GIS-BASED SITE SELECTION FOR WIND FARM IN KYIV REGION, UKRAINE AND ASSESSMENT OF WIND REGIMES USING HISTORICAL DATA AND CMIP6 CLIMATIC PROJECTIONS

Olga Kostur (1894160)

This dissertation is submitted as an Independent Geographical Study as a part of a BSc degree in Geography at King's College London

APRIL 21, 2021

KING’S COLLEGE LONDON

UNIVERSITY OF LONDON

DEPARTMENT OF GEOGRAPHY

INDEPENDENT GEOGRAPHICAL STUDY (6SSG0610)

I, Olga Kostur, hereby declare (a) that this dissertation is my own original work and that all source material used is acknowledged therein; (b) that it has been specially prepared for a degree of King’s College London; and (c) that it does not contain any material that has been or will be submitted to the Examiners of this or any other university, or any material that has been or will be submitted for any other examination.

This Dissertation is 9980 words.

Signed: Olga Kostur

Date 21/04/21

Abstract

Maximising the proportion of renewables in energy systems is an essential aim for each nation, given the accelerated depletion of fossil fuels and growth of greenhouse gas emissions. In Ukraine, wind energy has considerable unused potential, contributing significantly to climate change mitigation targets if exploited. This study explored the possibility of an embedded energy approach for the Kyiv region. By conduction a GIS-based suitability analysis, legally, socially, and environmentally appropriate sites for wind farms were selected. Using the Weibull distribution function to examine historical and CMIP6 future projections of wind speeds, sites’ potential and sustainability of energy production was examined.

Acknowledgments

Firstly, I would like to thank my mother – Nataliia Kostur, for supporting my decision in changing my architectural career to pursue a Geography degree at Kings College London. I can say with confidence that I made the most of this education.

Secondly, I would like to thank my supervisor Kristopher Chan for having intellectually stimulating discussions, providing me feedback and being honest about my weaknesses and mistakes.

Finally, I would like to thank my partner, Kostiantyn Kovalchuk, for your love, understanding and support during the most challenging parts of this work. Without you in my life, I would never stay as positive, motivated, and focused. You earn this degree as much as I do.

Table of Contents

List of figures

List of tables

List of Equations

1. Introduction

Non-renewable energy is responsible for 49% of all greenhouse gases emitted globally, contributing substantially to the current level of global warming: 1°C (Panda, 2021). Furthermore, Kalair (2019) predicted that fossil fuels will run out by 2088. However, to stay below the IPPC’s 2°C warming target (IPPC, 2018), only 33% of proven non-renewable energy sources can be burnt (Kalair, 2020), implying the urgent need for a transition to renewables. The effects of cumulative climate change, adaptation and prevention costs in addition to the exponential growth of energy demand - 778 Etta Joule by 2035 (Akbarzadeh, 2020) signals the unavoidable replacement of fossil fuels within the next 30 years (Soeder, 2021). Therefore, each nation should maximize the percentage of renewables within their national energy mix and aim for a future 100% transition to ensure that future generations avoid the financial burden of uncontrollable climate change.

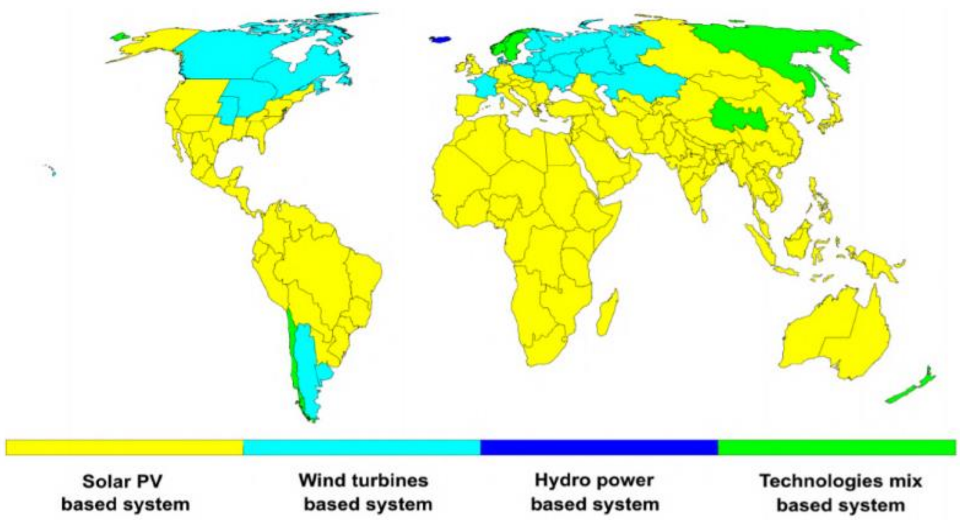

Figure 1: 100% renewable electricity energy system types per country. (Ram, 2019)

A simulation study by the Lappeenranta-Lahti University of Technology discovered that transitioning to a 100% renewable energy mix by 2050 is feasible, with wind and solar supplying 96% of the power. Figure 1 demonstrates the kinds of renewable energies that would make up that 100%, where Eastern European states, including Ukraine, are predicted to adopt a wind-based energy system.

Ukraine, despite its high wind energy potential (320 GW and sufficient speeds of 4-5ms^{-1}), is presently underutilizing its resources (IRENA, 2017). The country’s energy demand is satisfied mainly through non-renewable means, with only 8.2% coming from renewable sources. Moreover, 65% of fossil fuels are imported from Russia (IEA, 2020). Therefore, energy insecurity is a significant concern for Ukrainian geopolitics due to the conflict which has existed between the nations since 2014, leading to acute price and supply complications. Accordingly, national renewable energy development is crucial for energy self-sufficiency and the formation of a geopolitical coalition with the European Union. The EU-Ukraine Association Agreement of 2020 finalized Ukraine's strategy for becoming part of NATO and the European Union, obliging the nation to adopt European green deal strategies (Leonard, 2021). This infers the urgent need to achieve its renewables targets. By prioritizing this goal, Ukraine can demonstrate that they possess values in common with Europe, gaining political support and ultimately facilitating Ukrainian economic growth.

However, the idea of a wind-based energy system should be explored on a narrower spatial scale, as technical, economic, or environmental limitations could restrict the development of wind farms. Furthermore, the annexation of Crimea and prolonged military unrest in the Donbas region reduced Ukrainian wind energy generation from 514MW to 286MW. Besides, the installed capacity of solar power is higher than wind by 3,755MW, and amounts to 77% of the total amount of national renewables (UWEA, 2019). Thus, before selecting a wind-based renewable development trajectory, it is critical to analyze spatial suitability and sufficiency wind regimes in understudied areas to ensure that there is enough suitable space for the required turbines.

Planning for future climatic changes is a novel approach in energy sector, as most studies rely on historical data to determine the resource's sufficiency. Climate change can alter large-scale atmospheric circulation and the intensity of storms, which greatly affects the frequency and distribution of wind speed (Tobin, 2016). Wind turbine energy production is sensitive to climate change, and even a slight shift away of speed distribution from the turbine's operational range can render the wind farm impractical due to an inadequately low level of energy production (Pryor, 2010). It is crucial to understand this kind of data in order to ensure that throughout a wind farm's typical lifetime (25-30 years or up to 50 years for mag-lev models [Rubert, 2019]) energy production will remain stable and reliable.

Up-to-date research is based on CMIP5 models, projecting negligible wind speed decline over Eastern Europe, including Ukraine, with increased inter and intra-annual variability corresponding to energy supply irregularity (Reyers, 2016). These are valuable insights into future wind conditions, questioning the feasibility of wind-based energy systems in Ukraine and emphasizing the importance of regional scale analysis on the influence of climate change. However, recently launched CMIP6 models (2019) must be the base for future wind regimes assessments due to technological advancements such as lowered latitudinal correlation for historical wind speed data, corresponding to more reliable and robust projections (Morim, 2020). For Ukraine, where a transition to sustainable energy is a pivotal geopolitical step towards forming a coalition with the EU, implementation of CMIP6 data in wind farm proposals would demonstrate forward-thinking approach and yield reliable energy transition policies

1.1 Originality of this research

Compared with European countries of similar populations (France, Poland), Ukraine has a much lower installed wind capacity, regardless of having a larger territory, indicating vast unacquainted potential. Moreover, no research has explored the Kyiv region as its 20% slower than average winds are considered less favourable for turbines (Kharlamova, 2016). Nevertheless, the embedded generation approach of placing an energy supply’s source at a substantial energy demand site, was reported to be the most efficient tactic, minimizing the construction costs of high voltage transmission networks that connect wind farms to the main energy grid (Grant, 2007). Hence, this study aims to close the research gap and analyze the Kyiv region, which despite its lower wind speeds, contains the country’s largest metropolis with the highest energy demand. The vital contribution of this study is utilization of CMIP6 models to assess future wind speeds, as no assessment of the future climatic impacts on wind has been conducted in Ukraine using any models. Wind speeds, projected by CMIP6 models, have only been analyzed on a global scale (Deng, 2020) and in China (Zhang, 2021).

This research analyses regional characteristics in Kyiv, determining land suitability for wind farm construction. This study will examine the current and future wind speed characteristics to ensure the sustainability and reliability of wind-focused renewable energy. This dissertation combines an analysis of all the steps required for wind farm installation and considers future resource volatility, which is not always accounted for in wind power plant proposals. Henceforth, the presented rationale and methodology are valuable knowledge input for the Ukrainian energy transition, applicable globally.

1.2 Research Aims

1. To identify physical and environmental constraints for wind farm installation within the Kyiv region of Ukraine and select areas fulfilling legal and practical requirements.
2. To assess differences in historical regimes at potential sites to identify which location has the highest potential energy yields.
3. Examine each site's future wind speed potential using the data of CMIP6 models at four shared socioeconomic pathways: SSP1-2.6, SSP2-4.5, SSP3-7.0, and SSP5-8.5.

2. Literature Review

2.1 Necessity for, and the current progress of, Ukrainian wind energy development

A fossil-fuel dominant (91.8%) Ukrainian energy mix has led to severe socio-environmental problems, including the loss of 2,538 disability-adjusted life years (DALYs) in 2016 (WHO, 2016). The economic effect of this impact on health can be estimated by multiplying DALYs by the GDP per capita in 2016, resulting in $5,553,144; a strong financial incentive for renewables research and development. Furthermore, indirect negative externalities include a burden on the healthcare system, rising mental health issues and ethical concerns.

Additionally, in 2019, the yearly temperature was 2.9°C higher than the historical average, mainly due to energy sector emissions (climateactiontracker.org, 2021). The climatic shift led to a recurring absence of snow during winter, causing a substantial loss in the agricultural sector. Reduced yields per hectare force the spatial expansion of farming, resulting in biodiversity loss, reduced albedo and can catalyze further warming. The Ukrainian agricultural sector accounts for 12.4% of GDP (World Bank, 2019), highlighting the financial interest in limiting fossil fuel emissions to ensure stable income from climate-reliant sectors.

The 2050 Low Emission Development Strategy addressed the concerns setting the target of a 51-54% reduction in MtCO2/yr. (Savitsky, 2018). A pivotal legal step for implementing the goal was Law No. 810-IX, which aimed to 'improve the conditions for promoting electricity generation from renewable energy sources', including a capacity auction-based support scheme for wind energy production stimulation (Chemnik, 2019; Government of Ukraine, 2019). Respectively, 2019 showed the highest growth rate of installed wind energy capacity, reaching 637.1 MW - just below tenfold growth from 67.8MW in 2018. In 2019, 2.27 million tons of CO2 were saved, which was a practical step forward on a roadmap for climate change mitigation goals. The wind energy sector contributed financially to national and local budgets (65 million EUR) and mitigated unemployment by 1,500 jobs. These achievements are facilitating geopolitical integration with the EU. For example, BloombergNEF Climatescope 2019 ranked Ukraine 8th in terms of best investment conditions in emerging markets for renewables. The soar from 63rd place in 2018 is a vivid improvement for Ukraine. Indeed, the production of insightful knowledge on the country's underexploited areas for wind energy is crucial for identifying the most promising opportunities and the continuation of energy sector greenification.

Despite the rising share of wind energy, the sector’s portion is still small – just 2.15% (UWEA, 2019). While the necessity for an energy innovation is recognized, Ukrainian scientists have limited the scope of analysis to general national wind conditions, (Volkovaia, 2015), seasonal fluctuations (Velychko, 2003) and contain planning legislation references (Ukrecoconsult, 2018). Therefore, wind assessment in Ukraine needs development and investment to ensure that the transformation of the energy sector is founded upon a data-driven strategy and considerable scientific evidence. Consequently, identification of sites for sustainable embedded energy generation in Ukraine, such as this study, would be a valuable research contribution.

2.2 GIS-based spatial decision making: criteria and constraints

The site selection for renewable energy facilities is a multi-faceted process which requires selecting a site with the highest resource potential, without exceeding environmental carrying capacity while ensuring social wellbeing. Wind speed is the primary selection factor, but legal, ecological, and technical restrictions may also restrict the construction of wind farms (Haaren, 2011). Buffer zones are central to spatial multicriteria assessments as they protect sensitive areas while ensuring sustainable renewable resource development.

Ukrainian wind farm planning regulations are limited to the document ‘An Evaluation of the Environmental Impacts of Wind Farms' by UkrNDNC (2017), which defines only one spatial constraint:700m from all settlements. Furthermore, The Legal Regime of Special Zones of Energy Facilities No. 2755-VI allows the construction of electricity facilities on all land categories only if the original purpose of space remains unchanged. Moreover, the addition of article 19 of the Land Code of Ukraine states that *"objects of alternative energy can be located on areas assigned to transport, communication, energy and defense ".* Consequently, policy design eases the legal aspect of the building process and enhances investment interest by maximizing the availability of legally appropriate land for wind energy facilities.
Moreover, EU laws and EBRD recommendations are used as reliable guidance for safe and effective wind farm site selection if the national legislation is not complete (Slitz-Szkliniarz, 2011; Aydin, 2010). Thus, Ukrainian and international laws and academic papers ought to be reviewed to define suitable buffer distances for wind farms in the Kyiv region.

2.2.1 Wind Speed

Wind speed is the primary factor for determining site suitability because energy production is impossible outside of the operational speed range. The viscosity of air reduces near-surface wind speed, which increases with altitude, where the rate of increase depends on the roughness of the surface (Marvel, 2013). Therefore, turbine producers increase the hub height to capture faster speeds and ensure wind loads balance on blades via equal wind speed distribution over the rotor (Mariano, 2019). For instance, the Vestas brand increased their turbines’ height from 80m in 2011 to 117-149 in 2018.

Earlier studies (Sliz-Szkliniarz, 2011) recommended cut-in speeds ranging from 3.5 -5.5 ms-1, but modern studies suggest higher minimum wind speeds – 6 -8 ms-1 (Table 1). Overall, minimum wind speed requirements increase with time and are directly related to faster wind regimes, which turbines capture via hub height increases. Cut-off speed is it is 25ms^{-1} (Song, 2020) for most of the turbines. Producing taller turbines could allow more places in the world to capture wind energy, as wind speed increases by 4-8% during hub height increase of 80m to 140m and a further 1% increase from 140m to 160m (Lantz, 2019). Additionally, surface spaces that are rougher (e.g. urban territory, forested regions) experience more prominent wind velocity growth as altitude increases. Indeed, sufficient wind energy generation becomes feasible in various spaces, satisfying a higher proportion of energy demand with a renewable source. The global trends are beneficial for the Kyiv region, as wind speeds are lower than the country’s average and are expected to decrease further, according to CMIP5 models. While minimum wind speed constraint depends on the type of turbine, even at 3.5ms-1, energy generation is possible. However, to maximize potential energy output while ensuring sustainable power yields, taller turbines and higher cut-in speeds are favourable.

Table 1: Wind Speed Spatial Criteria in Literature

Study	Hub Height	Cut-in Speed
Baban S., Parry T. (2001) *Developing and applying a GIS-assisted approach to locating wind farms in the UK.*	Not specified	$5ms^{-1}$
Gorsevski, P. V., Cathcart, S. C., Mirzaei, G., Jamali, M. M., Ye, X., & Gomezdelcampo, E. (2013). *A group-based spatial decision support system for wind farm site selection in Northwest Ohio.* Energy Policy, 55, 374-385.	50m	5.6ms-1
Wang, Q., M'Ikiugu, M. M., & Kinoshita, I. (2014). *A GIS-based approach in support of spatial planning for renewable energy: A case study of Fukushima, Japan.* Sustainability, 6(4), 2087-2117.	70m	$6.0ms^{-1}$
Höfer T., Sunak Y., Siddique H., Madlener R. (2016) *Wind farm siting using a spatial Analytic Hierarchy Process approach: A case study of the Städteregion Aachen.*	135	$6.0ms^{-1}$
Bili A., Vagiona D.G. (2018) *Use of multicriteria analysis and GIS for selecting sites for onshore wind farms: The case of Andros Island (Greece)*	85m	$6.0m^{-1}$
Sánchez-Lozano, J. M., García-Cascales, M. S., & Lamata, M. T. (2014). *Identification and selection of potential sites for onshore wind farms development in Region of Murcia, Spain.* Energy, 73, 311-324.	80m	3.20-5.50 ms-1 = ‘good’ 5.50 – 7.00ms-1 = ‘very good’ >7.00ms-1 = ‘excellent’

2.2.2 Settlements

Settlements' buffer zones depend on whether an area is rural or urban, with the general trend of urban being double that of the rural length (Latinopoulos, 2015). Settlements' buffers range in the rural category from0.5 -1km, while in urban, 1 -2km (Table 2). In Poland in 2016, the legal settlement spatial constraint was reduced from 3km to 10 times the turbine height (usually 800m – 1km) to maximize the number of wind turbines and corresponding energy profits per site (Hajito, 2017). Similarly, we can observe the global trend of reducing buffer length as governments and energy experts realize the sustainability advantages of renewable clean energy generation.

