CLUSTERING VIA STATISTICAL TESTS

ŞADİ UYSAL

BOGAZICI UNIVERSITY

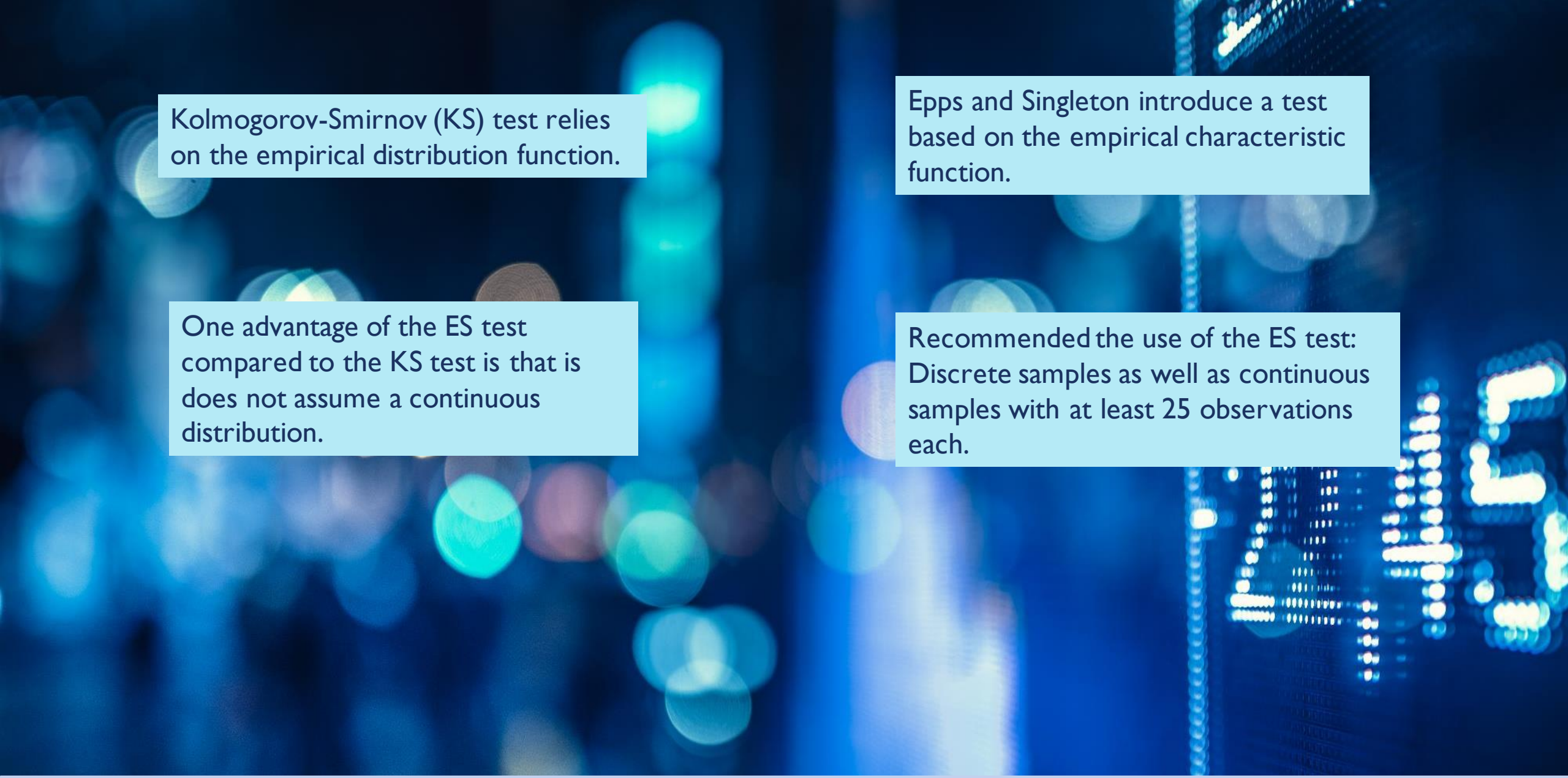
HEADLINES

- Statistical tests
- Coding Part
- Results

Testing whether two samples are generated by the same underlying distribution is a classical question in statistics.

Kolmogorov-Smirnov Test

Epps-Singleton (ES) Test



Kolmogorov-Smirnov (KS) test relies on the empirical distribution function.

Epps and Singleton introduce a test based on the empirical characteristic function.

One advantage of the ES test compared to the KS test is that is does not assume a continuous distribution.

