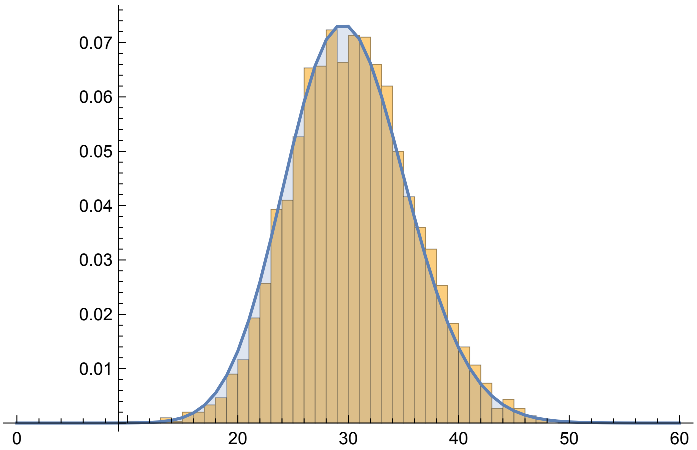
Social Network Analysis

Chen Avin
School of Electrical and Computer Engineering

Unit 9
Scale Free Graphs
and Power Laws Degree
Distributions

(Based on Networks: An Introduction. By M.E.J Newman, and Slides by Lada Adamic)

Last Time

- Random Graphs models
- Erdös-Rényi Graph model
  - Simple model
  - Giant Component
  - Short Path length
  - Poisson / Binomial Degree Distribution
- 

What happens in real life?

■ Network of Thrones

In[1139]:=

```
SetDirectory[NotebookDirectory[]];
file = Rest[Import["data/stormofswords.csv"]];
tribes = Import["data/tribes.csv"];
nodes = Flatten[tribes[[All, 1]]];
edges = #[[1]] ⟷ #[[2]] & /@ file [[All, {1, 2}]];
ThronesG = Graph[nodes, edges, VertexLabels → Placed["Name", Tooltip]]
```

Out[1144]=

In[1145]:=

```
maxDegree = Max[VertexDegree[ThronesG]];
avgDegree = Mean[VertexDegree[ThronesG]] // N;
minDegree = Min[VertexDegree[ThronesG]];
size = VertexCount[ThronesG];
```

■ Network Size

```
size
```

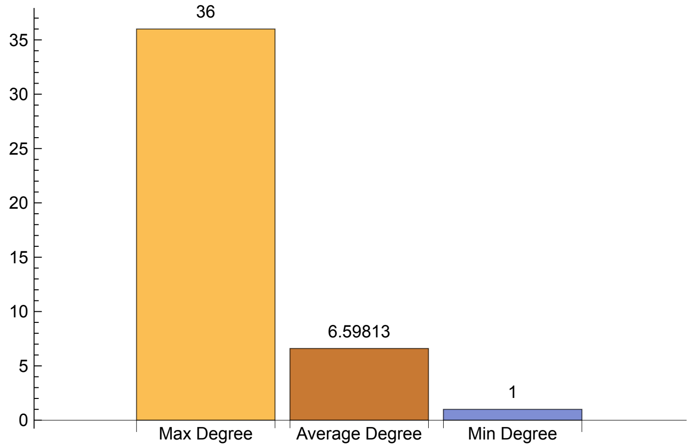
107

■ Max Degree, Avrg. Degree, Min Digree

In[1150]:=

```
BarChart[{{maxDegree, avgDegree, minDegree}},
 ChartLabels → {"Max Degree", "Average Degree", "Min Degree"},
 LabelingFunction → Above]
```

Out[1150]=

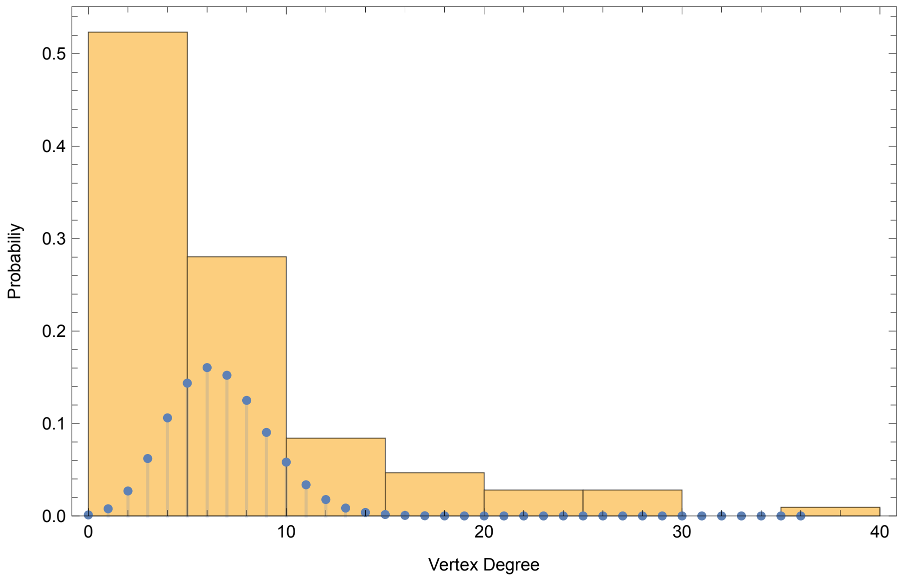


■ Let's try to fit Binomial Distribution

In[1151]:=

```
Show[Histogram[VertexDegree[ThronesG], Automatic, "Probability", PlotRange → All,
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True],
 DiscretePlot[Evaluate@PDF[BinomialDistribution[size, avgDegree / size], k],
   {k, 0, maxDegree, 1}, PlotRange → {0, maxDegree, 5},
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]]
```

Out[1151]=



■ Let's try to fit Exponential Distribution (looks good)

In[1152]:=

```
PDF[ExponentialDistribution[λ], x]
```

Out[1152]=

$$\begin{cases} e^{-x\lambda}\lambda & x \geq 0 \\ 0 & \text{True} \end{cases}$$

In[1153]:=

```
Mean[ExponentialDistribution[λ]]
```

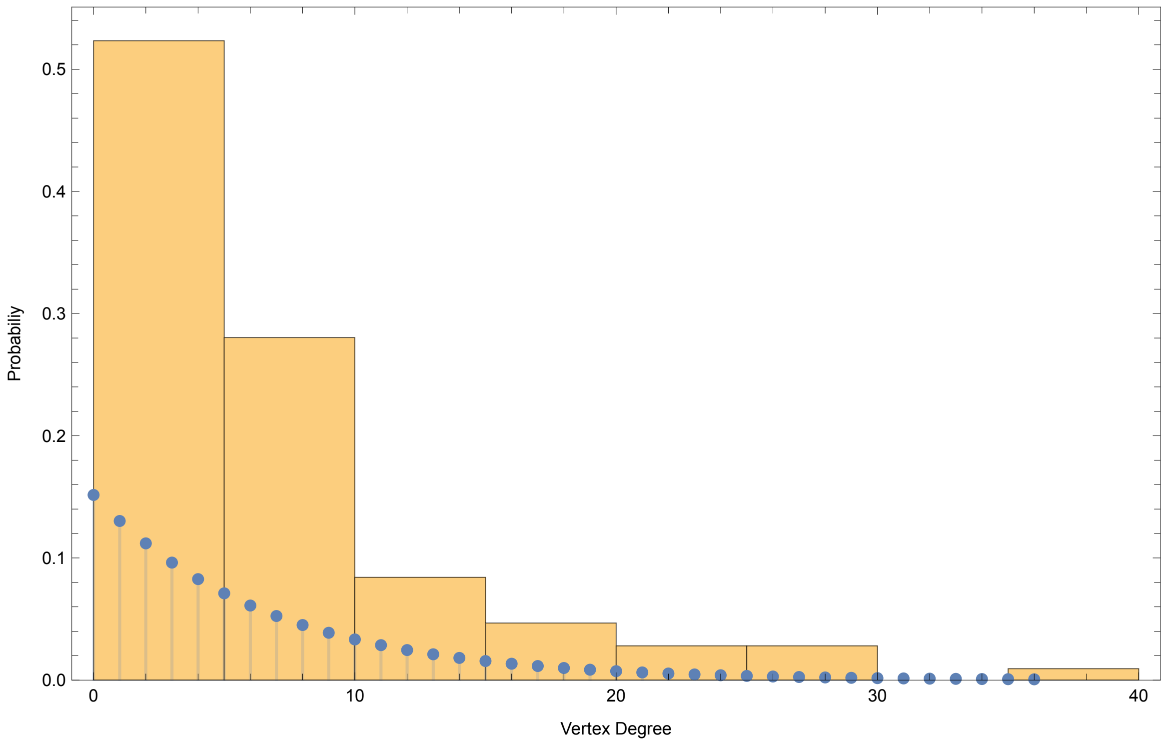
Out[1153]=

$$\frac{1}{\lambda}$$

In[1154]:=

```
Show[Histogram[VertexDegree[ThronesG], Automatic, "Probability", PlotRange → All,
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True],
 DiscretePlot[Evaluate@PDF[ExponentialDistribution[1 / avgDegree], k],
   {k, 0, maxDegree, 1}, PlotRange → All,
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]]
```

