# Perform Network Analysis

- Create a cinemanet matrix
- Create a cinemanet matrix using threshold filter with Random Matrix Theory
- Analyze properties of the network

**Questions to answer with alpha and beta diversity analysis:**

- Does the 50 year film lists for (a) or (b)
- Does the 50 year film lists for (a) or (b)
- Does data 20 compare to 50, 20 compare to 100
- Does data 20 compare to 50

```
In [1]: import numpy as np
        import pandas as pd
        import os
        %matrix inline
        %matrix inline pt_pt
```

## Import Data

```
In [2]: pt_pt = pd.read_csv('data/50_cinemanet_matrix.csv', index_col=0)
        pt_people_cinemanet = pd.read_csv('data/50_cinemanet_people_cinemanet.csv', index_col=0)
        pt_people = pd.read_csv('data/50_people.csv', index_col=0)
        pt_film = pd.read_csv('data/50_film.csv', index_col=0)
        pt_people_cinemanet.shape, pt_people.shape, pt_film.shape
        pt_people.shape, pt_film.shape, pt_pt.shape
        pt_pt.shape, pt_pt.sum(axis=1).sum()
```

```
In [3]: print('pt_people_cinemanet.shape', pt_people_cinemanet.shape)
        print('pt_people.shape', pt_people.shape)
        print('pt_film.shape', pt_film.shape)
        print('pt_pt.shape', pt_pt.shape)
        print('pt_pt.sum(axis=1).sum()', pt_pt.sum(axis=1).sum())
```

```
pt_people_cinemanet.shape (9, 9)
pt_people.shape (9, 9)
pt_film.shape (9, 9)
pt_pt.shape (9, 9)
pt_pt.sum(axis=1).sum() 0.0
```

## Network Construction

- create a pan-wide similarity (Pearson) of attendance across different samples
- create a film-wide similarity (Pearson) of attendance across different samples

```
In [4]: %matrix pearson_similarity_matrix pt_pt
```

## Use Random Matrix Theory To Threshold

Use Random Matrix Theory To Threshold by Tyler Kendall

TODO: Figure out if Log Pearson distribution is an RV, or Yes, RV, squared value series univar.

```
In [5]: %cd /home/jet/projects/jet_data_lib, %run -i -d -b -q -p -q -m -s -t -r -o -i -v -l -q -p -q -m -s -t -r -o -i -v -l -q -p -q -m -s -t -r -o -i -v -l
```

## Advanced NetworkX: Community detection with modularity

There are many ways of calculating communities, cliques, and clusters in your network, but the most popular method currently is modularity. Modularity is a measure of relative density in your network, a community pattern is relative if modularity shows that high density relative to other nodes within its module but low density with those outside. Modularity gives you an overall score of how fractious your network is, and it can be used to determine the optimal number of communities in your network.

```
In [1]: from networkx.algorithms.community.quality import modularity, coverage, performance, inter_c
```

```
In [2]: print(list(nx.connected_components(Ego_Twitter_Graph)))
[{'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18',
```

```
# an-perfect-index 20
# a-more-fun-index 20
# a-month-of-index 20
# a-month-of-index-2 20
total number of index: 20

# an-perfect-index 21
# a-more-fun-index 21
# a-month-of-index 21
# a-month-of-index-2 21
total number of index: 21

# an-perfect-index 22
# a-more-fun-index 22
# a-month-of-index 22
# a-month-of-index-2 22
total number of index: 22

# an-perfect-index 23
# a-more-fun-index 23
# a-month-of-index 23
# a-month-of-index-2 23
total number of index: 23

# an-perfect-index 24
# a-more-fun-index 24
# a-month-of-index 24
# a-month-of-index-2 24
total number of index: 24

# an-perfect-index 25
# a-more-fun-index 25
# a-month-of-index 25
# a-month-of-index-2 25
total number of index: 25

# an-perfect-index 26
# a-more-fun-index 26
# a-month-of-index 26
# a-month-of-index-2 26
total number of index: 26

# an-perfect-index 27
# a-more-fun-index 27
# a-month-of-index 27
# a-month-of-index-2 27
total number of index: 27
```

```
{'an-perfect-index': {'0.0000000000000000': 0.0, '0.0555555555555555': 0.0},
 'a-more-fun-index': {'0.0000000000000000': 0.0, '0.0555555555555555': 0.0},
 'a-month-of-index': {'0.0000000000000000': 0.0,
 'a-month-of-index-2': {'0.0000000000000000': 0.0,
 'aaliyah-ms': {'0.0000000000000000': 0.0,
 'aaliyah-ms-2': {'0.0000000000000000': 0.0,
 'aaliyah-ms-3': {'0.0000000000000000': 0.0,
 'aaliyah-ms-4': {'0.0000000000000000': 0.0,
 'aaliyah-ms-5': {'0.0000000000000000': 0.0,
 'aaliyah-ms-6': {'0.0000000000000000': 0.0,
 'aaliyah-ms-7': {'0.0000000000000000': 0.0,
 'aaliyah-ms-8': {'0.0000000000000000': 0.0,
 'aaliyah-ms-9': {'0.0000000000000000': 0.0,
 'aaliyah-ms-10': {'0.0000000000000000': 0.0,
 'aaliyah-ms-11': {'0.0000000000000000': 0.0,
 'aaliyah-ms-12': {'0.0000000000000000': 0.0,
 'aaliyah-ms-13': {'0.0000000000000000': 0.0,
 'aaliyah-ms-14': {'0.0000000000000000': 0.0,
```