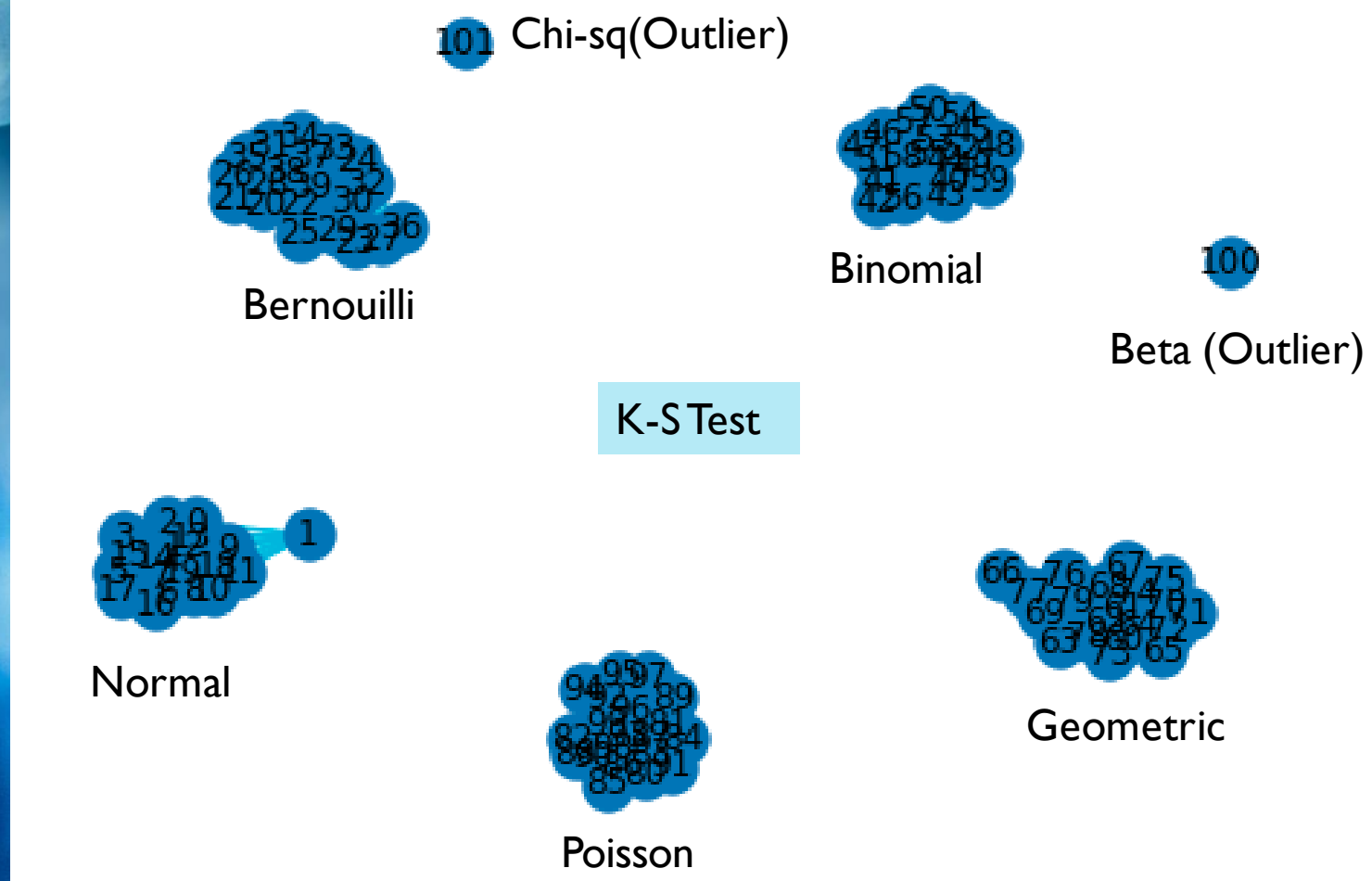
Recommended the use of the ES test: Discrete samples as well as continuous samples with at least 25 observations each.

```python
from scipy.stats import bernoulli,binom,geom,poisson,beta,chi2,ks_2samp,epps_singleton_2samp
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import random

G = nx.Graph()
H=nx.Graph()
n_of_samples=20
samples=[] #list to store generated discrete number samples as [random_numbers,sample_number,dist_type]
s_size=1000
n=random.randrange(50,100)   #binomial n,poisson mean
p=0.3     #bernouilli,binomial,geometric  p value
```

