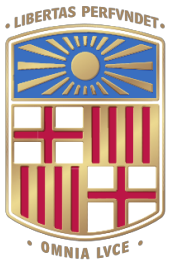

Practical Exercise 1: Structural Descriptors of Complex Networks

Complex Networks

Master in Artificial Intelligence

Students:
- Alam López
- Mario Rosas

Professors:
-Alex Arenas
-Sergio Gómez

Academic Year: 2023-2

Laboratory Group: 1

March 10, 2024

Contents

Structural Descriptors of complex Networks

1 | Introduction

In many fields and systems of study composed of individual parts with some relationship, the main focus is either the individual parts themselves or the connection between them. For social networks, for example, the first approach would consider the characteristics of each of the members, while the second approach would focus on the communication or interaction between their members. However, there is a third approach to be considered, as mentioned by M.E.J. Neman in his book, "Networks An Introduction", that is crucial for the behavior of the system, which is the pattern of connections between the components.

At first glance, it may not seem obvious, however, the actual structure of the pattern of connections, or the "shape of the network" itself can determine how its elements interact and can provide us with multiple insights about the behavior of the system. Over the years there have been multiple studies, to find the tools for analyzing the network representations and extracting these crucial insights of them.

Nowadays, there are multiple standardized file formats for representing and storing network data, such as GraphML format or Pajek NET Format, and some others. Also, there are multiple programming languages that have developed multiple built-in functions to read and analyze these formats like Python, Matlab, R, Julia, etc.

This first assignment is devoted to getting hands-on experience in the first steps of the process of dealing with a network, from reading the data, which in this case is in .NET format, to analyzing with a programming language the main structural descriptors seen in class for both the networks and the nodes, such as the Average Path length, Degree of nodes, assortativity, etc., and being able to interpret the results.

2 | Methodology

For the implementation of the three tasks in the assignment, even when our main coding knowledge was in Python, we decided to explore the Julia language. Our decision was made mainly to be able to compare the languages and libraries and the ease of use, and since the tasks were on an introductory level of complexity we did not oversee a possible risk on the delivery, on the other hand, we thought that it might be a good opportunity to explore the capabilities of Julia in the Networks domain and we could be very surprised on the result.

Even though the source code and all the outputs as tables, plots, and other results shown were implemented in Julia, mainly with the Graphs module, we also implemented the tasks in Python with the NetworkX library in parallel to be able to compare both the results of the tasks and the easiness of getting each task done. No specific results with NetworkX are provided in this report since it is not the main goal of the assignment, however, some reflections on the experience in working with both languages are detailed in the *Conclusions* section.

2.1 | Task A. Numerical descriptors of networks

We developed a systematic methodology to analyze complex networks. This analysis involved the calculation of key structural descriptors, crucial for understanding the intricate properties of the different networks provides as data to test our implementation.

Initially, our strategy commenced with the development of a function to accurately read and interpret network data stored in Pajek format files. The *read_network* function, was tasked with parsing network files to construct graph representations of the networks. It identified vertices, edges, and optionally, weights associated with the edges, incorporating them into a graph structure. This was achieved through by complementing the *Graphs* package *loadgraph* function with a line-by-line analysis of the file, distinguishing edge definitions and extracting relevant data such as source, destination, and weight for each edge. We additionally extracted the *vertex label* to be able to identify better each node in some networks such as the *airports_UW*.

After the construction of graphical representations, we turn to the computation of numerical descriptors, a crucial step in understanding the structural properties of networks. Tor this we created the *network_num_descriptors* function, the computations within this function includes the number of nodes and edges, degree distribution (minimum, maximum, and average degree), average clustering coefficient, assortativity, average path length, and network diameter. These descriptors help to get insights into the network's size, connectivity, clustering tendencies, degree correlations, navigability, and overall spread.

The structural network analysis can be done by running the *ExerciseA.jl* file which will perform the same descriptors analysis over all the networks contained in the folder. Results are presented in a tabulated format and saved into the *ExerciseA_Results.csv* file.

2.2 | Task B. Numerical descriptors of nodes

In this second taks the objective was to compute and present a detailed set of numerical descriptors for individual nodes within a specific network. The focus was on extracting and analyzing metrics that provide insights into the roles and importance of nodes within the network, highlighting the network's structure and connectivity from the perspective of its components.

For this task we implemented the *nodes_num_descriptors* function. By extracting the information of both the nodes and the links in the network, as well as the weights of each one, we were able to compute the correct computation of the properties of the nodes in the network. With the network graph constructed, we then proceeded to compute a suite of numerical descriptors for each node. These descriptors included:

- *Degree.* The number of connections a node has with other nodes.
- *Strength.* A measure of the node's connectivity weight, considering the total weight of a node's edges.
- *Clustering Coefficient.* Indicates how well the neighbors of a node are interconnected.
- *Average Path Length (ASPL).* The average distance of the shortest paths from a node to all other nodes in the network.
- *Maximum Path Length (LSPL).* The longest shortest path from a node to any other node.
- *Betweenness Centrality.* Reflects the extent to which a node lies on paths between other nodes.
- *Eigenvector Centrality.* Measures a node's influence based on the centrality of its neighbors.
- *PageRank.* Assesses a node's importance based on the structure of the entire network.

