Chun-Li Chuang, Nehar Poddar
MATH 7241
Professor Christopher King
2019/12/02

Project Report: Time Series of Wind Speed

Source

Before we started the search of the dataset, we had a mindset to search for a dataset that has an appropriate amount of entries, as well as enough number of states, while the general pattern of the time series not showing any upward or downward trend. However, as we started to look for the right dataset for our project, we realized that it was harder than we expected to find the specific dataset,as none of them were random enough to follow the markov chain trend. Luckily, we stumbled upon the dataset of application energy prediction. One of the variables measured the wind speed of Chièvres, Belgium. The data is collected on a ten-minute interval over 4.5 months and is measured by the Chièvres Airport Weather Station.

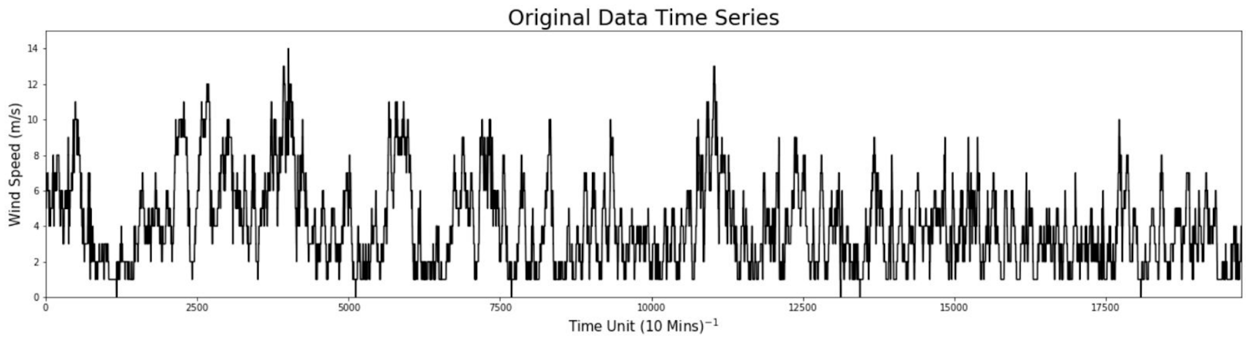
Application

The following Markov Chain can be used to predict and analyse the average wind speed, the rate at which it changes, the range it covers, the standard deviation from the mean, and also the number of times it reaches a certain speed. These values can be used in various ways :

1. To calculate the energy that can be generated using windmills.
2. Create wind maps
3. Plan tall buildings
4. Predict climate conditions

Data Preparation

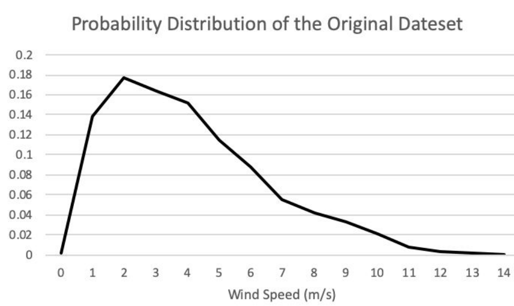
The dataset has a total of 19,735 entries. The wind speed is measured in meters per second. The value should be continuous in theory. However, the Chièvres Airport Weather Station presented the data in an increment of ⅙ m/s. The minimum value of this dataset is 0 m/s while the maximum is 14 m/s. This means that there are a total of 85 possible values within the dataset. To simplify the values, we rounded each entry to the nearest integer. This caused the data to have a total of 15 possible values, which was perfect for us to set them as the states of our Markov chain. The dataset did not have any erroneous entry, therefore there was no need to remove any of them. We imported the cleaned data and completed our analysis using Excel and Python. The time series plot is shown below.



By observing the diagram above before the analysis, we can see that the overall pattern of the time series shows minimal upward or downward trend. We have assumed that the dataset shows how wind speed at one point is independent from that of another point.

Empirical Distribution / Transition Matrix / Stationary Distribution

After importing to Excel, we used the COUNTIF function to compute the occupation frequencies. We then used the line chart within Excel to plot out the probability distribution and is shown below.

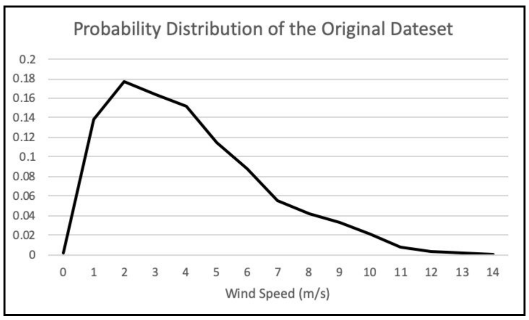


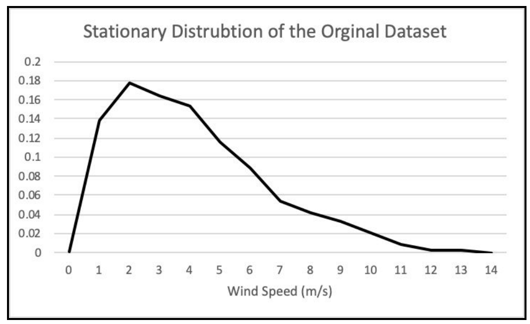
To find the transition matrix, we began first by formatting the values into two digits (Eg.: 1 was formatted as “01”) We then combined the current value with its next value together to create one entry (Eg.: 9 to 10 was represented as “0910”). This will then output a total of 19,734 entries since the last entry will not have a new state to jump to. We then used the function COUNTIF to tally up each pair of states. We then summed up the all the occupation frequencies that shares the same initial state. By dividing the jump frequencies by its corresponding occupation frequency, we created our transition matrix as the following.

