, **cov_diff_GW**.
- **cov_fulldiff**.
- **cov_match**, **cov_match_GW**.
- **cov_fullmatch**.
- **dyadcov**, **dyadcov_full**, **dyadcov_GW**.

Pour chaque effet, le pipeline complet est livré :

- initialiseur R (vérifications strictes, INPUT_PARAM compact) ;
- changestat C (un-toggle, recomptage local efficace) ;
- MWE minimal pour validation manuelle ;
- selftest autonome pour summary, traduction, et fits MLE/CD courts.

3. Tests et validations

Une suite étendue de tests couvre :

- reconstruction bipartie ;
- traduction ERPM→ERGM ;
- validation analytique (cliques, factorial, diff, match) ;
- comportements limites (groupes vides, covariées, dyadic failures) ;
- fits MLE et CD courts avec détection automatique des cas dégénérés.

Chaque effet dispose d'un selftest isolé.

4. Documentation

- blocs `roxygen2` et `doxygen` complets ;
- documentation LaTeX structurée : tableau des effets, définitions mathématiques, explication effet par effet ;

En l'état, le wrapper est abouti et l'ensemble du socle d'effets ERPM est implémenté, testé et documenté. Le projet est stabilisé et prêt pour l'intégration de nouveaux modules.

Tableau de correspondance des effets

Effet ERPM	**Correspondance ERGM**	**Alias/Termes ERGM**	**Si aucune : `InitErgmTerm` + C**	**Commentaire**	**Status**
`groups(from,to)`	**Directe**	`b2degrange(from,to)`	—	Taille des groupes = degré (mode 2).	✓
`squared_sizes(from,to,pow)`	**ErgmTerm**	—	`InitErgmTerm.squared_sizes` `c_squared_sizes`	Somme des tailles des groupes (mode 2) élevées à la puissance `pow`.	✓
`log_factorial_sizes`	**ErgmTerm**	—	`InitErgmTerm.log_factorial_sizes` `c_log_factorial_sizes`	Somme, sur les groupes du mode 2, des $\log(n_g!)$ où n_g est la taille de chaque groupe.	✓
`cliques(clique_size,normalized)`	**ErgmTerm**	—	`InitErgmTerm.cliques` `c_cliques`	Compte, pour chaque groupe du mode 2, le nombre de k-cliques d'acteurs qu'il induit dans la projection 1-mode ; chaque groupe de taille n_g contribue $\binom{n_g}{k}$.	✓
`cliques_GW(lambda)`	**ErgmTerm**	—	`InitErgmTerm.cliques_GW` `c_cliques_GW`	Version géométriquement pondérée de `cliques` : chaque groupe de taille n_g contribue $S(n_g,\lambda) = \lambda\,[1-((\lambda-1)/\lambda)^{n_g}]$.	✓
`cov_ingroup(cov,size,category)`	**ErgmTerm**	—	`InitErgmTerm.cov_ingroup` `c_cov_ingroup`	Mesure, pour chaque groupe, l'effectif d'acteurs portant l'attribut nodal *catégorie* ciblé, avec filtre optionnel `size` sur les tailles de groupes.	✓
`cov_match(cov)`	**ErgmTerm**	—	`InitErgmTerm.cov_match` `c_cov_match`	Mesure l'homophilie sur une covariée catégorielle en fonction des effectifs par modalité dans chaque groupe (cliques « monochromatiques »).	✓
`cov_match_GW(cov,lambda)`	**ErgmTerm**	—	`InitErgmTerm.cov_match_GW` `c_cov_match_GW`	Version géométriquement pondérée de `cov_match` ; applique un poids contrôlé par $\lambda > 1$ aux contributions liées aux effectifs par modalité dans chaque groupe.	✓
`cov_fullmatch(cov)`	**ErgmTerm**	—	`InitErgmTerm.cov_fullmatch` `c_cov_fullmatch`	Compte les groupes unanimement homogènes sur une covariée catégorielle, avec filtre `size` et catégorie ciblée optionnelle.	✓

Effet ERPM	Correspondance ERGM	Alias/Termes ERGM	Si aucune : `InitErgmTerm` + C	Commentaire	Status
`cov_diff(cov)`	**ErgmTerm**	—	`InitErgmTerm.cov_diff` `c_cov_diff`	Mesure la dispersion intra-groupe d'une covariée numérique via une statistique de type différence entre valeurs d'acteurs, avec filtre optionnel des tailles de groupes.	✓
`cov_diff_GW(cov)`	**ErgmTerm**	—	`InitErgmTerm.cov_diff_GW` `c_cov_diff_GW`	Version géométriquement pondérée de `cov_diff` qui agrège les contributions sur toutes les tailles de cliques $k \geq 2$ via des poids fonction de λ.	✓
`cov_fulldiff(cov)`	**ErgmTerm**	—	`InitErgmTerm.cov_fulldiff` `c_cov_fulldiff`	Mesure, pour chaque groupe, la dispersion d'une covariée numérique via l'écart max − min, avec filtre explicite des tailles de groupes.	✓
`dyadcov(dyadcov,clique_size, normalized)`	**ErgmTerm**	—	`InitErgmTerm.dyadcov` `c_dyadcov`	Pour une matrice dyadique z_{ij} sur le mode acteurs, agrège par groupe les valeurs z_{ij} sur les dyades internes des k-cliques ciblées, avec option de normalisation par groupe.	✓
`dyadcov_full(dyadcov,size)`	**ErgmTerm**	—	`InitErgmTerm.dyadcov_full` `c_dyadcov_full`	Somme, pour chaque groupe, des valeurs z_{ij} d'une matrice dyadique numérique sur toutes les dyades internes $i < j$, avec filtre optionnel sur les tailles de groupes.	✓
`dyadcov_GW(dyadcov,lambda)`	**ErgmTerm**	—	`InitErgmTerm.dyadcov_GW` `c_dyadcov_GW`	Version géométriquement pondérée de `dyadcov` qui combine les contributions des cliques de toutes tailles $k \geq 2$ via un schéma de poids contrôlé par λ.	✓

Explication effet par effet

0. Notations mathématiques

Pour éviter les ambiguïtés, on fixe ici les notations communes à toutes les sous-sections.

Partitions et biparti.

On travaille avec une partition stricte d'acteurs en groupes, représentée par un graphe biparti

$$B = (A, G, E),$$

où :

- $A = \{1, \dots, N_1\}$ est l'ensemble des acteurs (mode 1) ;
- G est l'ensemble des groupes (mode 2) ;
- $E \subseteq A \times G$ est l'ensemble des arêtes d'appartenance : $(i, g) \in E$ signifie que l'acteur i appartient au groupe g.

Pour $g \in G$, on note

$$A(g) = \{\, i \in A : (i, g) \in E \,\}, \qquad n_g = |A(g)|$$

la taille (degré mode 2) du groupe g. Quand on indexe les acteurs par leur groupe unique, on écrit $\gamma(i) \in G$ pour le groupe de l'acteur i.

Crochets d'Iverson et indicatrices.

On utilise systématiquement les *crochets d'Iverson* pour les tests logiques :

$$[P] = \begin{cases} 1, & \text{si la proposition } P \text{ est vraie,} \\ 0, & \text{sinon.} \end{cases}$$

Dans certaines formules, on écrit aussi $\mathbf{1}[P]$ pour la même quantité (fonction indicatrice). Les deux notations sont interchangeables :

$$\mathbf{1}[P] \equiv [P],$$

Tailles admissibles.

Plusieurs effets utilisent un ensemble de tailles de groupes admissibles $S \subset \mathbb{N}_{\geq 1}$. On note alors

$$[\, n_g \in S \,] \quad \text{ou} \quad \mathbf{1}[\, n_g \in S \,]$$

pour indiquer que le groupe g ne contribue que si sa taille appartient à S.

Combinatoire.

On adopte la convention standard pour les coefficients binomiaux :

$$\binom{n}{k} = \begin{cases} \dfrac{n!}{k!\,(n-k)!}, & \text{si } n \geq k \geq 0, \\ 0, & \text{si } n < k \text{ ou } k < 0, \end{cases}$$

de sorte que les groupes de taille $n_g < k$ ne contribuent pas aux sommes sur les k-cliques.

Formule du binôme de Newton.

Pour deux éléments x et y d'un anneau commutatif (par exemple des réels, complexes, polynômes, ou matrices carrées qui commutent, i.e. $xy = yx$), la puissance n-ième de leur somme s'écrit, pour tout entier naturel n,

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k\, y^{n-k} = \sum_{k=0}^{n} \binom{n}{k} x^{n-k}\, y^k,$$

Cette identité est notamment utilisée pour obtenir des formes fermées dans les effets de type `GW` (pondérations géométriques).

Sommes locales vs globales.

Beaucoup d'effets se présentent comme des sommes de contributions locales par groupe :

$$T = \sum_{g \in G} T_g,$$

où T_g dépend typiquement de n_g, d'un attribut agrégé sur g (e.g. somme, max, min) ou de cliques d'acteurs à l'intérieur de g. Les change-statistics sont alors *locales* : un toggle (i, g) ne modifie que les termes associés au groupe g (et éventuellement au groupe quitte par i).

1 `groups(from, to)` : comptage de groupes par taille

Principe.

L'effet `groups(from, to)` compte le nombre de groupes dont la taille appartient à un ensemble de tailles admissibles. Il fournit un contrôle direct sur la distribution des tailles de groupes, indépendamment de tout attribut d'acteur.

Cadre et notations biparties.

On considère un biparti $B = (A, G, E)$ issu d'une partition stricte :

- A : ensemble des acteurs ;
- G : ensemble des groupes ;
- $E \subseteq A \times G$: arcs d'appartenance (i, g).

Pour un groupe $g \in G$, on note sa taille (nombre d'acteurs du groupe)

$$n_g = |g|$$

On fixe un ensemble de tailles admissibles

$$S \subset \mathbb{N}_{\geq 1},$$

Définition formelle.

La statistique associée à un ensemble S est

$$\boxed{T(B; S) = \sum_{g \in G} \mathbf{1}[\, n_g \in S\,],}$$

Cas particuliers et liens.

