Network Estimation: : **CHEAT SHEET**

Packages

package	description
qgraph	plot networks
bootnet	estimate networks
mgm	Estimate mixed graphical networks/network prediction
psychonetrics	(confirmatory) network modeling

```
install.packages("qgraph","bootnet", "psychonetrics")
```

Data Format

Columns: Variables (Nodes)

Rows: Participants

Network Estimation

Gaussian Graphical Model (Continuous Data)

```
GMM_net <- estimateNetwork(data, default = "EBICglasso", corMethod = "cor", missing = "pairwise")
```

Regularization

Based on correlations

Pairwise deletion of missing values

Ising Model (Binary Data)

```
Ising_net <- estimateNetwork(data, default = "IsingFit", rule = "OR", missing = "listwise")
```

eLASSO

Listwise deletion of missing values

"OR" Rule

Mixed Graphical Model (Mixed Data, e.g., Continuous, Count, Categorical)

Type of Variable
g = Continuous
p = Count
c = Categorical

```
Fit_obj <- mgm(data, type = rep("g", 2), level = rep(1,2), lambdaSel = "EBIC", lambdaGam = 0.25
```

Level of Variable (continuous = 1)

Type of Model Selection

Network Prediction

Determines how much variance is explained by other nodes in the network

```
pred_obj <- predict(object = Fit_obj, data, errorCon = "R2")
```

Use fitted network via mgm

predict works only with networks fitted with the mgm package and no NAs.

Network Plots

Basic Plotting

Several possibilities to plot network models.

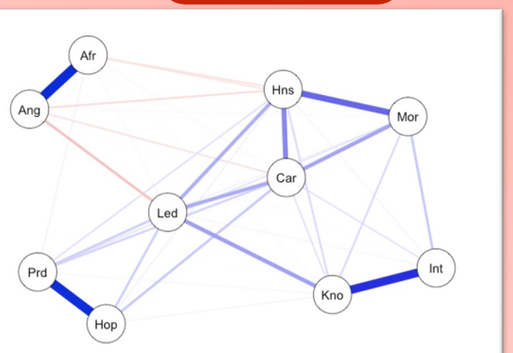
Both codes produce identical results.

```
plot(GGM_net)
```

```
qgraph(GGM_net$graph, layout="spring", theme = colorblind)
```

Object from *estimate Network* function

Fruchterman-Reingold



Advanced Plotting

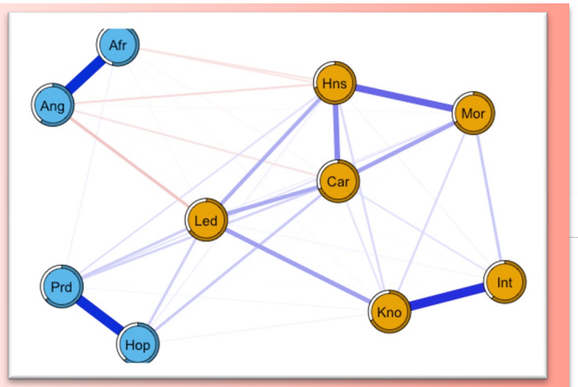
```
groups <- c(rep("Beliefs",6), rep("Feelings",4))
```

Differentiate between groups

Define groups

```
plot(GGM_net, groups = groups, legend = false, pie = abs(pred_obj$errors[,2]), labels = colnames(data))
```

Object via *predict*



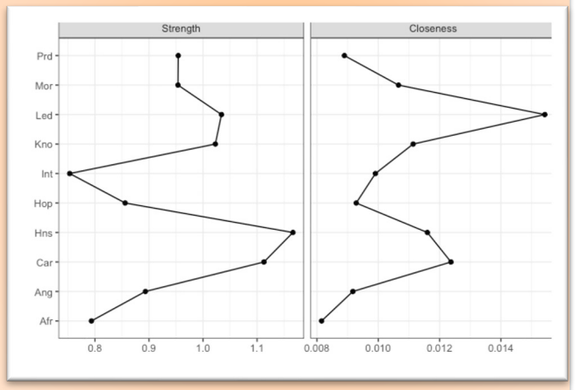
Plot including grouping and R^2 for each node (defined by the ring around the node).

Network Centrality

Object from *estimateNetwork*

```
centralityPlot(GGM_net, scale = "raw", include = c("Strength","Closeness"))
```

Define centrality indices




```
#determine values
cent <- centrality(GGM_net)
```

Network Stability

Use also "closeness" or "strength"

```
set.seed(4815162342)
boot1 <- bootnet(GGM_net, statistics = c("edge", "Strength", "Closeness"), nboots = 1000, nCores = 2, type = "nonparametric")
```

Set Seed for reproducibility

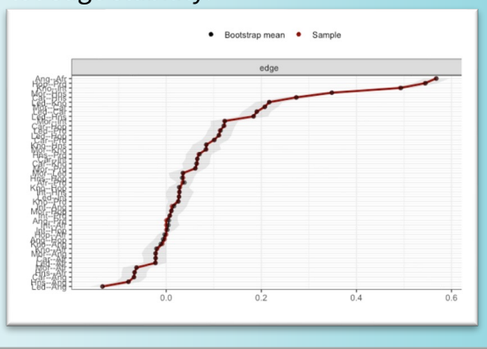
Edge comparison

```
boot2 <- bootnet(GGM_net, statistics = c("Strength", "Closeness"), nboots = 1000 nCores = 2, type = "case")
```

Centrality Stablity

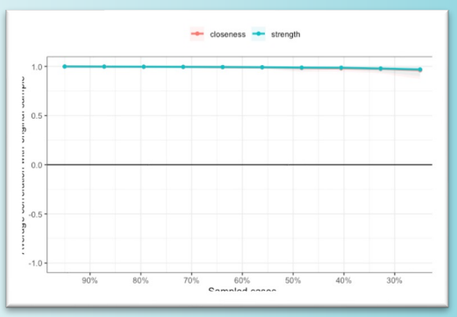
```
plot(boot1, statistics = "edge", labels = TRUE, order = "sample")
```

Plot edge stability.




```
plot(boot2, statistics = c("Strength","Closeness"))
```

Plot centrality stability.



Difference Tests

```
plot(boot1, statistics = "edge", plot = "difference", onlyNonZero = TRUE, order = "sample")
```

specify difference