Transition Matrix															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0.76923077	0.23076923	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.00219861	0.93917186	0.05862953	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0.04585841	0.89595873	0.05818286	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0.06288724	0.87639405	0.06071871	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0.06522463	0.87221298	0.0625624	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.08307556	0.85152452	0.06539991	0	0	0	0	0	0	0	0
6	0	0	0	0	0.00057471	0.08505747	0.85172414	0.06264368	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0.10110294	0.83088235	0.06801471	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0.08883553	0.83793517	0.07322929	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0.09284627	0.83866058	0.06849315	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.11002445	0.84352078	0.04645477	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0.11656442	0.8404908	0.04294479	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0.10769231	0.84615385	0.04615385	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0.08108108	0.89189189	0.02702703
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5

The transition matrix shows the pattern that each states (except state 4) is only able to jump to its current state or states that are 1 higher or lower than its current state. This simplified the calculation of the stationary distribution. However, we used `numpy.linalg.solve` within Python to solve the system of equations and get the following result.

Stationary Distribution	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
w	0.001	0.139	0.178	0.164	0.153	0.115	0.088	0.054	0.042	0.033	0.02	0.008	0.003	0.002	0



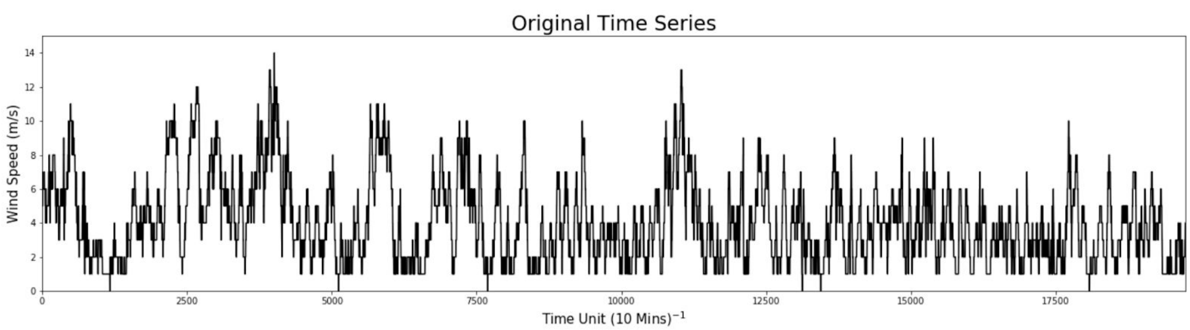


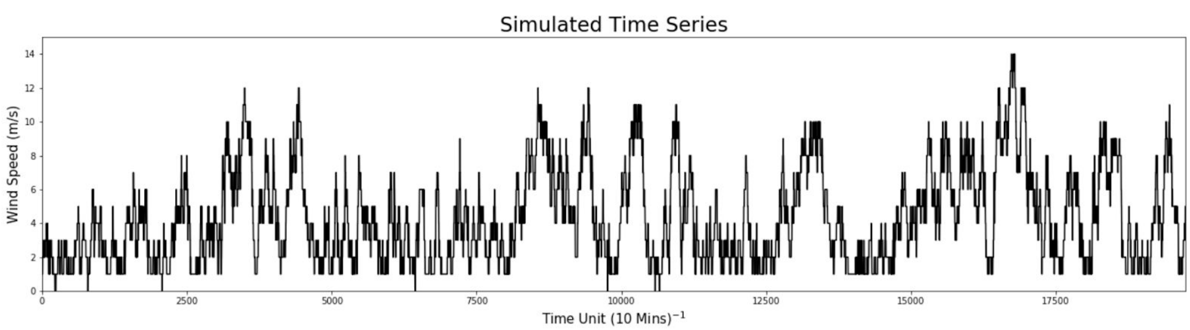
After plotting the stationary distribution and comparing with the empirical distribution, it is not hard to notice the strong similarity between the two with both distribution skewed right and peak at state 2. However, there’s a slight difference between the two at state 12, possibly due to rounding.

Simulation

The simulation is built using Python.