- Si $S = \mathbb{N}_{\geq 1}$, tous les groupes non vides sont comptés. Cela correspond à l'appel `groups` dans `erpm()`.
- Si $S = \{k\}$ pour un entier $k \geq 1$, seuls les groupes de taille exactement k sont comptés. Cela correspond à `groups(k)`.

- Si l'on fixe deux entiers a, b avec $1 \leq a < b$ et

$$S = \{\, n \in \mathbb{N}_{\geq 1} : a \leq n < b \,\},$$

alors on récupère l'intervalle $[a, b)$ implémenté par `groups(from = a, to = b)` dans `erpm()`. Sur le biparti, cela correspond au terme `b2degrange(from = a, to = b)` appliqué sur le mode des groupes.

Change-stat locale.

Un toggle (i, g) modifie uniquement la taille du groupe g. On note

$$n_g^- \quad \text{la taille avant le toggle,} \qquad n_g^+ \quad \text{la taille après le toggle,}$$

avec $n_g^+ = n_g^- \pm 1$ selon qu'il s'agit d'un ajout ou d'un retrait.

La variation de la statistique s'écrit

$$\boxed{\Delta T(B; S) = \mathbf{1}[\, n_g^+ \in S \,] - \mathbf{1}[\, n_g^- \in S \,],}$$

Le calcul est en temps constant, car il ne dépend que de la taille du seul groupe impacté.

Interprétation.

Un coefficient positif associé à `groups(from, to)` favorise des partitions où un grand nombre de groupes ont une taille dans l'ensemble S (c'est-à-dire dans l'intervalle implicite défini par `from`, `to`). Un coefficient négatif les pénalise.

Exemples `erpm()`.

```
# Tous les groupes non vides  : S = N_{>=1}
erpm(partition ~ groups)

# Groupes de taille exactement 3 : S = {3}
erpm(partition ~ groups(3))

# Intervalle implicite S = { n : 2 <= n < 5 }
erpm(partition ~ groups(from = 2, to = 5))
```

Résumé.

$$\boxed{\texttt{groups(from, to)} : \quad T(B; S) = \sum_{g \in G} \mathbf{1}[\, n_g \in S \,], \quad n_g = |g|,}$$

2 `squared_sizes(from, to, pow)` : somme de puissances de tailles

Principe.

L'effet `squared_sizes` généralise `groups` en remplaçant le simple comptage des groupes par une somme de puissances des tailles. Il permet de contrôler plus finement l'influence des groupes volumineux.

Cadre et notations biparties.

On considère le même biparti $B = (A, G, E)$ issu d'une partition stricte. Pour un groupe $g \in G$, on note sa taille

$$n_g = |g|$$

On fixe :

$$S \subset \mathbb{N}_{\geq 1} \quad \text{(ensemble de tailles admissibles)}, \qquad p > 0 \quad \text{(exposant réel)},$$

Définition formelle.

La statistique associée à un ensemble S et à un exposant p est

$$\boxed{T(B; S, p) = \sum_{g \in G} \mathbf{1}[\, n_g \in S \,]\, n_g^p,}$$

Cas particuliers et liens.

- Si $p = 1$, on obtient la somme des tailles des groupes dont la taille appartient à S.
- Si $p = 2$ (valeur par défaut), on obtient une somme de carrés de tailles, fortement sensible aux grands groupes.
- Dans `erpm()`, les paramètres `from` et `to` définissent implicitement un ensemble de tailles de la forme

$$S = \{\, n \in \mathbb{N}_{\geq 1} : a \leq n < b \,\},$$

et le paramètre `pow` fixe l'exposant p.

Change-stat locale.

Un toggle (i, g) modifie uniquement la taille du groupe g. On note

$$n_g^- \quad \text{la taille avant le toggle,} \qquad n_g^+ \quad \text{la taille après le toggle,}$$

avec $n_g^+ = n_g^- \pm 1$.

La variation de la statistique s'écrit alors

$$\boxed{\Delta T(B; S, p) = \mathbf{1}[\, n_g^+ \in S \,]\,(n_g^+)^p - \mathbf{1}[\, n_g^- \in S \,]\,(n_g^-)^p,}$$

Le calcul est en temps constant, puisqu'il ne dépend que de la taille du seul groupe impacté.

Interprétation.

Plus p est grand, plus la statistique est dominée par les groupes de grande taille. Un coefficient positif avec $p = 2$ favorise des partitions où quelques groupes concentrent beaucoup d'acteurs. Un coefficient négatif pousse au contraire vers des répartitions plus homogènes des tailles.

Exemples `erpm()`.

```
# Somme des carrés pour tous les groupes : S = N_{>=1}, p = 2
erpm(partition ~ squared_sizes)

# Somme des cubes pour tailles dans S = { n : 2 <= n < 5 }
erpm(partition ~ squared_sizes(from = 2, to = 5, pow = 3))

# Vectorisation sur deux ensembles de tailles
erpm(partition ~ squared_sizes(from = c(1, 4),
                               to   = c(4, Inf),
                               pow  = c(2, 2)))
```

Résumé.

`squared_sizes(from,to,pow)` :	$T(B; S, p) = \sum_{g \in G} \mathbf{1}[\, n_g \in S \,]\, n_g^p, \quad n_g = \lvert g \rvert,$

3 `log_factorial_sizes` : logs de factorielles de tailles

Principe.

L'effet `log_factorial_sizes` associe à chaque groupe une quantité combinatoire basée sur le logarithme du factoriel décalé de sa taille, puis somme ces contributions sur l'ensemble des groupes.

Cadre et notations biparties.

On considère un biparti $B = (A, G, E)$ issu d'une partition stricte. Pour tout groupe $g \in G$, on note

$$n_g = |g|,$$

Définition formelle.

La statistique globale est

$$\boxed{T(B) = \sum_{g \in G} \log\big((n_g - 1)!\big),}$$

Interprétation combinatoire.

Le terme $(n_g - 1)!$ correspond au nombre de permutations circulaires possibles des n_g membres d'un groupe. Ainsi,

$$T(B) = \log\left(\prod_{g \in G} (n_g - 1)!\right)$$

mesure la richesse combinatoire totale de la structure interne des groupes.

Cas asymptotiques.

En utilisant l'approximation stirlingienne

$$\log((n-1)!) \approx (n - \tfrac{1}{2}) \log n - n + C,$$

on obtient

$$T(B) \approx \sum_{g \in G} \big[(n_g - \tfrac{1}{2}) \log n_g - n_g\big] + C',$$

Pour une somme totale $\sum_g n_g$ fixée, cette forme montre que $T(B)$ est minimale lorsque les tailles (n_g) sont aussi égales que possible et augmente lorsque la distribution des tailles devient plus déséquilibrée.

Change-stat locale.

Un toggle (i, g) modifie uniquement la taille du groupe g. On note

$$n_g^- \quad \text{avant le toggle,} \qquad n_g^+ \quad \text{après,} \qquad n_g^+ = n_g^- \pm 1,$$

La variation locale est donc

$$\boxed{\Delta T = \log\big((n_g^+ - 1)!\big) - \log\big((n_g^- - 1)!\big),}$$

Liens comparatifs.

Statistique	Croissance en n_g
`groups`	constante
`squared_sizes`	n_g^p
`log_factorial_sizes`	$n_g \log n_g$

Appel typique dans `erpm()`.

```
erpm(partition ~ log_factorial_sizes)
```

Résumé.

$$\boxed{\texttt{log\_factorial\_sizes}(B) = \sum_{g \in G} \log((n_g - 1)!), \quad n_g = |g|,}$$

4 `cliques(k, normalized)` : k-cliques sur la projection 1-mode

Principe.

L'effet `cliques` compte les k-cliques d'acteurs induites par les groupes de la partition dans la projection 1-mode. Il capture la concentration d'acteurs dans des groupes suffisamment grands pour générer des cliques d'ordre k.

Cadre et notations biparties.

On part d'un biparti $B = (A, G, E)$ associé à une partition stricte. Pour chaque groupe $g \in G$, on note

$$A(g) = \{\, i \in A : (i, g) \in E \,\}, \qquad n_g = |A(g)|, \qquad N_1 = |A|,$$

La projection sur le mode des acteurs relie deux acteurs s'ils partagent au moins un groupe.

Définition formelle.

Pour chaque groupe g, le nombre de k-cliques d'acteurs qu'il engendre dans la projection est $\binom{n_g}{k}$ (avec convention $\binom{n}{k} = 0$ si $n < k$). La statistique non normalisée est

$$\boxed{C_k(p) \;=\; \sum_{g \in G} \binom{n_g}{k},}$$

Une version *normalisée globalement* par le maximum possible sur le mode acteurs est

$$\boxed{C_k^{\mathrm{norm}}(p) \;=\; \binom{N_1}{k}^{-1} \sum_{g \in G} \binom{n_g}{k},}$$

Cas particuliers et liens ERGM.

- $k = 1$: $C_1(p) = \#\{\, g \in G : n_g = 1 \,\}$, *le nombre de groupes de taille 1* (ce n'est **pas** $\sum_g n_g = N_1$, qui est constant).
- $k = 2$: nombre de paires d'acteurs co-groupés $\sum_g \binom{n_g}{2}$, égal au nombre d'arêtes de la projection 1-mode lorsque chaque paire n'est reliée que par co-appartenance.

Il n'existe pas d'effet `ergm` standard qui reproduise directement C_k pour $k > 2$; d'où une implémentation dédiée (`InitErgmTerm.cliques` + change-stat C native).

Change-stat locale (un-toggle).

Un toggle (i, g) ne modifie que la taille d'un seul groupe g. Si n_g^- et n_g^+ désignent respectivement les tailles avant et après le toggle, la variation non normalisée est

$$\boxed{\Delta C_k \;=\; \binom{n_g^+}{k} - \binom{n_g^-}{k},}$$

Forme pièce-par-pièce utilisée en C :

$$\Delta C_k = \begin{cases} +1 & \text{si } k = 1 \text{ et } (n_g^- {=} 0 \to 1) \text{ ou } (n_g^- {=} 2 \to 1), \\ -1 & \text{si } k = 1 \text{ et } (n_g^- {=} 1 \to 0) \text{ ou } (n_g^- {=} 1 \to 2), \\ \binom{n_g^-}{k-1} & \text{si } k \geq 2 \text{ et ajout}, \\ -\binom{n_g^- - 1}{k-1} & \text{si } k \geq 2 \text{ et retrait}, \\ 0 & \text{sinon.} \end{cases}$$

La version normalisée multiplie systématiquement ΔC_k par $\binom{N_1}{k}^{-1}$.

