

Package	Estimation Methods	Input	Output	Wichtige Eigenschaften - Weiteres
JGL	Joint	Mehrere NxP Matrizen, gespeichert in Liste. Diese Netzwerke sollen sich „ähnlich“ sein, also gleiche Variablen. Lambda1 (Glasso-Parameter) und Lambda2 (Joint/Fused – Parameter) müssen gewählt werden.	Objekt der Klasse 'jgl'. Für Jedes Netzwerk wird die jeweilige Konzentrationsmatrix ausgegeben.	Plot Funktion basiert auf Paket iGraph. Darstellung ist schon bei 10 Beobachtungen sehr unübersichtlich. Zusätzlich hat das Paket einige Funktionen aus iGraph übernommen.
	Fused			
Huge	GLasso	NxP Data oder DxD Covarianzmatrix. Vektor von Lambdas absteigend wird benötigt.(Lambdas werden vorgegeben falls nicht angegeben)	Objekt der Klasse 'huge' Für jedes Lambda wird eine Konzentrationsmatrix berechnet. Es gibt außerdem sowohl plot als auch Auswahlfunktion – nach StARS, rotation information criterion (ric) und extended bayesian information criterion (ebic)	Für Jede Methode wird von Normalverteilten Daten ausgegangen. Falls diese Annahme nicht gegeben ist, ist ein „Normalizer“ gegeben. Huge.npn Es ist außerdem noch ein Inferenztest implementiert, welcher für NxP und berechnete Konzentrationsmatrix eine Matrix von p-werten berechnet.
	Meinerhausen & Bühlmann			
	Tiger			
Fastclime	Fastclime fastclime(...)	NxP Data oder DxD Covarianzmatrix. Außerdem wird kleinstes Lambda angegeben für welches die Konzentrationsmatrix berechnet werden soll. Außerdem muss die Länge, welcher jeder Pfad maximal haben soll, angegeben werden.	Objekt der Klasse 'fastclime' Für einen Vektor von Strafparametern Lambda wird jeweils ein graphisches Modell berechnet. Auch hier gibt es eine selektor Funktion, es ist aber nicht klar nach welchem Kriterium ausgewählt wird.	In dem Paket sind noch 'Dantzig' estimator. Allerdings ist in der Vignette nicht aufgeführt was das ist. Es wird von normalverteilten Daten ausgegangen Der plot funktioniert für fastclime-Objekt, sowie eines der ausgewählten Modelle separat (nicht so übersichtlich wie huge)

Flare	<p>Linear Regression - slim()</p> <p>Standard Lasso Dantzig Selektor Lq-Lasso</p>	<p>Notwendiger Input sind Outcome Vektor, sowie Designmatrix. Falls der Vektor an absteigenden positiven Lambdas nicht gewählt wird, so wird er aus dem Input nlambda (Anzahl) und lambda.min.value (kleinster Wert den Lambda annehmen soll) errechnet.</p>	<p>Objekt der Klasse 'slim'</p> <p>Für jedes Lambda wird ein Schätzer für den Effekt beta berechnet.</p>	<p>Flare beinhaltet sowohl Berechnungen für die Präzisionsmatrix, als auch für lineare Regression (Spärlich)</p> <p>Auch hier wird von normalverteilten Daten ausgegangen</p>
	<p>Sparse Graph Estimation -</p> <p>Tiger Clime</p>	<p>Wie schon bei den anderen Algorithmen wird eine NxP Matrix von Daten, bzw eine DxD Kovarianzmatrix benötigt. Lambda wird genauso wie bei der Linearen Regression angegeben</p> <p>Die Wahl zwischen method = 'clime' oder 'tiger' führt dann zur Berechnung der Konzentrationsmatrix</p>	<p>Objekt der Klasse 'sugm'</p> <p>Für jedes Lambda wird Konzentrationsmatrix und somit ein graphisches Modell berechnet.</p> <p>Auch hier gibt es Selektionsfunktion – diese nutzt StARS oder Cross Validierung</p>	