Social acceptance of wind farms is an uncertain factor when defining the buffer distance. Some perceive them as causing a severe degradation of nature, while others have positive attitudes towards clean energy and therefore condone them (Haaren, 2011). The perception survey by Ferreira (2019) concluded that wind farms located on the visual periphery of settlements, in proximity to transportation networks, benefit public opinion, triggering investment and expansion. Therefore, constructing wind farms within the Kyiv region could catalyze public support for the clean energy revolution, further facilitating investment.

Table 2: Settlement Spatial Constraints in Literature

Study	Location	Rural Buffer	Urban Buffer
Höfer T., Sunak Y., Siddique H., Madlener R. (2016) *Wind farm siting using a spatial Analytic Hierarchy Process approach: A case study of the Städteregion Aachen.*	Germany	550	550
Tegou, L., Polatidis, H. and Haralambopoulos, D. (2010) *Environmental Management Framework for Wind Farm Siting: Methodology and Case Study.* Journal of Environmental Management, 91, 2134-2147.	Greece	500	1000
Latinopoulos D., Kechagia K. (2015) *A GIS-based multi-criteria evaluation for wind farm site selection. A regional scale application in Greece Renew*. Energy, 78, pp. 550-560.	Greece	500	1000
Baban S., Parry T. (2001) *Developing and applying a GIS-assisted approach to locating wind farms in the UK.*	UK	Not considered	2000
Sliz-Szkliniarz, B. and Vogt, J. (2011) *GIS-Based Approach for the Evaluation of Wind Energy Potential: A Case Study for the Kujawsko-Pomorskie Voivodeship.* Renewable and Sustainable Energy Reviews, 15, 1696-1707.	Poland	500	2000
Değirmenci S., Bingöl F., Sofuoglu S.C. (2018) MCDM analysis of wind energy in Turkey: decision making based on environmental impact	Turkey	Not considered	1000
Georgiou A., Polatidis H., Haralambopoulos D., *"Wind Energy Resource Assessment and Development: Decision Analysis for Site Evaluation and Application"*, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 34, pp. 1759, 2012.	Cyprus	Not considered	850

Noise has a quantifiable impact and is one of the core reasons for buffer distance. The sound level regulations are well established in Ukraine and apply to wind farms, even if no wind-energy specific protocols exist. Noise is regulated by the law No.580-VIII ‘Population Protection from noiseous hazards’, where daytime (7am – 11pm) threshold - 55dB; nigh (11pm – 7am) – 55dB. According to a technical report for the European Commission (Dalla, 2018), standard turbines (with a capacity of 3MW, a hub height of 80m and a rotor diameter of 90m), produce an acceptable level of noise at a distance starting from 500m. Noise spread has a logarithmic relation to sound power at the source (the turbine) and sound pressure at the location (the nearest settlement). However, the setback distance for reaching an acceptable level of noise increases with the number of turbines (Figure 2).

Furthermore, turbine noise amplitude is modulated at the blade route frequency, implying that peak sound is 3 to 5 dBA higher than the 45/55dBA average, on which policy has been based (van den Berg, 2005). Vast evidence advocates that the 500m buffer is inadequately low, causing annoyance and sleep deprivation for nearby residents (Frey, 2007; McMurty, 2009). A 1 -2 km buffer is recommended based on field studies in the Netherlands (Pedersen, 2009) and Sweden (Persson, 2007). Placement regulations for Ukrainian wind farms are still being developed, so considering European standards could prevent potential mitigation measures.

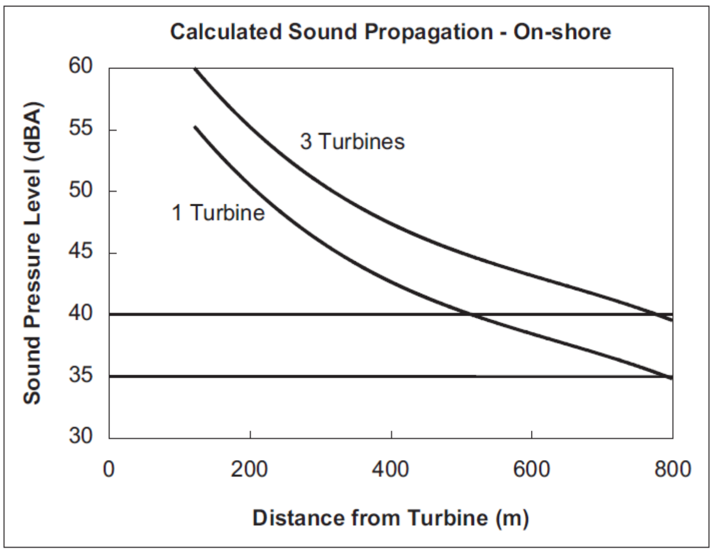

Figure 2: Sound level changes with distance from the turbine (Harrisson, 2011)

Lastly, Ukrainian winter temperatures usually drop to -10°C - -15°C -, so ice, potentially forming on turbine blades, could get scattered during rotation, causing a hazard to animals and people walking or living in proximity. Ice can be thrown at a maximum of 350m (Kjeller, 2017), but most ice fragments reach just 69% of that distance, so this hazard is automatically accounted for by implementation of settlement buffer. Even though winters are getting progressively milder in Ukraine, the high inter-annual variability of temperatures leads to severe uncertainty when it comes to predicting ice formation, warranting a buffer hazard mitigation function.

2.2.3 Water Bodies

The setback distance of water bodies prevents contamination during the construction phase; a disturbance of groundwater and nearby vegetation in addition to potential diesel, oil and propane spillage is a hazard for the habitat of aquatic biota. Furthermore, most rivers and streams have inherent functionalities as fish bypass channels, so construction pollution and noise can put local aquafauna under stress, resulting in biodiversity loss (Dai, 2015). Overall, international regulations rarely define buffer zones for inland water bodies above 500m (Haugen, 2011), while Canada's required buffer is 30m.

Table 3: Inland Water Bodies Buffers in Literature

Study	Buffer for Inland Water Bodies
Ameri M, Ghadiri M, Hosseini M. *Recent advances in the implementation of wind energy in Iran.* In: The 2nd joint international conference on sustainable energy and environment (SEE 2006) 21–23 November 2006, Bangkok, Thailand, 2006	500m
Bennui, A., Rattanamanee, P., Puetpaiboon, U., Phukpattaranont, P., & Chetpattananondh, K. (2007, May). *Site selection for large wind turbine using GIS.* In PSU-UNS international conference on engineering and environment (pp. 561-566).	200m
Baffoe, P. E., & Sarpong, D. (2016). *Selecting suitable sites for wind energy development in Ghana.* Ghana Mining Journal, 16(1), 8-20.	200m
Satkin, M., Noorollahi, Y., Abbaspour, M., & Yousefi, H. (2014). *Multi criteria site selection model for wind-compressed air energy storage power plants in Iran.* Renewable and Sustainable Energy Reviews, 32, 579-590.	500m

The water Code of Ukraine (1995) requires a safety zone for rivers, reservoirs, and lakes between 100m and 25m. Despite larger buffers being recommended by European academics, the Ukrainian 100m is within an acceptable range. Moreover, the maximization of available space for clean energy is a Ukrainian development strategy, facilitated by selecting the lowest possible buffers. Thus, the buffer distance must be kept within the lower end of the legal range to align with the strategic roadmap of the clean energy transition.

2.2.4 Roads and Railways

The distance between the turbine and any roads or railways must exceed the height of the turbine itself in case of a fall. Wizelius (2007) recommends a buffer distance equal to the sum of the hub height and rotor radius with an extra 50m. If the exact turbine height is unknown at the planning stage, the average height of turbines available on the market should be used as the most appropriate reference. However, with the global trend of turbines increasing in height, setting an even larger buffer would ensure the safe positioning of a site in the long term, should the site be upgraded with newer and taller turbines. In contrast, roads and railways should be in relative proximity to wind farms for the cost effectiveness of construction.

Therefore, multiple European studies set a maximum distance from the roads instead of a minimum

Table 4: Transportation constraints in Literature

Study	Minimum Buffer	Maximum Distance
Baban S., Parry T. (2001) *Developing and applying a GIS-assisted approach to locating wind farms in the UK.*	X	10000m
Latinopoulos D., Kechagia K. (2015) *A GIS-based multi-criteria evaluation for wind farm site selection. A regional scale application in Greece* Renew. Energy, 78, pp. 550-560.	150m	X
Atici, K.B., Simsek, A.B., Ulucan, A, et al. (2015) A *GIS-based multiple criteria decision analysis approach for wind power plant site selection.* Utilities Policy 37: 86-96.	500	X
Nguyen KQ (2007) *Wind energy in Vietnam: resource assessment, development status and future implications.* Energy Policy 35: 1405–13.	100	X
Georgiou A., Polatidis H., Haralambopoulos D., *"Wind Energy Resource Assessment and Development: Decision Analysis for Site Evaluation and Application",* Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 34, pp. 1759, 2012.	X	5000m

Though the road network is critically underdeveloped in Ukraine (Kozak, 2019), an investment in its expansion ($3.5 billion) has already been planned by The Cabinet of Ministers. Of course, the potential wind farm can guide the road network’s expansion, create jobs, and facilitate further investment into clean energy. The Dnipro-Buzka wind farm in Southern Ukraine holds a roads and railways buffer of 200m, stating that site suitability decreases by 20% for every extra 2,000m from the nearest transportation facility due to additional construction expenses. Thus, there is a tradeoff between minimizing expenses and maximizing wind energy generation. Nevertheless, the costs of a dependency on fossil fuels certainly prioritizes the development of wind farms and corresponding roads.

2.2.5 Slope

The topography is a critical technical factor because slopes which are too steep make the operation of tracks and cranes problematic, prolonging the construction process. Moreover, a hilly region can disturb the flow of air mass, leading to low energy production. Such a phenomenon is facilitated via flow separation, which is endorsed via increased surface roughness, e.g., developed vegetation cover over the summer (Teunissen, 1987). However, increased surface roughness affects only a limited region which is close to the ground (Neff, 1998). Indeed, a high turbine height implies that the slope constraint is mostly a construction challenge rather than a wind energy production limitation. Furthermore, hilly terrain can be advantageous for energy generation due to significant wind acceleration (Archer, 2005). The stable air mass moving over a complex topography can experience oscillatory motion, powered by temperature differences between the air flowing over the mountains and the surrounding atmosphere. Consequently, air moving downwards is warmed more rapidly than the surrounding air, driving rapid upwards flow through the buoyancy force restoration.

Table 5: Slope spatial constraints in literature

Study	Slope Constraint
Baban S., Parry T. (2001) *Developing and applying a GIS-assisted approach to locating wind farms in the UK.*	10%
Watson J.J., Hudson M.D. (2015) *Regional Scale wind farm and solar farm suitability assessment using GIS-assisted multi-criteria evaluation.* Landscape and Urban Planning 138: 20-31.	10%
Atici, K.B., Simsek, A.B., Ulucan, A, et al. (2015) *A GIS-based multiple criteria decision analysis approach for wind power plant site selection.* Utilities Policy 37: 86–96.	10%
Georgiou A., Polatidis H., Haralambopoulos D., (2012) *"Wind Energy Resource Assessment and Development: Decision Analysis for Site Evaluation and Application",* Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 34, pp. 1759.	10%
Villacreses G, Gaona G, Martinez-Gomez J, Jijon DJ. (2017) *Wind farms suitability location using geographical information system (GIS), based on multi-criteria decision making (MCDM) methods: The case of continental Ecuador*, Renewable Energy, 109, 275-286.	15%
Wang, Q.; M'Ikiugu, M.M.; Kinoshita, I. A (2014*) GIS-Based Approach in Support of Spatial Planning for Renewable Energy: A Case Study of Fukushima, Japan.* Sustainability, 6, 2087-2117.	20%
Höfer T., Sunak Y., Siddique H., Madlener R. (2016) *Wind farm siting using a spatial Analytic Hierarchy Process approach: A case study of the Städteregion Aachen.*	30%

The same process can occur in reverse order, further increasing wind-wave oscillations and enhancing energy yields (Smith 1979). Still, although increasing turbine height can reduce topographic limitations (Dale, 2016) and higher turbulence increases the power output, especially with yaw-control strategies in place (Campagnolo, 2016), turbulence also intensifies the fatigue loads on the rotor. This ultimately reduces the operational lifetime of the turbines (Mycek, 2014). Despite some promising energy outputs from complex topographical

environments, a vast uncertainty regarding annual power output has led academics to recommend lower slopes for wind farm construction.
The slope is measured in percent, where 45^{o} corresponds to 100%. Most academics recommend maximum slopes of around 10 -30% but tend to favour the lower end of the range. No studies recommend slopes less than 10%, and only Miller (2014) investigated a 40% slope constraint.

2.2.6 Natural Reserves

The environmental impacts of wind farms are usually negligible when compared to their positive effects. However, negative influences on birds are common when inappropriately placing wind farms. Birds may collide with the rotors, wildlife may be displaced, barrier effects can be observed, migration routes can be disturbed and in some cases, habitat loss can occur (Zimmerling, 2013). Bird mortality is the most widely discussed environmental concern due to its ambiguous quantification. Some studies report that death directly due to turbine blade collision represents 0.01 – 0.02% of the total (Erickson, 2001), while others report that as much as 9.33 birds are killed, per turbine, annually. (Barclay, 2012). Overall, the turbines’ impact on birds is largely uncertain. However, most academics suggest that mortality is minor compared to other kinds of collisions, e.g., with vehicles, buildings and glass windows. Limited research exists on the estimation of buffer distances from natural reserves as most studies exclude protected areas without further analysis of the effectiveness of buffers. Instead, research is concentrated on technical methods of environmental impact mitigation, e.g. the Norwegian Institute for Nature Research identified that painting one of the blades black reduces collisions by 70% (May, 2020).

Table 6: Natural Reserves Buffers in Literature

Study	Buffer
Aydin N.Y., Kentel E., Duzgun S. (2010) GIS-based environmental assessment of wind energy systems for spatial planning: A case study from Western Turkey, Renewable and Sustainable Energy Reviews, Volume 14, Issue 1, 2010, Pages 364-373.	1000m
Höfer T., Sunak Y., Siddique H., Madlener R. (2016) Wind farm siting using a spatial Analytic Hierarchy Process approach: A case study of the Städteregion Aachen.	0m for natural reserves; 300 m for bird protection areas
Yue, C. D., & Wang, S. S. (2006). GIS-based evaluation of multifarious local renewable energy sources: a case study of the Chigu area of southwestern Taiwan. Energy Policy, 34(6), 730-742.	500m for bird habitats
Yue CD, Wang SS. GIS-based evaluation of multifarious local renewable energy sources: a case study of the Chigu area of southwestern Taiwan. Energy Policy 2006; 34:730–42.	500m from wildlife conservation areas
Clarke A. Wind energy progress and potential. Energy Policy 1991; 19:742–55.	300m from birds protection reserves

Ukrainian Law 'On the Nature Reserve Fund' forbids construction on the territory of natural reserves, but no legal obligation to observe buffer zones exists. Still, it is beneficial to implement buffers to ensure environmental safety.

2.2.7 Airports

Wind turbines pose a direct threat to aviation safety, as their height can reach up to 180m; all structures above 150m pose a physical obstacle risk, that is most severe during take-off and landing. (International Civil Aviation Organization, Annex 14). However, the main issue is the turbines' impact on Primary Surveillance Radar (PSR). PSRs are used to detect moving objects with reflective characteristics, such as inbuilt metallic lighting conductors in the moving blades of the turbine's rotors (Jackson, 2013).

Table 7: Airport Buffers in Literature

Study	Airport Buffer
Wang, Q.; M'Ikiugu, M.M.; Kinoshita, I. A (2014) GIS-Based Approach in Support of Spatial Planning for Renewable Energy: A Case Study of Fukushima, Japan. Sustainability, 6, 2087-2117.	2500m
Villacreses G, Gaona G, Martinez-Gomez J, Jijon DJ. Wind farms suitability location using geographical information system (GIS), based on multi-criteria decision making (MCDM) methods: The case of continental Ecuador, Renewable Enegy, 2017; 109, 275-286.	2500m
Bili, A., & Vagiona, D. G. (2018). Use of multicriteria analysis and GIS for selecting sites for onshore wind farms: the case of Andros Island (Greece). European Journal of Environmental Sciences, 8(1), 5-13.	2500m
Değirmenci S., Bingöl F., Sofuoglu S.C. (2018) MCDM analysis of wind energy in Turkey: decision making based on environmental impact.	3000m
Georgiou A., Polatidis H., Haralambopoulos D., "Wind Energy Resource Assessment and Development: Decision Analysis for Site Evaluation and Application", Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 34, pp. 1759, 2012.	4000m

While most PSRs can discriminate between moving and stationary objects, blades move at very close speeds to airplanes, notably the Boeing 747. Due to a lack of processing power for filtering the incoming signals, radar displays can become saturated with noise signals, making air traffic controllers unable to differentiate between real aircraft targets and wind turbines. Such an effect confuses air traffic controllers and therefore results in a safety hazard, implying a critical need to implement a buffer.

An airport buffer is the most extensive spatial constraint in all studied papers. Most authors use a 2,500m buffer, but some studies use a weighting approach and define 100% site suitability for maximum buffer as 6,000m, which gradually falls as proximity to the airport increases (Aydin,2010). However, the guidelines set out by the Civil Aviation Authority (2016) suggest that using a single buffer is the most effective hazard prevention tactic, resulting in minimal airport and wind farm operations interference.