Out[1154]=



Class Facebook Network

In[1155]:=

```
SetDirectory["/Users/avin/Dropbox/SocialNetworkClass/FacebookStudents/MyTry"];
fbNodes = Import["ClassVertexList.csv", CharacterEncoding → "UTF8"];
fbEdges = Import["ClassEdgeList.csv" , CharacterEncoding → "UTF8"];
class = Import["students.csv" , CharacterEncoding → "UTF8"];
ClassFBall = Graph[Flatten[fbNodes],
   #[[1]] ⟷ #[[2]] & /@ fbEdges, VertexLabels → Placed["Name", Tooltip]];
maxDegree = Max[VertexDegree[ClassFBall]];
avgDegree = Mean[VertexDegree[ClassFBall]] // N;
minDegree = Min[VertexDegree[ClassFBall]];
size = VertexCount[ClassFBall];
```

■ Network Size

In[1164]:=

```
size
```

Out[1164]=

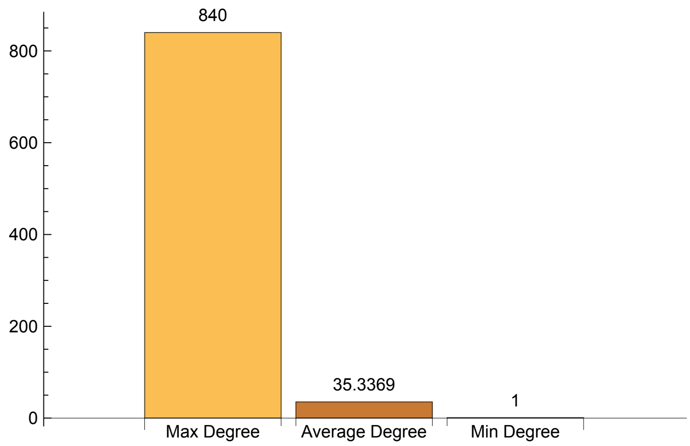
10 095

■ Max Degree, Avrg. Degree, Min Digree

In[1165]:=

```
BarChart[{{maxDegree, avgDegree, minDegree}},
 ChartLabels → {"Max Degree", "Average Degree", "Min Degree"},
 LabelingFunction → Above]
```

Out[1165]=

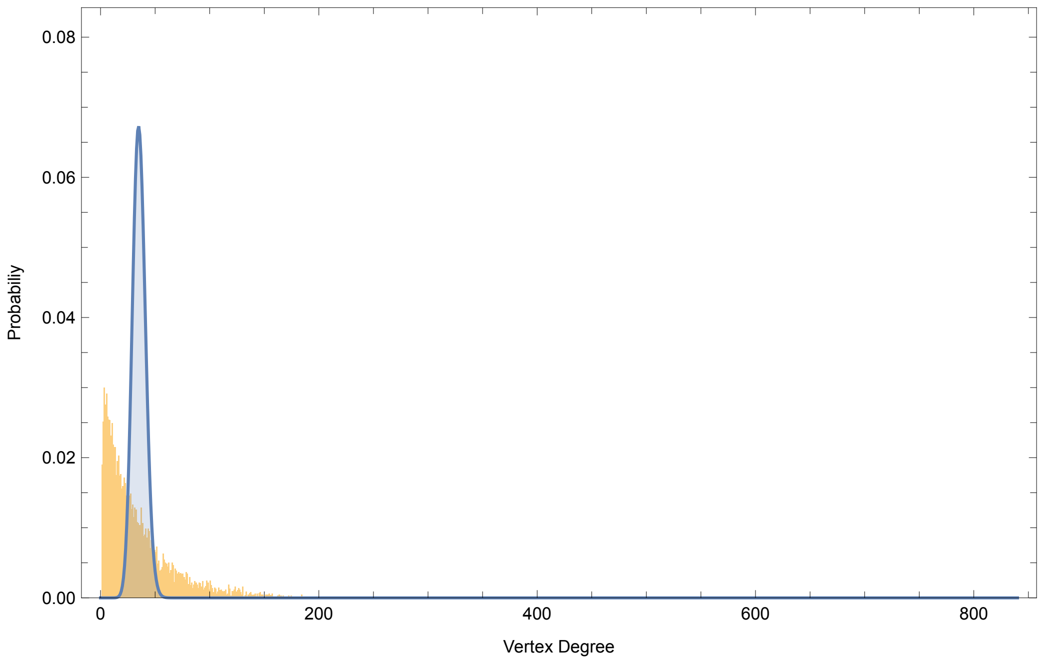


■ Let’s try to fit Binomial Distribution

In[1166]:=

```
Show[Histogram[VertexDegree[ClassFBall], {0, maxDegree, 1},
  "Probability", PlotRange → {{0, maxDegree}, {0, 0.08}},
  FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True],
 DiscretePlot[Evaluate@PDF[BinomialDistribution[size, avgDegree / size], k],
  {k, 0, maxDegree, 1}, PlotRange → {{0, maxDegree}, {0, 0.08}},
  FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]]
```

Out[1166]=



- Let's try to fit Exponential Distribution (looks good)

In[1167]:=

```
PDF[ExponentialDistribution[λ], x]
```

Out[1167]=

$$\begin{cases} e^{-x\lambda}\lambda & x \geq 0 \\ 0 & \text{True} \end{cases}$$

In[1168]:=

```
Mean[ExponentialDistribution[λ]]
```

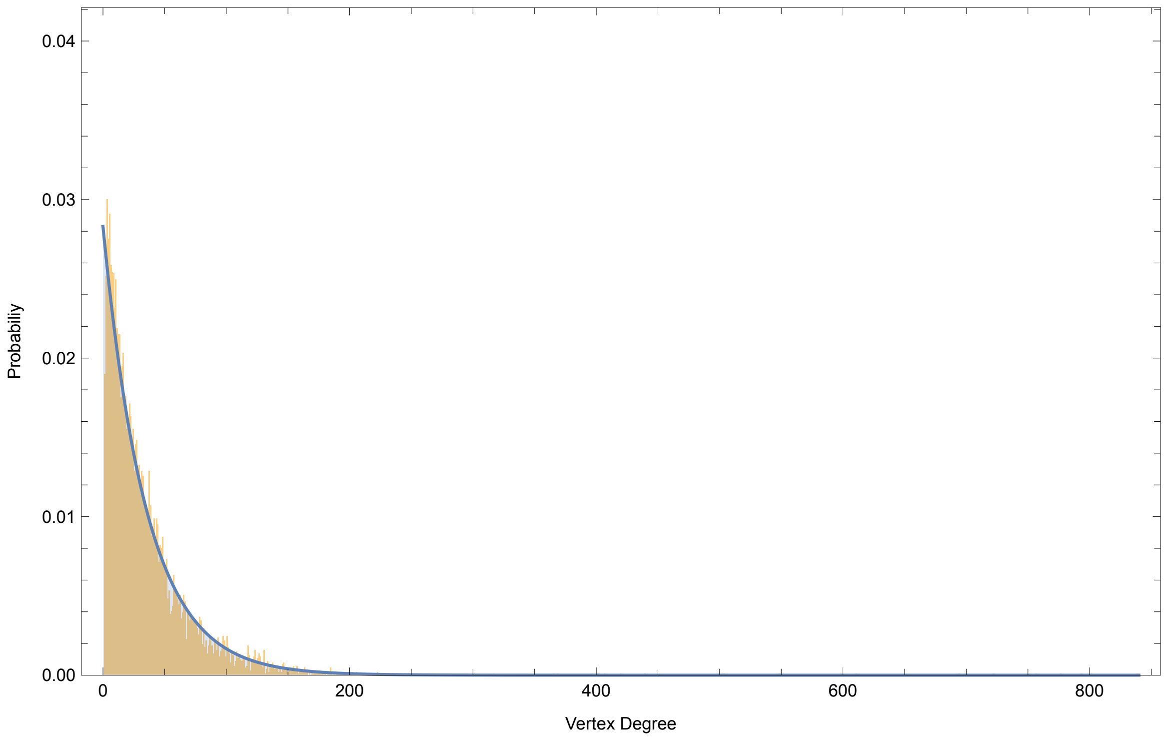
Out[1168]=

$$\frac{1}{\lambda}$$

In[1169]:=

```
Show[Histogram[VertexDegree[ClassFBall], {0, maxDegree, 1},
   "Probability", PlotRange → {{0, maxDegree}, {0, 0.04}},
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True],
  DiscretePlot[Evaluate@PDF[ExponentialDistribution[1 / avgDegree], k],
   {k, 0, maxDegree, 1}, PlotRange → {{0, maxDegree}, {0, 0.04}},
   FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]]
```