The results are presented in a tabulated format structured to include each node's identifier and its corresponding metrics. For nodes identified by specific vertex labels, we include in de index each vertex label instead of the node ID. This selective aggregation highlighted key nodes of interest, providing focused insights into their structural and functional roles within the network.

The node descriptors analysis for the requested network can be run using the *ExerciseB.jl*. Results are presented in a tabulated format in the console and saved into the *ExerciseB_Results.csv* file.

2.3 | Task C. PDF and CCDF of networks

In this task the main goal was to be able to plot the Probability Distribution Function (PDF) and the Complimentary Cumulative Distribution Function (CCDF) of the degrees of the nodes in the networks analyzed. The main challenge was to find the appropriate number of bins, ranging from 10 to 30, and also the scale to show the plot, linear or log-log scale.

To be able to select the appropriate visualization for both the PDF and CCDF, we decided that the optimal way was to create a custom function in Julia with parameters that could be easily manipulated to explore the different visualizations while changing the parameters to look for the best approach.

This function is ***degree_pdf(network,num_bins,log_scale, ccdf)*** which contains 4 parameters:

- **graph:** The name of the element that contains the network data
- **num_bins:** The selected number of bins to show in the histogram. Set to 10 by default.
- **log_scale:** Boolean parameter that plots a linear scale when set to false, and a log-log scale when set to true. It is set to false by default.
- **ccdf:** Boolean parameter that plots the PDF when set to false, and the CCDF when set to true. It is set to false by default.

Once the function was created, it was only necessary to follow a simple 3-step process to determine the optimal parameters for plotting both distribution functions.

First, an exploratory plot with the default parameters was run: **degree_pdf(network)**

With this first approach, which outputs a PDF in a linear scale with 10 bins, we could detect if the range of the degrees shown on the x-axis was either wide or narrow. From this execution, it was evident for all the networks if the scale that best fitted was linear or log-log.

Once the best scale was selected and therefore fixed in the parameter of the function, some iterations on the num_bins parameter were necessary to perform. We want to avoid having empty bins in the middle of the distribution and find a well-distributed probability as possible. Since the default parameter is 10 bins, a 5 step of increasing bins was performed: *degree_pdf(network,15,false,false)*, *degree_pdf(network,20,false,false)*, *degree_pdf(network,25,false,false)*...

When the best range was observed, we changed the step to 1 bin: *degree_pdf(network,20,false,false)*, *degree_pdf(network,21,false,false)*, *degree_pdf(network,22,false,false)*...

After this step, the best PDF plot of the network was found.

In the last step, we considered the same scale found in step 1 and the same number of bins found in step 2, and we only changed the type of plot to CCDF with the fourth parameter. Best PDF plot:

degree_pdf(network,22,false,false) -> degree_pdf(network,22,false,true)

We decided to keep the same parameters to maintain the consistency of the number of bins in the PDF and CCDF plots, however, it is important to note that in some cases we noted that a smother plot of the CCDF when increasing the number of bins for the CCDF.

In the *Results & Discussion* section a well-detailed of this process with all the plots for each step is provided for the first network *"model/ER5000k8.net"*. However, for the sake of the clarity and length of the report, for the other four networks only the best plots of PDF and CCDF are shown, but the same process was followed.

3 | Results & Discussion

3.1 | Task A. Numerical descriptors of networks

After performing the network numerical descriptors analysis over all the given networks, the results obtained are presented in table 1.1. This exercises helped us to understand the general structure of the networks and their differences between the ones classified toy, model, and real-world systems. The results in the presented table include node count (# Nodes), edge count (# Edges), degrees (Dmin, Dmax and Davg), average clustering coefficient (ACC), assortativity (Assort), average path length (APL), and diameter.

The variance in *node counts* and *edge counts* across networks underscores the diversity in network size and connectivity density, in the toy models usually the number of nodes is lower but also the number of edges, which in general indicated less complexity compared to the model and real networks. For instance, networks such as *PGP* and *ER5000K8* exhibit a large number of nodes and edges, indicating complex systems with potentially rich interaction patterns. On the other hand, smaller networks like *circle9* and *star* offer a more focused view on localized structural properties.

The *degree metrics (minimum, maximum, and average)* provide valuable information about the node connectivity within each network. Networks with a high maximum degree, such as *airports_UW* (Dmax = 250), highlight the presence of highly connected nodes, which can be of that can strongly influence the dynamics and information flow. The variation in average degrees allows us to clarify the range of connectivity patterns, making possible to identify from densely connected networks (e.g. *256_4_4_4_13_18_p*) to more sparse configurations (e.g. *SF_1000_g2.5*).