Interprétation.

Pour $k \geq 2$, `cliques(k)` mesure à quel point les acteurs sont concentrés dans des groupes capables de générer de nombreuses cliques d'ordre k. Un coefficient positif favorise des groupes volumineux ; un coefficient négatif les décourage. La normalisation corrige l'effet mécanique de N_1.

Appels typiques dans `erpm()`.

```
# k par défaut = 2, non normalisé
erpm(partition ~ cliques())

# k explicite
erpm(partition ~ cliques(k = 3))

# Forme abrégée (argument positionnel)
erpm(partition ~ cliques(4))
```

```
# Normalisation globale par C(N1, k)
erpm(partition ~ cliques(k = 2, normalized = TRUE))
```

Résumé.

$$\texttt{cliques(k)}: \quad C_k(p) = \sum_g \binom{n_g}{k}, \qquad C_k^{\mathrm{norm}}(p) = \binom{N_1}{k}^{-1} C_k(p),$$

5 `cliques_GW(lambda)` : cliques pondérées géométriquement

Principe.

L'effet `cliques_GW` lisse le comptage discret des `cliques(k)` en combinant les contributions de toutes les tailles de cliques $k \geq 1$ via une série géométrique contrôlée par $\lambda > 1$.

Cadre et notations biparties.

On considère le même biparti $B = (A, G, E)$ et les tailles de groupes n_g. Pour chaque $k \geq 1$, on rappelle

$$c_k(P) = \sum_{g \in G} \binom{n_g}{k},$$

Définition formelle.

La version « suite géométrique » est

$$T(\lambda) = \sum_{k \geq 1} \Big(-\tfrac{1}{\lambda} \Big)^{k-1} c_k(P),$$

En inversant les sommes, on obtient une forme par groupe :

$$T(\lambda) = \sum_{g \in G} \sum_{k=1}^{n_g} \Big(-\tfrac{1}{\lambda} \Big)^{k-1} \binom{n_g}{k},$$

On définit la contribution individuelle

$$S(n_g, \lambda) = \sum_{k=1}^{n_g} \Big(-\tfrac{1}{\lambda} \Big)^{k-1} \binom{n_g}{k},$$

et la statistique globale

$$\boxed{T(\lambda) = \sum_{g \in G} S(n_g, \lambda),}$$

Forme fermée.

En posant $r = -1/\lambda$ et en appliquant le binôme de Newton, on obtient

$$S(n_g, \lambda) = \frac{(1+r)^{n_g} - 1}{r} = \lambda \Big[1 - \Big(\tfrac{\lambda - 1}{\lambda} \Big)^{n_g} \Big],$$

Donc

$$\boxed{T(\lambda) = \sum_{g \in G} \lambda \Big[1 - \Big(\tfrac{\lambda - 1}{\lambda} \Big)^{n_g} \Big],}$$

Cas particuliers.

- $\lambda = 1$ (limite) : $S(n_g, 1) = 1$ pour tout groupe non vide. La statistique compte simplement le nombre de groupes.
- $\lambda \to \infty$: $(1 - 1/\lambda)^{n_g} \approx 1 - n_g/\lambda$, d'où $S(n_g, \lambda) \approx n_g/\lambda$.

Change-stat locale.

Un toggle (i, g) modifie une seule taille n_g. On note n_g^- avant, $n_g^+ = n_g^- \pm 1$ après. La variation est

$$\boxed{\Delta T(\lambda) = S(n_g^+, \lambda) - S(n_g^-, \lambda),}$$

Interprétation.

$T(\lambda)$ résume dans une même statistique toutes les `cliques(k)` pour $k \geq 1$, avec un poids géométrique $(-1/\lambda)^{k-1}$.

Résumé.

$$\boxed{T(\lambda) = \sum_{g \in G} \lambda \Big[1 - \Big(\tfrac{\lambda - 1}{\lambda} \Big)^{n_g} \Big], \quad \lambda > 1,}$$

6 `cov_ingroup` : poids intra-groupe d'une covariée

Principe.

L'effet `cov_ingroup` agrège, au sein de chaque groupe, la somme d'un attribut d'acteur multipliée par la taille du groupe. Il pondère donc les acteurs porteur de la covariée par l'ampleur du groupe.

Cadre et notations biparties.

On considère $B = (A, G, E)$ et un attribut d'acteur $x = (x_i)_{i \in A}$ (numérique ou indicatrice d'une modalité). Chaque acteur i appartient à un groupe unique $g(i)$, de taille $n_{g(i)}$. On choisit un ensemble de tailles admissibles $S \subset \mathbb{N}_{\geq 1}$.

Définition formelle.

En notation par acteur :

$$\boxed{T_{\text{in}}(B; x, S) = \sum_{i \in A} x_i \, n_{g(i)} \, \mathbf{1}[\, n_{g(i)} \in S \,],}$$

En notation par groupe, avec $X_g = \sum_{i \in g} x_i$,

$$T_{\text{in}}(B; x, S) = \sum_{g \in G} n_g X_g \, \mathbf{1}[\, n_g \in S \,],$$

Cas particuliers et liens.

- Sans filtre ($S = \mathbb{N}$) et $x_i \equiv 1$, on obtient $\sum_g n_g^2$, proche de `squared_sizes`.
- Si x est binaire $\mathbf{1}[c_i = \kappa]$, $\sum_{i \in g} x_i$ est l'effectif d'une modalité ciblée.

Change-stat locale.

On note $X_g = \sum_{j \in g} x_j$. Un toggle (i, g) déplace éventuellement i entre deux groupes a et b (cas général « $a \to b$ »). Avant déplacement :

$$n_a, n_b, \quad X_a, X_b,$$

Après :

$$n'_a = n_a - 1, \quad X'_a = X_a - x_i; \qquad n'_b = n_b + 1, \quad X'_b = X_b + x_i,$$

La variation est

$$\boxed{\Delta T_{\text{in}} = n'_b X'_b \mathbf{1}[\, n'_b \in S \,] - n_b X_b \mathbf{1}[\, n_b \in S \,] + n'_a X'_a \mathbf{1}[\, n'_a \in S \,] - n_a X_a \mathbf{1}[\, n_a \in S \,],}$$

Sans filtre ($S = \mathbb{N}$), cette expression se simplifie en

$$\Delta T_{\text{in}} = (X_b - X_a) + (n_b - n_a + 2)\, x_i,$$

Interprétation.

`cov_ingroup` est sensible à la fois à la concentration de la covariée dans un groupe (X_g) et à la taille n_g. Un coefficient positif favorise de grands groupes fortement dotés en x, un coefficient négatif les pénalise.

Exemples `erpm()`.

```
1  # Covariée numérique
2  erpm(partition ~ cov_ingroup("age", size = 2:3), nodes = nodes
       )
3
4  # Covariée catégorielle ciblée (binaire implicite)
5  erpm(partition ~ cov_ingroup("dept", category = "A", size = 3)
       ,nodes = nodes)
```

Résumé.

$$\boxed{\texttt{cov\_ingroup}: \quad T(B;x,S)=\sum_{g} n_g\Big(\sum_{i\in g} x_i\Big)\mathbf{1}[\,n_g\in S\,],}$$

7 `cov_fullmatch` : groupes unanimement homogènes

Principe.

L'effet `cov_fullmatch` compte les groupes dont tous les membres partagent la même modalité d'une covariée catégorielle, éventuellement restreints à un ensemble de tailles et/ou à une modalité ciblée.

Cadre et notations biparties.

On considère une partition $B=(A,G,E)$ et un attribut catégoriel $c=(c_i)_{i\in A}$ à valeurs dans un ensemble fini L. Pour un groupe g, on note

$$n_g=|g|, \qquad n_{g,r}=|\{\, i\in g : c_i=r\,\}|,$$

On fixe un ensemble de tailles admissibles $S\subset\mathbb{N}_{\geq 1}$.

Définition formelle.

Un groupe est homogène s'il existe une modalité $r\in L$ telle que $n_{g,r}=n_g$. La statistique générale est

$$\boxed{T(B;c,S)=\sum_{g\in G}\mathbf{1}[\,n_g\in S\,]\,\mathbf{1}\big[\,\exists r\in L : n_{g,r}=n_g\,\big],}$$

Variante avec modalité cible (`category = κ`) :

$$\boxed{T_\kappa(B;c,S)=\sum_{g\in G}\mathbf{1}[\,n_g\in S\,]\,\mathbf{1}[\,n_{g,\kappa}=n_g\,],}$$

Cas particuliers et liens.

- Si $S=\{1\}$, tous les singletons sont comptés (valeur de c quelconque).
- Sur la projection sur le mode des acteurs, l'homogénéité totale d'un groupe équivaut à l'égalité $\sum_r \binom{n_{g,r}}{2} = \binom{n_g}{2}$, c'est-à-dire « toutes les paires concordent sur c ».

Change-stat locale.

Un toggle (i,g) modifie uniquement deux quantités pour le groupe g :

- sa taille n_g ;
- les compteurs par modalité $(n_{g,r})_{r\in L}$, en particulier celui de la modalité de l'acteur i, notée $r^*=c_i$.

Avant le toggle, l'indicateur d'homogénéité vaut

$$F_g=\mathbf{1}[\,n_g\in S\,]\,\mathbf{1}\big[\,\exists r\in L : n_{g,r}=n_g\,\big],$$

Après mise à jour de n_g et de n_{g,r^*}, on recalcule

$$F'_g=\mathbf{1}[\,n'_g\in S\,]\,\mathbf{1}\big[\,\exists r\in L : n'_{g,r}=n'_g\,\big],$$

La variation due au toggle est donc simplement

$$\Delta T=F'_g-F_g,$$

Pour la version sans filtre sur une modalité, la variation s'écrit :

$$\boxed{\Delta T=F'_g-F_g}$$

pour la version avec filtre sur une modalité, la variation s'écrit :

$$\boxed{\Delta T_\kappa=F_g^{(\kappa)\prime}-F_g^{(\kappa)}, \quad F_g^{(\kappa)}=\mathbf{1}[\,n_g\in S\,]\mathbf{1}[\,n_{g,\kappa}=n_g\,]}$$

Interprétation.