Out[1169]=

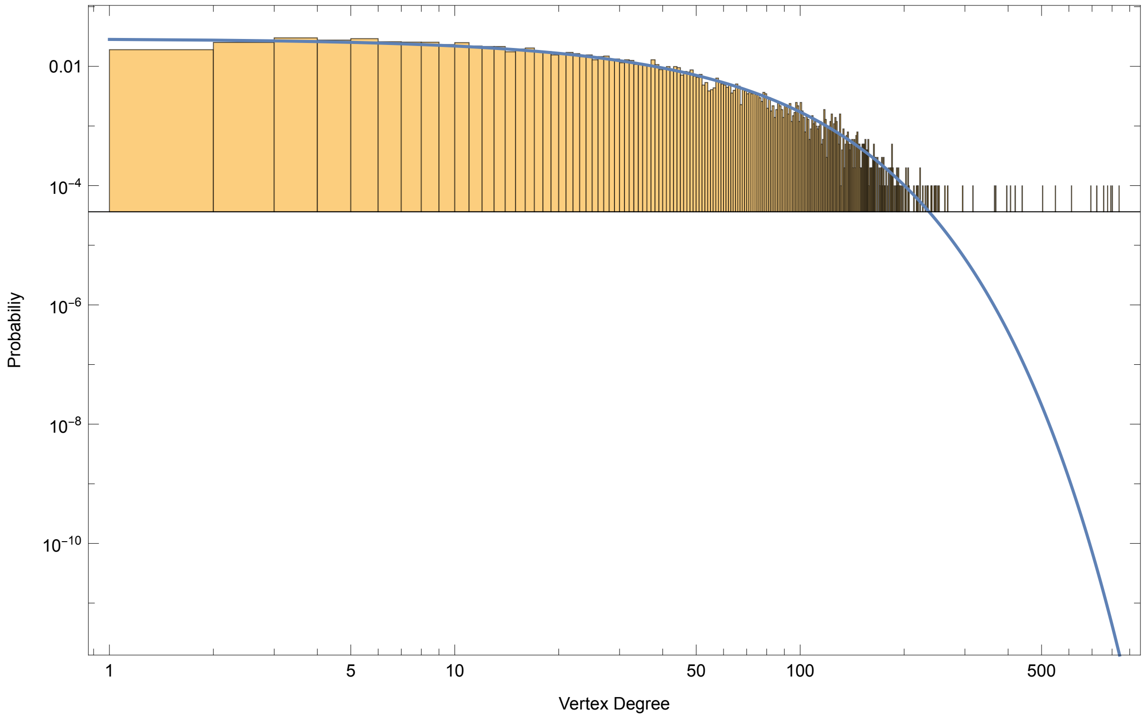


- Let’s take a closer look

In[1170]:=

```
Show[Histogram[VertexDegree[ClassFBall],
  {0, maxDegree, 1}, "Probability", ScalingFunctions → {"Log", "Log"},
  FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True],
 ListLogLogPlot[Table[Evaluate@PDF[ExponentialDistribution[1 / avgDegree], k],
   {k, 0, maxDegree, 1}], Joined → True,
  FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]]
```

Out[1170]=



Power Laws and Scale-Free Networks

Let's consider the AS graph

In[1171]:=

```
ExampleData[{"NetworkGraph", "Internet"}, "LongDescription"]
```

Out[1171]=

```
A symmetrized snapshot of the structure of the Internet
  at the level of autonomous systems, reconstructed from BGP
  tables posted by the University of Oregon Route Views Project.
```

In[1217]:=

```
ASgraph = ExampleData[{"NetworkGraph", "Internet"}];
```

- Size, Max Degree, Average Degree, Min Degree

In[1173]:=

```
sizeAS = VertexCount[ASgraph];
maxAS = Max[VertexDegree[ASgraph]];
avgAS = Mean[VertexDegree[ASgraph]] // N;
minAS = Min[VertexDegree[ASgraph]];
Style[TextGrid[{{"", "Size", "Max Degree", "Average Degree", "Min Degree"},
   {"AS Graph", sizeAS, maxAS, avgAS, minAS}}, Frame → All], 20]
```

Out[1177]=

	Size	Max Degree	Average Degree	Min Degree
AS Graph	22 963	2390	4.21861	1

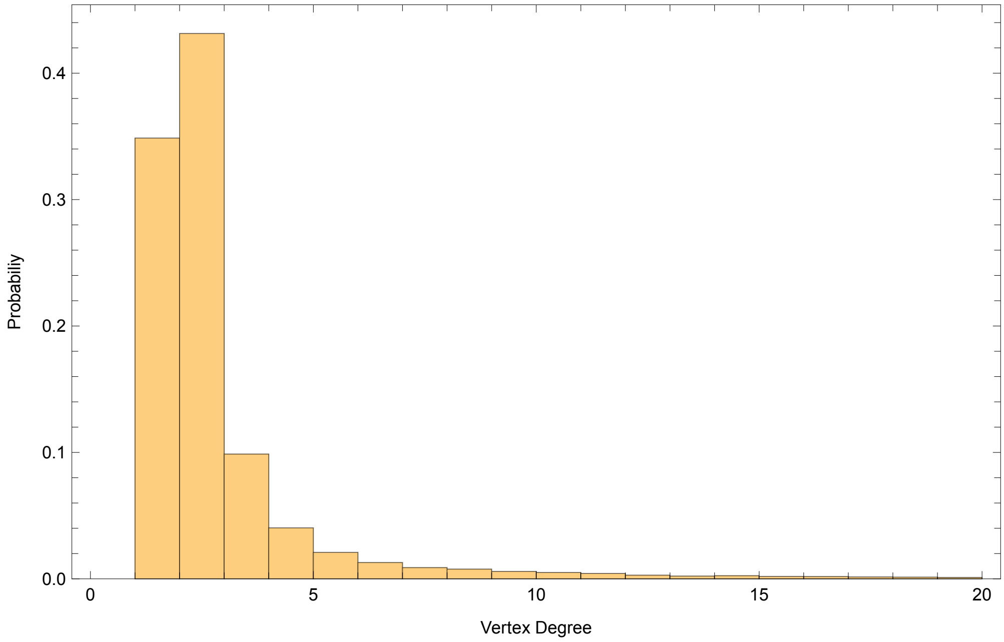
- Wow! one BGP router is connected to 10% of the network

■ Regular Histogram

In[1178]:=

```
Histogram[VertexDegree[ASgraph], {0, 20, 1},
 "Probability", PlotRange → {{0, 20}, Automatic},
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]
```

Out[1178]=



■ Wow! Almost 80% are degree 1 or 2

In[1179]:=

```
Total[Sort[Tally[VertexDegree[ASgraph]]][[{1, 2}, 2]]] / VertexCount[ASgraph] // N
```

Out[1179]=

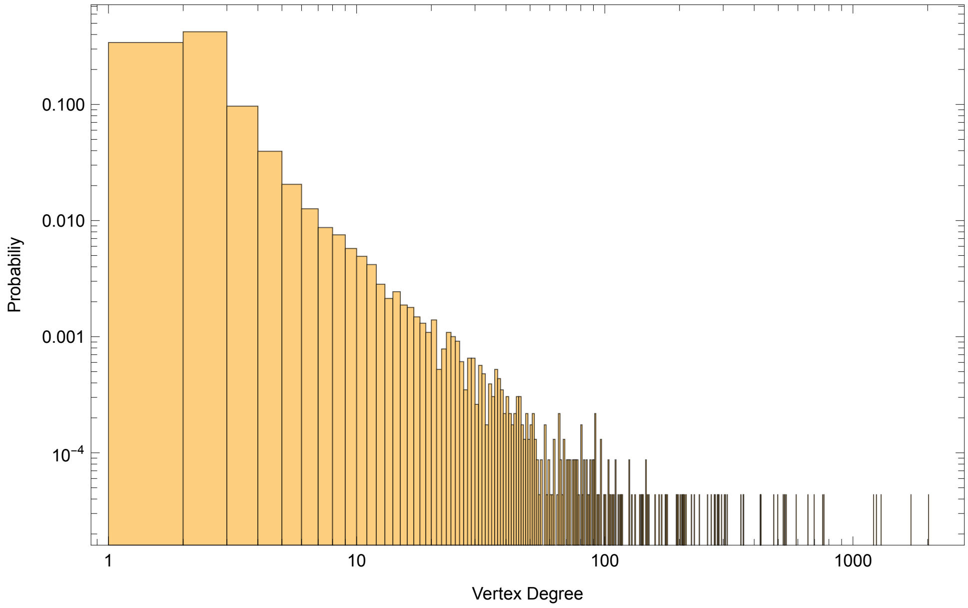
```
0.763837
```

■ How does is looks on a log-log scale?