According to out results *Average Clustering Coefficient* values vary significantly, with *rb25* showing an extremely high clustering coefficient, indicative of a tightly-knit structure, whereas networks like *ER5000k8* display very low ACC values which suggest it has sparse connectivity. This descriptor helps with the understanding of the potential for cluster formation within the network.

Using the *Assortativity* we can measure the tendency of nodes to connect with other nodes that are similar in some way, such as degree. Networks like (20x2+5x2) show strong positive assortativity, indicating that its nodes are linked to others with similar degrees, while for example (zachary_unwh) which displays negative assortativity, suggesting a propensity for nodes to connect with others of dissimilar degrees. For the assortativity value some networks exhibit values *NaN* this can occur when there is no variability in degree values between nodes, which could probably mean, for example, that each node is connected to all other nodes.

The *Average Path Length* and *Diameter* provide insights into the fact that there are certain types of networks where most nodes can be reached from every other by a small number of steps, despite the network's size.,Shorter path lengths and diameters indicating an ease of navigation and communication across the network. For example, the *PGP* network presents a large diameter which suggests it spans a broad scope. In general the analzed networks present a wide range of scenarios, from the highly clustered and assortative networks to the sparse and node-centric topologies others.

Table 1.1: Networks Numerical Descriptors

Index	# Nodes	# Edged	Dg(min)	Dg(max)	Dg(avg)	ACC	Assort.	APL	Diameter
20x2+5x2	50	404	4	22	16.16	0.9716	0.9186	2.3878	4
256_4_4_2_15_18_p	256	2274	15	23	17.7656	0.7331	0.0286	2.7821	5
256_4_4_4_13_18_p	256	2299	10	25	17.9609	0.5113	0.0007	2.6511	4
BA1000	1000	3990	4	115	7.98	0.0354	-0.0542	3.1833	5
ER1000k8	1000	3956	1	17	7.912	0.008	-0.0168	3.5698	6
ER5000k8	5000	19980	4	17	7.9918	0.0014	-0.0555	4.3797	6
PGP	10680	24316	1	205	4.5536	0.2659	0.2382	7.4855	24
SF_1000_g2.5	1000	1905	2	30	3.81	0.0096	0.02	4.6149	10
SF_1000_g2.7	1000	1668	2	24	3.336	0.0067	-0.002	5.4688	12
SF_1000_g3.0	1000	1517	2	26	3.034	0.0052	-0.0085	5.9651	13
SF_500_g2.7	500	859	2	22	3.436	0.0078	-0.0256	4.8759	12
airports_UW	3618	14142	1	250	7.8176	0.4957	0.0462	4.4396	17
circle9	9	9	2	2	2	0	NaN	2.5	4
dolphins	62	159	1	12	5.129	0.259	-0.0436	3.357	8
graph3+1+3	7	8	2	3	2.2857	0.6667	-0.6	2.1905	4
graph4+4	8	13	3	4	3.25	0.875	-0.0833	1.8571	3
grid-p-6x6	36	72	4	4	4	0	NaN	3.0857	6
homorand_N1000_K4_0	1000	2000	4	4	4	0.002	NaN	5.64	9
homorand_N1000_K6_0	1000	2994	5	6	5.988	0.0038	0.1919	4.1913	6
rb125	125	410	4	100	6.56	0.8373	-0.173	2.3032	4
rb25	25	66	4	20	5.28	0.9023	-0.1635	2.0333	4
star	9	8	1	8	1.7778	0	-1	1.7778	2
wheel	9	16	3	8	3.5556	0.6243	-0.3333	1.5556	2
ws1000	1000	3000	3	13	6	0.0044	-0.0999	4.0913	6
ws2000	2000	6000	3	13	6	0.0033	-0.0762	4.5111	7
zachary_unwh	34	78	1	17	4.5882	0.5706	-0.4756	2.4082	5

3.2 | Task B. Numerical descriptors of nodes

These descriptors presented in the table 1.2 include Degree, Strength, Average Shortest Path Length (ASPL), Longest Path Length (LPL), Clustering Coefficient, Betweenness, Eigenvector Centrality, and PageRank, collectively paint a detailed picture of each node's importance, connectivity, and potential influence within the network.

The node descriptor analysis displays that airports shuch as PAR, LON, and FRA exhibit high *degrees* and *strength*, indicating they have numerous direct connections and handle a large volume of traffic, respectively. This suggests these airports serve as major hubs within the network, facilitating a significant portion of the network's connectivity and flow. The high strength values, particularly for *LON* airport, underscore their role in handling extensive traffic.