`cov_fullmatch` favorise ou pénalise la création de groupes entièrement homogènes sur une covariée.

Exemples `erpm()`.

```
# Tous les groupes homogènes
erpm(partition ~ cov_fullmatch("dept"))

# Filtre de tailles
erpm(partition ~ cov_fullmatch("dept", size = 2:3))

# Variante ciblée : unanimité sur "A"
erpm(partition ~ cov_fullmatch("dept", category = "A", size =
    2:3))
```

Résumé.

$$\texttt{cov\_fullmatch}: \quad T(B;c,S) = \sum_{g} \mathbf{1}[n_g \in S]\mathbf{1}[\exists r : n_{g,r} = n_g],$$

8 `cov_match(k)` : homophilie discrète par cliques

Principe.

`cov_match(k)` mesure l'homogénéité d'un attribut catégoriel à un ordre de clique fixé k. Pour chaque groupe, il compte les sous-ensembles de k acteurs partageant la même modalité.

Cadre et notations biparties.

On reprend $B = (A, G, E)$ et un attribut catégoriel $c = (c_i)$. Pour chaque groupe g et modalité r, $n_{g,r}$ est l'effectif de la modalité r dans g.

Définition formelle.

Pour un ordre $k \geq 1$,

$$S_k(B;c) = \sum_{g \in G} \sum_{r \in L} \binom{n_{g,r}}{k},$$

Variante ciblée (`category = κ`) :

$$S_k^{(\kappa)}(B;c) = \sum_{g \in G} \binom{n_{g,\kappa}}{k},$$

Cas particuliers et projection sur le mode des acteurs.

- $k = 1$: $S_1(B;c) = N$, nombre total d'acteurs.
- $k = 2$:

$$S_2(B;c) = \sum_{i<j} \mathbf{1}[c_i = c_j]\, w_{ij},$$

où w_{ij} est le nombre de groupes communs à (i, j). Sur la projection sur le mode des acteurs simple ($w_{ij} \in \{0, 1\}$), cela correspond à un `nodematch` pondéré par co-appartenance.

Change-stat locale.

Un toggle (i, g) modifie l'appartenance de l'acteur i au groupe g. On note $r^* = c_i$ la modalité de i. Ce toggle n'affecte que deux quantités dans le groupe g :

- la taille du groupe n_g (qui augmente ou diminue de 1) ;
- le compteur de la modalité ciblée n_{g,r^*} (nombre d'acteurs de modalité r^* dans g).

Avant le toggle, on pose

$$m = n_{g,r^*},$$

Deux cas se présentent :

- **Ajout** : $i \notin g \to i \in g$. Le compteur devient $m' = m + 1$. Les nouvelles k-cliques monochromatiques créées dans g sont exactement celles qui utilisent l'acteur ajouté i et $k-1$ des m acteurs déjà présents avec la même modalité. D'où la contribution
$$\Delta S_k = \binom{m}{k-1},$$
- **Retrait** : $i \in g \to i \notin g$. Le compteur devient $m' = m - 1$. Toutes les k-cliques monochromatiques qui disparaissent étaient formées d'acteur i et de $k-1$ des $m-1$ acteurs restants de modalité r^*. Cela retire donc
$$\Delta S_k = -\binom{m-1}{k-1},$$

On adopte la convention $\binom{n}{t} = 0$ pour $n < t$, ce qui couvre naturellement les cas où il n'existe pas assez d'acteurs de modalité r^* pour former une k-clique.

Normalisations possibles.

Trois variantes :

$$S_k^{\text{none}} = \sum_{g,r} \binom{n_{g,r}}{k}, \quad S_k^{\text{by\_group}} = \sum_g \frac{\sum_r \binom{n_{g,r}}{k}}{\binom{n_g}{k}}, \quad S_k^{\text{global}} = \frac{\sum_{g,r} \binom{n_{g,r}}{k}}{\binom{N}{k}},$$

Interprétation.

Un coefficient positif fait émerger des groupes où les modalités sont fortement concentrées (beaucoup de cliques monochromatiques d'ordre k). Plus k est grand, plus la statistique est sensible aux groupes massivement homogènes.

Exemples `erpm()`.

```
# Paires monochromatiques
erpm(partition ~ cov_match("dept", clique_size = 2))

# Cliques d'ordre 3, normalisation par groupe
erpm(partition ~ cov_match("dept", clique_size = 3,
                           normalized = "by_group"))

# Modalité ciblée
erpm(partition ~ cov_match("dept", clique_size = 2,
                           category = "A"))
```

Résumé.

`cov_match(k)` :	$S_k(B;c) = \sum_{g,r} \binom{n_{g,r}}{k}$,	cliques monochromatiques d'ordre k,

9 `cov_match_GW(lambda)` : homophilie géométriquement pondérée

Principe.

`cov_match_GW` est une extension lissée de `cov_match(k)`. Elle agrège l'homophilie sur tous les ordres de cliques via une série géométrique contrôlée par $\lambda > 1$.

Cadre et notations biparties.

Même cadre $B = (A, G, E)$, covariée catégorielle c, effectifs $n_{g,r}$. On note
$$S_k(B;c) = \sum_{g,r} \binom{n_{g,r}}{k}$$
la statistique d'ordre k (voir ci-dessus).

Définition formelle.

La définition sous forme "*série géométrique*" est

$$S_{\mathrm{GW}}(B;c,\lambda)=\sum_{n\geq 1}\Big(-\tfrac{1}{\lambda}\Big)^{n-1}\sum_{g\in G}\sum_{r\in L}\binom{n_{g,r}}{n},$$

En sommant sur n et en posant $r_\lambda=(\lambda-1)/\lambda$, on obtient une forme fermée par groupe et modalité :

$$\boxed{S_{\mathrm{GW}}(B;c,\lambda)=\sum_{g\in G}\sum_{r\in L}\lambda\Big[1-r_\lambda^{\,n_{g,r}}\Big],\qquad r_\lambda\in[0,1),}$$

Lien avec `cov_match`.

On a

$$S_{\mathrm{GW}}(B;c,\lambda)=\sum_{k\geq 1}\Big(-\tfrac{1}{\lambda}\Big)^{k-1}S_k(B;c),$$

soit une combinaison géométrique de tous les `cov_match(k)`. Il ne peut pas être reproduit en général par une somme finie d'effets `cov_match(k)` avec coefficients libres.

Change-stat locale.

Un toggle (i,g) modifie uniquement l'effectif de la modalité $r^*=c_i$ dans le groupe g. Avant le toggle, on note

$$m=n_{g,r^*},\qquad n_g=\sum_{r\in L}n_{g,r},$$

Lors d'un *ajout* $(i\notin g\to i\in g)$:
- l'effectif de la modalité r^* devient $m'=m+1$;
- la contribution du couple (g,r^*) passe de $\lambda(1-r_\lambda^{\,m})$ à $\lambda(1-r_\lambda^{\,m'})$;
- le reste du groupe n'est pas affecté.

La variation associée à ce seul couple (g,r^*) est donc

$$\lambda\big[(1-r_\lambda^{\,m+1})-(1-r_\lambda^{\,m})\big]=\lambda(r_\lambda^{\,m}-r_\lambda^{\,m+1})=r_\lambda^{\,m},$$

Lors d'un *retrait* $(i\in g\to i\notin g)$:
- l'effectif devient $m'=m-1$;
- on passe de $\lambda(1-r_\lambda^{\,m})$ à $\lambda(1-r_\lambda^{\,m-1})$.

La variation est alors

$$\lambda\big[(1-r_\lambda^{\,m-1})-(1-r_\lambda^{\,m})\big]=-\,r_\lambda^{\,m-1},$$

En combinant les deux cas, la change-stat locale s'écrit

$$\boxed{\Delta S_{\mathrm{GW}}=\begin{cases}r_\lambda^{\,m}, & \text{ajout},\\ -\,r_\lambda^{\,m-1}, & \text{retrait},\end{cases}}$$

La mise à jour ne dépend que de la modalité r^* et de son effectif local, ce qui garantit une complexité $\mathcal{O}(1)$, indépendante de la taille du groupe ou de $|L|$.

Normalisations.

Trois variantes disponibles :

$$S_{\mathrm{GW}}^{\mathrm{none}}=\sum_{g,r}\lambda(1-r_\lambda^{\,n_{g,r}}),$$

$$S_{\mathrm{GW}}^{\mathrm{by\_group}}=\sum_{g}\frac{\sum_r\lambda(1-r_\lambda^{\,n_{g,r}})}{\lambda(1-r_\lambda^{\,n_g})},$$

$$S_{\mathrm{GW}}^{\mathrm{global}}=\frac{\sum_{g,r}\lambda(1-r_\lambda^{\,n_{g,r}})}{\lambda(1-r_\lambda^{\,N})},$$

Interprétation.

λ grand focalise la statistique sur les petits ordres de cliques, λ proche de 1 renforce la contribution des grandes cliques. La forme fermée assure une change-stat locale et stable.

Exemples `erpm()`.

```
# Forme de base
erpm(partition ~ cov_match_GW("dept", lambda = 2))

# Normalisation et modalité ciblée
erpm(partition ~ cov_match_GW("dept", lambda = 3,
                              category   = "A",
                              normalized = "by_group"))
```

Résumé.

$$\texttt{cov\_match\_GW}(\lambda): \quad \sum_{g,r} \lambda\big[1 - r_\lambda^{n_{g,r}}\big], \quad r_\lambda = (\lambda - 1)/\lambda,$$

10 `cov_fulldiff` : dispersion max–min par groupe

Principe.

`cov_fulldiff` mesure, pour chaque groupe, l'écart maximal d'une covariée numérique entre ses membres, puis somme ces écarts sur les groupes dont la taille appartient à un ensemble donné.