In[1180]:=

```
Histogram[VertexDegree[ASgraph], {0, maxAS, 1},
 "Probability", ScalingFunctions → {"Log", "Log"},
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]
```

Out[1180]=

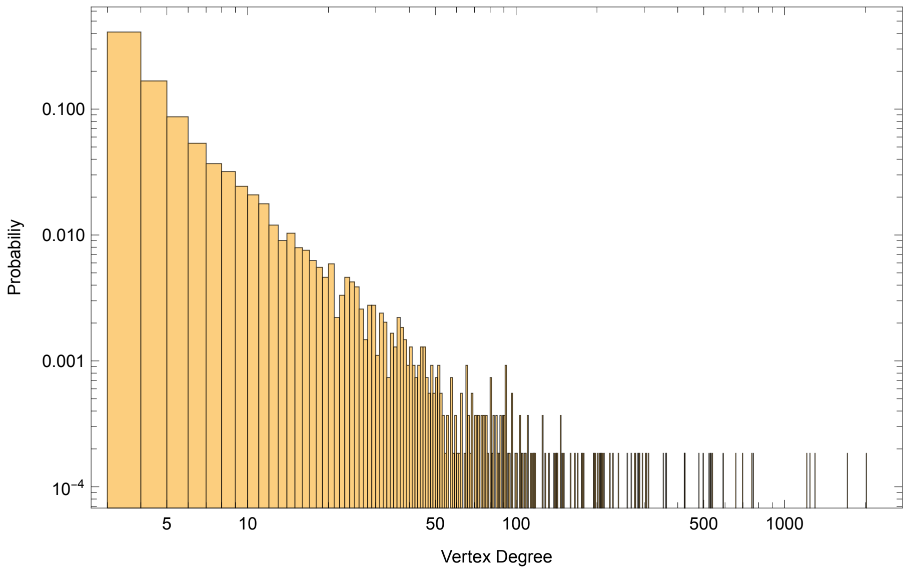


- “cleaning” the low degree nodes

In[1181]:=

```
Histogram[VertexDegree[ASgraph], {3, maxAS, 1},
 "Probability", ScalingFunctions → {"Log", "Log"},
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]
```

Out[1181]=



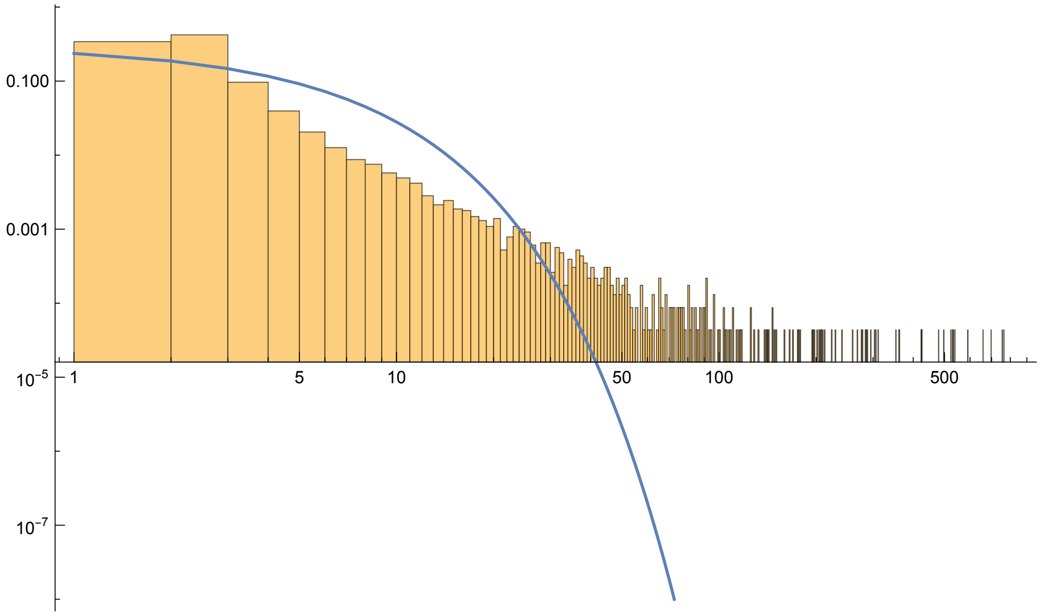
- Can you “see” the straight line?
- Who are the “hubs” (superstars) of the network?

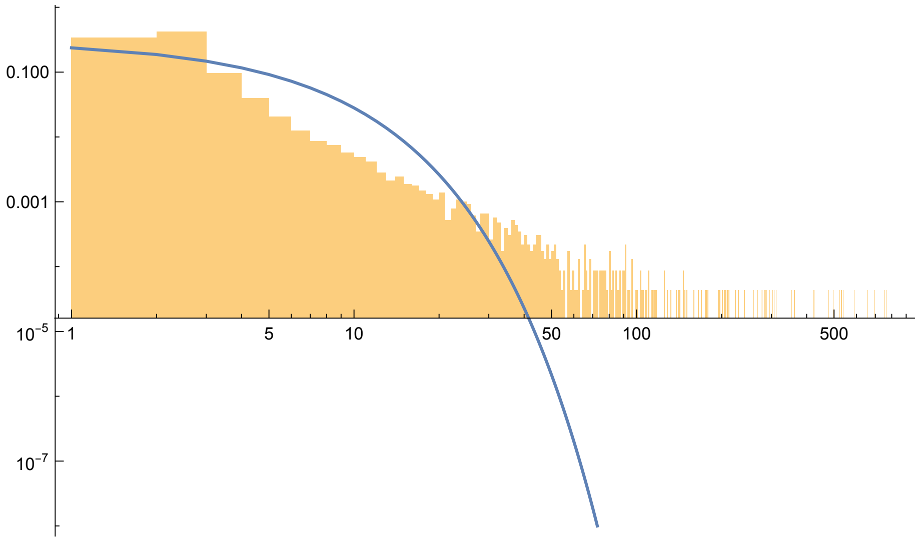
■ Let's check again the Exponential Distribution

In[1183]:=

```
Show[Histogram[VertexDegree[ASgraph], {0, maxDegree, 1},
  "Probability", ScalingFunctions → {"Log", "Log"}], ListLogLogPlot[
  Table[Evaluate@PDF[ExponentialDistribution[1 / avgAS], k], {k, 0, maxAS / 5, 1}],
  Joined → True, PlotRange → {Automatic, {1, 10^-8}}]]
```

Out[1183]=





Power Law Distribution

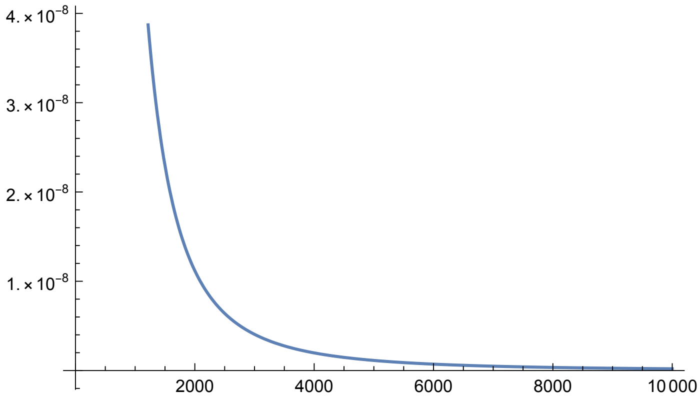
- p_k *is Pr(deg = k)*
  - The probability of a random node to be of degree k
- $p_k \approx C\, k^{-\beta}$
- The logarithm of the degree distribution p_k is a linear function of the logarithm of the degree k
- *so*

$$\ln p_k = -\beta \ln k + c$$

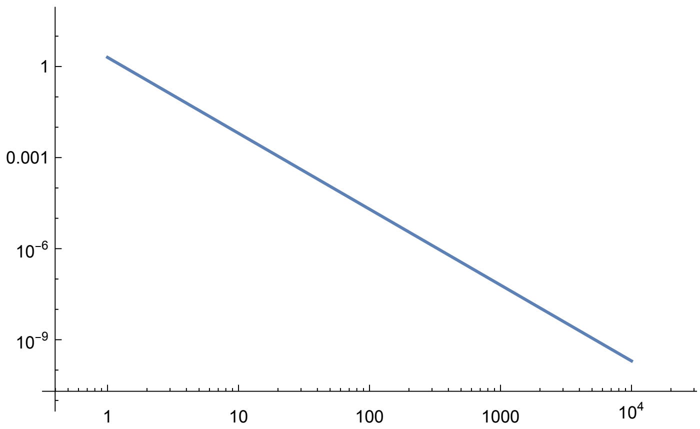
And taking the exponent on both sides

$$p_k = C\, k^{-\beta} \text{ where } C = e^c$$

```
Plot[2 k^-2.5, {k, 1, 10 000}]
```




```
LogLogPlot[2 k^-2.5, {k, 1, 10 000}]
```