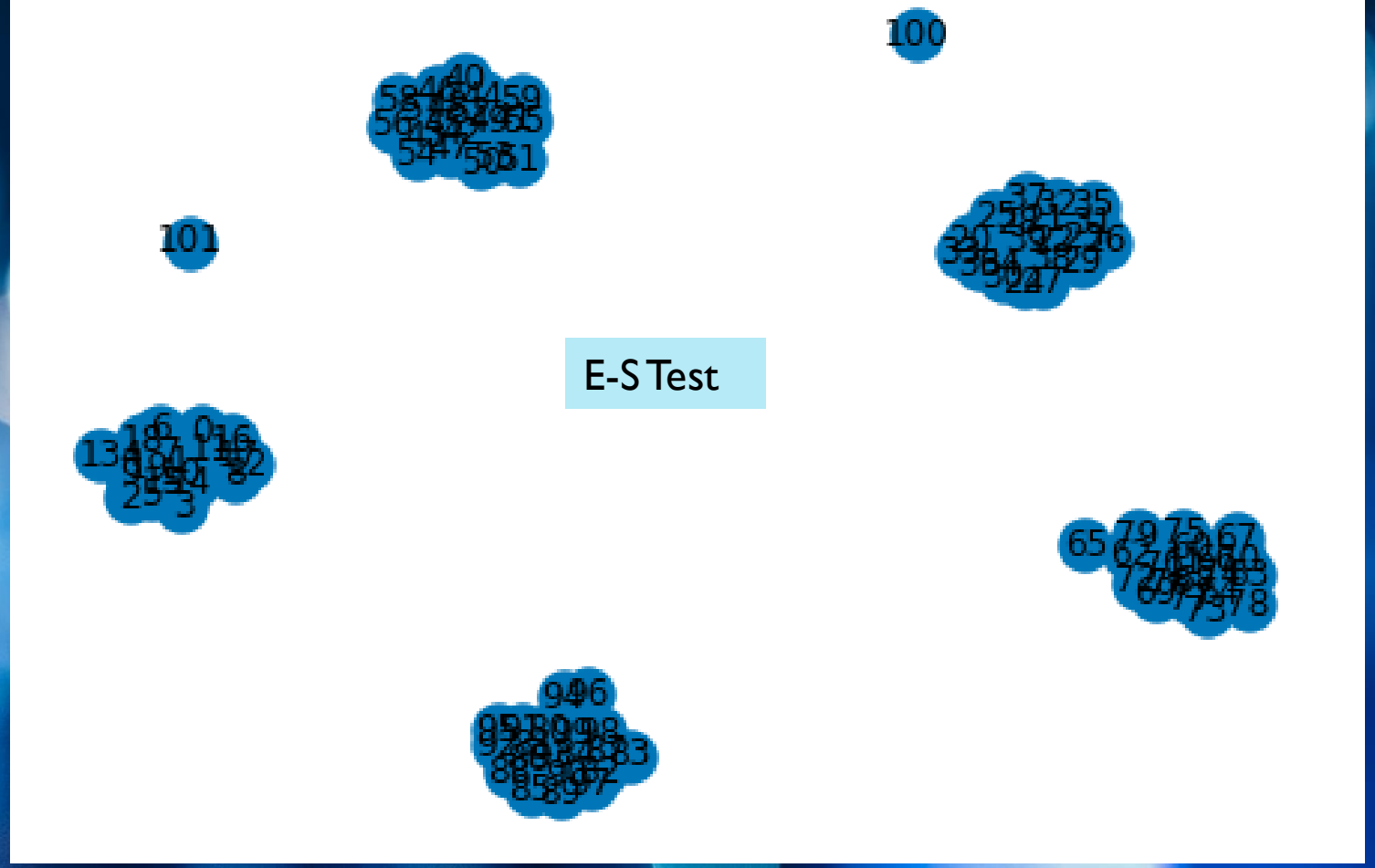
```
for i in range(n_of_samples):
    y = np.random.normal(0, 1, s_size)
    samples.append([y,i,"normal"])
for i in range(n_of_samples,2*n_of_samples):
    y = bernoulli.rvs(p, size=s_size)
    samples.append([y,i,"bernoulli"])
for i in range(2*n_of_samples,3*n_of_samples):
    y=binom.rvs(n,p, size=s_size)
    samples.append([y,i,"binomial"])
for i in range(3*n_of_samples,4*n_of_samples):
    y = geom.rvs(p, size=s_size)
    samples.append([y,i,"geometric"])
for i in range(4*n_of_samples,5*n_of_samples):
    y = poisson.rvs(n, size=s_size)
    samples.append([y,i,"poisson"])
outlier_1 = beta.rvs(1, 10, size=1000)
outlier_2 = chi2.rvs(n, size=1000)
samples.append([outlier_1,5*n_of_samples,"beta"])
samples.append([outlier_2,5*n_of_samples+1,"chi_square"])
```

```
for i in range(len(samples)):
    for j in range(i,len(samples)):
        ks_test_pvalue=ks_2samp(samples[i][0], samples[j][0])[1]
        epps_singleton_pvalue=epps_singleton_2samp(samples[i][0], samples[j][0])[1]

        if ks_test_pvalue>0.05:
            G.add_edge(i, j, weight=0.01/(ks_test_pvalue)) #0.01 scaling factor here
        if epps_singleton_pvalue>0.05:
            H.add_edge(i, j, weight=0.01/(epps_singleton_pvalue)) #0.01 scaling factor here
```

```
nx.draw(G,with_labels=True, edge_color='#00b4d9')
```



```
1  nx.draw(H,with_labels=True, edge_color='#00b4d9')
```



THANK YOU

Şadi Uysal
Contact info:
sadiuysalsadi@gmail.com