

در بررسی شبکه شهر رفسنجان به این نتیجه رسیدیم که scale free نیست و Random می باشد، چگونه به این نتیجه رسیدیم؟

```
degree_sequence = sorted([d for n, d in G.degree()], reverse=True)
```

```
import powerlaw
fit = powerlaw.Fit(degree_sequence,xmin=1)
fit.alpha
```

میدانیم شبکه ای scale free می باشد که توزیع درجات آن power law باشد، ابتدا لازم هست که degree sequence شبکه را بدست بیاوریم سپس به کمک پکیج power law دنباله درجات را روی توزیع power law فیت می کنیم. سپس مقدار degree exponent را بدست میاریم که برابر با 2.516 شد. از طرف دیگر ممکن است توزیع power law نباشد و توزیع دیگری مثل log linear یا exponential باشد، حال باید چه کنیم؟ باید مقایسه بین توزیع ها کنیم که رشته کد زیر این امکان را برای ما فراهم می کند. طبق داکيومنت کد زیر در خروجی یک ضریب loglikelihood به ما می دهد و ما R را چک میکنیم اگر بزرگتر از صفر شد یعنی دنباله درجات از توزیع نخست یعنی exponential پیروی می کنند و در غیر این صورت از توزیع دوم پیروی خواهند کرد. در اینجا مقدار R بزرگتر از صفر به دست آمد پس دنباله درجات شبکه شهر رفسنجان از توزیع exponential پیروی می کنند و شبکه ما Random می باشد.

```
R, p = fit.distribution_compare('exponential','power_law',normalized_ratio=True)
R
```

**R** : float

Loglikelihood ratio of the two distributions' fit to the data. If greater than 0, the first distribution is preferred. If less than 0, the second distribution is preferred.

## Change idea

در اینجا آدام توزیع exponential و log normal را مقایسه کردم دیدم که توزیع log normal فیت تر هست نسبت به exponential بر روی دنباله درجات و از آن جایی که توزیع های log normal و power law از یک خانواده هستند پس شبکه ما scale free می باشد و Random می باشد.

```
R, p = fit.distribution_compare('exponential','lognormal',normalized_ratio=True)
R
```

-41.647079340760854


برای به دست آوردن قطر شهر از دو راه رفتم یک راه با کمک کتابخانه گفی بود که مقدار 131 را در خروجی نمایش داد و یک راه هم از کتابخانه networkx که به خروجی 153 را داد که من در مقاله قطری که به کمک گفی به دست آمده بود رو در نظر گرفتم.

```
diameter = max([max(j.values()) for (i,j) in nx.shortest_path_length(G)])
```

diameter

153

networkx

 HTML Report



## Graph Distance Report

### Parameters:

Network Interpretation: undirected

### Results:

Diameter: 131

Radius: 66

Average Path length: 46.54301022914224

Gephi

```
✓ [17] nx.info(G)
0s 'MultiDiGraph with 7963 nodes and 20072 edges'
```

```
✓ [18] k = sum([v for k, v in G.degree()]) / len(G)
0s k
5.041316086901921
```

```
✓ [19] G = nx.Graph(G)
0s nx.info(G)
'Graph with 7963 nodes and 11217 edges'
```

```
✓ [20] k = sum([v for k, v in G.degree()]) / len(G)
0s k
2.8172799196282807
```

در مقاله توضیح داده شد که ما گراف را غیر جهت دار کردیم و به گراف ساده تبدیل کردیم در اینجا کدش را مشاهده می کنید