2.3 An assessment of wind regimes

Wind speed is inherently stochastic, and is influenced by location-dependent factors, such as topography, pressure gradients and atmospheric circulation (Elma, 2019). Characterizing wind profiles and turbulence is highly challenging, while the cubic dependency of wind speed and power output - Equation 1 (Mostafaeipour, 2014), implies the critical importance of this knowledge. Even a minor underestimation of loads on turbines can cause a severe safety hazard. In contrast, if turbines are not switched off during inadequately low wind speeds (0-2ms^{-1}), significant or economic losses may occur. Moreover, apart from seasonal variations, interannual fluctuations of wind speed can reach 10 -30%. Thus, a thorough knowledge of wind regimes is essential for the effective utilization of wind energy.

$$P = \frac{1}{2}\rho v^3 \ \left(\frac{W}{m^2}\right)$$

ρ – density of air
v – wind speed
P – wind power per unit area

Equation 1: Cubic dependency of power on wind speed

Wind energy developers use mathematical probability distribution functions (PDFs) to model wind speed to historically measured data and obtain functional relationships (Agelin-Chaab, 2018). PDFs statistically project the proportion of time for which the wind is within an optimal velocity range and estimate periods when a wind farm is shut down due to inadequately low speeds. Distribution functions for wind regime modelling include Weibull, Rayleigh and Johnson. However, most researchers use the two-parameter Weibull function. It is most appropriate for live data-based reliability engineering (Monahan, 2011) and gives the best agreement with experimental data (Akdag, 2010)

In case of a high number of null values in dataset, a generalized version of the Weibull - Rayleigh function becomes more appropriate due to its fixed shape parameter of 2 (Akpinar, 2005). The International Energy Agency recommends the Rayleigh function for the national annual wind energy production assessment (IEC, 1998). However, for examining resources at a proposed site, the Weibull function is more suitable due to its shape parameter variation, exposing more fluctuations in speed while staying robust to the noise of wind gusts (Jamil, 1995).

Limitations of the Weibull function are mostly related to its application for brittle fracture modelling (Basu, 2009, Wong, 2006). In wind speed research, multiple studies revealed that the addition of location parameters can significantly improve the Weibull distribution fitting to observed data, implying that it is still the most appropriate wind analytics method (Marshall, 1997).

2.4 CMIP6 climate projections

Climate change's effect on wind has not been studied in Ukraine, so European studies are instead reviewed to uncover large spatial scale projections. CMIP5-based publications have robust projections (2%-6% in magnitude with an increase towards the end of 21st century [Wenz, 2017]) of annual average wind speed, demonstrating an increase over Northern Europe and around the Baltic and Aegean seas while displaying a slight decrease over Southern Europe, particularly around the Mediterranean (Reyers, 2015; Carvalho, 2017; Bloom, 2008). These findings are consistent with large-scale climatic transitions; poleward enlargement of the Hadley cell enhances the North-South pressure gradient with lowered pressure over Northern Europe and increased pressure over South-Eastern Europe (Mizuta, 2012). Seasonality has robust projections from most models across Europe with increased wind speeds during winter and decreased speeds during summer (Reyers, 2016).

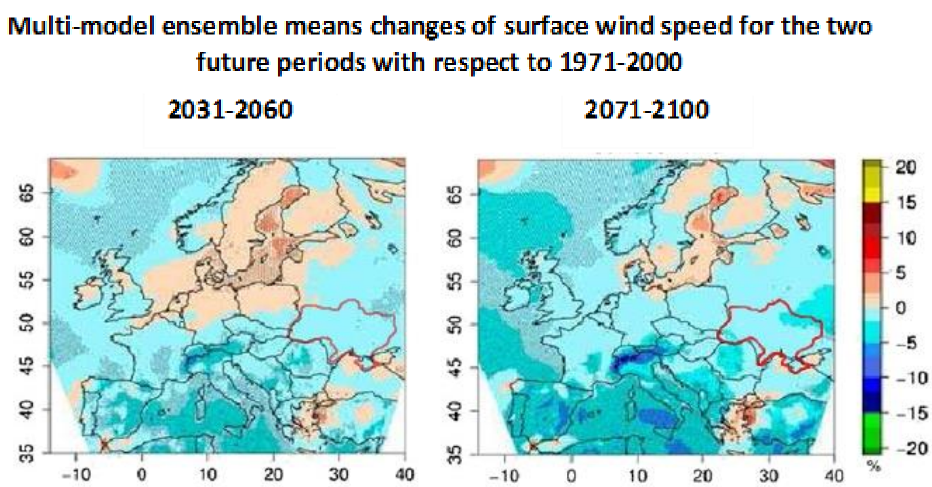

Figure 3: Mean projections of surface wind speed over Europe based on 15 simulations from ENSEMBLES European FP6 project. (Tobin, 2018)

The uncertainties are most distinct for central and eastern Europe, where change varies between models, generating a negligible net difference (Moemken, 2018). Thus, in ensembles, such as EURO-CORDEX (which consists of 5 GCMs), individual projections cancel out by averaging and forecasting stable wind resource availability (±5%) (Tobin, 2016). However, local-scale changes of single models can reach up to 15%, an enormous effect on power outputs. This could lead to critical energy shortages due to inadequate planning. Thus, it is vital to exploit climate model data on a small spatial scale and incorporate it in wind farm planning to select locations where wind speeds will remain within the turbines' operational range.

Tobin (2018) presented the spatial distribution of surface wind speed changes in the mid-to-late 21st century. According to this research, Ukrainian territory has experienced a uniformly minor slowing-down, apart from a small territory in the east where winds have dropped by up to $5ms^{-1}$. Therefore, in the long-term, eastern areas with faster winds, considered superior according to historical data, may become less favourable than central regions because of climate change. Still, the high inter-model variability of CMIP5 ensembles indicates the importance to investigate CMIP6 projections. CMIP6 is superior in comparison to CIMP5; it conceptually frames climate trajectories and offers a more precise estimation of climatic sensitivity. Thus, relying on the most advanced climate models for Kyiv is of crucial importance, as more innovative scenarios and developed relationships will provide for more reliable energy planning projections.

Thus, while CMIP5-based stability of wind regimes over the Kyiv Region (central Ukraine) may imply future superiority for wind energy generation, over historically faster winds in eastern Ukraine, conclusions must be validated with new models.

3. Methods & Data

The developed methodology for evaluating the site's suitability and sustainability for wind energy generation is a set of sequential steps, evaluating spaces that satisfy legal, technical, geographical, and climatic restrictions (Sliz-Szklarz, 2011). Actions performed in this research follow the hierarchy of the site selection process (Spyridonidou, 2020; Gao, 2020). There are several studies that have adopted this methodology for the analysis of specific locations (Latinopoulos, 2015; Van Haaren, 2011) and some research revealing the impacts of climate change on wind resources e (Solaun, 2020).

3.1 GIS

Factors playing a role in wind farm site selection are spatially dependent, so GIS is the most appropriate analysis tool due to its capabilities for managing and analyzing spatial data. Firstly, sitting criteria were selected and defined based on the above literature review. Moreover, the necessary data corresponding to each restraint criterion were downloaded.

Table 8: Spatial Constraints and Corresponding Data

	Factors	Buffer Zone/Criteria constraint	Data Source	Data Format
1.	Wind Speed	$5ms^{-1}$	Global Wind Atlas	Raster (250m resolution)
2.	Elevation (Transformed to Slope)	10%	CGIAR SRTM (3 seconds resolution)	Raster (1000m resolution)
3.	Roads	1000m	Digital Chart of the World (DIVA GIS)	Vector
4.	Railways	150m	Digital Chart of the World (DIVA GIS)	Vector
5.	Administrative Areas (Cities, Towns & Villages)	2500m	GADM, version 1.0	Vector
6.	Water Bodies	500m	Digital Chart of the World (DIVA GIS)	Vector
7.	Airports	2500m	Humanitarian OpenStreetMap Team (HOT)	Vector
8.	Natural Reserves/ Protected Areas	1000m	ProtectedPlanet.net	Vector

The analysis was performed in QGIS3.10, as its open-source access and faster geoprocessing capabilities, in addition to its advanced intern structure, are advantageous over ArcGIS. Figure 4 illustrates all steps taken to create a suitability map.

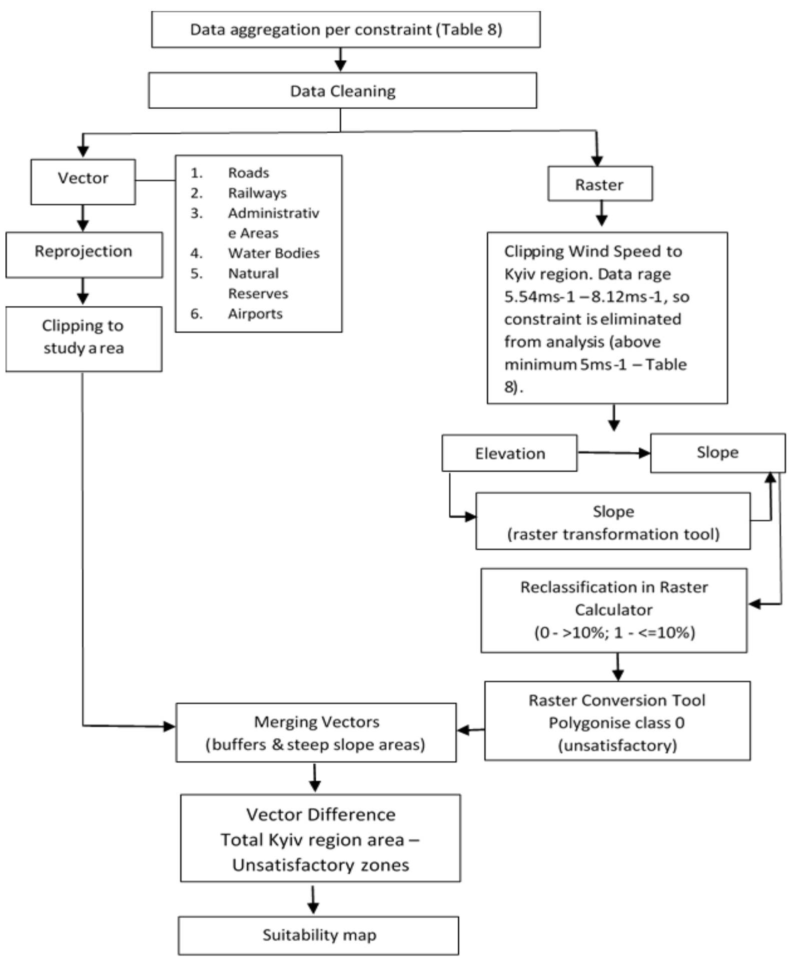

Figure 4: Flowchart of GIS analysis

Data was reprojected *to 'EPSG:6385 – UCS-2000 / Ukraine TM zone 11'*, and *the Clip* tool selected only data within the Kyiv region. The *Buffer* tool was used to create mandatory safety zones around the vector data. Wind Speed data at a 100m turbine hub height, averaged over 10 years, ranged between 5.54ms-1 – 8.12ms-1. Thus, all study areas satisfied the primary selection conditions at this stage. Elevation raster data was transformed to slope in percentages via *the Slope* tool in *Raster Analysis*. Then *the Raster Calculator* was used to reclassify areas according to the criteria. Unsatisfactory raster data was vectorized using the *Polygonise* tool in *Raster Conversion*. Unsuitable areas were merged into one vector layer and subtracted from the total Kyiv region vector layer through *the Vector Difference* tool to yield a vector layer of appropriate areas. Each polygon area was added as a column in *the Attribute table*, sorted in descending order, and the three largest polygons were moved to the final layer, *Potential Sites*.

3.2 3.2 An examination of historical data

As specified in the literature review, wind regimes at potential sites were assessed with the two-parameter Weibull function.

Equation 2: Weibull Distribution Formulas

$f(v) = \left[\left(\frac{k}{c}\right)\right]\left(\frac{v}{c}\right)^{k-1}\exp\left[-\left(\frac{v}{c}\right)^{k}\right]$	*Weibull probability density*
$F(v) = \left[1 - \exp\left[-\left(\frac{v}{c}\right)^{k}\right]\right]$	*Weibull cumulative distribution function*

k – dimensionless Weibull shape parameter

c – Weibull scale parameter (m/s)

f (v) – probability of observing wind speed equal to v

F(V) – probability of observing wind speed equal or less then V

Data was obtained from *meteoblue.com*, as the source has provided recordings since 1985 of hourly frequency. Such a sizable temporal extent increases the reliability of findings and allows the examination of trends where the impact of climate change could be visible. However, the data could be downloaded only as a CSV file for a specific set of coordinates and costs €270. This limits the possibility of examining wind speed throughout the whole site. Centroids were plotted in QGIS by *Centroid* toll in *Vector Geometry*, coordinates were recorded, and data was obtained.

Using Python, data was firstly cleaned - N/As and non-numerical symbols were deleted, and the data format was set to float. A 30% random sample of data was taken for further analysis, as

Equation 3: The wind profile power law

$$\frac{U(z)}{U(z_r)} = \left(\frac{z}{z_r}\right)^{\alpha}$$

the whole data set had over 300,000 records, corresponding to the arbitrary high probability of detecting a significant pair (Taherdoost, 2016). The sample's representativeness was checked with the Kolmogorov-Smirnov test (p-value = 0.68), indicating the sample's statistical significance. The distribution of data was then checked graphically, indicating a non-Gaussian spread. The Levene test revealed that variance is homogenous; the primary assumption of the non-parametric Kruskal-Wallis test, indicating its appropriateness for comparison between sites (Levene, 1960). Lastly, vertical extrapolation to 100m hub height was performed by Equation 3. This study used the suggestion of the International Electrical Commission (2005) regarding a power-law exponent of 0.2 for onshore areas. However, α is a highly variable factor, influenced by various environmental variables such as elevation or pressure (Manwell, 2009), thus setting it to a standard value is certainly a limitation of this study. Alternatively, a more complex approximation method by Justus (1978), calculatesα in relation to recorded velocity and corresponding hub height. Despite this, a review of a method by Moemken (2018) revealed an overestimation issue, so using this method would cause more uncertainty than setting to standard value of 0.2. For the Weibull probability distribution, the dataset was separated into intervals of $1ms^{-1}$, that is the same as in papers presented in the literature review, and so probability of occurrence of each value was calculated accordingly.
Three data characteristics were evaluated:

a. Historical trends

A multiple-line graph was plotted to detect time-series dynamics per site, to identify trends and compare rates against climate projections.

b. Differences between sites

The Kruskal-Wallis test was used to detect differences in wind speed distribution between sites over the whole 30-year period. Box plots were created to show variability between sites graphically. F(v) and f(V) were plotted by calculating constants (k&c), and then these were substituted using corresponding formulas. Weibull plots were examined to identify the most likely wind speed at each site, the proportion of time when wind speed is within the turbine's operational range and potential power output. For power output calculation, air density was taken as 1.225 kg/m^3, following MIT Technology Review (2011) recommendations, which is appropriate for areas with an average temperature of 15°C (in Ukraine annual air temperature mean is 15.5°C [climate-data.org, 2021])

c. Seasonality within each site

Kruskal-Wallis and Dunn-Bonferroni tests detected the significance of monthly wind speed differences between sites, and p-values were subsequently plotted on the graph. Output was supported with a compound bar and line chart for each site's monthly mean with a standard deviation to characterize the differences. Weibull plots were created for each month, and the same aspects were calculated as in section **a**.

3.3 CMIP6 future projections

Three CMIP6 scenarios were selected to represent the most diverse pathways of our future society. SSP1-2.6 and SSP5-8.5 assume rapid economic growth, featuring substantial investment in education and health. SSP1-2.6 presumes a shift towards sustainability practices, low material growth with mitigation-orientated climate change policies. In contrast, SSP5-8.5 assumes intensive fossil-fuel exploitation with a geoengineering approach to climate change management. SSP2-4.5 is commonly referred to as the 'middle of the road' scenario, where development patterns and climate change management are consistent with historical trends.

Data were filtered to surface wind speed variable, monthly frequency (for quicker computation) and lowest available nominal resolution (50km to capture variation between sites). The CNRM-CERFACS CNRM-CM6-1-HR model ensemble satisfied the filtering criteria for SSP1-2.6 and SSP5-8.5, while only a 100km resolution dataset was available for SSP5-8.5. The first simulations,'r1i1p1f2', were selected for all scenarios, a common practice in climate model assessment (Kim, 2020). The netCDF4 data format was then transformed to time-series (2015-2100) by extracting values at the centroid coordinates of potential sites, stored in a data frame, and extrapolated to 100m hub height via Equation 3. The same characteristics (**a, b, c**) were evaluated to look at the differences in past and future wind regimes.

4. <u>Results</u>

4.1 Site Selection

Figures 4-8 represent each spatial constraint groups with corresponding buffers, and Figure 9 shows spaces in the Kyiv region satisfying every constraint where the three largest areas analysed further. Suitable areas constitute 28.64% of the territory, where selected sites are 560.86km^2, 172km^2 and 193km^2 are 72.31% of the total. Towns and villages with 1000m buffer totalled 8553.89km^2, that is 30.4% of Kyiv region (28121km^2) – biggest constraint studied. Slope’s spatial distribution was flatter areas in the south and steeper in the north (Polesian Lowland) and east (Dnieper Lowlands). However, only within Kyiv city slope ranges from 0 to 60.35% corresponding to the Dnieper Upland, particularly the Kyiv plateau, the rest of the region has slopes up to 27.4%. Selected sites fall within Dnieper upland's geographic region, thus having higher elevation but uniformly flat topography, corresponding to faster wind speeds.