The values for *clustering coefficient* are varied accross the studied nodes, airports like BCN and WAW show higher values, indicating that these nodes are part of tightly-knit communities within the network. The *Betweenness centrality* values seems particularly high for nodes like PAR and NYC, suggesting that these airports can be important points of flow within the network, acting as bridges between different parts of the network.

The *Eigenvector centrality* and *PageRank* offer insights into the influence and importance of nodes within the network. Nodes like LON and PAR, with high eigenvector centrality are well-connected in general but also connected to other well-connected nodes, amplifying their influence. *PageRank* values contribute to this analysis, with LON and PAR again appear significant because of their structural importance.

For the *ASPL* and *LPL* the TBO and ZVA airports show extremes in high values in ASPL, suggesting that while some nodes are relatively isolated, they are still reachable within the network.

From this descriptors in general we can identify that major hubs like PAR, LON, and FRA are central to the network's operation, offering robust connectivity and influencing overall network dynamics. In contrast, nodes like TBO and ZVA represent less connected but essential for network completeness and accessibility.

Table 1.2: Networks Numerical Descriptors

Index	Degree	Strength	ASPL	LPL	Clust Coeff	Betweeness	Eigenvector	PageRank
PAR	250	1.02E+06	2.68841581	10	0.08915663	0.09342038	0.1803367	0.00609169
LON	242	1.46E+06	2.63588609	10	0.11234183	0.08498898	0.20040458	0.00521484
FRA	237	697513.5	2.68288637	10	0.11696346	0.06557771	0.1955994	0.00491882
AMS	192	481335	2.73209842	10	0.14283377	0.04049213	0.17150222	0.00397053
CHI	184	1.33E+06	2.80868123	11	0.13417676	0.0444435	0.1380112	0.00401931
NYC	179	1.52E+06	2.70915123	11	0.15755445	0.06928349	0.1605246	0.00385891
ATL	172	1.13E+06	2.91622892	11	0.1378349	0.02489618	0.12197559	0.00377999
HOU	144	654154.5	2.98396461	11	0.16336441	0.01745704	0.0961918	0.00330531
BCN	80	289105	3.27398396	11	0.32848101	0.0019323	0.08919432	0.00158248
WAW	55	86836.5	3.24440144	11	0.45858586	0.00155695	0.07523491	0.00114297
CHC	20	64158.5	3.5662151	10	0.25263158	0.00336745	0.00418821	0.00084705
DJE	20	10198.5	3.57920929	11	0.7	0.00014586	0.03184926	0.000405
ADA	7	10704	3.63339784	11	0.71428571	1.32E-05	0.0106939	0.00018467
AGU	7	7678	3.66546862	11	0.76190476	5.76E-06	0.00512863	0.00019228
TBO	2	234	4.58446226	12	1	0	0.00012311	0.000122
ZVA	1	19	7.57727398	15	0	0	0	0.00013394

3.3 | Task C. PDF and CCDF of networks

As it was mentioned in the previous section, the same methodology was followed for each of the 5 networks proposed. However, only a detailed explanation will be given for the first network, while for the rest of them, just the best plot for the PDF and the CCDF will be provided.

The first step was to run the network in the custom function without any parameters, which means that the plot will be a PDF in linear scale and 10 bins. As can de seen in the top left image in figure 1.1. In this case, is evident that the scale that was the most proper for this network is the linear one instead of the log-log. However, the number of bins could not be shown as the specific 10 as indicated in the function. This when further investigation, could be a result of the specific number of different degrees in the network. What we can get from this firs run, is that the linear scale is the more adequate for the network.

Therefore, an increase of the number of bins was needed. As shown in fig 1.1 in the top right and bottom images, the number of bins was increased to 15 and to 25. Where it is very clear that we empty bins while increasing the number of them. So the best range to look for the correct number of bins was between 15 and 25, which in fact, ended up being 15.

Just to give a little more explanation on the idea that only by running the function for the first time we can properly say that the right scale was linear and not log-log. In fig. 1.2 we can observe the behavior of the different scales while selecting the same amount of bins: 15. In the left we can note