Cadre et notations biparties.

On considère une partition $B = (A, G, E)$ et un attribut numérique $x = (x_i)_{i \in A}$.

Pour chaque groupe g,

$$x_g^{\min} = \min_{i \in g} x_i, \quad x_g^{\max} = \max_{i \in g} x_i, \quad D_g = x_g^{\max} - x_g^{\min},$$

Par convention, $D_g = 0$ si $n_g \leq 1$. On fixe un ensemble de tailles admissibles $S \subset \mathbb{N}_{\geq 1}$.

Définition formelle.

$$T_{\text{fulldiff}}(B; x, S) = \sum_{g \in G} D_g \, \mathbf{1}[\, n_g \in S\,] = \sum_{g \in G} (x_g^{\max} - x_g^{\min}) \mathbf{1}[\, n_g \in S\,],$$

Lien avec `absdiff`.

Sur la projection du mode des acteurs, un `absdiff` classique calculerait

$$T_{\text{absdiff},g} = \sum_{i<j,\, i,j \in g} |x_i - x_j|,$$

En général, il n'existe pas de relation linéaire simple entre D_g et $T_{\text{absdiff},g}$. `cov_fulldiff` n'est donc pas réductible à un empilement de `absdiff`.

Change-stat locale.

On considère un déplacement d'un acteur i d'un groupe a vers un groupe b (ou un ajout/retrait particulier). Avant :

$$n_a, n_b, \quad D_a, D_b,$$

Après :

$$n'_a, n'_b, \quad D'_a, D'_b$$

avec réactualisation locale des min/max. La variation générale est

$$\Delta T_{\text{fulldiff}} = D'_b \mathbf{1}[\, n'_b \in S\,] - D_b \mathbf{1}[\, n_b \in S\,] + D'_a \mathbf{1}[\, n'_a \in S\,] - D_a \mathbf{1}[\, n_a \in S\,],$$

Interprétation.

`cov_fulldiff` valorise les groupes où l'attribut s'étale fortement, et ignore ceux dont la taille n'appartient pas à S. Un coefficient positif pousse vers des groupes hétérogènes, un coefficient négatif favorise des groupes plus homogènes.

Appels `erpm()`.

```
# Dispersion max-min pour groupes de taille >= 2
erpm(partition ~ cov_fulldiff("age"))

# Filtre explicite de tailles
erpm(partition ~ cov_fulldiff("age", size = 3:5))
```

Résumé.

$$\texttt{cov\_fulldiff}: \quad T(B;x,S)=\sum_{g}(x_g^{\max}-x_g^{\min})\,\mathbf{1}[\,n_g\in S\,],$$

11 `cov_diff` : dispersion dans les k-cliques

Principe.

`cov_diff` mesure l'hétérogénéité d'un attribut numérique dans toutes les k-cliques d'acteurs co-groupés. Pour chaque clique, on considère la différence max–min de l'attribut.

Cadre et notations biparties.

On considère une partition stricte d'acteurs A en groupes G et un attribut numérique $x=(x_i)$. Pour un groupe g, on note $\mathcal{C}_k(g)$ l'ensemble des sous-ensembles $S\subset g$ de taille k (éventuellement vide si $n_g<k$). On définit

$$D(S)=\max_{i\in S}x_i-\min_{i\in S}x_i,$$

Définition formelle.

$$T_k(B;x)=\sum_{g\in G}\sum_{S\in\mathcal{C}_k(g)}D(S),$$

Cas particulier $k=2$ et lien avec `absdiff`.

Pour $k=2$, chaque clique est une paire $\{i,j\}$ et

$$T_2(B;x)=\sum_{g\in G}\sum_{\substack{i<j\\ i,j\in g}}|x_i-x_j|,$$

En introduisant $\gamma(i)$ le groupe de i, on obtient

$$T_2(B;x)=\sum_{i<j}\mathbf{1}[\gamma(i)=\gamma(j)]\,|x_i-x_j|,$$

Sur la projection du mode des acteurs, ceci coïncide avec `absdiff(cov)` appliqué à x.

Normalisation par groupe.

Une version normalisée corrige l'effet de $\binom{n_g}{k}$:

$$T_k^{\text{by\_group}}(B;x)=\sum_{g\in G}\frac{1}{\binom{n_g}{k}}\sum_{S\in\mathcal{C}_k(g)}D(S),$$

avec convention $\binom{n_g}{k}=0$ si $n_g<k$.

Change-stat locale.

Un toggle (i,g) modifie uniquement les cliques dont i est un membre dans le groupe g. En effet, toutes les autres cliques du groupe (celles ne contenant pas i) gardent exactement les mêmes ensembles d'acteurs avant et après l'opération. La change-stat est donc entièrement déterminée par les cliques où i apparaît.

Cas $k=2$. Les cliques de taille 2 sont exactement les paires $\{i,j\}$ avec $j\in g$. Avant l'ajout, $i\notin g^-$, donc aucune paire $\{i,j\}$ n'existe. Après l'ajout, pour chaque $j\in g^-$, la paire $\{i,j\}$ apparaît et contribue $|x_i-x_j|$ à la statistique.

Inversement, lors d'un retrait, les paires $\{i,j\}$ disparaissent pour tout $j\in g^+\setminus\{i\}$.

On obtient donc

$$\Delta T_2 = \begin{cases} \displaystyle\sum_{j \in g^-} |x_i - x_j|, & \text{ajout de } i, \\ \displaystyle -\sum_{j \in g^+ \setminus \{i\}} |x_i - x_j|, & \text{retrait de } i, \end{cases}$$

Cas $k > 2$. Une clique de taille k contenant i est de la forme $\{i\} \cup C$, où $C \subset g^-$ est un sous-ensemble de taille $k-1$.

- Lors d'un *ajout*, toute nouvelle clique de taille k est formée en combinant i avec un sous-ensemble $C \subset g^-$ de cardinal $k-1$. Le nombre de nouvelles cliques est donc $\binom{|g^-|}{k-1}$.
- Lors d'un *retrait*, toutes les cliques de la forme $\{i\} \cup C$, $C \subset g^+ \setminus \{i\}$, disparaissent. Leur nombre est $\binom{|g^+|-1}{k-1}$.

Les contributions individuelles dépendent de la statistique considérée (par exemple produit, somme ou agrégation des valeurs x), mais dans tous les cas, la change-stat ne dépend que du contenu du groupe hors et des cliques où i.

Ainsi, pour tout $k > 2$, la variation provient exclusivement des cliques contenant i, et le reste de la structure du groupe est inchangé.

Interprétation.

`cov_diff` est une mesure de dispersion intra-groupe à l'échelle des cliques de taille k. Pour $k = 2$, on retrouve une forme dyadique proche de `absdiff` ; pour $k > 2$, on capture des hétérogénéités sur des sous-ensembles plus larges, sans équivalent direct en termes purement dyadiques.

Exemples `erpm()`.

```
# Dispersion dyadique (k = 2)
erpm(partition ~ cov_diff("age", clique_size = 2))

# Variante normalisée par groupe
erpm(partition ~ cov_diff("age", clique_size = 2,
                          normalized = TRUE))
```

Résumé.

$$\texttt{cov\_diff} : \quad T_k(B; x) = \sum_g \sum_{S \in \mathcal{C}_k(g)} (\max_{i \in S} x_i - \min_{i \in S} x_i),$$

12 `cov_diff_GW(lambda)` : dispersion numérique multi-échelle

Principe.

`cov_diff_GW` étend `cov_diff` en combinant, dans une même statistique, les dispersions à toutes les tailles de cliques $k \geq 2$, pondérées géométriquement par un paramètre $\lambda > 1$.

Cadre et notations biparties.

Pour un groupe g, on note $\mathcal{C}_k(g)$ l'ensemble des sous-ensembles $S \subset g$ de taille k (éventuellement vide si $n_g < k$).

$$D(S) = \max_{i \in S} x_i - \min_{i \in S} x_i,$$

On reprend $B = (A, G, E)$, un attribut numérique x, et la définition de

$$c_k(\mathrm{cov}, p) = \sum_{g \in G} \sum_{S \in \mathcal{C}_k(g)} D(S)$$

comme statistique de base de type `cov_diff` d'ordre k.

Définition formelle.

On note $K_{\max} = \max_{g \in G} n_g$. La statistique `cov_diff_GW` est

$$\boxed{T_{\mathrm{GW}}(p; x; \lambda) = \sum_{k=2}^{K_{\max}} \Big(-\tfrac{1}{\lambda} \Big)^{k-1} c_k(\mathrm{cov}, p), \qquad \lambda > 1,}$$

Les contributions de grande taille k sont amorties par $\lambda^{-(k-1)}$.

Lien avec `cov_diff`.

Pour chaque k, $c_k(\mathrm{cov}, p)$ coïncide avec la version non normalisée de `cov_diff` d'ordre k. En particulier, $c_2(\mathrm{cov}, p)$ est exactement la `cov_diff` dyadique :

$$c_2(\mathrm{cov}, p) = \sum_{g \in G} \sum_{i<j \in g} |x_i - x_j|,$$

`cov_diff_GW` peut ainsi être vu comme une somme pondérée

$$T_{\mathrm{GW}} = \sum_{k \geq 2} (-1/\lambda)^{k-1} T_k(B; x),$$

Change-stat locale.

Un toggle (i, g) n'affecte que les cliques du groupe g qui contiennent i. Toutes les autres cliques conservent les mêmes ensembles d'acteurs.

Pour $k = 2$. Les cliques sont les paires $\{i, j\}$, $j \in g$.

$$\Delta c_2 = \begin{cases} \displaystyle\sum_{j \in g^-} |x_i - x_j|, & \text{ajout}, \\ \displaystyle -\sum_{j \in g^+} |x_i - x_j|, & \text{retrait}, \end{cases}$$

Pour $k \geq 3$. Chaque clique contenant i est de la forme $\{i\} \cup C$ avec $C \subset g$ et $|C| = k - 1$.

$$\Delta c_k = \sum_{\substack{C \subset g^+ \\ |C| = k-1}} D(\{i\} \cup C) - \sum_{\substack{C \subset g^- \\ |C| = k-1}} D(\{i\} \cup C),$$

Variation globale. Les poids géométriques sont ensuite appliqués :

$$\boxed{\Delta T_{\mathrm{GW}} = \sum_{k=2}^{K_{\max}} \big(-1/\lambda\big)^{k-1} \Delta c_k,}$$

La change-stat reste strictement locale au groupe g et ne dépend que des cliques contenant i.