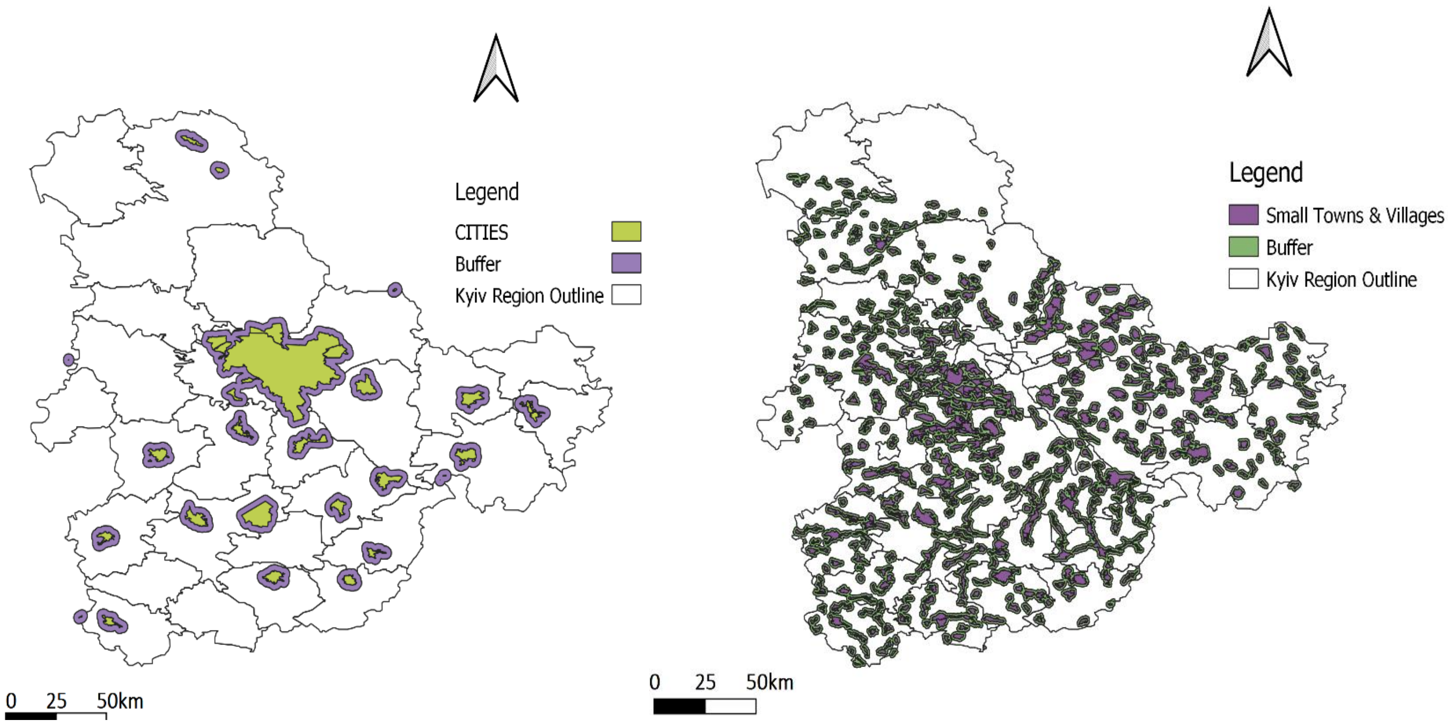

Figure 6: Administrative Areas & Buffers

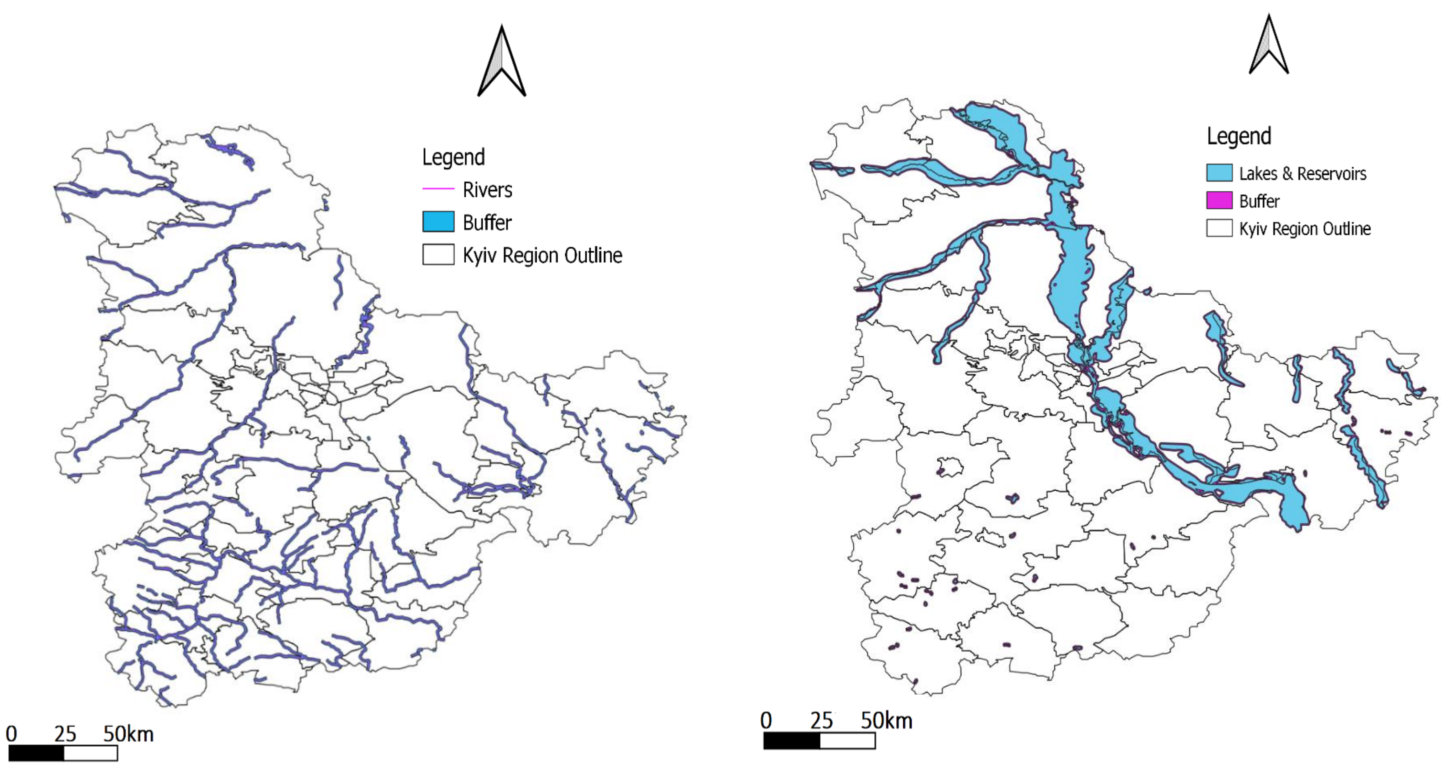

Figure 7: Waterbodies & Buffers

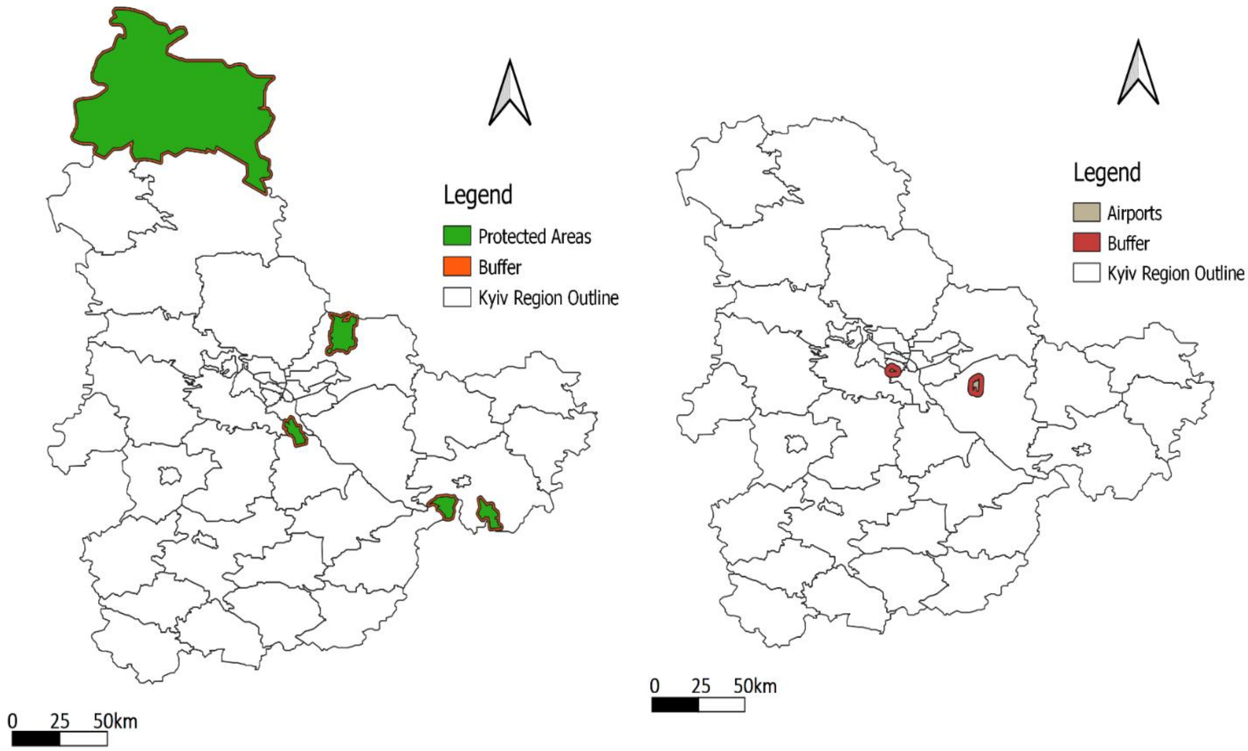

Figure 7: Safety-related areas & Buffers

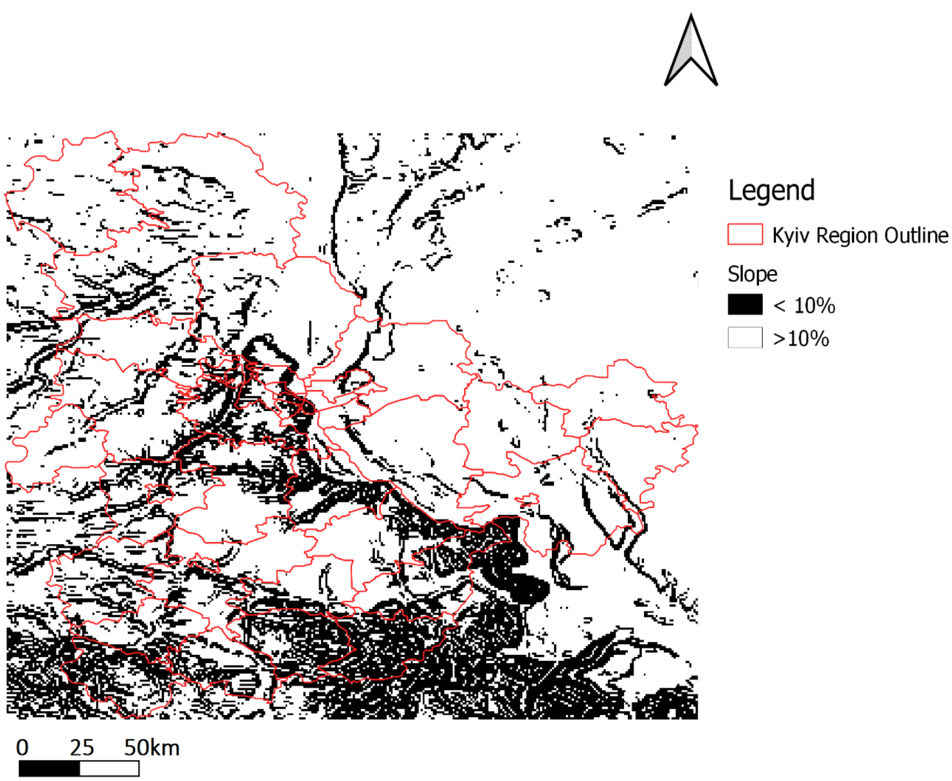

Figure 8: Slope Map

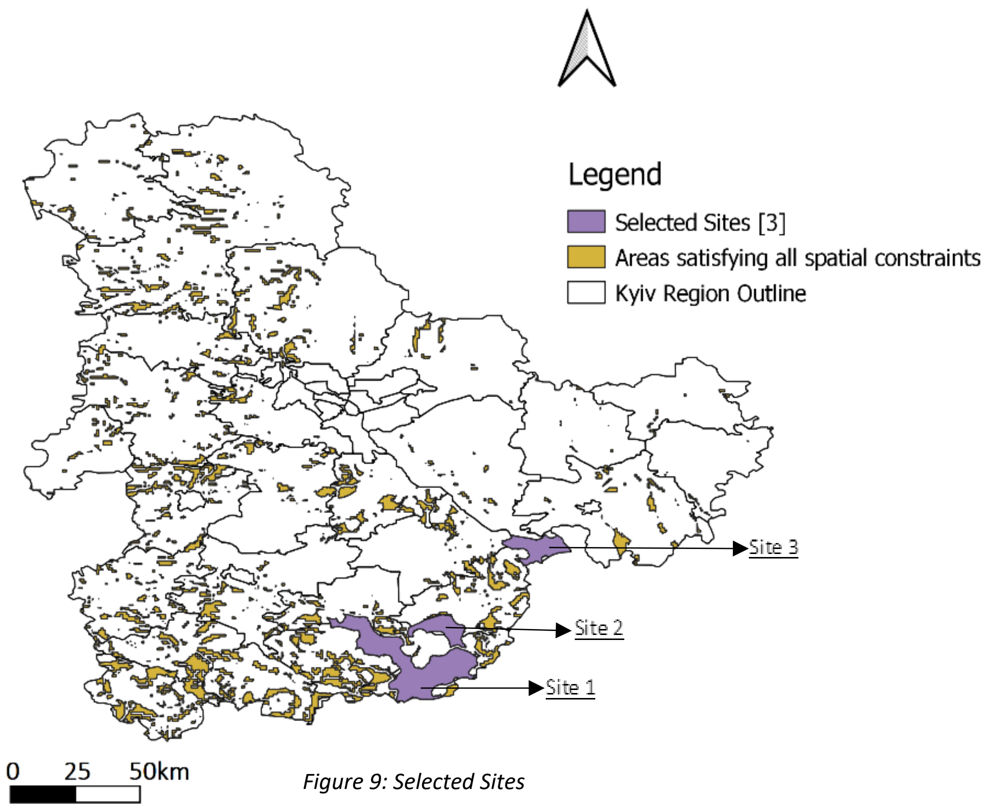

Figure 9: Selected Sites

4.2 ***An assessment of historical wind regimes***

Figure 10 shows a time-series of average wind speeds at each site. Variability is very prominent, notably, a spike of 11.57ms^{-1} in 1993 and a trough of 9.5 ms^{-1} in 2011. The deviations from the means are below 1ms^{-1}, while the annual rates of change range between -8.6% to +8.4% for the first and second sites, and between -10.22 % and 8.75% for the third site. High fluctuations lead to a negligibly low mean rate of change for the whole data time-series, but since 2000, the slowdown accelerates (Table 9). Low standard deviation values suggest consistent wind speed behaviour and given that results relate to a 100m hub height, wind turbines are likely to deliver steady output (Ozay, 2016).

Table 9: Descriptive Statistics of historical data

	Mean ms^{-1}	Median ms^{-1}	Standard Deviation	Change Rate ms^{-1}/year	Mean (since 2000)	Median (since 2000)	Standard Deviation (since 2000)	Change Rate ms^{-1}/year (since 2000)
Site 1	10.58	10.57	0.48	-0.30	10.34	10.38	0.37	-0.57
Site 2	10.54	10.53	0.50	-0.36	10.27	10.31	0.36	-0.63
Site 3	10.49	10.47	0.49	-0.38	10.22	10.27	0.34	-0.64

The Kruskal-Wallis test determined that differences in sites’ historical wind speed regimes are statistically insignificant, as shown in Figure 11. Boxplots exposed high magnitude outliers, indicating the presence of wind gusts, constituting 1.3% of all recordings for sites 1 and 2 and 1.4% for site 3.

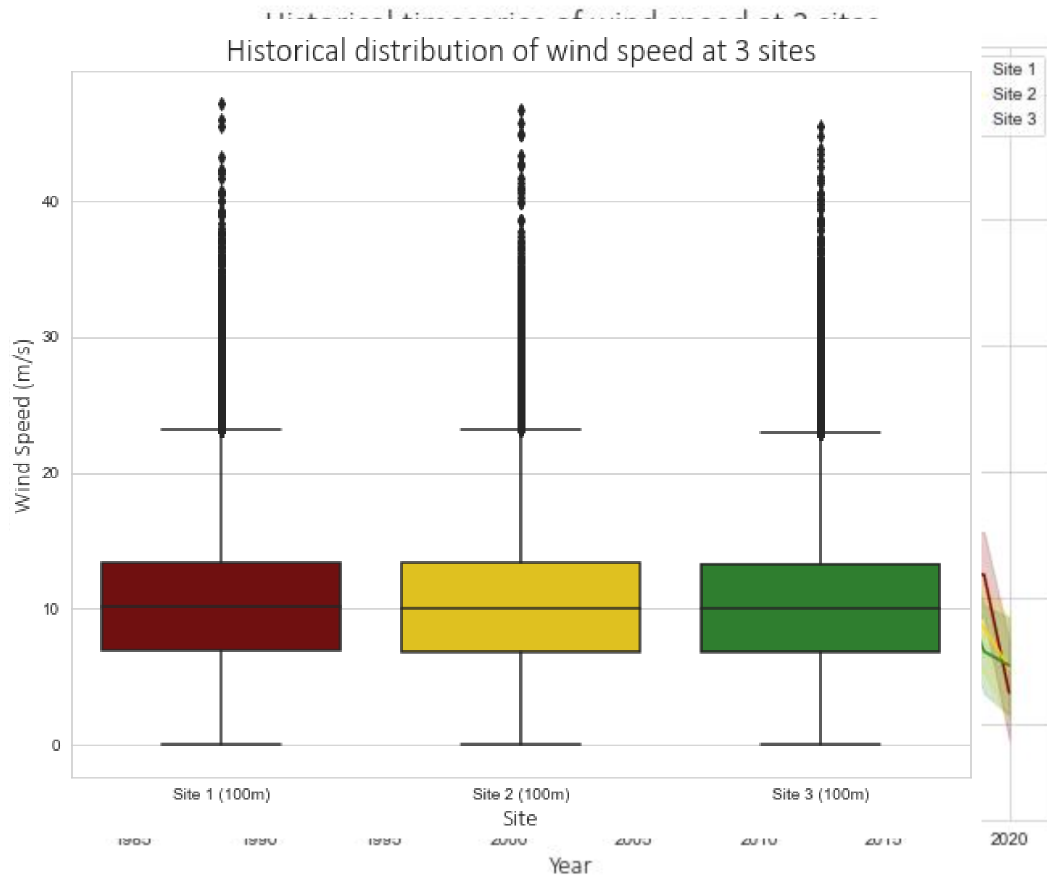

Figure 11: Data distribution at each site

Weibull distribution plots identify the frequency of different wind speed levels and determine the dominant wind speed value per site. The peak of the Weibull distribution - f(v) is the most frequently occurring wind speed per site, while the cumulative Weibull distribution – F(v) represents the proportion of time when wind speed is below a specific value. Figure 12 graphically represents the distribution function, showing that each wind speed value's probability of occurrence is the same at all sites. Speeds of 8ms-1 and 9ms-1 are the most frequent and occur 7.5% of the time. The probability of wind speeds equal to or above the cut-off speed - 25ms^{-1} is 1.27%. Moreover, values equal to or below the cut-in speed of 5ms^{-1} have a 9.66% probability. Thus, wind speeds within the turbine's operational range occur 89.07% of the time for all sites.