- β is the exponent of the power law

- β is the slop in the log-log plot
- Typically in the range $2 \leq \beta \leq 3$
- Usually a power law in the tail of the distribution (large degrees)
- Networks with power law degree distributions are called "scale free" network
- Why "scale free"?

$$\frac{\Pr(bx)}{\Pr(x)} = f(b) \quad \forall\, b, x$$

- In our case

$$\frac{C\,(bx)^{-\beta}}{C\,(x)^{-\beta}} = b^{-\beta} \quad \forall\, b, x$$

Power Law in real world

- In many places....
- Sort and plot (log-log scale)

In[1184]:=

```
LargeCities = CityData[{Large, "UnitedStates"}];
Pop = CityData[#, "Population"] & /@ LargeCities;

Length[LargeCities]
```

306

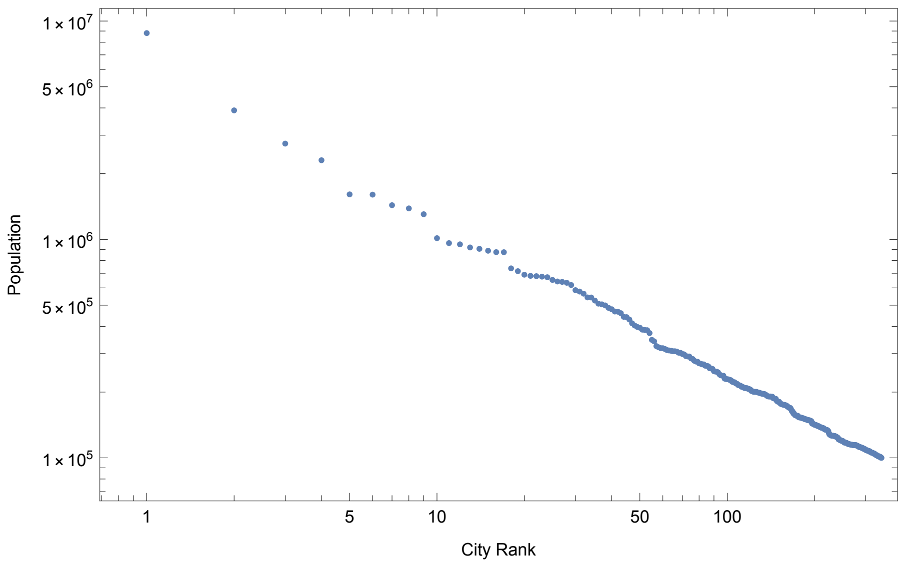
In[1186]:=

```
Cities = Table[Tooltip[Pop⟦i⟧, LargeCities⟦i⟧], {i, 1, Length[LargeCities]}];
```

In[1187]:=

```
ListLogLogPlot[Cities, PlotRange → All,
 Frame → True, FrameLabel → {"City Rank", "Population"}]
```

Out[1187]=




```
{Pop⟦1⟧, LargeCities⟦1⟧}
```

{8 336 697 people, New York City}

```
{Pop⟦2⟧, LargeCities⟦2⟧}
```

{3 857 799 people, Los Angeles}

```
{Pop⟦3⟧, LargeCities⟦3⟧}
```

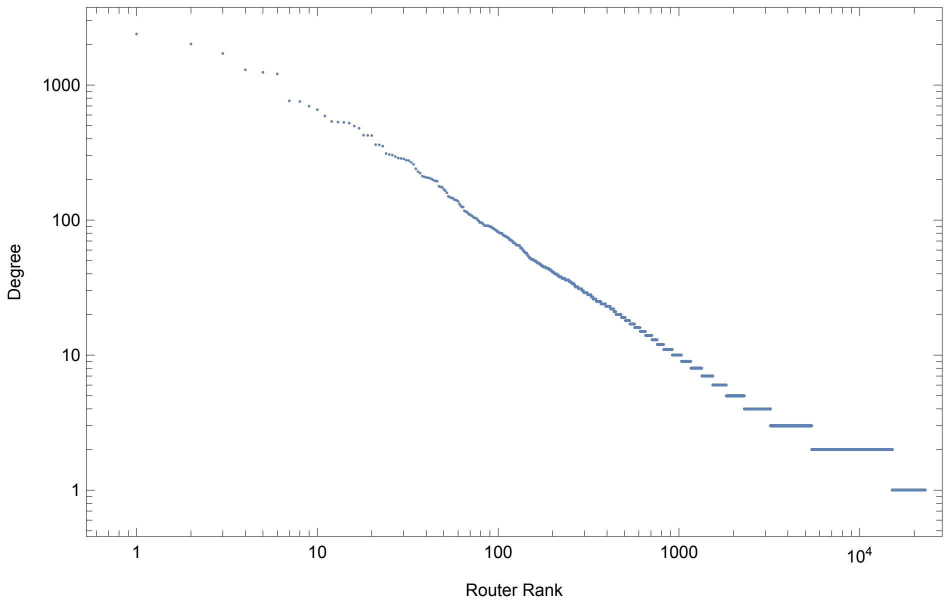
{2 714 856 people, Chicago}

- AS graph

In[1188]:=

```
ListLogLogPlot[Sort[VertexDegree[ASgraph], Greater],
 PlotRange → All, Frame → True, FrameLabel → {"Router Rank", "Degree"}]
```

Out[1188]=

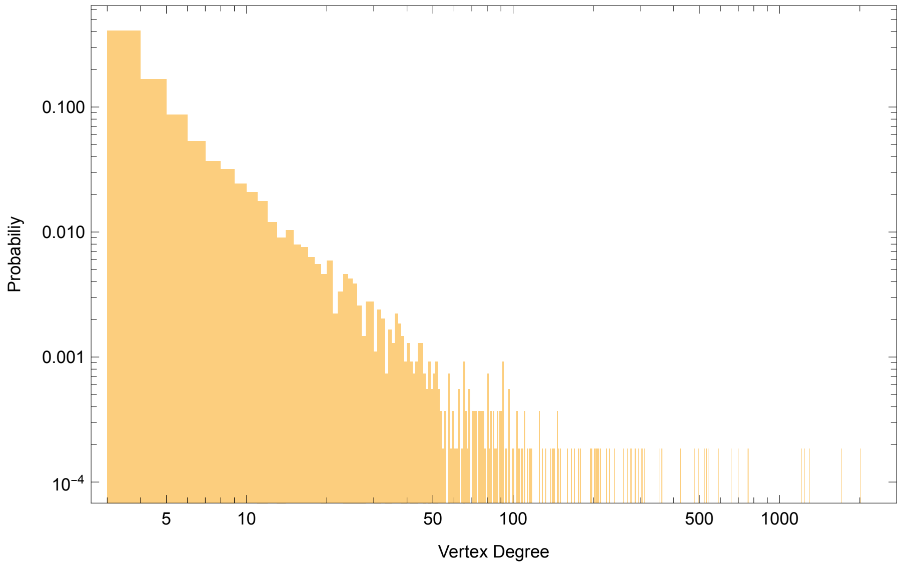


- Why if sorting a power law the distribution is a power law?
  - If after sorting we have the sizes C$(\frac{1}{1}, \frac{1}{2^\beta}, \frac{1}{3^\beta}, \frac{1}{4^\beta},\ldots)$
  - There are i elements larger than $\frac{C}{i^\beta}$
  - $\Pr(k \geq \frac{C}{i^\beta}) = \frac{i}{N}$
  - $\Pr(k > x) = C' \, x^{\left(-\frac{1}{\beta}\right)}$