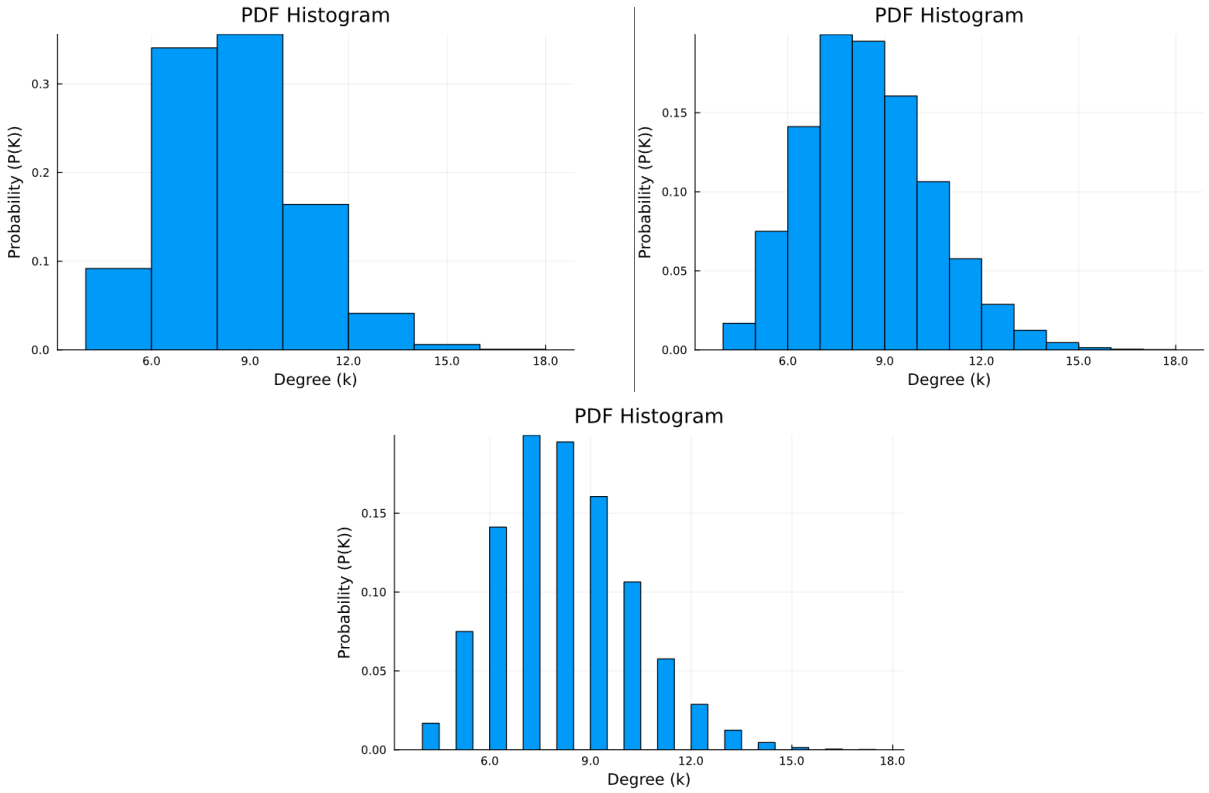

Figure 1.1: Network: ER5000k8, Scale: Linear. Comparison of number of bins. Top left: 10, Top right:15, Bottom:25

a smooth distribution with the linear scale, while in the right there are big gaps between the bins in the log-log scale.

After having both the scale and the specific number of bins. It is only a matter of changing the type of distribution to CCDF with the fourth parameter of the function to be able to plot the right CCDF. This specific decision was made to maintain the consistency between the number of bins between the two distributions. However, as it can be seen in fig.1.3 even tough a greater number of bins was not a good approach for the PDF, it may not seem as bad in the CCDF. This, of course, is due to the nature of the complementary cumulative distribution function itself which does not allow to have empty bins since the result is cumulative, and this results in a smoother curve on the right figure with 25 bins, than the one on the left with 15 bins. It is a good topic for further analysis of the networks if it is more valuable to maintain the same number of bins in both distribution functions or have a smoother function.

Finally, after these 3 steps, we can see that the chosen parameters for the first network are: linear

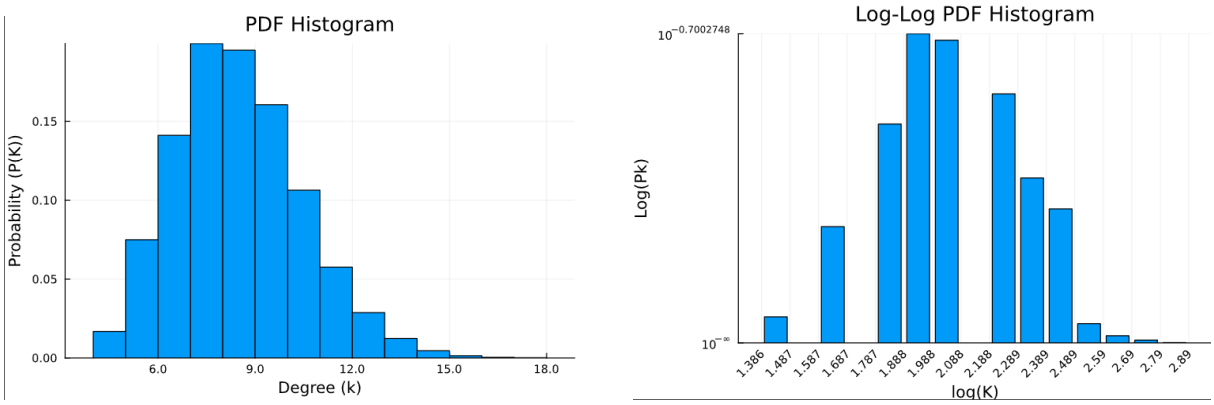

Figure 1.2: Network: ER5000k8, Bins: 15, Comparison of linear scale vs log-log scale. Left: Linear, Right:Log-log