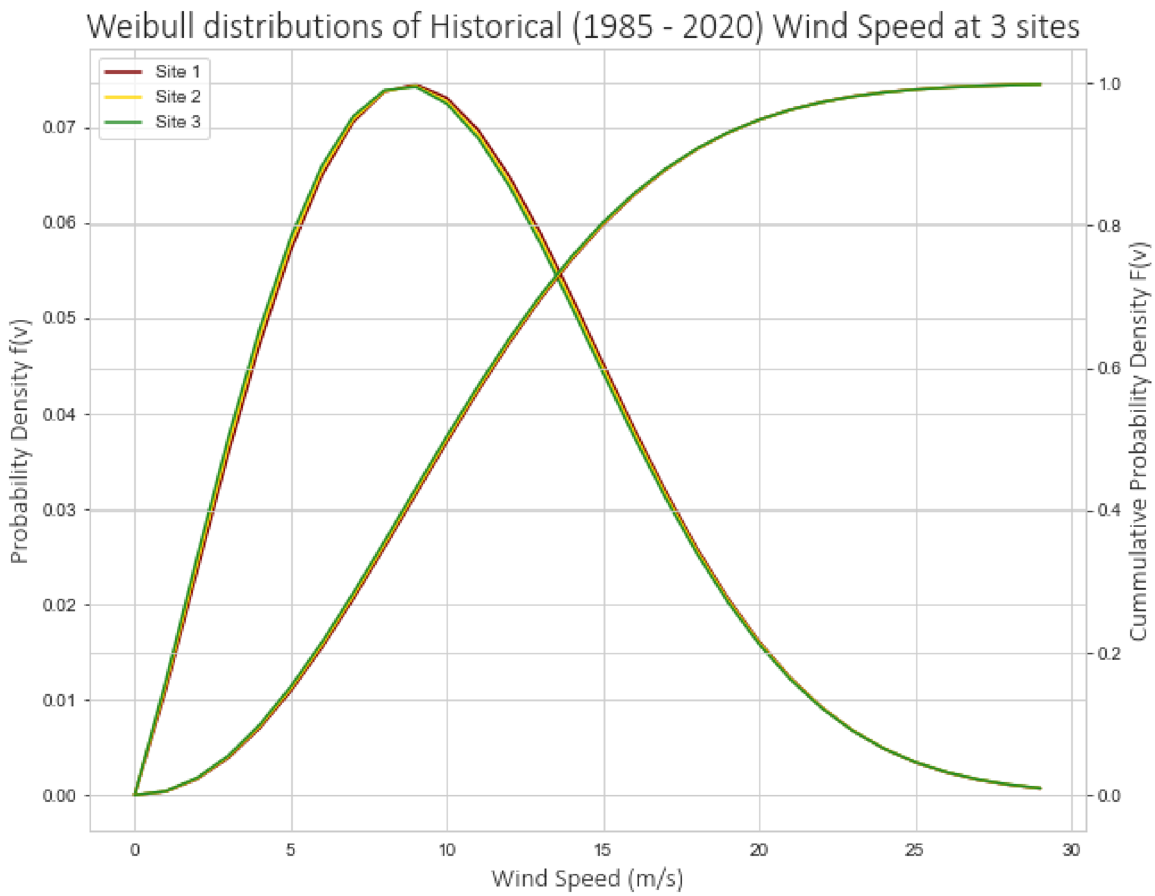

Figure 12: Weibull distribution of Historical wind speed data

The potential power output was calculated by applying Equation 1 to each absolute value of wind speed and multiplying by the corresponding probability (f[v]). Power outputs were very similar between sites due to near-identical wind regimes and averaged at 1284.39Wm^{-2}. However, site 1 had slightly higher results of 1292.71Wm^{-2}, which was 17.07Wm^{-2} and 7.88Wm^{-2} higher than site 3 and site 2 respectively. Wind speed carrying the largest amount of power was 16ms^{-1} resulting in an identical value of 96.38Wm^{-2} at each site.

During a seasonality analysis, the Kruskal-Wallis test showed that there are only statistically significant differences between sites throughout the summer period (July-September). Figure 13 visualized the post-hoc test results, where statistically significant differences are the most prolonged (3months) between sites 1 and 3 and the shortest (1month) for sites 2 and 3.

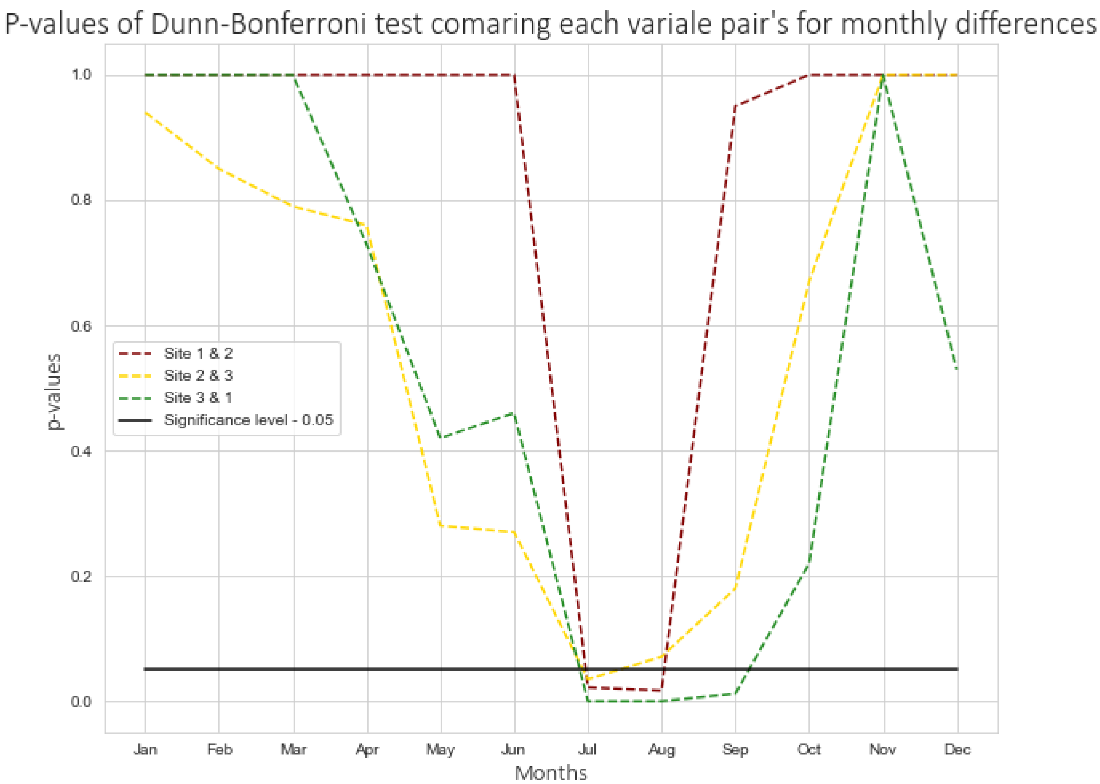

Figure13: Dunn-Bonferroni test output for monthly differences

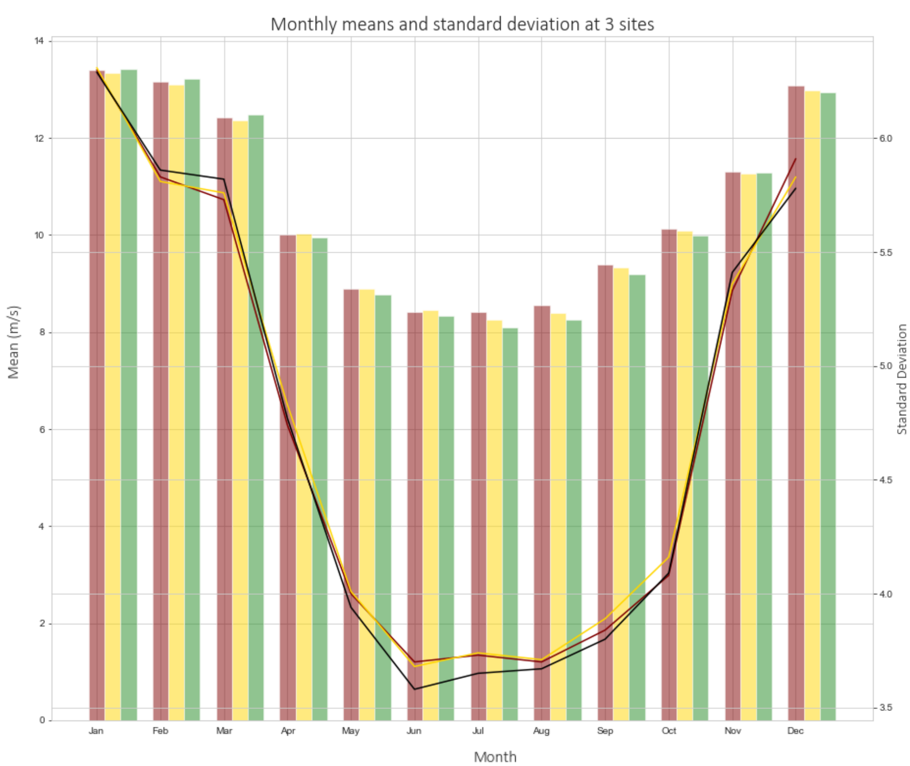

Figure14: Monthly mean and standard deviation

Figure 14 depicts the annual cycle of wind speeds, analyzed in monthly averages. Specifically, summer slowdown, following a winter speed up. During summer, the standard deviation is approximately half of the winter period for all sites, and during May and September, it is close to 60% of the average winter value (– 6°C). Thus, during summer, resource supply is more stable but lower in magnitude. Simultaneously, during winter, average wind speed is higher but less stable with a greater probability of wind gusts, corresponding to larger power output but with higher variability.

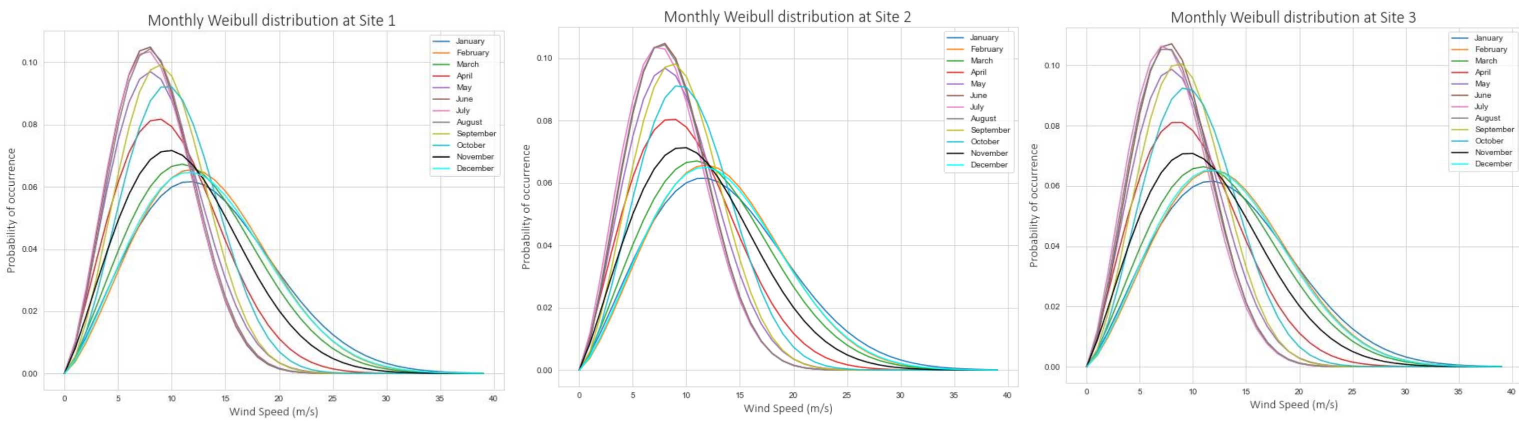

Figure15: Monthly Weibull distribution at each site

Monthly Weibull distribution plots, found in Figure 15, disclose that historical seasonality fluctuations are near-identical at three sites. Thus, average f(v) and F(v) was used to examine trends and estimate variations in power output. During the summer season, the line plot peaks at lower wind speed values,7-8ms^{-1}, while during autumn and winter, range of wind speeds increases, where the most frequent values are 12-14ms^{-1}. The probability of winds above the cut-off speed in winter is 5%, and during summer, it is negligible - 3.9 x 10-5%. Conversely, winds below cut-in speeds are more frequent in summer. Specifically, 18.3%; more probable than in winter by7.7%. Nevertheless, in the winter season, winds stay within the turbine’s operational range 87% of the time, whereas in summer, this figure drops to 82%, confirming the uneven distribution of power output throughout the year. Table 10 documents monthly fluctuations of power output and average values by season. In summer, power output drops by approximately $\frac{2}{3}$rds, when compared to winter, while during spring and autumn, power quantities are 30% lower than the winter measure. High fluctuations are related to the cubic relationship of wind speed and power.

Table 10: Power output per season

Power Output (Wm^{-2})	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
	1676.62	1690.86	1703.80	1541.94	999.48	689.31	575.09	559.56	580.29	755.21	943.41	1302.16
	Winter			Spring			Summer			Autumn		
	1690.43			1076.91			571.49			1000.26		

4.3 Examination of the climate model's data

Figure 16 shows a wind speed time-series at each site per climatic scenario. Due to the sites' proximity to each other and 50km spatial resolution of climate data, centroids of sites 1 and 2 fell within the same grid for SSP1-2.6 and SSP2-4.5. In SSP5-8.5 sites fell within the same grid due to 100km resolution. Identical data distributions are illustrated via overlapping lines in time-series and boxplots (Figure 17). Yearly means vary around $7ms^{-1}$ for all scenarios with a +- 0.5ms-1 deviations from the mean. Due to monthly recordings instead of hourly, the climate models data's 95% confidence interval (shaded region) has a larger range ~8.5-$6ms^{-1}$ and there are no recordings of wind gusts, as in hourly historical data. For an overlapping period (2015-2020), average correlation coefficients per site are high in all scenarios – 0.72 – SSP1-2.6; 0.63 – SSP2-4.5; 0.59 – SSP5-8.5, indicating the reliability of the model ensemble chosen for the analysis. Climate models generate means 0.37 -$0.57ms^{-1}$ lower, while annual rates of change are slightly higher than in historical recordings, ranging between -9.5 - + 10.6% (SSP1-2.6); -10.32 - + 13.76%(SSP2-4.5) and -10.16 - +12.96%(SSP5-8.5), yielding smaller average changes (Table 11). Each scenario projects relative wind speed consistency throughout the 21st century, but for the highest emission scenario (SSP5-8.5), the net change is negative, while the other two climatic projections assume a negligible mean increase rate. Standard deviation values are lower than in historical data and similar between projections, implying more steady wind speeds and future power output.