Detecting Power Laws

▪ Regular Bins

```
Histogram[VertexDegree[ASgraph], {3, maxAS, 1},
 "Probability", ScalingFunctions → {"Log", "Log"},
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]
```



▪ Noise

▪ Scatter Plot

In[1189]:=

```
DegreeFreq[g_] := Map[{#[[1]], #[[2]] / VertexCount[g]} &, Tally[VertexDegree[g]]];
```

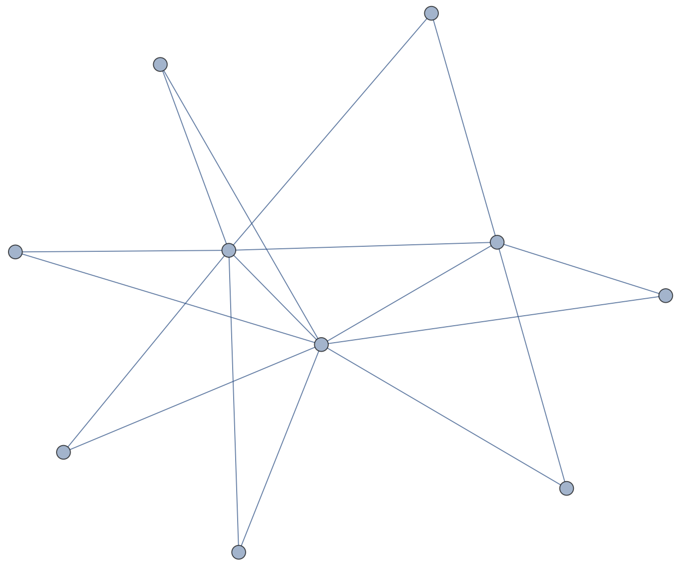
In[1219]:=

```
CumDigDist[g_, min_, max_] := Module[{df, cm, minpos, maxpos},
  df = Sort[DegreeFreq[g], #1[[1]] < #2[[1]] &];
  minpos = First@First@Position[df[[All, 1]], min];
  maxpos = First@First@Position[df[[All, 1]], max];
  cm = Reverse[Accumulate[Reverse[df[[All, 2]]]]];
  Transpose[{df[[minpos ;; maxpos, 1]], cm[[minpos ;; maxpos]]}]]
```

In[1220]:=

```
rg = RandomGraph[BarabasiAlbertGraphDistribution[10, 2]]
TableForm[Transpose[
  {CumDigDist[rg, Min[VertexDegree[rg]], Max[VertexDegree[rg]]][[All, 1]],
   Sort[Tally[VertexDegree[rg]], #1[[1]] < #2[[1]] &][[All, 2]],
   Sort[DegreeFreq[rg], #1[[1]] < #2[[1]] &][[All, 2]],
   CumDigDist[rg, Min[VertexDegree[rg]], Max[VertexDegree[rg]]][[All, 2]]}],
 TableHeadings → {None, {"Degree", "DegreeCount",
    "DegreeFrequency", "Cumulative"}}]
```

Out[1220]=



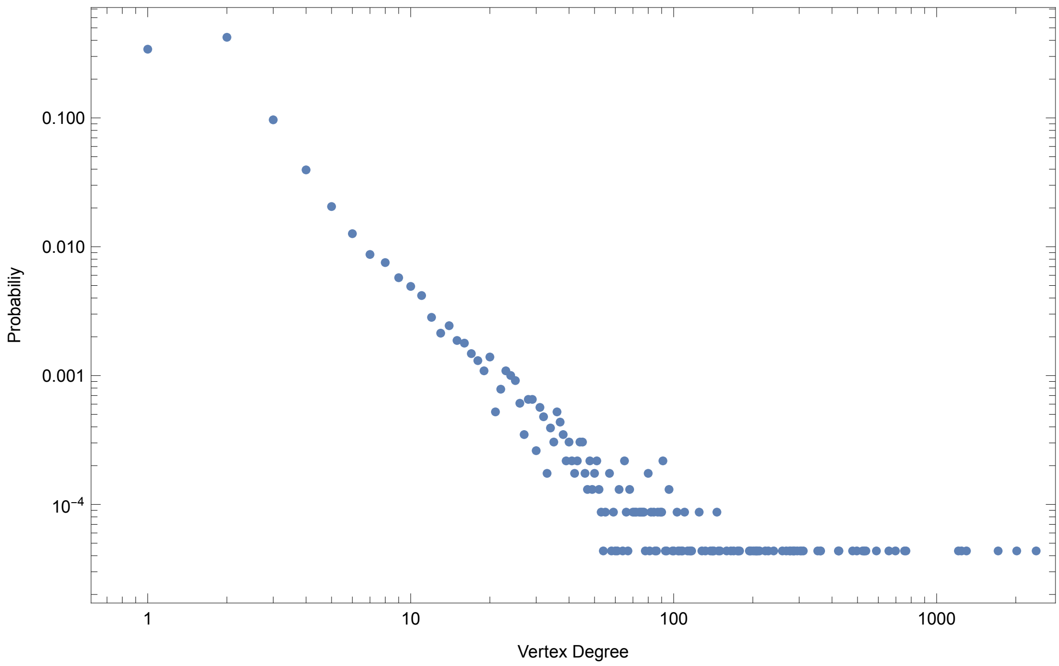
Out[1221]//TableForm=

Degree	DegreeCount	DegreeFrequency	Cumulative
2	7	$\frac{7}{10}$	1
5	1	$\frac{1}{10}$	$\frac{3}{10}$
7	1	$\frac{1}{10}$	$\frac{1}{5}$
8	1	$\frac{1}{10}$	$\frac{1}{10}$

In[1200]:=

```
ListLogLogPlot[DegreeFreq[ASgraph],
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True, PlotRange → All]
```

Out[1200]=

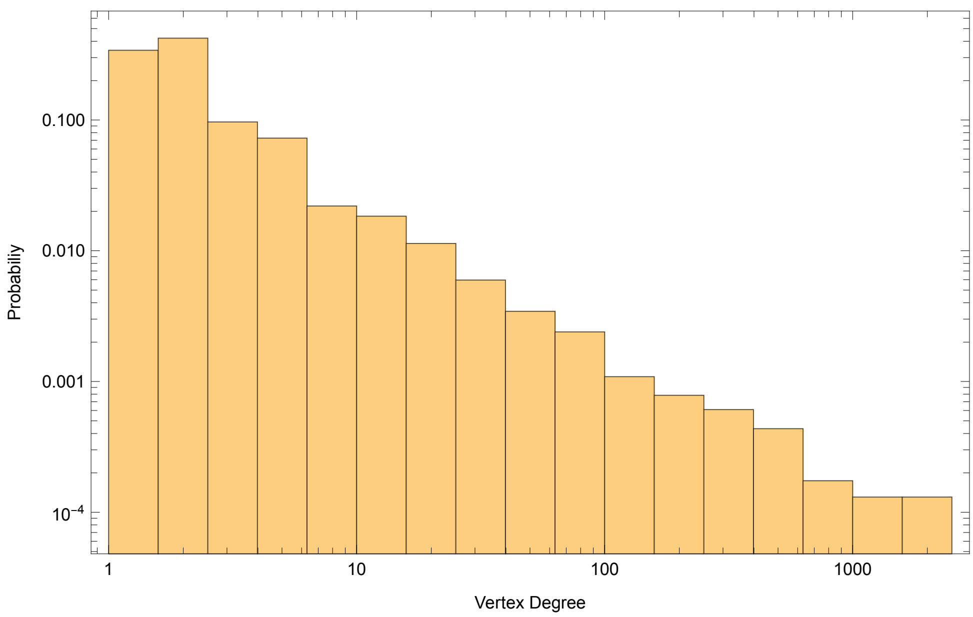


■ Logarithmic Binning (reduce noise)

In[1201]:=

```
Histogram[VertexDegree[ASgraph], "Log", {"Log", "Probability"},
 FrameLabel → {"Vertex Degree", "Probabiliy"}, Frame → True]
```