Interprétation.

`cov_diff_GW` produit une mesure d'hétérogénéité multi-échelle sur un attribut numérique. Les tailles de cliques élevées contribuent avec un poids amorti et un signe alterné, ce qui permet d'ajuster finement la sensibilité aux structures denses et hétérogènes.

Exemples `erpm()`.

```
# Version par défaut (lambda = 2)
erpm(partition ~ cov_diff_GW("age"))

# Lambda plus grand (pondération plus courte des grandes
    cliques)
erpm(partition ~ cov_diff_GW("age", lambda = 3))
```

Résumé.

$$\boxed{\texttt{cov\_diff\_GW}(\lambda) : \quad T_{\mathrm{GW}} = \sum_{k \geq 2} (-1/\lambda)^{k-1} c_k(\mathrm{cov}, p), \quad \lambda > 1,}$$

13 `dyadcov_full` : covariée dyadique intra-groupe

Principe.

`dyadcov_full` agrège une covariée dyadique numérique définie entre acteurs, en ne retenant que les couples d'acteurs appartenant au même groupe de la partition, avec un filtre optionnel sur les tailles. La covariée peut être non symétrique : on utilise alors les deux sens (i, j) et (j, i).

Cadre et notations biparties.

On considère une partition stricte $B = (A, G, E)$, un groupe g de taille n_g, et une covariée dyadique $Z = (z_{ij})_{1 \le i,j \le N}$ avec $z_{ii} = 0$. On note

$$A(g) = \{i \in A : \gamma(i) = g\},$$

On choisit un ensemble de tailles admissibles $S \subset \mathbb{N}$.

Définition formelle.

La contribution locale d'un groupe g est une somme sur paires ordonnées :

$$S_g(Z) = \sum_{\substack{i \neq j \\ i,j \in A(g)}} z_{ij} = \sum_{\substack{i < j \\ i,j \in A(g)}} \big(z_{ij} + z_{ji}\big),$$

On introduit un filtre de taille

$$\phi_{\text{size}}(n_g; S) = \begin{cases} 1, & \text{si } S = \varnothing, \\ 1, & \text{si } S \neq \varnothing \text{ et } n_g \in S, \\ 0, & \text{si } S \neq \varnothing \text{ et } n_g \notin S, \end{cases}$$

avec la convention que les groupes de taille $n_g \le 1$ ne contribuent jamais (leur somme interne est nulle). La statistique globale vaut

$$\boxed{T_{\text{full}}(p; Z, S) = \sum_{g \in G} \phi_{\text{size}}(n_g; S) \sum_{\substack{i \neq j \\ i,j \in A(g)}} z_{ij} = \sum_{g \in G} \phi_{\text{size}}(n_g; S) \sum_{\substack{i < j \\ i,j \in A(g)}} \big(z_{ij} + z_{ji}\big),}$$

Lien avec `edgecov`.

Sans filtre ($S = \varnothing$),

$$T_{\text{full}}(p; Z, \varnothing) = \sum_{g \in G} \sum_{\substack{i \neq j \\ i,j \in A(g)}} z_{ij} = \sum_{i < j} \mathbf{1}[\gamma(i) = \gamma(j)]\,(z_{ij} + z_{ji}),$$

Sur la projection 1-mode où $\tilde{y}_{ij} = \mathbf{1}[\gamma(i) = \gamma(j)]$, on récupère

$$T_{\text{full}}(p; Z, \varnothing) = T_{\text{edgecov}}(\tilde{Y}; Z^{(\text{sym})}), \qquad Z^{(\text{sym})}_{ij} = z_{ij} + z_{ji},$$

En particulier, si Z est symétrique, on a $T_{\text{full}}(p; Z, \varnothing) = 2\, T_{\text{edgecov}}(\tilde{Y}; Z)$.

Change-stat locale.

Un toggle (i, g) modifie uniquement la contribution du groupe g : ni la matrice Z, ni les autres groupes ne changent. La change-statistic se réécrit

$$\Delta T_{\text{full}} = \phi_{\text{size}}(n_g^+; S)\, S_g^+ - \phi_{\text{size}}(n_g^-; S)\, S_g^-,$$

où S_g^- et S_g^+ sont les sommes internes avant et après toggle, en agrégeant pour chaque paire non orientée $\{i, j\} \subset A(g)$ la quantité $z_{ij} + z_{ji}$. L'implémentation C recalcule localement S_g pour le seul groupe touché, sans jamais recomposer la statistique globale.

Interprétation.

`dyadcov_full` pèse la partition par la somme de z_{ij} sur les couples ordonnés internes aux groupes (ou, de façon équivalente, par $\sum_{i<j}(z_{ij} + z_{ji})$ sur les paires non orientées), éventuellement restreints par taille. Un coefficient positif favorise des regroupements dans lesquels les dyades de forte covariée, éventuellement asymétrique, se retrouvent co-groupées.

Exemples `erpm()`.

```
# Covariée dyadique intra-groupe sans filtre
erpm(partition ~ dyadcov_full("friendship"))

# Variante avec filtre sur les tailles (3 ou 4)
erpm(partition ~ dyadcov_full("friendship", size = c(3, 4)))
```

Résumé.

$$\boxed{\texttt{dyadcov\_full}: \quad T(p;Z,S)=\sum_{g}\phi_{\text{size}}(n_g;S)\sum_{i\neq j\in A(g)} z_{ij}}$$

$$\boxed{T(p;Z,S)=\sum_{g}\phi_{\text{size}}(n_g;S)\sum_{i<j\in A(g)}(z_{ij}+z_{ji})}$$

14 `dyadcov` : covariée dyadique sur k-cliques intra-groupe

Principe.

`dyadcov` généralise `dyadcov_full` et `cliques(k)`. Il agrège une covariée dyadique réelle sur toutes les cliques d'acteurs de taille fixée k à l'intérieur des groupes, avec une option de normalisation par taille de groupe.

Cadre et notations biparties.

On considère une partition stricte des acteurs $A = \{1,\dots,N\}$ en groupes G, représentée par le graphe biparti $B = (A, G, E)$, et une covariée dyadique $Z = (z_{ij})_{1\le i,j\le N}$ définie sur les paires d'acteurs. On autorise Z non symétrique ; on impose seulement $z_{ii} = 0$. Pour chaque paire non orientée $\{i,j\}$ avec $i<j$, l'effet travaille avec la quantité

$$z^{\star}_{ij} = z_{ij} + z_{ji},$$

Pour une taille de clique `clique_size` $= k \ge 2$ et un groupe g de taille n_g, on note

$$\mathcal{C}_k(g) = \{\, C \subseteq A(g) : |C| = k \,\}$$

l'ensemble des k-cliques d'acteurs dans g.

Définition formelle.

Pour une clique $C = \{i_1,\dots,i_k\}$,

$$P(C;Z) = \prod_{\substack{i<j\\ i,j\in C}} (z_{ij}+z_{ji}),$$

La contribution locale non normalisée du groupe g est

$$S_g^{(k)}(Z) = \sum_{C\in\mathcal{C}_k(g)} P(C;Z) = \sum_{C\in\mathcal{C}_k(g)} \prod_{\substack{i<j\\ i,j\in C}} (z_{ij}+z_{ji}),$$

et la statistique globale non normalisée

$$\boxed{T^{(k)}(p;Z) = \sum_{g\in G} S_g^{(k)}(Z) = \sum_{g\in G}\sum_{C\in\mathcal{C}_k(g)} \prod_{\substack{i<j\\ i,j\in C}} (z_{ij}+z_{ji}),}$$

Pour la version normalisée (`normalized = TRUE`), on divise par le nombre théorique de cliques $\binom{n_g}{k}$ lorsque $n_g \ge k$:

$$\bar{S}_g^{(k)}(Z) = \begin{cases} \dfrac{1}{\binom{n_g}{k}} S_g^{(k)}(Z), & n_g \ge k,\\ 0, & n_g < k, \end{cases} \qquad T^{(k)}_{\text{norm}}(p;Z) = \sum_{g\in G} \bar{S}_g^{(k)}(Z),$$

Cas particulier $k = 2$ et lien avec `dyadcov_full`.

Pour $k = 2$, une clique C est une dyade $\{i,j\}$ et

$$P(\{i,j\};Z) = z_{ij}+z_{ji}, \quad S_g^{(2)}(Z) = \sum_{\substack{i<j\\ i,j\in A(g)}} (z_{ij}+z_{ji}),$$

On obtient alors

$$T^{(2)}(p;Z) = \sum_{g\in G} \sum_{\substack{i<j \\ i,j\in A(g)}} \big(z_{ij} + z_{ji}\big),$$

ce qui coïncide, par construction, avec l'effet `dyadcov_full(dyadcov, size = NULL)` dans sa version actuelle (prise en compte des deux orientations de chaque dyade).

Change-stat locale.

Un toggle (i, g) dans le biparti ne modifie que : (i) la taille n_g, (ii) l'ensemble des cliques $\mathcal{C}_k(g)$ qui contiennent l'acteur i et (iii) leurs produits $\prod(z_{ij} + z_{ji})$. La matrice Z reste fixe. On note $S_g^{(k)-}(Z)$ et $S_g^{(k)+}(Z)$ les contributions locales avant et après le toggle. La variation non normalisée est

$$\boxed{\Delta T^{(k)} = S_g^{(k)+}(Z) - S_g^{(k)-}(Z),}$$

et, pour la version normalisée, on remplace simplement $S_g^{(k)}$ par $\bar{S}_g^{(k)}$, ce qui ne fait intervenir en plus que les facteurs $\binom{n_g^-}{k}$ et $\binom{n_g^+}{k}$. La change-stat reste strictement locale au groupe g et aux cliques incluant i.

Interprétation.