Here is a comparison of the original time series and the simulated result

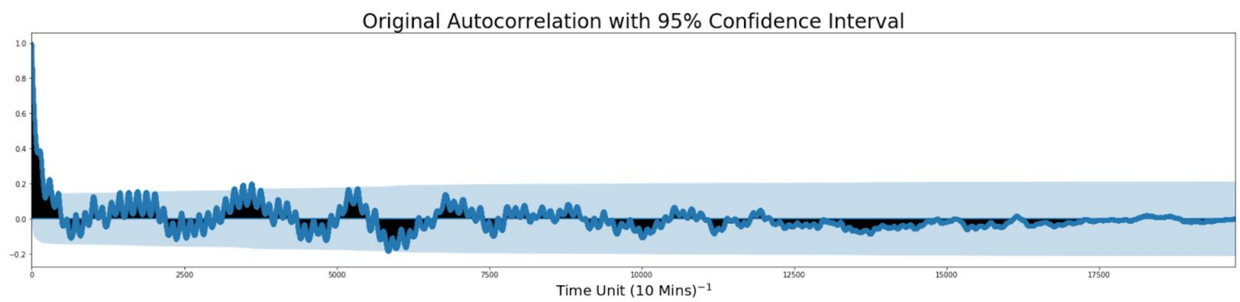


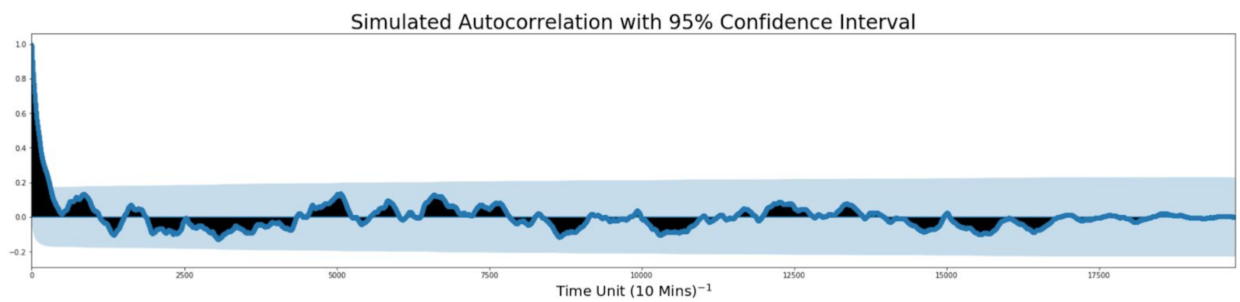


Observing the two diagrams before any analytical work, we believe that the simulation performed well in recreating the time series. The overall pattern of the simulated time series shows no trend. Still, one flaw we noticed within the simulated time series is that local spikes happen less often compared to the original time series. In other words, the original entries increases and decreases more dramatically compared to the simulated entries.

<u>Autocorrelation</u>

We analyzed both time series by using the autocorrelation function. This was implemented through Python using available package function `statsmodels.graphics.tsaplots.plot_acf`. The result is shown as follows.





As we can see above, there are some similarities and differences between the two graphs. Both time series have autocorrelations within the confidence interval after a similar size of lags and mostly stayed within the confidence interval as the number of lags increases. This tells us that the spikes of both models are mostly not statistically significant within the 95% significance level, and therefore indicates that wind speed is largely independent of each other. This agrees with what we observed earlier, and our simulation successfully recreated this characteristic. However, comparing the two autocorrelations more carefully, we noticed that the original dataset shows a zigzag pattern as the number of lags increases while the simulated model fails to do so. We believe that the pattern tells us that wind speed has minor correlations within short intervals but independent when viewing within a large interval.

Conclusion
In this project, we cleaned the data, categorized them into 15 states, analyzed the pattern and generated a transition matrix. We then compared the probabilities of the data occurring and the transition matrix generated and found them to be nearly identical. We found out how the wind would behave by calculating the stationary distribution and also generated a similar markov chain by simulating the transition matrix. Finally, we analyzed both the original and the simulated time series by the autocorrelation function to verify the accuracy of the simulation.

To conclude, through the use of Markov chains, we have analyzed the behavior of wind flow. By using Markov chains, we can predict the probability of the magnitude of wind speed. Although the wind speed varies randomly, the overall trend can be analyzed by converting the data into a transition matrix and calculating the stationary distribution. Through the transition matrix, we could analyze that the chain generally jumps or moves to its immediate neighbors and hence is denser near the diagonal, which is true as the wind speed in reality changes gradually and cannot jump states. The stationary state tells us that the wind speed visits state 1 to 4 the most and the frequency of the higher states gradually decreases.

Do you consider that the Markov chain method produces a good model for this time series?

Whenever we simulate the Markov chain to produce a time series, it is going to be different due to the randomness involved. Although we can not predict exactly what the wind speed is going to be like, we were able to get a sense of the overall behavior over a larger span of time. Hence, we think Markov chain is a good model for approximations and assumptions. Still, it will not satisfy with 100 percent accuracy at any given point of time. To state another important observation, the original autocorrelation follows a wriggly pattern and has steeper ridges as it is in a way dependent on the previous states but the simulation is completely independent and random. The wind speed phenomenon is not independent of the previous states as shown by the almost diagonal transition matrix and the wriggly autocorrelation. Thus, we conclude that Markov chain is a good model for our data set but not the perfect way to represent it as the data set does not completely follow the Markov chain property.