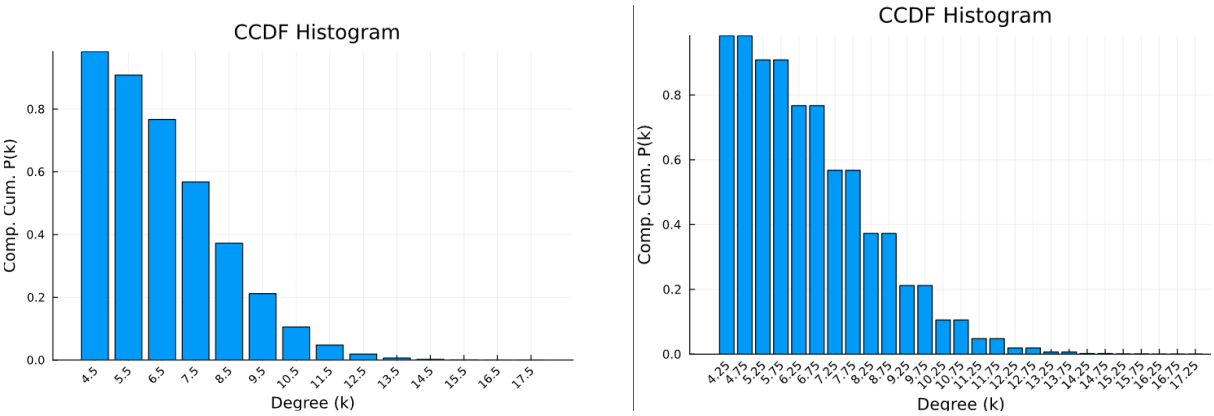

Figure 1.3: Network: ER5000k8, Scale: Linear, Comparison of num bins. Left: 15, Right:25

scale and 15 bins, as shown in the fig1.4. Having the Probability Distribution Function on the left, and the Complementary Cumulative Distribution Function on the right. This specific arrangement of figures with the final parameters is the one that we have adopted for the other networks and the ones that are shown in further pages.

To conclude with the analysis of the first network we can add that from the PDF histogram it is clear that since the scale is linear, there is not a wide range of degrees within the network. Also a bell-shaped can be noted on the data, with some outliers on the right of it. That suggests a normal distribution of the degrees of the nodes that it is centered in the 7-9 degrees, this is also clear in the CCDF histogram where we can see an accelerated decrease of the probability in the same range. A very useful piece of information that we are missing is what specifically represents this network, both the nodes and the edges. Using this analysis in conjunction with the domain knowledge we could extract more insightful conclusions.

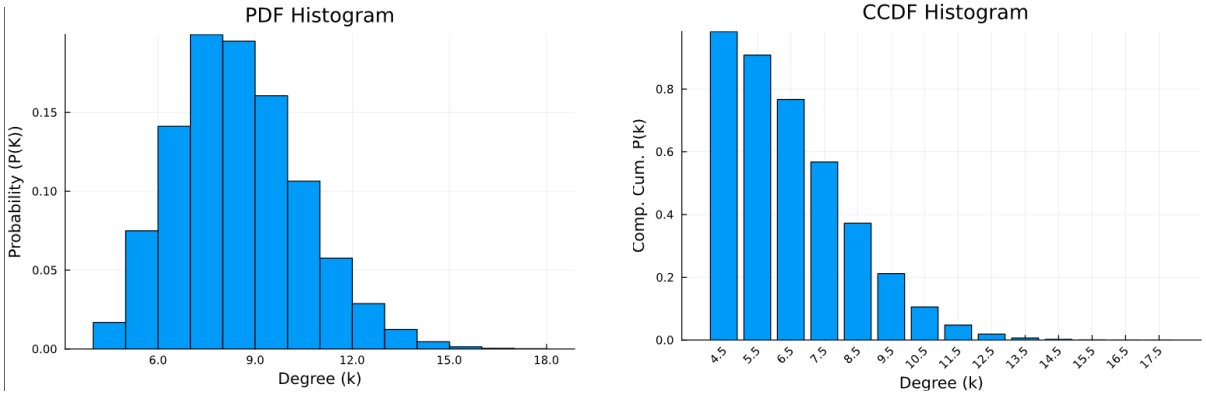

Figure 1.4: Network: ER5000k8, Bins: 15, Scale: Linear. PDF plot on the left and CCDF on the right