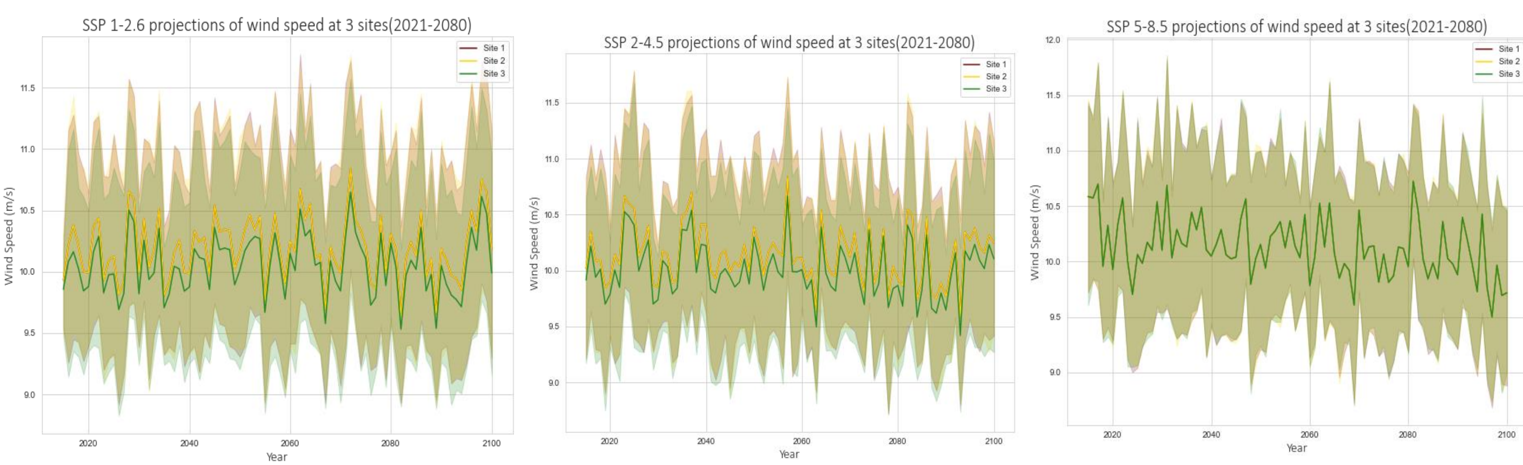

Figure16: Timeseries f wind speed per site at tree SSP scenarios

Table 11: Descriptive statistics of climatic scenarios

		Mean ms^{-1}	Median ms^{-1}	Standard Deviation	Change Rate %/year
SSP 1-2.6	Site 1 & 2	10.20	10.09	0.26	0.14
	Site 3	10.06	9.95	0.24	0.12
SSP2-4.5	Site 1 & 2	10.14	10.12	0.26	0.15
	Site 3	10.00	10.00	0.25	0.15
SSP5-8.5	Site 1, 2 & 3	10.13	10.03	0.27	-0.02

The Kruskal-Wallis test confirmed that there are significant differences between historical wind regimes and climatic projections for each site. Figure 17 presents the primary differences in data distribution, demonstrating the disappearance of wind gust and maximum speeds reaching $11.5ms^{-1}$, which is less than half of historical maximums. The minimum speed is around $4ms^{-1}$ for all scenarios, compared with $0ms^{-1}$ found in historical data, due to monthly frequency instead of hourly, when windless periods are more likely.

Weibull distribution plots (Figure 18) are leptokurtic (high kurtosis - 0.65 mean) for each climatic scenario, peaking at 7ms-1 with an average probability of 18.7%. Winds are projected not to reach a cut-off speed of 25ms-1, so only winds slower than the cut-in speed of $5ms^{-1}$ limit the operational range, occurring on average 3.19% of the time at sites 1 and 2, and 3.29% of the time at site 3. Thus, winds are expected to remain in the turbine's operational range 96.81% of the time at sites 1 and 2, and 96.71% of the time at site 3. Thus, in future operational periods are expect to rise by ~7.69%. However, slower winds ultimately resulted in 41% less power output, an average of $763.12Wm^{-2}$. While power output differences between sites are minor, sites 1 and 2 have slightly higher values ($784.38Wm^{-2}$ – SSP1-2.6 ; $770.9Wm^{-2}$ – SSP2-4.5) , than site 3 (751.11 – SSP1-2.6; 740.23 – SSP2-4.5). As all sites fall within the same grid for SSP5-8.5, there is no differentiation between sites and the approximate power output is $768.99Wm^{-2}$.

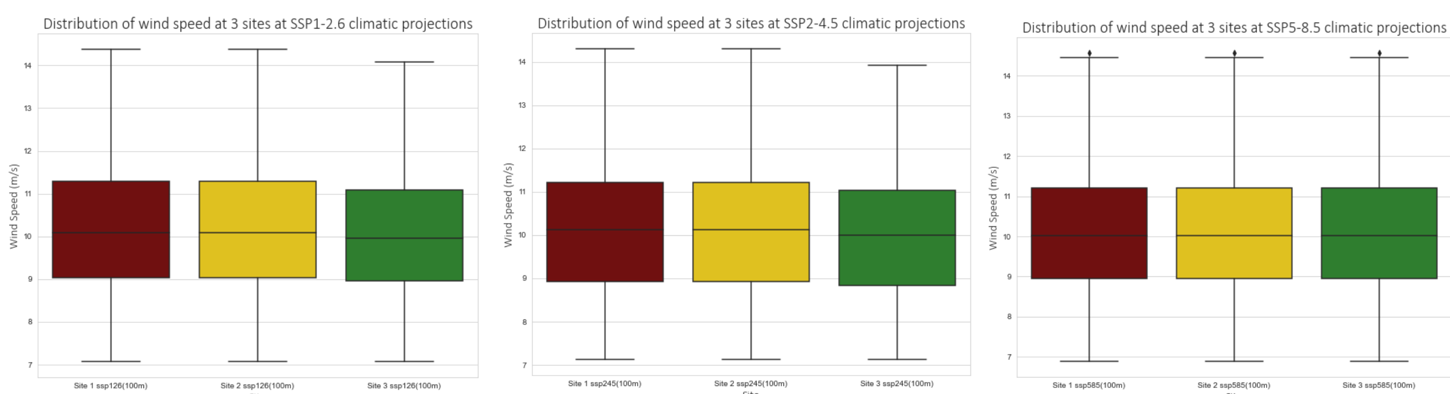

Figure 17: Data Distribution per site and per SSP scenario

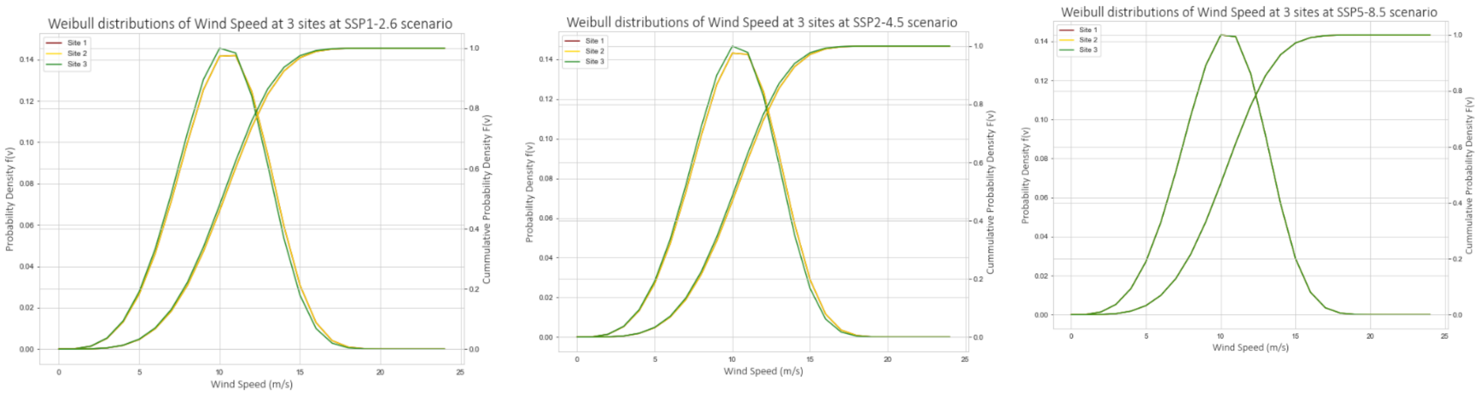

Figure 18: Weibull distributions per scenario and per site

Seasonality analysis with Kruskal-Wallis and Dunn-Bonferroni tests showed no prolonged differences between sites during summer, but some significant differences can be observed for separate months. Sites 1 and 2 and site 3 at SSP2-4.5 were significantly different in March and August, and sites 1 and 2 at SSP1-2.6 and site 3 at SSP2-4.5 were significantly different in March. The combined bar and line plot shows that the wind regimes of summer slowdown and winter speed up will remain under all climatic scenarios. Standard deviation is lowest in July (mean of 0.59) and highest in January (mean of 1.08), indicating increased wind speed stability throughout the year. Furthermore, the mean wind speed difference between winter and summer becomes smaller for climate projections, when compared to historical data. On average, from 1985-2020, wind speed dropped by 40% between summer and winter, while

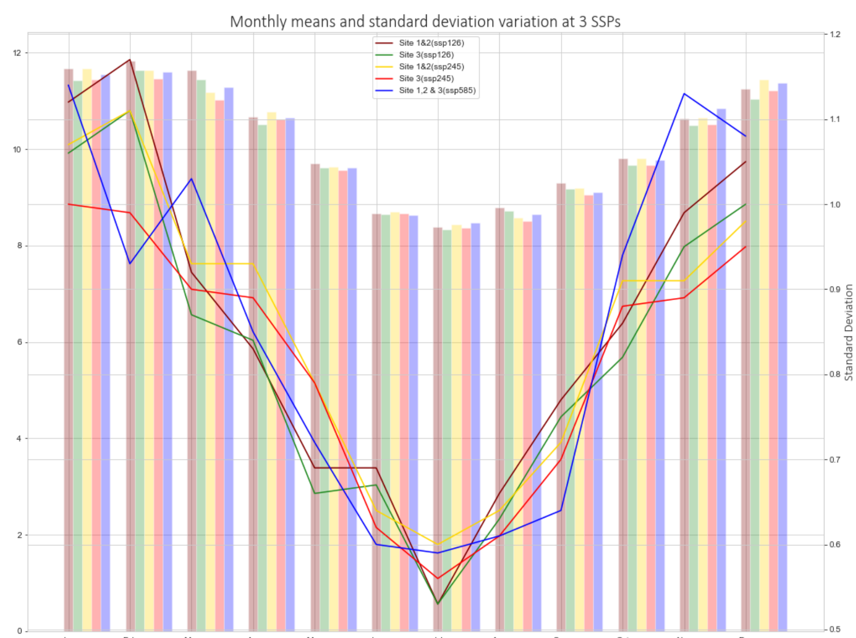

Figure 198: Monthly means and standard deviation at each SSP scenario

when observing climate projections, the average decrease is 28%, indicating more steady resource availability throughout the year.

Monthly Weibull distribution plots per SSP (Figure 20-22), disclose that historical seasonal fluctuations are preserved at each scenario without significant dissimilarities between sites. This is similar when we turn to historical data distribution, where kurtosis is higher during summer (0.11), resulting in high peak probability (18.1% of lower values [8-9ms^{-1}]), while during winter, kurtosis lowers (-0.87) resulting in a larger range of wind speeds and a lower peak probability (-13.3% of higher values [12-13 ms^{-1}]). Winds above the cut-off speed do not occur at any site at any SSP and winds below the cut-in speed have only a minor probability of 1.14% in summer and 0.3% in winter. Correspondingly, the operational range is consistent throughout the year, unlike in historical recordings, implying stable power output.

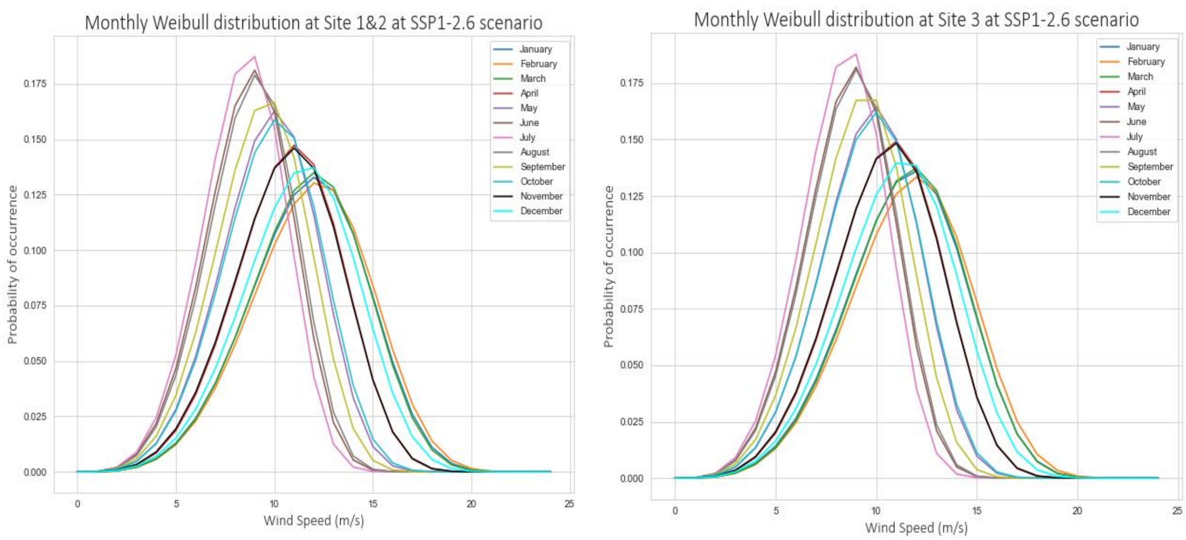

Figure 21: Monthly Weibull distributions at SSP1-2.6

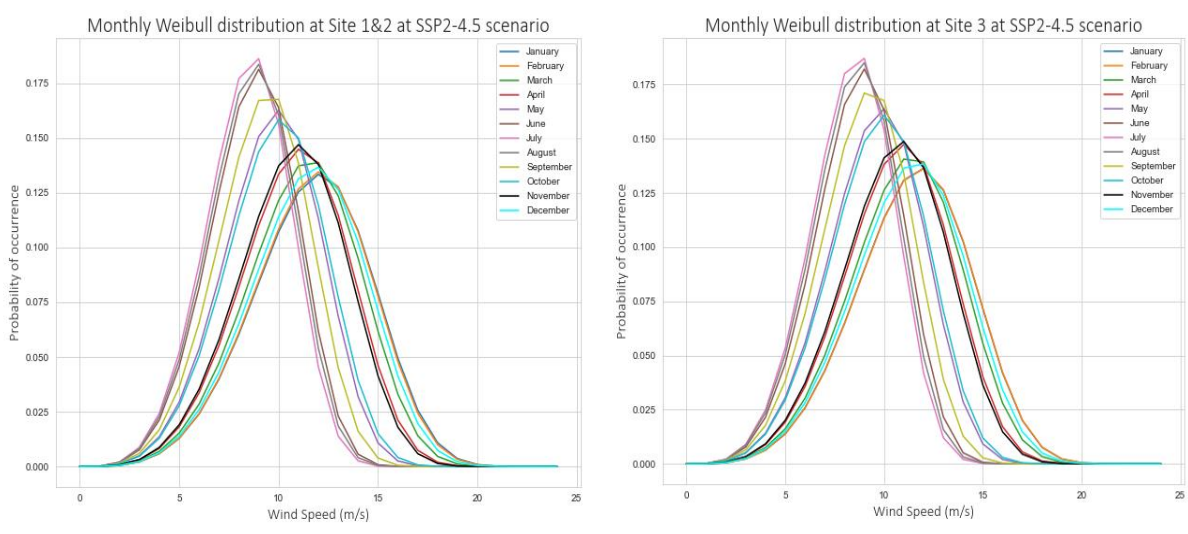

Figure 20: Monthly Weibull distributions at SSP2-4.5

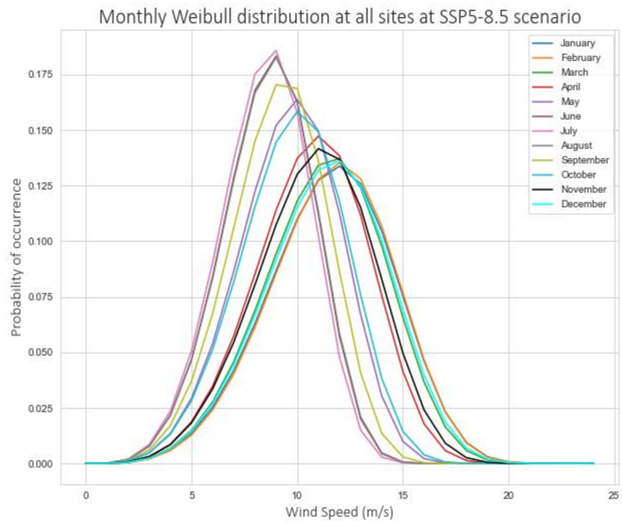

Figure 22: Monthly Weibull Distribution at SSP5-8.5

Table 12: Power output per season, site and SSP

<table>
<tr><td rowspan="9">Power Output (Wm^{-2})</td><td rowspan="3">SSP1-2.6</td><td>Dec</td><td>Jan</td><td>Feb</td><td>Mar</td><td>Apr</td><td>May</td><td>Jun</td><td>Jul</td><td>Aug</td><td>Sep</td><td>Oct</td><td>Nov</td></tr>
<tr><td>1005.8</td><td>1119</td><td>1175.9</td><td>1116.1</td><td>860.8</td><td>652.1</td><td>469.3</td><td>422.9</td><td>485.8</td><td>570.6</td><td>669.5</td><td>856.7</td></tr>
<tr><td colspan="3">Winter</td><td colspan="3">Spring</td><td colspan="3">Summer</td><td colspan="3">Autumn</td></tr>
<tr><td></td><td colspan="3">1100.2</td><td colspan="3">876.3</td><td colspan="3">459.3</td><td colspan="3">698.9</td></tr>
<tr><td rowspan="2">SSP2-4.5</td><td>1054.9</td><td>1122.1</td><td>1116.9</td><td>993.3</td><td>888.7</td><td>641.8</td><td>473.9</td><td>428.9</td><td>449.8</td><td>550.4</td><td>672.1</td><td>860.2</td></tr>
<tr><td colspan="3">1098</td><td colspan="3">841.3</td><td colspan="3">450.9</td><td colspan="3">694.2</td></tr>
<tr><td rowspan="2">SSP5-8.5</td><td>1069.9</td><td>1123.1</td><td>1132.5</td><td>1043.8</td><td>878.3</td><td>644.2</td><td>464.7</td><td>440.2</td><td>468.2</td><td>546.9</td><td>679.8</td><td>928.4</td></tr>
<tr><td colspan="3">1108.5</td><td colspan="3">855.4</td><td colspan="3">457.7</td><td colspan="3">718.4</td></tr>
</table>

Despite the much longer operational time periods of climate model projections, slower winds within it yield significantly lower power output. Table 12 documents seasonal power output fluctuations per climatic scenario, where average f(v) was substituted in Equation 1, due to insignificant differences between sites. Seasonal differences in power output are like the historical data for each scenario, as summer values are approximately 60% lower than in winter. Overall, power outputs of climate projections are lower for each season: 35% in winter, 20% in spring and summer and 30% in autumn. For autumn and winter, the largest power outputs are predicted within the SSP5-8.5 scenario, while for spring and summer the SSP1-2.6 scenario provides the most optimistic results. However, 35Wm^{-2} is the maximum difference, which is statistically insignificant. Consequently, when observing all projections, a near-identical decrease in wind resource is expected at every site examined.