Out[1201]=



Cumulative Distribution

$$\hat{P}_k = \sum_{k'=k}^{\infty} p_k$$

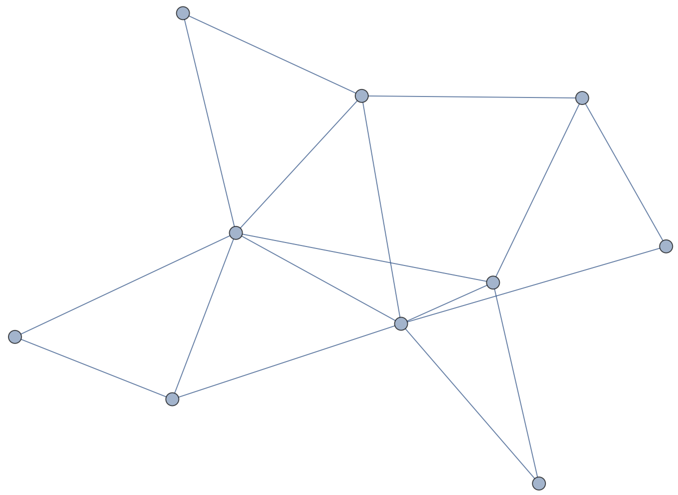
- Fraction of vertices that have degree k or larger
- Notice that for power law distribution

$$p_k = C\, k^{-\alpha}$$

$$\hat{P}_k = \sum_{k'=k}^{\infty} C\, k^{-\alpha} \simeq C \int_{k'=k}^{\infty} k'^{-\alpha}\, dk' = \frac{C}{\alpha - 1}\, k^{-(\alpha-1)}$$

- Also a power law!
- Easy to generate

```
rg = RandomGraph[BarabasiAlbertGraphDistribution[10, 2]]
TableForm[Transpose[
  {CumDigDist[rg, Min[VertexDegree[rg]], Max[VertexDegree[rg]]][[All, 1]],
   Sort[Tally[VertexDegree[rg]], #1[[1]] < #2[[1]] &][[All, 2]],
   Sort[DegreeFreq[rg], #1[[1]] < #2[[1]] &][[All, 2]],
   CumDigDist[rg, Min[VertexDegree[rg]], Max[VertexDegree[rg]]][[All, 2]]}],
 TableHeadings → {None, {"Degree", "DegreeCount",
    "DegreeFrequency", "Cumulative"}}]
```



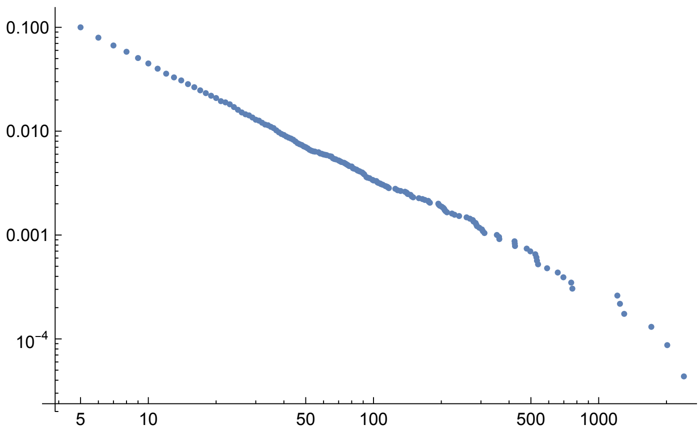
Degree	DegreeCount	DegreeFrequency	Cumulative
2	4	$\frac{2}{5}$	1
3	2	$\frac{1}{5}$	$\frac{3}{5}$
4	2	$\frac{1}{5}$	$\frac{2}{5}$
6	2	$\frac{1}{5}$	$\frac{1}{5}$

■ The Cumulative degree distribution of the AS graph

In[1202]:=

```
ListLogLogPlot[CumDigDist[ASgraph, 5, maxAS]]
```

Out[1202]=



■ Can we find the slop? The exponent

In[1203]:=

```
ASfit = Fit[Log[CumDigDist[ASgraph, 5, maxAS]], {1, x}, x]
```

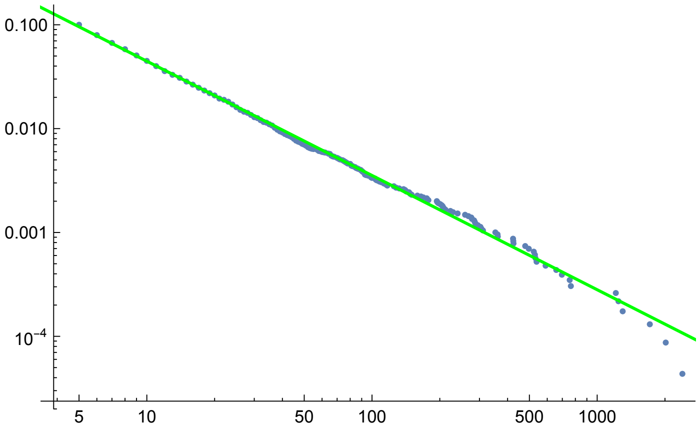
Out[1203]=

```
-0.576474 - 1.10031 x
```

In[1204]:=

```
Show[ListLogLogPlot[CumDigDist[ASgraph, 5, maxAS]],
 Plot[ASfit, {x, 1, 1000}, PlotStyle → Green]]
```

Out[1204]=



- So the slop (of the degree distribution) is about -2.1 and $\alpha \approx$ 2.1.

```
-ASfit[[2]][[1]] + 1
```

```
2.10031
```

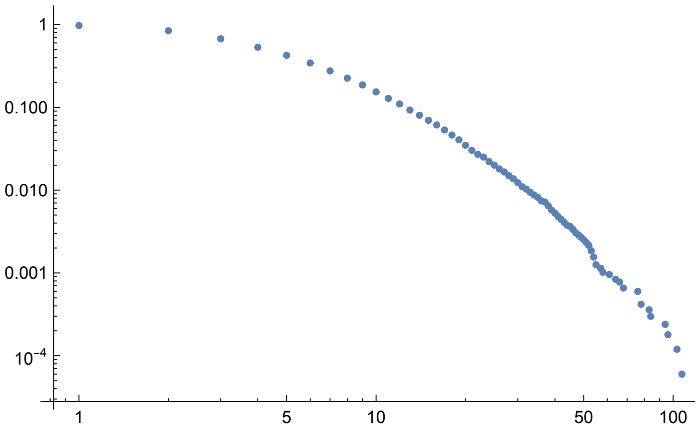
In[1205]:=

```
socialNet = ExampleData[{"NetworkGraph", "CondensedMatterCollaborations"}];
```

In[1206]:=

```
ListLogLogPlot[CumDigDist[socialNet,
  Min[VertexDegree[socialNet]], Max[VertexDegree[socialNet]]]]
```

Out[1206]=



In[1207]:=

```
Powerfit = Fit[Log[CumDigDist[socialNet, 5, 58]], {1, x}, x]
```

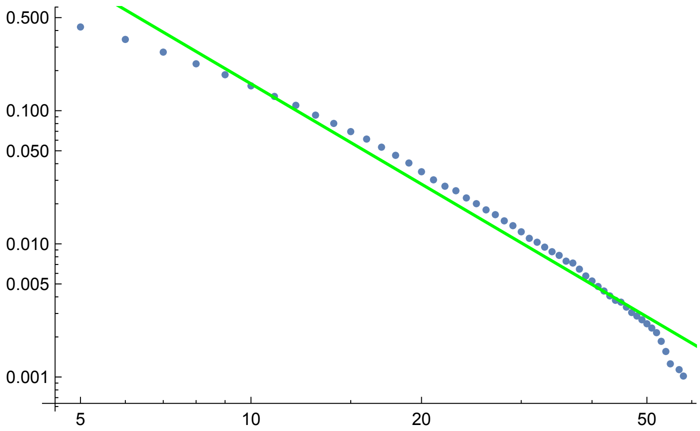
Out[1207]=

```
3.92705 - 2.50348 x
```

In[1208]:=

```
Show[ListLogLogPlot[CumDigDist[socialNet, 5, 58]],
 Plot[Powerfit, {x, 1, 58}, PlotStyle → Green]]
```

Out[1208]=



Properties of Power Laws Distributions

- Normalization

$$p_k = C\, k^{-\beta}$$

$$1 = C \int_{k'=k_{min}}^{\infty} k'^{-\beta}\, dk' = \left[\frac{C}{1-\beta}\, k^{-(\beta-1)}\right]_{k_{min}}^{\infty}$$

$$C = (\beta - 1)\, k_{min}{}^{\beta-1}$$

$$p_k = \frac{(\beta-1)}{k_{min}} \left(\frac{k}{k_{min}}\right)^{-\beta}$$

- First Moment

$$< k \geq C \int_{k_{min}}^{\infty} k \cdot k^{-\beta}\, dk =$$

$$C \int_{k_{min}}^{\infty} k^{-\beta+1}\, dk = \left[\frac{C}{2-\beta}\, k^{-\beta+2}\right]_{k_{min}}^{\infty}$$

- If $\beta > 2$

$$< k \geq \frac{C}{\beta-2}\, k_{min}{}^{-\beta+2} = \frac{\beta-1}{\beta-2}\, k_{min}$$

- Second Moment

$$< k \geq C \int_{k_{min}}^{\infty} k^2 \cdot k^{-\beta}\, dk =$$

$$C \int_{k_{min}}^{\infty} k^{-\beta+2}\, dk = \left[\frac{C}{3-\beta}\, k^{-\beta+3}\right]_{k_{min}}^{\infty}$$

- α must be s.t. $\beta > 3$ for a meaningful variance....