`dyadcov` mesure la façon dont la covariée dyadique se combine à l'intérieur des cliques de taille k dans chaque groupe, en tenant compte des deux orientations de chaque dyade via $\big(z_{ij} + z_{ji}\big)$. Un coefficient positif favorise des partitions où les cliques ciblées présentent des produits élevés de ces sommes, quitte à pondérer ou non par le nombre de cliques possibles selon `normalized`.

Exemples `erpm()`.

```
# Produit des covariées dyadiques sur toutes les paires intra-
    groupe (k = 2)
erpm(partition ~ dyadcov("friendship", clique_size = 2,
                         normalized = FALSE))

# Version normalisée par groupe pour des cliques de taille 3
erpm(partition ~ dyadcov("friendship", clique_size = 3,
                         normalized = TRUE))
```

Résumé.

$$\boxed{\texttt{dyadcov}: \quad T^{(k)}(p;Z) = \sum_{g\in G} \sum_{C\in\mathcal{C}_k(g)} \prod_{i<j\in C} \big(z_{ij} + z_{ji}\big)}$$

$$\boxed{T^{(k)}_{\text{norm}}(p;Z) = \sum_{g\in G} \mathbf{1}[n_g \geq k]\, \frac{1}{\binom{n_g}{k}} \sum_{C\in\mathcal{C}_k(g)} \prod_{i<j\in C} \big(z_{ij} + z_{ji}\big)}$$

15 `dyadcov_GW(lambda)` : covariée dyadique multi-échelle

Principe.

`dyadcov_GW` combine, pour une même partition et une même covariée dyadique, les contributions de toutes les tailles de cliques $n \geq 2$ via une pondération géométrique en fonction de $\lambda > 1$. La covariée dyadique $Z = (z_{ij})$ n'a pas besoin d'être symétrique : l'effet travaille systématiquement sur la version symétrisée

$$w_{ij} = z_{ij} + z_{ji}$$

pour chaque paire non orientée $\{i, j\}$ avec $i < j$.

Cadre et notations biparties.

On considère $B = (A, G, E)$, une covariée dyadique réelle $Z = (z_{ij})_{1\leq i,j\leq N}$ avec $z_{ii} = 0$, et les cliques $\mathcal{C}_n(g)$ de taille n dans chaque groupe. Pour chaque $n \geq 2$ et groupe g,

$$S_g^{(n)}(Z) = \sum_{C\in\mathcal{C}_n(g)} \prod_{\substack{i<j \\ i,j\in C}} w_{ij} = \sum_{C\in\mathcal{C}_n(g)} \prod_{\substack{i<j \\ i,j\in C}} \big(z_{ij} + z_{ji}\big),$$

et

$$c_n(Z,p) = \sum_{g\in G} S_g^{(n)}(Z),$$

Définition formelle.

On définit les poids

$$a_n(\lambda) = \Big(-\frac{1}{\lambda}\Big)^{n-1}, \qquad n \geq 2,\ \lambda > 1,$$

et la statistique globale

$$\boxed{T_{\mathrm{GW}}(p; Z, \lambda) = \sum_{n=2}^{N_{\max}} a_n(\lambda)\, c_n(Z,p),}$$

avec $N_{\max} = \max_{g\in G} n_g$. On peut également écrire

$$T_{\mathrm{GW}}(p; Z, \lambda) = \sum_{g\in G} \sum_{n=2}^{n_g} \Big(-\frac{1}{\lambda}\Big)^{n-1} S_g^{(n)}(Z),$$

où chaque $S_g^{(n)}(Z)$ est calculé à partir des termes symétrisés $w_{ij} = z_{ij} + z_{ji}$.

Cas particulier et lien avec `dyadcov_full`.

Pour $\lambda = 2$,

$$T_{\mathrm{GW}}(p; Z, 2) = c_2(Z,p) - \tfrac{1}{2}c_3(Z,p) + \tfrac{1}{4}c_4(Z,p) - \dots$$

Le terme $c_2(Z,p)$ coïncide avec la statistique de `dyadcov_full` sans filtre de taille, construite elle aussi sur la covariée symétrisée $w_{ij} = z_{ij} + z_{ji}$. Les termes d'ordre supérieur fournissent des corrections liées aux cliques de taille $n \geq 3$.

Change-stat locale.

Un toggle (i, g) modifie la taille n_g et les cliques contenant i dans ce groupe. Pour chaque taille n, on calcule

$$\Delta S_g^{(n)}(Z) = S_g^{(n)+}(Z) - S_g^{(n)-}(Z),$$

d'où

$$\Delta c_n(Z,p) = \Delta S_g^{(n)}(Z),$$

et la variation globale

$$\boxed{\Delta T_{\mathrm{GW}}(p; Z, \lambda) = \sum_{n=2}^{N_{\max}} a_n(\lambda)\, \Delta c_n(Z,p),}$$

En pratique, l'implémentation C recalcule localement, pour le seul groupe touché, les produits de $w_{ij} = z_{ij} + z_{ji}$ dans les cliques impliquant l'acteur i, puis les combine avec les coefficients $a_n(\lambda) = (-1/\lambda)^{n-1}$.

Interprétation.

`dyadcov_GW` fournit une mesure multi-échelle des structures dans lesquelles la covariée dyadique *symétrisée* w_{ij} se renforce au sein des cliques d'acteurs. Les petites cliques dominent, les grandes cliques contribuent avec un poids amorti et des signes alternés, ce qui permet de moduler finement la sensibilité à des configurations très denses, même lorsque Z est initialement non symétrique.

Exemples `erpm()`.

```
# Forme de base : combinaison géométrique des c_n(Z,p), lambda = 2
erpm(partition ~ dyadcov_GW("friendship"))

# Poids plus courts (atténuation plus rapide des grandes cliques)
erpm(partition ~ dyadcov_GW("friendship", lambda = 3))
```

Résumé.

$$\boxed{\texttt{dyadcov\_GW}(\lambda): \quad T_{\mathrm{GW}}(p; Z, \lambda) = \sum_{n \geq 2} \left(-\tfrac{1}{\lambda}\right)^{n-1} c_n(Z, p)}$$

$$\boxed{c_n(Z, p) = \sum_{g \in G} \sum_{C \in \mathcal{C}_n(g)} \prod_{i<j \in C} \left(z_{ij} + z_{ji}\right)}$$

Annexe — ERGM et Metropolis–Hastings

A.1 Rappel : qu'est-ce qu'un ERGM ?

Un modèle de graphe aléatoire exponentiel (ERGM, *Exponential Random Graph Model*) définit une loi de probabilité sur l'ensemble des graphes possibles y :

$$\Pr_\theta(Y = y) \propto \exp\big(\theta^\top g(y)\big),$$

où :

- $g(y)$ est le **vecteur de statistiques du graphe** : il regroupe des caractéristiques mesurées sur le réseau, par exemple le *nombre d'arêtes* (statistique dénombrable ou « brute ») et le *degré moyen* (statistique agrégée ou « moyenne »).
- θ est le **vecteur de paramètres du modèle** : chaque composante pondère la propension du réseau à présenter la structure correspondante (par ex. triangles, réciprocité).
- $\Pr_\theta$ désigne la **distribution de probabilité induite par** θ : c'est la loi selon laquelle un graphe est susceptible d'être généré.

Motivation.

On dispose d'un **graphe observé** y_{obs} (données empiriques). Ce graphe est fixe : on calcule ses statistiques $g(y_{\text{obs}})$ une fois pour toutes. L'objectif est de trouver θ tel que les graphes simulés depuis $\Pr_\theta$ aient, *en moyenne*, les mêmes statistiques que y_{obs} :

$$\mathbb{E}_\theta[g(Y)] \approx g(y_{\text{obs}}),$$

afin de reproduire les régularités structurelles du réseau (densité, distribution des degrés, triangles, homophilie, etc.).

Principe général.

L'estimation de θ se fait par deux boucles :

- **Boucle interne (MCMC–MH)** : pour un θ fixé, on échantillonne des graphes selon $\Pr_\theta$ à l'aide de **Metropolis–Hastings (MH)**. Concrètement :

  1. partir d'un graphe initial $y^{(0)}$;
  2. proposer une modification locale donnant un candidat y' (ajout/suppression d'une arête) ;
  3. calculer la variation de log-probabilité [1] :

  $$\Delta = \theta^\top\big[g(y') - g(y)\big],$$

  4. accepter y' avec probabilité $\alpha = \min\big(1, \exp(\Delta) \times \frac{q(y' \to y)}{q(y \to y')}\big)$, où q est la loi de proposition.

  En répétant ces étapes, on obtient une chaîne de Markov dont la loi stationnaire est précisément $\Pr_\theta$. L'échantillon $\{y^{(s)}\}$, constitué de plusieurs graphes simulés indépendamment (après burn-in et thinning), sert à approximer les moyennes des statistiques $g(y)$ sous la distribution $\Pr_\theta$.

- **Boucle externe (mise à jour de θ)** : après simulation, on compare la moyenne simulée $\mathbb{E}_{\theta^{(t)}}[g(Y)]$ à $g(y_{\text{obs}})$ et on ajuste

$$\theta^{(t+1)} = \theta^{(t)} + \alpha \left(g(y_{\text{obs}}) - \frac{1}{S} \sum_{s=1}^{S} g(y^{(s)}) \right),$$

où α est un **pas d'apprentissage**. Un pas trop grand peut faire diverger l'algorithme ; trop petit, la convergence devient lente.

1. Pour deux graphes y et y' sous un même θ :

$$\frac{\Pr_\theta(Y = y')}{\Pr_\theta(Y = y)} = \exp\Big\{ \theta^\top [g(y') - g(y)] \Big\} = \exp(\Delta), \quad \Delta = \theta^\top [g(y') - g(y)],$$

Si $\Delta > 0$, y' est plus probable que y (toutes choses égales par ailleurs).

Lien avec les algorithmes de gradient.

La mise à jour de θ est analogue à une **montée de gradient** sur la log-vraisemblance

$$\log L(\theta) = \theta^\top g(y_{\text{obs}}) - \log Z(\theta),$$

car $g(y_{\text{obs}}) - \mathbb{E}_\theta[g(Y)]$ joue le rôle d'un **gradient stochastique**. Le pas α influence vitesse et stabilité de convergence. La fonction de vraisemblance peut être **non convexe** dans l'espace des graphes : la convergence dépend du point de départ et de la structure du réseau.