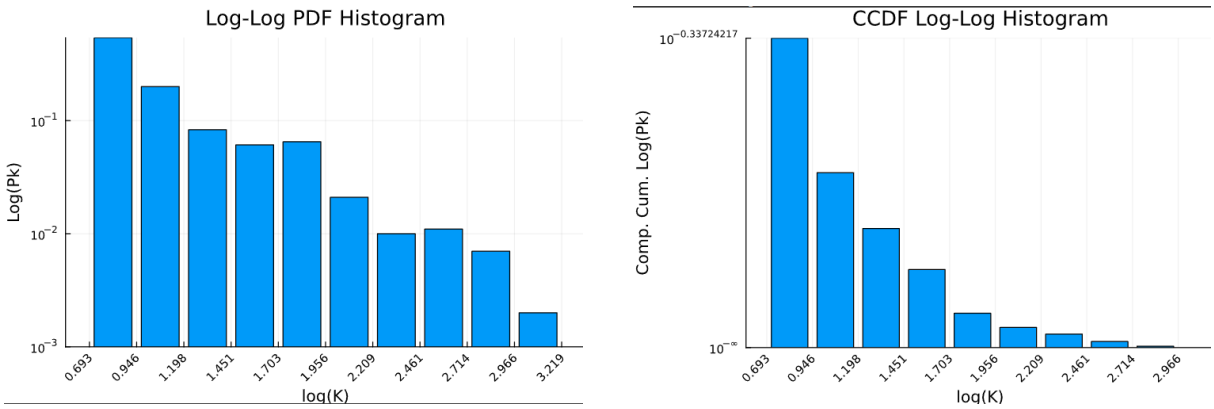

Figure 1.5: Network: SF_1000_g2.7, Bins: 10, Scale: Log-log. PDF plot on the left and CCDF on the right

As it can be seen in fig1.5, the network Network: SF_1000_g2.7, the scale is log-log and 10 bins. This suggests a very spread distribution of the degree of the nodes having a more concentrated number of nodes within the lower degrees. We have a positive skewed distribution since we have a longer tail on the right. As it can also be seen in the CCDF histogram.

As it was also said in the previous network, a very useful piece of information that we are missing is the real representation of the nodes and edges of the networks, that could help us to better interpret the results.

In fig1.6, the network Network: ws2000, the scale is linear and 10 bins. This suggests a narrow distribution of the degree of the nodes, presenting a normal distribution with higher concentration 5 and 7 degrees, and slightly skewed to the right as can be observed in the elements with degrees higher than 11. This is also present in the CCDF histogram.

For the fourth network on fig1.7, the network Network: airports_UW, the scale is log-log and 11 bins. This suggests a very spread distribution of the degree of the nodes having a more concentrated

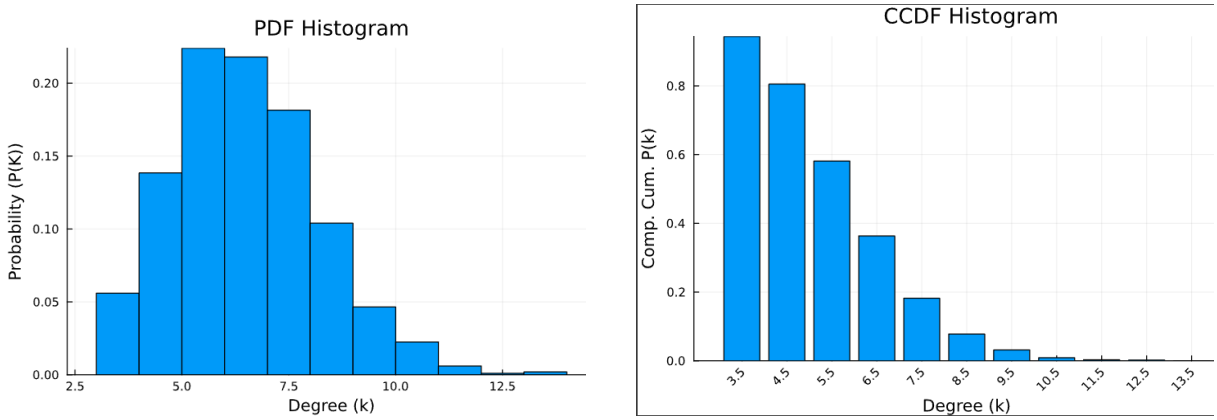

Figure 1.6: Network: ws2000, Bins:10, Scale: Linear. PDF plot on the left and CCDF on the right