5. Discussion

5.1 Site suitability

The suitability map generated (Figure 9) exposed sufficient availability of space for wind farm development– 28.64% of the study area (8,053km), in contrast to other studies examining spaces around or within metropolitan areas, e.g., New York State (Haaren, 2011). Interestingly, the wind speed was not a constraint in the Kyiv region, as its minimum value of $5.54ms^{-1}$ is above the turbines' cut-in speed of $5ms^{-1}$. Thus, the Kyiv region has excellent conditions for wind energy generation, compared to other recently analysed areas within developing economies, e.g., Wafangdian, China (Xu, 2020), where only 3.36% of the study area ($16km^2$) was suitable for turbines installation or Saudi Arabia, where only 1.85% of the country has annual mean wind speeds above $5ms^{-1}$.

In contrast, the slope factor had the largest influence on the location of selected sites and their spatial extent. This map layer changed the distribution of suitable areas from complete spatial randomness to a noticeable clustering in the South. As a higher gradient corresponds to higher construction and maintenance costs (Elmahmoudi, 2020), a lower end of slope constraint was chosen for this study to minimise these costs. The decision was influenced by a 4.4% economy contract in 2020 due to COVID-19 (NBU, 2021), implying a potentially limited amount of national funds for. Nevertheless, a computational fluid dynamic model designed by Antonini (2020) suggests that optimal positioning of turbines in complex topography can improve energy yield by up to 6%. Therefore, by increasing initial investment and adopting innovative strategies, other locations within the Kyiv region could become suitable for a wind farm.

Thus, spatial analysis of this research generated legally, technically, and environmentally suitable sites for the development of large-scale wind farms, that are most cost-effective in terms of construction. While the largest sites are concentrated in the South (purple polygons – Figure 9), smaller areas of suitable land are distributed around the whole region (yellow polygons – Figure 9). These areas can be used for smaller-scale wind farms to increase renewable energy generation in Ukraine and facilitate the goal of 51-54% reduction in annual CO_2 emissions set by the 2050 Low Emission Development Strategy (Savitsky, 2018).

5.2 Uncertainties of Spatial Analysis

The uncertainty of the final suitability map is related to errors in each dataset transmitted during the mathematical combination of map layers, coordinate systems reprojections and transformation of data formats (raster-vector), yielding compounding effect inaccuracies the final map. Correspondingly, the suitability map is less accurate than the layer with the most significant errors used for its generation. These limitations affect the reliability of further research steps, as wind speed data was downloaded from centroids of map polygons, shapes, and positions, which are subject to uncertainty. Most importantly, the proximity of sites (50km apart) causes high similarity of wind regimes and makes comparison difficult.

5.3 Historical and future wind regimes assessment

Historical data has shown good wind regimes with little wind gust, implying low hazard risks and brief periods of inadequately slow winds, suggesting a low wind farm shutdown probability. The wind slowdown trend, detected since 2000, is not projected to accelerate throughout the 21st century for any SSP scenarios. Thus, the expected relative stability of wind resources is beneficial for wind energy expansion in the studied area, due to consistent and reliable power outputs However, the means of each SSP projection are approximately 4% lower than the historical averages. This outcome is consistent with the findings of Tobin (2018), who suggests a slight slowdown of wind in Eastern Europe throughout the 21st century. Therefore, it is uncertain whether the 100% renewable energy system based on wind proposed by a simulation study by the Lappeenranta-Lahti University of Technology (2020) is feasible for Ukraine. The cubic relationship of wind speed and power output is critical, yielding a 20% -35% decrease of power generated and a sufficient impetus for the diversification of renewable energy sources in Ukraine.

No significant differences were detected between sites for historical or future projections due to their proximity locations. However, the seasonal analysis showed that site 1's wind speed is significantly higher during the summer season over the 1985-2020 period, which is only preserved for March and August when we look at SSP1-2.6 and SSP2-4.5. The mean wind speed is the highest for site 1 when we refer to historical data and the highest for sites 1 and 2 when we refer to future projections. Furthermore, the standard deviation at site 1 is the lowest in the past and nearly identical for all sites in the future. Thus, while statistically insignificant, there is a slight superiority of site 1 due to faster winds speeds that are more stable, implying the evenest energy production.

The Weibull probability distribution revealed that operational ranges are very high for both datasets, reaching a good value of 96% in the future, where most frequently occurring wind speeds are approximately double that of cut-in speeds (10.0ms^{-1} – 10.58ms^{-1}). According to the International Electricity Commission standards (2005), winds above 10ms^{-1} belong to Wind Class 1 (High wind) and correspond to excellent power generation conditions. Therefore, wind conditions are sufficient at all three sites, allowing energy production, with only a slight superiority of sites 1 and 2 over site 3.

Neither wind regimes nor power outputs differ significantly between sites, and wind speeds at all sites are projected to slow down by the same magnitude in the future. Therefore, the size of sites plays a decisive role in the recommendation of site 1 or sites 1 and 2 combined. These sites are more spatially extensive than site 3 and allow the installation of more turbines, producing more energy in total.

Ranges of speeds were high for historical data and future projections with significant differences in data distributions between seasons. Consequently, power output densities are very uneven throughout the year in the selected site, consistent with natural aerodynamic processes. In winter, cold fronts that reach a geographical area are of much lower temperatures than local air masses. During the displacement of warmer air masses, the temperature gradient becomes very high, resulting in increased pressure difference and faster wind speeds (Lazaridis, 2011). In Ukraine, where seasonality is very prominent (summer temperature maximum + 30^{o}C; winter temperature minimum -20^{o}C), seasonal differences in wind speed are significant. Respectively, for such climatic regions, the installation of variable speed wind turbines is suggested (Hansen, 2004; Yin, 2018). These turbines contain synchronous generators and power converters that allow the absorption of wind speed fluctuations caused by wind gusts (Krüger, 2001). Additionally, variable wind speed turbines have reduced noise compared to conventional fixed wind speed turbines (Sedaghat, 2017), inferring a possible reduction of the Ukrainian legal minimum settlement buffer of 700m. Consequently, the site's spatial extent could be extended further, resulting in additional turbine installation and extra power capture.

5.4 Limitations

An important limitation of this study is that wind speed data used to create Weibull distributions did not capture the spatial variations within site. Historical data could be downloaded only for specific coordinates (site centroids). Simultaneously, it was possible to compute a spatial average of wind speed in climatic data, a more reliable method in wind regime assessment (Cellura, 2008). However, different data deriving methodologies would result in much less reliable comparisons. Thus, decreased representativeness of data is a limitation of this study that could be addressed in the future by allocating more funds to attain data from metoblue.com, where data for a set of three coordinates costs €272. While spatially averaged over ten years wind speed data used as a spatial analysis layer was an alternative option, it would not uncover historical trends, which is a more critical limitation.

It is important to note that wind direction is also subject to climate change, but its analysis was beyond this study's scope, imposing a significant limitation. Wind farms are planned according to the primary wind direction (Chowdhury, 2013) to maximise the orographic speed up effect

(Devis, 2018). Consequently, any changes in wind route due to climate change can reduce the power capture at the proposed site.

The international system of wind classification (Ko, 2015) only exists for 10m, 30m and 50m hub heights, while power densities in this research were calculated using a hub height of 100m. Thus, to develop this research, the different power densities at hub heights documented in the classification should be calculated, allowing one to determine the level of the site's suitability for the development of a wind farm. So, another limitation is the inability to classify the potential power production, which is vital for the project development's economic evaluation. Nevertheless, the international standards themselves require an update, as according to the literature review, modern wind turbines are usually at least 100m tall and are expected to become higher. Correspondingly, the addition of power density classes at the height of 100m+ would be beneficial for benchmarking power generation at the hub height of modern turbines. Though to put the results in context, the highest power density (class 7) at a 50m hub height has a range of 800-2,000Wm^{-2} and is referred to as having 'excellent' resource potential, while the annual average power density for historical data was 1,292.71Wm^{-2}, and for climatic projections – 768.99Wm^{-2}.

5.5 Research Contributions

In most published climate change papers, the focus is skewed towards the highest warming projections, particularly RCP8.5 in CMIP5. Although it is the most realistic scenario, given the current magnitude and growth rate of emissions (Devis, 2018), it is critical to illustrate the alternative possibilities of future environmental states, as done in this dissertation. While no significant differences in wind speed under contrasting SSPs were detected, variations of effects may be more prominent for other areas of Ukraine or the world. Thus, the designed methodology is a valuable contribution to scientific research that could be developed further by adding climate models in the ensemble and the remaining SSPs.

Finally, the implementation of CMIP6 models in this study is a crucial originality aspect. High correlation coefficients correspond to the reliability of the projections; however, it is only time that will reveal how well CMIP6 were able to predict climatic changes. To date to issues of wind speed variable were not captured by the ES-DOCs Errata Service. However, CMIP6 are the newest models, so when more research will get published, more uncertainties and miscalculations of projections could be detected. Thus, the reliability of findings in this study is subject to change with improving knowledge and understanding of CMIP6 models era.

6. Conclusion

The outcome of the presented dissertation is three-fold. Firstly, a set of spatial restriction criteria for wind farm development was defined for Ukraine based on academic research and international policies. This outcome is a valuable research contribution that could be used for the development of legal requirements for wind farms in Ukraine. The spatial analysis based on the defining factor was applied in the Kyiv region, confirming the availability of appropriate space for wind farm development.

Secondly, an assessment of wind regimes was performed for the three largest sites selected during GIS analysis. Seasonal wind regime dynamics and corresponding power output fluctuations were examined, revealing the wind resource's sufficiency at all selected sites. In Ukraine, no similar analysis was published to date, so this research can potentially aid investment interest in the national renewables sector. Lastly, this study contributed to the overall scientific knowledge of future wind speed and power potential by analysing projections of the newest CMIP6 models.

Results confirm the expected negligible slowdown of wind speed over the Kyiv region and reveal that the period when wind speeds are within the turbine's operational range has been extended by 10%. However, a 20-35% reduction in power output confirms that the standard climate-change-related uncertainty of +-2% used by the wind industry is a severe underestimation (Solaun, 2019). Thus, future wind climatology must be analysed in every proposed wind farm project. The methodology presented in this study covers essential steps for selecting the location of a wind farm that is technically and legally appropriate and has sufficient wind resources that will not decrease significantly in future. No study covering all three steps was found in the published literature. Thus, this dissertation is unique in terms of its research extent and is a valuable contribution to the Ukrainian wind energy sector and general wind energy research. To improve this study , an economic assessment would be necessary to confirm that enough power could be generated and sold at the selected site, outweighing the construction and maintenance expenses. Moreover, the magnitude of power output reduction under climate change found in this study is substantial. As such, there is a strong incentive to investigate mitigation measures to limit future climate-change-related wind energy loss.

7. Bibliography

1. A.H. Monahan, Y. He, N. MC Farlane, A. Dai, (2011) The probability distribution of land surface wind speeds, J. Clim. 24 3892e3909.

2. Ackerman, F., & Stanton, E. (2006). *Climate change: the costs of inaction*. Global Development and Environment Institute, Tufts University.

3. Akbarzadeh, S., & Valipour, M. S. (2020). *Energy and exergy analysis of a parabolic trough collector using helically corrugated absorber tube.* Renewable Energy, 155, 735-747.

4. Ameri M, Ghadiri M, Hosseini M. (2006), *Recent advances in the implementation of wind energy in Iran.* In: The 2nd joint international conference on sustainable energy and environment (SEE 2006) 21–23 November 2006, Bangkok, Thailand, 2006

5. Antonini, E. G., Romero, D. A., & Amon, C. H. (2020). *Optimal design of wind farms in complex terrains using computational fluid dynamics and adjoint methods*. Applied Energy, 261, 114426.

6. Aydin NY, Kentel E, Duzgun S. (2010) GIS-based environmental assessment of wind energy systems for spatial planning: a case study from Western Turkey. Renewable and Sustainable Energy Reviews 14:364–73.

7. Baffoe, P. E., & Sarpong, D. (2016). Selecting suitable sites for wind energy development in Ghana. Ghana Mining Journal, 16(1), 8-20.

8. Barclay, Robert; E. F. Baerwald; J.C. Gruver (2007). *"Variation in bat and bird fatalities at wind energy facilities"* (PDF). Canadian Journal of Zoology. 85 (3): 381–87. doi:10.1139/Z07-011.

9. Baseer, M. A., Rehman, S., Meyer, J. P., & Alam, M. M. (2017). *GIS-based site suitability analysis for wind farm development in Saudi Arabia.* Energy, 141, 1166-1176.

10. Basu, B., Tiwari, D., Kundu, D., & Prasad, R. (2009). *Is Weibull distribution the most appropriate statistical strength distribution for brittle materials?.* Ceramics International, 35(1), 237-246.

11. Bennui, A., Rattanamanee, P., Puetpaiboon, U., Phukpattaranont, P., & Chetpattananondh, K. (2007, May). *Site selection for large wind turbine using GIS.* In PSU-UNS international conference on engineering and environment (pp. 561-566).

12. Bhadare, S. (2020) *CMS Energy buys majority stake in Aviator Wind* [online], Available at: https://iclg.com/ibr/articles/14193-cms-energy-buys-majority-stake-in-aviator-wind, Accessed (18.02.2021)

13. Billinton, R., & Wangdee, W. (2007). *Reliability-based transmission reinforcement planning associated with large-scale wind farms.* IEEE transactions on power systems, 22(1), 34-41.

14. Bosello, F., & Parrado, R. (2014). *Climate change impacts and market driven adaptation: the costs of inaction including market rigidities.*

15. CAA Policy and Guidelines on Wind Turbines, Sixth Edition (February 2016) [online] Available at: https://publicapps.caa.co.uk/docs/33/CAP764%20Issue6%20FINAL%20Feb.pdf, Accessed 25.02.2021

16. Cellura, M., Cirrincione, G., Marvuglia, A., & Miraoui, A. (2008). *Wind speed spatial estimation for energy planning in Sicily: A neural kriging application.* Renewable energy, 33(6), 1251-1266.

17. Cess, R. D., Hameed, S., & Hogan, J. S. (1980). *Response of the global climate to changes in atmospheric chemical composition due to fossil fuel burning.*

18. Chowdhury, S., Zhang, J., Messac, A., & Castillo, L. (2013). *Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions*. Renewable Energy, 52, 273-282.

19. Climate Change statistics for Ukraine, Country Summary[online], Available at: https://climateactiontracker.org/countries/ukraine/, Accessed: 13.02.2021

20. Constitution of Ukraine [Ukraine], Article 85, Article 102, Article 116 - Clause 14 of Section 15 "Transitional Provisions" (excluded) 28 June 1996, available at: https://www.refworld.org/docid/44a280124.html [accessed 16 February 2021]

21. Dai, K., Bergot, A., Liang, C., Xiang, W. N., & Huang, Z. (2015). *Environmental issues associated with wind energy–A review.* Renewable Energy, 75, 911-921.

22. Deng, K., Azorin-Molina, C., Minola, L., Zhang, G., & Chen, D. (2020). *Global Near-Surface Wind Speed Changes over the Last Decades Revealed by Reanalyses and CMIP6 Model Simulations*. Journal of Climate, 1-53.

23. Devis, A., Van Lipzig, N. P., & Demuzere, M. (2018). *Should future wind speed changes be taken into account in wind farm development?*. Environmental Research Letters, 13(6), 064012.

24. Elma O, Gabbar HA. (2019) *Design and analysis of mobile hybrid energy system for off-grid applications.* In: 2019 International conference on power generation systems and renewable energy technologies; 2019. p. 1–6.

25. Elmahmoudi, F., Abra, O. E. K., Raihani, A., Serrar, O., & Bahatti, L. (2020). *Elaboration of a Wind Energy Potential Map in Morocco using GIS and Analytic Hierarchy Process.* Engineering, Technology & Applied Science Research, 10(4), 6068-6075.

26. Energy Information Administration. (2013). *Average retail price of electricity to ultimate customers*.

27. Erickson, W. P., Johnson, G. D., Strickland, D. M., Young Jr, D. P., Sernka, K. J., & Good, R. E. (2001). *Avian collisions with wind turbines: a summary of existing studies and comparisons to other sources of avian collision mortality in the United States* (No. DOE-00SF22100-). Western EcoSystems Technology, Inc., Cheyenne, WY (United States); RESOLVE, Inc., Washington, DC (United States).

28. EVwind (2020), PowerChina and Ukrainian WindFarm signed a contract on building a $1 billion wind farm in Ukraine, 4.11.2020 Available at: https://www.evwind.es/2020/11/02/powerchina-and-ukrainian-windfarm-signed-a-contract-on-building-a-1-billion-wind-farm-in-ukraine/77973, Accessed: 12.02.2021

29. Ferguson-Martin, C. J., & Hill, S. D. (2011). *Accounting for variation in wind deployment between Canadian provinces*. Energy Policy, 39(3), 1647-1658.

30. Ferreira, P., Lima, F., Ribeiro, F., & Vieira, F. (2019). *A mixed-method approach for the assessment of local community perception towards wind farms.* Sustainable Energy Technologies and Assessments, 33, 44-52.

