Install packages

```
!pip install python-igraph leidenalg cairocffi
```

```
     Collecting python-igraph
       Downloading python_igraph-0.9.6-cp37-cp37m-manylinux2010_x86_64.whl (3.2 MB)
          |████████████████████████████████| 3.2 MB 5.2 MB/s
     Collecting leidenalg
       Downloading leidenalg-0.8.7-cp37-cp37m-manylinux2010_x86_64.whl (1.4 MB)
          |████████████████████████████████| 1.4 MB 65.2 MB/s
     Collecting cairocffi
       Downloading cairocffi-1.2.0.tar.gz (70 kB)
          |████████████████████████████████| 70 kB 8.2 MB/s
     Collecting texttable>=1.6.2
       Downloading texttable-1.6.4-py2.py3-none-any.whl (10 kB)
     Requirement already satisfied: cffi>=1.1.0 in /usr/local/lib/python3.7/dist-packages
     Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packages (f
     Building wheels for collected packages: cairocffi
       Building wheel for cairocffi (setup.py) ... done
       Created wheel for cairocffi: filename=cairocffi-1.2.0-py3-none-any.whl size=89562 s
       Stored in directory: /root/.cache/pip/wheels/e2/ca/86/9db2824f203afe4bdf5aa6ead017f
     Successfully built cairocffi
     Installing collected packages: texttable, python-igraph, leidenalg, cairocffi
     Successfully installed cairocffi-1.2.0 leidenalg-0.8.7 python-igraph-0.9.6 texttable-
```

Import packages

```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import igraph as ig
```

Load network

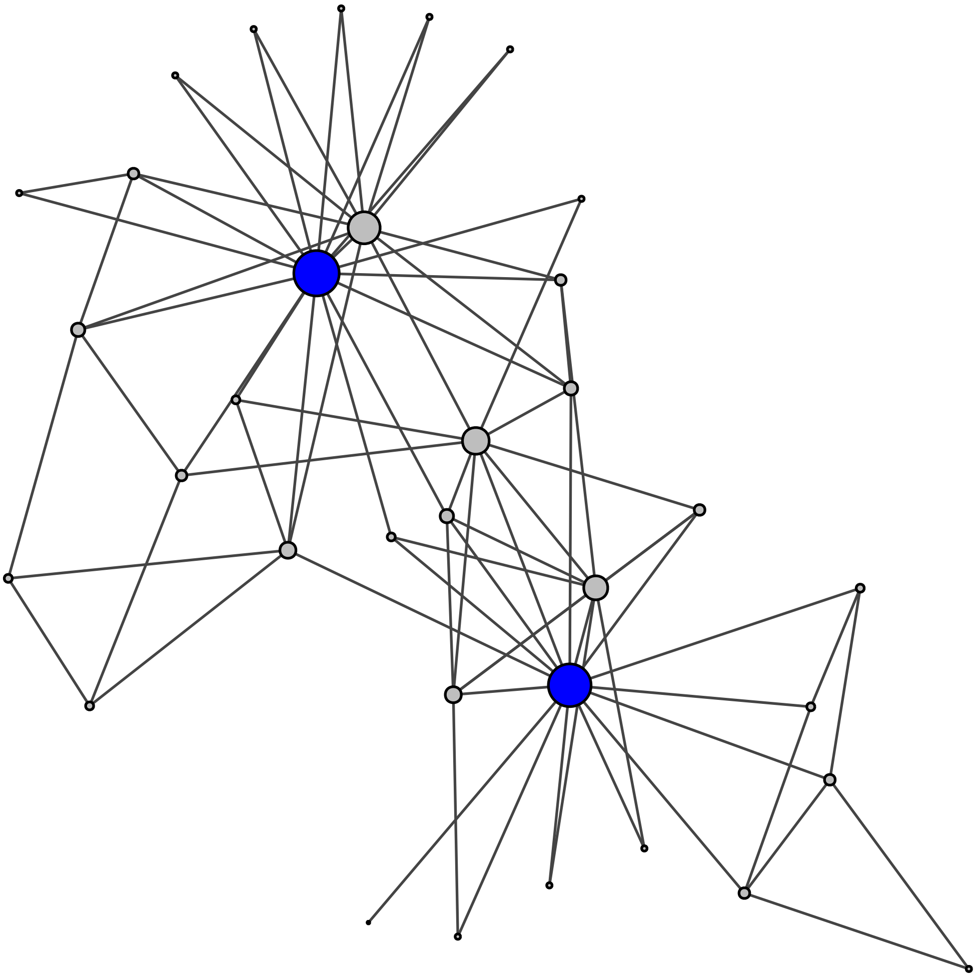
```
g = ig.Graph.Read_GML('drive/MyDrive/Notebooks/Master_Degree/Complex_Networks/Datasets/kar
```