Preferential Attachment Model - Generating Power law networks

- Network Formation
- Nodes arrive one by one (people, pages, routers)
- Preferential Attachment (PA) - “rich get richer”
  - high degree nodes attracts new nodes
- New nodes connects to existing nodes with probability

$$\Pi(i) = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2m}$$

- If node i has twice the degree of node j than is has twice the probability to be chosen!
- Generates a scale free network (with $\beta = 3$)
- Show Example (netlogo)
- (aka) BarabasiAlbertGraphDistribution

Let’s generate the Internet....

In[1214]:=
```
PaAS = RandomGraph[BarabasiAlbertGraphDistribution[
    VertexCount[ASgraph], Round[EdgeCount[ASgraph] / VertexCount[ASgraph]]]];
```

In[1215]:=
```
VertexCount[PaAS]
```

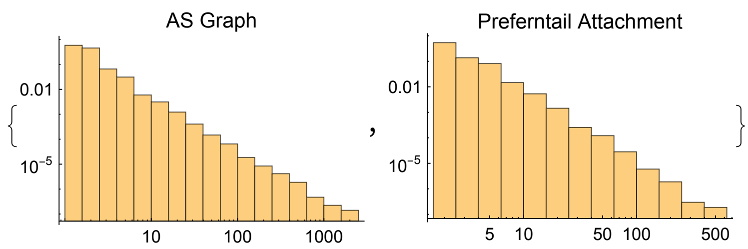
Out[1215]=
```
22 963
```

In[1216]:=
```
Mean[VertexDegree[PaAS]] // N
```

Out[1216]=
```
3.99974
```

In[1212]:=
```
{Histogram[VertexDegree[ASgraph], {"Log", 15}, {"Log", "PDF"},
  PlotLabel → "AS Graph"], Histogram[VertexDegree[PaAS],
  {"Log", 15}, {"Log", "PDF"}, PlotLabel → "Preferntail Attachment"]}
```

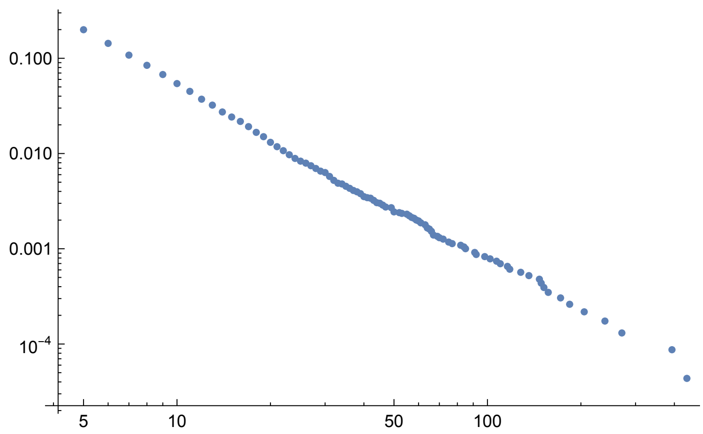
Out[1212]=



In[1235]:=
```
maxPaAS = Max[VertexDegree[PaAS]];
```

In[1236]:=
```
ListLogLogPlot[CumDigDist[PaAS, 5, maxPaAS]]
```

Out[1236]=



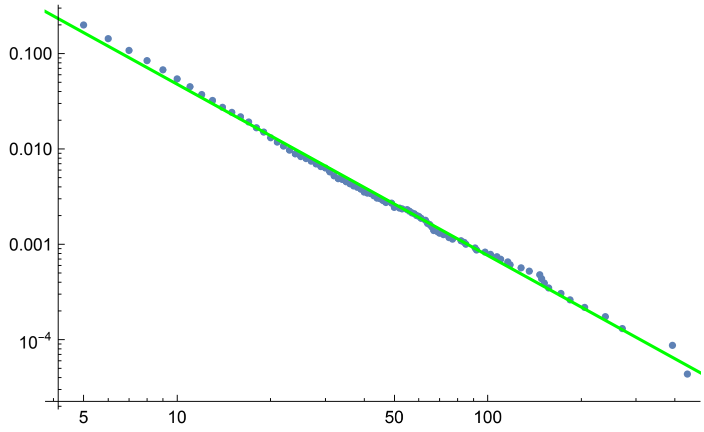
In[1237]:=
```
PAfit = Fit[Log[CumDigDist[PaAS, 5, maxPaAS]], {1, x}, x]
```

Out[1237]=
```
1.09582 - 1.79692 x
```

In[1239]:=

```
Show[ListLogLogPlot[CumDigDist[PaAS, 5, maxPaAS]],
 Plot[PAfit, {x, 1, 1000}, PlotStyle → Green]]
```

Out[1239]=



- α is:

```
-PAfit⟦2⟧⟦1⟧ + 1
```

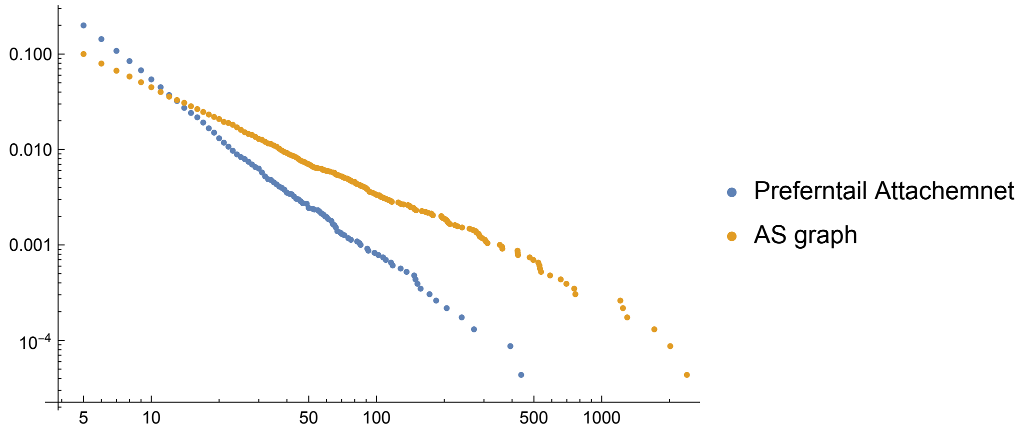
2.91542

- Close, but not the same....

In[1242]:=

```
ListLogLogPlot[{CumDigDist[PaAS, 5, maxPaAS], CumDigDist[ASgraph, 5, maxAS]},
 PlotLegends → {"Preferntail Attachemnet", "AS graph"}]
```

Out[1242]=



- One last thing....

```
N[GlobalClusteringCoefficient[PaAS]]
```

0.000741043

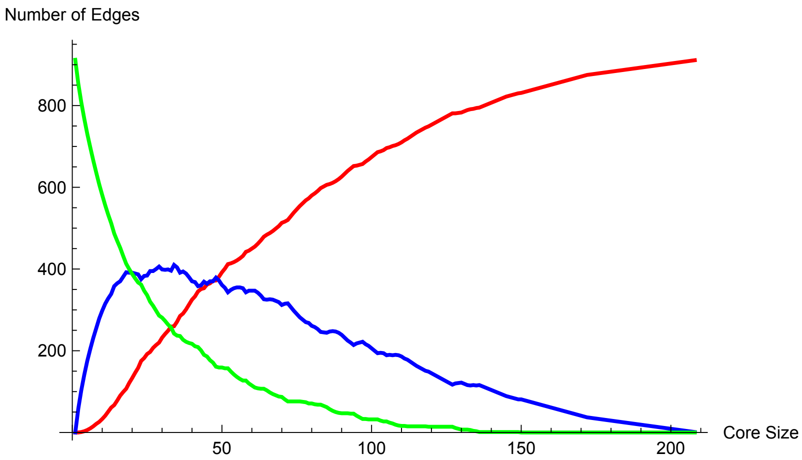
```
N[GlobalClusteringCoefficient[ASgraph]]
```

0.0111464

Class Homework

1. Find a network with power law degree distribution. Plot the degree distribution and try to find power exponent β.

2. Generate the symmetry point graph for PA with sevral sizes (on what fraction it is happends? what is the "core" fraction)



3. Add Preferential Attachment to the models you check “Network of Throne” with.