A.2 Log-vraisemblance, déviance, AIC et BIC

Constante de normalisation $Z(\theta)$.

Dans un ERGM, l'écriture exacte de la probabilité de tirage est :

$$\Pr_\theta(Y = y) = \frac{\exp\{\theta^\top g(y)\}}{Z(\theta)}, \qquad Z(\theta) = \sum_{y' \in \mathcal{Y}} \exp\{\theta^\top g(y')\},$$

où $\mathcal{Y}$ est l'ensemble de *tous* les graphes possibles sur le même jeu de nœuds. $Z(\theta)$ **assure la normalisation** (les probabilités somment à 1). Comme $\mathcal{Y}$ est gigantesque, $Z(\theta)$ n'est pas calculable exactement ; on l'approxime par MCMC.

Log-vraisemblance.

Idée générale.

La **vraisemblance** $L(\theta)$ mesure, pour un θ donné, à quel point le modèle juge plausible le graphe observé y_{obs} :

$$L(\theta) = \Pr_\theta(Y = y_{\text{obs}}) = \frac{\exp\{\theta^\top g(y_{\text{obs}})\}}{Z(\theta)},$$

Maximiser $L(\theta)$ (ou son logarithme) revient à choisir les paramètres qui rendent y_{obs} le plus probable dans la famille des modèles.

Pourquoi travailler au log.

On utilise la **log-vraisemblance**

$$\ell(\theta) = \log L(\theta) = \theta^\top g(y_{\text{obs}}) - \log Z(\theta),$$

car (i) les produits de probabilités deviennent des *sommes* (plus stables numériquement), et (ii) le calcul des dérivées/gradients est direct.

Déviance et critères d'information.

La **déviance** d'un modèle est définie par

$$D = -2\,\ell(\hat\theta),$$

où $\hat\theta$ est l'estimateur (par exemple le maximum de vraisemblance). Sous conditions régulières, les **rapports de vraisemblance** ont une loi asymptotique de type χ^2, d'où le facteur -2 qui permet d'interpréter les *différences* de déviance dans des tests de comparaison de modèles.

Déviance et AIC/BIC (version unifiée).

Dans un ERGM, $D = -2\,\ell(\hat\theta)$ sert d'indicateur global d'ajustement : plus $\ell(\hat\theta)$ est grand, plus D est petit. La déviance mesure toutefois l'ajustement *pur* : un modèle plus complexe a presque toujours une déviance plus faible. Pour **équilibrer ajustement et complexité**, on utilise les critères d'information AIC/BIC :

$$\text{AIC} = 2k - 2\,\ell(\hat\theta), \qquad \text{BIC} = k \log n - 2\,\ell(\hat\theta),$$

où k est le nombre de paramètres et n une taille effective (en réseau, souvent proche du nombre de dyades potentielles). On compare des modèles ajustés sur les mêmes données et on retient en pratique celui dont l'AIC/BIC est **minimal** (meilleur compromis « fidélité/parcimonie »). AIC/BIC permettent en outre de comparer des modèles *non nécessairement emboîtés*, ce que ne permet pas un test basé uniquement sur la différence de déviance.

A.3 Sorties usuelles de `summary(ergm)`

- **Call** : l'appel exact (traçabilité, reproductibilité).
- **Coefficients** : estimés $\hat{\theta}_i$ (un par statistique), **erreur-type** (SE), **statistique** z ($z = \hat{\theta}_i/\mathrm{SE}(\hat{\theta}_i)$) et **p-valeur bilatérale** pour $H_0 : \theta_i = 0$. Un grand $|z|$ suggère un effet éloigné de 0 au regard de l'incertitude.
- **MCMC %** : fraction (en %) de l'erreur-type due au *bruit Monte-Carlo. Interprétation et actions* : un MCMC% élevé indique que l'échantillon simulé est peu informatif (forte autocorrélation). **Burn-in** = nombre d'itérations initiales *jetées* avant d'enregistrer des échantillons (laisser la chaîne se stabiliser) ; **Thinning** = ne garder qu'une itération sur m pour **décorréler les échantillons** entre eux (réduire l'autocorrélation sérielle) ; **Taille d'échantillon** = nombre total d'états conservés pour estimer les moyennes. Si la chaîne *mélange mal*, il peut être nécessaire d'ajuster les contraintes ou le mécanisme de proposition.
- **Log-vraisemblance, déviance** D**, AIC, BIC** : indicateurs globaux (définitions ci-dessus). AIC/BIC plus petits $\Rightarrow$ meilleur compromis.

A.4 Contraintes et mécanisme de proposition (MH) en biparti

Contraintes (restriction de l'espace d'état).

Les **contraintes** (par exemple `b1part`) restreignent l'espace des graphes explorés par MH aux configurations *valides* : en biparti, interdire les arêtes intra-mode, imposer certaines bornes de degrés, etc. Elles ne constituent pas un « assouplissement » : elles **excluent** simplement les états invalides et garantissent que chaque graphe visité respecte la structure voulue.

Mécanisme de proposition.

Le **mécanisme de proposition** (souvent appelé « proposal ») génère un candidat y' à partir de y (par exemple tirer une dyade admissible et toggler l'arête), en *respectant* les contraintes. Il définit la distribution $q(y \to y')$ qui intervient dans le **ratio de Hastings**. Si q est asymétrique (par exemple plus de façons de détruire que de créer une structure), le terme $\frac{q(y' \to y)}{q(y \to y')}$ **corrige** cette asymétrie dans la probabilité d'acceptation.

Pourquoi ces corrections sont indispensables.

Le couple « Δ du modèle » + « ratio de Hastings » assure la **réversibilité** (détail de balance) de la chaîne et garantit que la **loi stationnaire** visée est bien $\Pr_\theta$. Sans cette correction, la chaîne convergerait vers une distribution biaisée (dépendante du mécanisme de proposition) au lieu de la loi ERGM recherchée.

A.5 Exemple minimal (R)

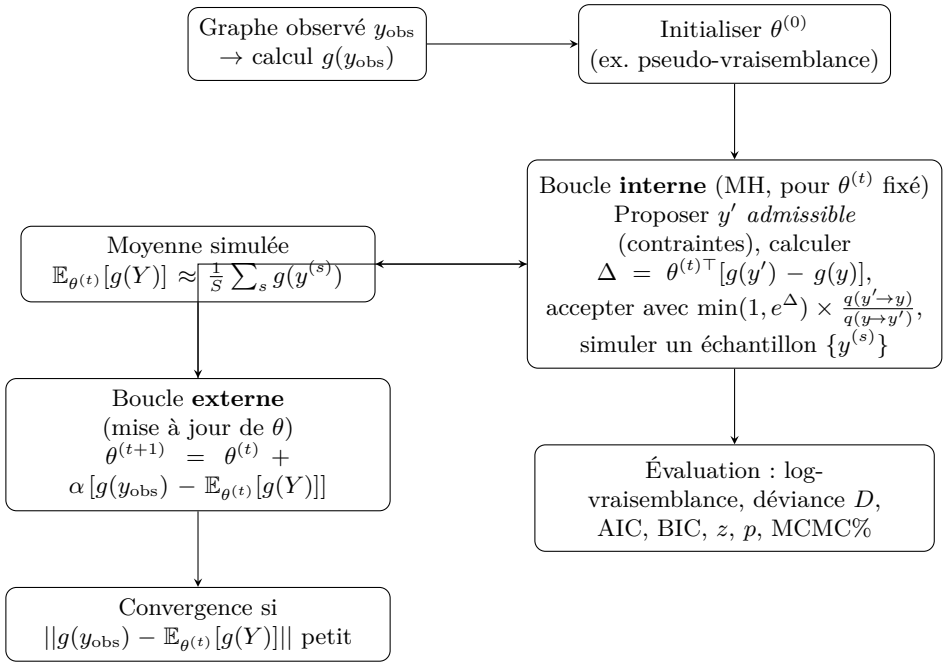
```r
library(ergm)
# Réseau biparti jouet : 4 acteurs -> 2 groupes
m <- matrix(c(1,0,
              1,0,
              0,1,
              0,1), nrow=4, byrow=TRUE)
nw <- network::network(m, bipartite=2, directed=FALSE)

# Modèle 1 : densité (edges)
fit1 <- ergm(nw ~ edges)

# Modèle 2 : densité + distribution de degrés côté groupes (mode 2)
fit2 <- ergm(nw ~ edges + b2degrange(from=2, to=3))

summary(fit1)
summary(fit2)  # comparer log-vraisemblance, AIC, BIC, z, p-values, MCMC%
```

A.6 Schéma du processus d'estimation



Deux boucles imbriquées : **interne** (MH : échantillonnage pour θ fixé) et **externe** (mise à jour de θ jusqu'à convergence).

A.7 Points d'attention

Choix de $g(y)$.

Les statistiques doivent refléter des mécanismes plausibles (densité, degrés, triangles, homophilie, effets d'attributs, etc.) sans surcharger le modèle : empiler trop de termes peut entraîner instabilités, colinéarités ou non-identifiabilité. Une approche progressive (noyau simple, diagnostics, puis complexification) est souvent utile.

Diagnostics MCMC.

Surveiller l'autocorrélation et la stabilité des moyennes simulées. Un **MCMC%** élevé signale une forte dépendance entre échantillons. Augmenter le **burn-in** (laisser la chaîne se stabiliser avant de collecter), appliquer du **thinning** (garder une itération sur m pour **décorréler les échantillons entre eux**), et/ou accroître la **taille d'échantillon**. Si le mélange reste faible, ajuster contraintes et mécanisme de proposition peut aider.

Comparaison de modèles.

AIC/BIC comparent des modèles ajustés sur les mêmes données en pénalisant la complexité. L'information est dans les **écarts** d'AIC/BIC (pas dans leur niveau absolu). Les conclusions gagnent à être croisées avec une **évaluation de type goodness-of-fit** : comparer, entre réseaux simulés et observé, des distributions de degrés, distances géodésiques, motifs triadiques, etc.