Plot network

```
#creating labels
labels = np.arange(1,g.vcount()+1)
labels = [str(i) for i in labels]

#setting the new labels
# g.vs['label'] = labels
```

```
#plotting network
visual_style = {}
visual_style["vertex_size"] = [i for i in g.vs.degree()]
visual_style["vertex_color"] = ['gray' if i<15 else 'blue' for i in g.vs.degree()]
visual_style["bbox"] = (400, 400)
visual_style["margin"] = 20
visual_style['vertex_shape'] = 'circle'
ig.plot(g, **visual_style)
```



Get parameters

```
print(f'The network has: \n')
print(f'nodes -> {g.vcount()} ')
print(f'links -> {g.ecount()} edges')
print(f'maximum degree -> {g.vs.maxdegree()}')
print(f'minimum degree -> {min(g.vs.degree())} ')
print(f'average degree -> {int(sum(g.vs.degree())/g.vcount())} ')
print(f'maximum links -> {(g.vcount()*(g.vcount()-1))/2} ')
print(f'density -> {(sum(g.vs.degree()))/(g.vcount()*(g.vcount()-1)):.6} ')
```

```
    The network has:
```

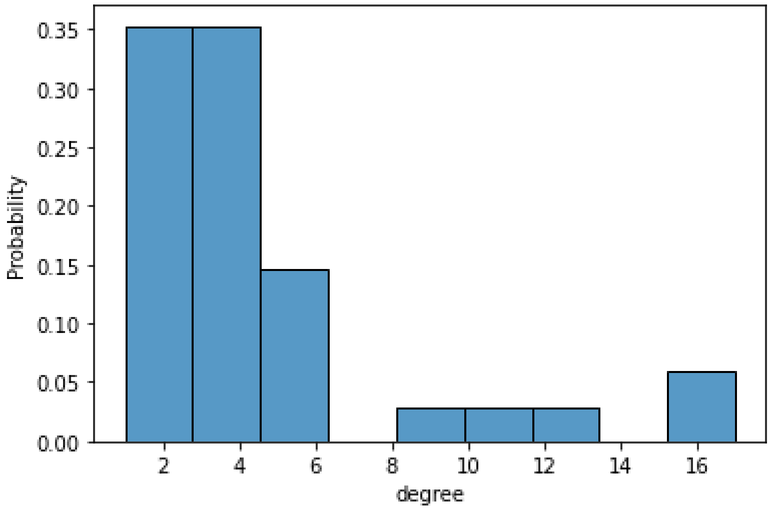
```
nodes -> 34
links -> 78 edges
maximum degree -> 17
minimum degree -> 1
average degree -> 4
maximum links -> 561.0
density -> 0.139037
```

Degree distribuition

```
df = pd.DataFrame(data=g.vs.degree(), columns=['degree'])
```

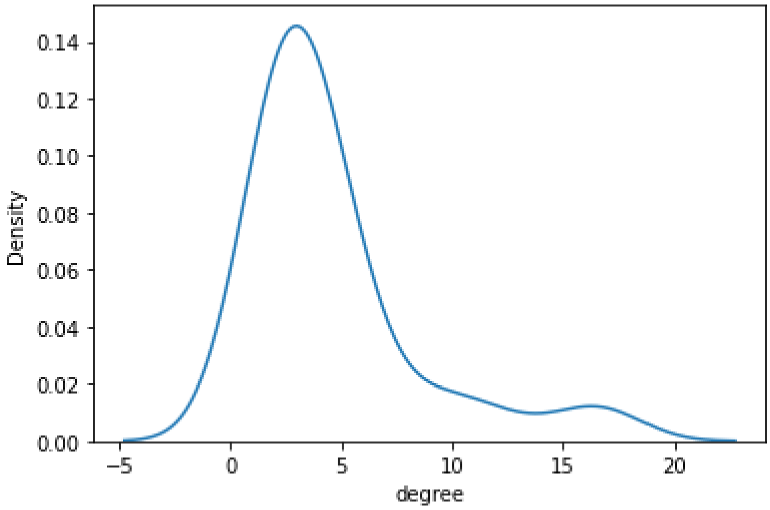
```
sns.histplot(df['degree'],stat='probability')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f595a0b2990>
```



```
sns.kdeplot(df['degree'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5959ffc810>
```



✓ 0s completed at 2:12 PM ● ×