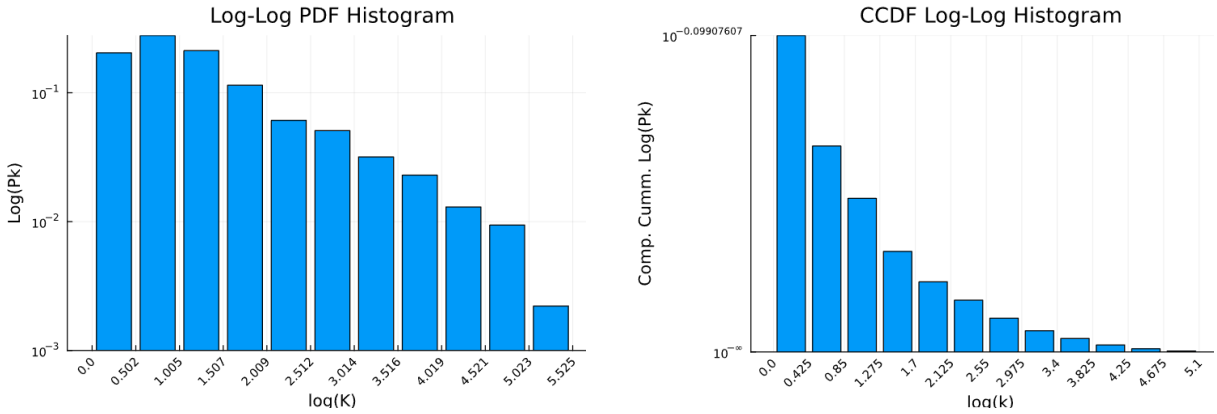

Figure 1.7: Network: airports_UW Bins: 11, Scale: Log-log. PDF plot on the left and CCDF on the right

number of nodes within the lower degrees. It is interesting how the second bin from left to right is the higher one.

In contrast with other networks, it is clear the element that the nodes and the edges are representing. In this case, the nodes represent the airports while the edges represent the connection between them. Is interesting to see how even when we have a log-log scale the distribution looks more uniform and smoother than in network 4. We can interpret that there is a higher number of airports that have fewer connections, and there are few airports that have a higher number of connections.

Therefore, we could then rate these airports are the most important in the case of a number of passengers and then are the ones that have the highest flow of people and money.

As it can be seen in fig1.7, the network Network: PGP, the scale is log-log and 14 bins. The number of bins being greater in the log-log scale than the previous networks suggests an even wider distribution of the degree of the nodes. Also the last bin seen in the PDF figure, shows how the

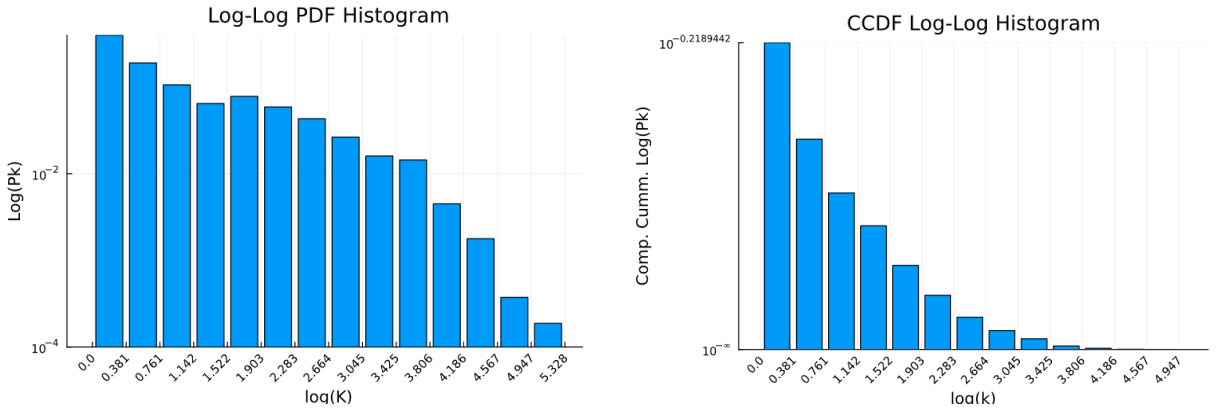

Figure 1.8: Network: PGP, Bins: 14, Scale: Log-log. PDF plot on the left and CCDF on the right

number dramatically decreased being a very small amount of node with the higher degrees.

4 | Conclusions

This assignment provided a comprehensive and deeper understanding of the analysis of complex network structures, emphasizing both overall network properties and the detailed roles of individual nodes. This is a very important step in networks since it is necessary for being able to implement more complex tasks such as the prediction of the behavior of the network. With a deep examination of both network and node-specific metrics within varied network examples, we gained profound insights into the connectivity patterns of these networks. The analysis highlighted the crucial balance between highly connected hubs and peripheral nodes, showcasing their collective importance in ensuring robustness and efficient information flow across networks.

This exercise helped us to get a better understanding and hands-on experience of complex network dynamics and demonstrated the applicability of complex network analysis in real-world systems, such as air transportation networks. The creation of histograms also contributed to a deeper and more visual understanding of the complexities of networks in various domains. The skills and knowledge gained through this assignment allow us to improve our analytical skills and contribute to our understanding of complex systems.

Also, we highlight the relevance of the domain knowledge of the network for both the nodes and the edges. Since it is very important for the deeper interpretation of the analysis.