31. Frey, B. J., & Hadden, P. J. (2007). *Noise radiation from wind turbines installed near homes: Effects on health*[online], Available at: http://docs.wind-watch.org/wtnoisehealth.pdf, Accessed (22.02.2021)

32. Gao, J., Guo, F., Ma, Z., Huang, X., & Li, X. (2020*). Multi-criteria group decision-making framework for offshore wind farm site selection based on the intuitionistic linguistic aggregation operators*. Energy, 204, 117899.

33. Gobiet, A., Kotlarski, S., Beniston, M., Heinrich, G., Rajczak, J., & Stoffel, M. (2014). *21st century climate change in the European Alps—A review.* Science of the Total Environment, 493, 1138-1151.

34. Hansen AD, Sørensen P, Iov F, Blaabjerg F. (2004*). Control of Variable Speed Wind Turbines with Doubly-Fed Induction Generators.* Wind Engineering. 2004;28(4):411-432. doi:10.1260/0309524042886441

35. Haugen, K. M. (2011). *International Review of Policies and Recommendations for Wind Turbine Setbacks from Residences: Setbacks, Noise, Shadow Flicker, and Other Concerns*. Minnesota Department of Commerce: Energy Facility Permitting.

36. Hennessey Jr, J. P. (1977). *Some aspects of wind power statistics*. Journal of Applied Meteorology and Climatology, 16(2), 119-128.

37. IEA (2020), Ukraine energy profile, IEA, Paris https://www.iea.org/reports/ukraine-energy-profile

38. IEA International Energy Agency, World energy outlook 2019: electricity [WWW document] URL https://www.iea.org/reports/world-energy-outlook-2019/electricity#abstract (2020), Accessed 14th Jan 2020

39. IEC 61400-12:1998(E) Wind turbine generator systems – Part 12: Wind turbine Power

40. IPCC, 2018: Summary for Policymakers. In: *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty* [Masson-Delmotte, V., P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R.

Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield (eds.)]. *World Meteorological Organization, Geneva, Switzerland, 32 pp.*

41. J. Ryan Zimmerling, Andrea C. Pomeroy, Marc V. d'Entremont and Charles M. Francis,(2013), *"Canadian estimate of bird mortality due to collisions and direct habitat loss associated with wind turbine developments",* Avian Conservation & Ecology, 2013, v.8 n.2.
42. Jackson, C. A. (2013) *Wind farms and radars living together?* [online] Availabel at: https://www.internationalairportreview.com/article/12201/wind-farms-and-radars-living-together/, Accessed: 25.02.2021
43. Jamil M, Parsa S, Majidi M. *Wind power statistics and an evaluation of wind energy density*. Renew Energy 1995;6:623–8.
44. Kalair, A. R., Abas, N., Hasan, Q. U., Kalair, E., Kalair, A., & Khan, N. (2019). *Water, energy and food nexus of Indus Water Treaty: Water governance.* Water-Energy Nexus, 2(1), 10-24.
45. Kalair, A., Abas, N., Saleem, M. S., Kalair, A. R., & Khan, N. (2020). *Role of energy storage systems in energy transition from fossil fuels to renewables*. Energy Storage, e135.
46. Khan N, Kalair E, Abas N, Kalair AR, Kalair A. (2019) *Energy transition from molecules to atoms and photons.* Engineering Science and Technology, an International Journal. 2019 Feb 1;22(1):185-214.
47. Kim, Y. H., Min, S. K., Zhang, X., Sillmann, J., & Sandstad, M. (2020). *Evaluation of the CMIP6 multi-model ensemble for climate extreme indices.* Weather and Climate Extremes, 29, 100269.
48. Kjeller Vindteknikk AS (2017) *Icethrow from wind turbines*. Available at: http://winterwind.se/wp-content/uploads/2015/08/3_3_28_Bredesen_IEA_Task_19__IceRisk_Review_of_current_knowledge_and_the_way_forward_in_risk_assessments_associated_with_ice_throw_from_wind_turbine_blades_Pub_v1-1.pdf Accessed (19.01.2021)
49. Ko, Donghui & Jeong, Shin & Kang, Keum-Seok. (2015). *Assessment of Offshore Wind Power Potential in the Western Seas of Korea.* Journal of Korean Society of Coastal and Ocean Engineers. 27. 266-273. 10.9765/KSCOE.2015.27.4.266.
50. Kozak, M. (2019) Poor roads are impeding Ukraine's economic development. [online] Available at: https://www.obserwatorfinansowy.pl/in-english/poor-roads-are-impeding-ukraines-economic-development/ Accessed: 10.03.2021
51. Kudria, S., Ivanchenko, I., Tuchynskyi, B., Petrenko, K., Karmazin, O., & Riepkin, O. (2021). *Resource potential for wind-hydrogen power in Ukraine.* International Journal of Hydrogen Energy, 46(1), 157-168.

52. Lantz, E. J., Roberts, J. O., Nunemaker, J., DeMeo, E., Dykes, K. L., & Scott, G. N. (2019). *Increasing wind turbine tower heights: Opportunities and challenges.*

53. Latinopoulos, D., & Kechagia, K. (2015). *A GIS-based multi-criteria evaluation for wind farm site selection. A regional scale application in Greece.* Renewable Energy, 78, 550-560.

54. Lazaridis, M. (2011). *First principles of meteorology*. In First Principles of Meteorology and Air Pollution (pp. 67-118). Springer, Dordrecht.

55. Leonard, M., Shapiro, J., Pisani-Ferry, J., Tagliapietra, S., & Wolff, G. B. (2021). *The geopolitics of the European Green Deal.* Bruegel-Policy Contributions, NA-NA.

56. Levene, H. (1960). *In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, I. Olkin et al. eds., Stanford University Press, pp. 278-292.

57. Levine, M. D., & Steele, R. V. (2021). *Climate change: What we know and what is to be done*. Wiley Interdisciplinary Reviews: Energy and Environment, 10(1), e388.

58. Li, X. (2021). *Future projections of offshore wind energy resources in China using CMIP6 simulations and a deep learning-based downscaling method.* Energy, 217, 119321.

59. Machol, B., & Rizk, S. (2013). *Economic value of US fossil fuel electricity health impacts*. Environment international, 52, 75-80.

60. Mariano, A. (2019) *Wind turbines are getting bigger and taller*. [online] Available at: https://www.pagerpower.com/news/wind-turbines-are-getting-bigger-and-taller/. Accessed : 20.02.2021

61. Marray, M. (2020) *PowerChina and WindFarm team up on US$1 billion project in Ukraine* [online], Available at: https://www.theasset.com/article/42138/powerchina-and-windfarm-team-up-on-us-1-billion-project-in-ukraine, Accessed, 12.02.2021

62. Marshall, A. W., & Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. Biometrika, 84(3), 641-652.

63. Marvel, K., Kravitz, B., & Caldeira, K. (2013). *Geophysical limits to global wind power*. Nature Climate Change, 3(2), 118-121.

64. May, R., Nygård, T., Falkdalen, U., Åström, J., Hamre, Ø., & Stokke, B. G. (2020). Paint it black: Efficacy of increased wind turbine rotor blade visibility to reduce avian fatalities. Ecology and evolution, 10(16), 8927-8935.

65. McMurtry R. (2009). Deputation to the Standing Committee on General Government regarding Bill C-150. [online], Available at: http://windconcernsontario. files.wordpress.com/2009/04/deputation-to-standingcommittee-mcmurtry.pdf, Accessed (22.02.2021)

66. Mostafaeipour, A., Jadidi, M., Mohammadi, K., & Sedaghat, A. (2014). *An analysis of wind energy potential and economic evaluation in Zahedan, Iran.* Renewable and Sustainable Energy Reviews, 30, 641-650.

67. National Bank of Ukraine (NBU), (2021). *In 2021, the economy will almost completely make up for the losses from the coronavirus crisis, according to the NBU Inflation Report* [online], Available at: https://bank.gov.ua/en/news/all/ekonomika-mayje-povnistyu-nadolujit-vtrati-vid-koronakrizi-u-2021-rotsi--inflyatsiyniy-zvit-nbu, Accessed (19.04.2021)

68. Panda, R., & Maity, M. (2021). *Global Warming and Climate Change On Earth: Duties and Challenges of Human Beings*. International Journal of Research in Engineering, Science and Management, 4(1), 122-125.

69. Pedersen, E., van den Berg, G. P., Bakker, R., & Bouma, J. (2009). *Response to noise from modern wind farms in the Netherlands.* Journal of the Acoustical. Society of America, 126, 634-643.

70. Persson Waye, K. (2007). *Wind turbine noise, annoyance and self-reported health and well-being in different living environments*. Occupational and Environmental Medicine, 64, 480-486.

71. Pleßmann, G., Erdmann, M., Hlusiak, M., & Breyer, C. (2014). *Global energy storage demand for a 100% renewable electricity supply.* Energy Procedia, 46, 22-31.

72. Puig, D., Olhoff, A., Bee, S., Dickson, B., & Alverson, K. (Eds.) (2016). *The Adaptation Finance Gap Report*. United Nations Environment Programme.

73. Rubert, T., Zorzi, G., Fusiek, G., Niewczas, P., McMillan, D., McAlorum, J., & Perry, M. (2019). *Wind turbine lifetime extension decision-making based on structural health monitoring.* Renewable Energy, 143, 611-621.

74. S.A. Akdag, H.S. Bagiorgas, G. Mihalakakou, *Use of two-component Weibull mixtures in the analysis of wind speed in the Eastern Mediterranean*, Appl. Energy 87 (2010) 2566e2573.

75. Satkin, M., Noorollahi, Y., Abbaspour, M., & Yousefi, H. (2014). *Multi criteria site selection model for wind-compressed air energy storage power plants in Iran.* Renewable and Sustainable Energy Reviews, 32, 579-590.

76. Sedaghat, A., Hassanzadeh, A., Jamali, J., Mostafaeipour, A., & Chen, W. H. (2017). *Determination of rated wind speed for maximum annual energy production of variable speed wind turbines*. Applied energy, 205, 781-789.

77. Snaff, E. (2020) *New heights in wind energy Generating electricity from wind power is picking up more and more momentum, increasing all around the world.* [online] Available at; https://www.eni.com/en-IT/low-carbon/wind-turbines.html, Accessed: 20.02.2021

78. Sobchenko, A., & Khomenko, I. (2015). Assessment of regional wind energy resources over the Ukraine. Energy Procedia, 76, 156-163.

79. Soeder, D. J. (2021). *Fossil Fuels and Climate Change*. *In Fracking and the Environment (pp. 155-185). Springer, Cham.*

80. Solaun, K., & Cerdá, E. (2020). *Impacts of climate change on wind energy power–Four wind farms in Spain*. Renewable energy, 145, 1306-1316.

81. Song, D., Yang, Y., Zheng, S., Deng, X., Yang, J., Su, M., ... & Joo, Y. H. (2020). *New perspectives on maximum wind energy extraction of variable-speed wind turbines using previewed wind speeds.* Energy Conversion and Management, 206, 112496.

82. Spyridonidou, S., & Vagiona, D. G. (2020). *Systematic Review of Site-Selection Processes in Onshore and Offshore Wind Energy Research.* Energies, 13(22), 5906.

83. Stathopoulos, T., Alrawashdeh, H., Al-Quraan, A., Blocken, B., Dilimulati, A., Paraschivoiu, M., & Pilay, P. (2018). *Urban wind energy: Some views on potential and challenges.* Journal of Wind Engineering and Industrial Aerodynamics, 179, 146-157.

84. Stroiko, T., & Bondar, V. (2017). *Transport infrastructure of Ukraine: The modern realities and development prospects.* Baltic Journal of Economic Studies, 3(2).

85. Taherdoost, H. (2016). *Sampling methods in research methodology; how to choose a sampling technique for research.* How to Choose a Sampling Technique for Research (April 10, 2016).

86. Ukrainian Law № 580-VIII ‘Population protection from noise hazards’, part of CABINET OF MINISTERS OF UKRAINE DECREE of December 4, 2019 № 1186 Kiev About closing of the Technical regulation of noise radiation in environment from the equipment that is used outdoors [online] Available at: https://zakon.rada.gov.ua/laws/main?find=2&dat=00000000&user=a&text=%D1%88%D1%83%D0%BC+&textl=1&bool=and&org=0&typ=0&datl=0&yer=0000&mon=00&day=00&numl=2&num=&minjustl=2&minjust=, Accessed: 19.02.2021

87. Ukrecoconsult (2018) *Envrionmental Assessment of Dnipro-Buzka wind station*. Available at: https://res2.weblium.site/res/5b901b36bd5979002396d07d/5c063d6d14e32a00234c3cdc, Accessed: 14.02.2021

88. Van den Berg, G. P. (2004). *Effects of the wind profile at night on wind turbine sound.* Journal of Sound and Vibration, 277, 955-970.

89. Van der Loo, G., Van Elsuwege, P., & Petrov, R. (2014*). The EU-Ukraine Association Agreement: assessment of an innovative legal instrument.* EUI Department of Law Research Paper, (2014/09).

90. Van Haaren, R., & Fthenakis, V. (2011). *GIS-based wind farm site selection using spatial multi-criteria analysis (SMCA): Evaluating the case for New York State.* Renewable and sustainable energy reviews, 15(7), 3332-3340.

91. Velychko S. (2003) *Energy of the environment of Ukraine (with electronic maps).* Teaching and methodological manual for the MSc students. Scientific editor: Prof. I.G. Chervanyov ‐ Kharkiv: Kharkiv Karazin National University. (p. 52)

92. Volkovaia O.O., Tretyakov O.S., Chervaniov I.G. (2015) *Local forest-steppe area wind potential modeling for the wind energy needs with the use of GIS technology*, Ukrainian Geographical Journal (in Ukrainian), p.10-16.

93. Wong, T. F., Wong, R. H., Chau, K. T., & Tang, C. A. (2006). *Microcrack statistics, Weibull distribution and micromechanical modeling of compressive failure in rock.* Mechanics of Materials, 38(7), 664-681.

94. World Bank, (2019), Agriculture, forestry, and fishing, value added (% of GDP), Data for Ukraine in 2019. Available at: https://data.worldbank.org/indicator/NV.AGR.TOTL.ZS, Accessed: 12.02.2021

95. World Health Organization. (2016). *Ambient air pollution: A global assessment of exposure and burden of disease.*

96. World health statistics 2020: monitoring health for the SDGs, sustainable development goals. Geneva: World Health Organization; 2020. Licence: CC BY-NC-SA 3.0 IGO.

97. Xu, Y., Li, Y., Zheng, L., Cui, L., Li, S., Li, W., & Cai, Y. (2020). *Site selection of wind farms using GIS and multi-criteria decision-making method in Wafangdian, China.* Energy, 207, 118222.

98. Yin, M., Yang, Z., Xu, Y., Liu, J., Zhou, L., & Zou, Y. (2018). *Aerodynamic optimization for variable-speed wind turbines based on wind energy capture efficiency.* Applied Energy, 221, 508-521.

99. Zelinka, M. D., Myers, T. A., McCoy, D. T., Po-Chedley, S., Caldwell, P. M., Ceppi, P., ... & Taylor, K. E. (2020). *Causes of higher climate sensitivity in CMIP6 models*. Geophysical Research Letters, 47(1), e2019GL085782.

100. Ziegler, L., Gonzalez, E., Rubert, T., Smolka, U., & Melero, J. J. (2018). *Lifetime extension of onshore wind turbines: A review covering Germany, Spain, Denmark, and the UK.* Renewable and Sustainable Energy Reviews, 82, 1261-1271.

8. Data Sources

1. CGIAR SRTM, Elevation raster file [srtm_42_03.zip], Available at: https://srtm.csi.cgiar.org/download, Accessed on 12.03.2020

2. Digital Chart of the World, Ukraine -Railways shapefile, Downloaded from aggregator website – DIVA GIS, Available at: https://www.diva-gis.org/datadown, Accessed on 12.03.2020

3. Digital Chart of the World, Ukraine -Roads shapefile, Downloaded from aggregator website – DIVA GIS, Available at: https://www.diva-gis.org/datadown, Accessed on 12.03.2020

4. Digital Chart of the World, Ukraine -Water Bodies shapefile, downloaded from aggregator website – DIVA GIS, Available at: https://www.diva-gis.org/datadown, Accessed on 12.03.2020

5. GADM, version 1.0, Ukrainian Administrative Areas shapefile, Available at: https://gadm.org/download_country_v3.html, Accessed on 12.03.2020

6. Global Wind Atlas, Wind Speed Map of Ukraine (100 height) Available at: https://globalwindatlas.info/api/gis/country/UKR/wind-speed/100, Accessed on 12.03.2020

7. Humanitarian OpenStreetMap Team (HOT), HOTOSM Ukraine (north) Airports (OpenStreetMap Export) Available at: https://data.humdata.org/dataset/hotosm_ukr_north_airports, Accessed on 12.03.2020

8. Meteoblue.com, Site 1 Historic Data (1985-2020), Downloaded from: https://www.meteoblue.com/en/weather/archive/export/49.48%C2%B0N+30.8%C2%B0E_49.476N30.799E220_Europe%2FKiev, Unique access since 20.10.2020

9. Meteoblue.com, Site 2 Historic Data (1985-2020), Downloaded from: https://www.meteoblue.com/en/weather/archive/export/49.6%C2%B0N+30.95%C2%B0E_49.603N30.946E174_Europe%2FKiev, Unique access since 20.10.2020

10. Meteoblue.com, Site 3 Historic Data (1985-2020), Downloaded from: https://www.meteoblue.com/en/weather/archive/export/49.89%C2%B0N+31.29%C2%B0E_49.886N31.294E180_Europe%2FKiev, Unique access since 20.10.2020

11. UNEP-WCMC (2021). Protected Area Profile for Ukraine from the World Database of Protected Areas, March 2021. Available at: www.protectedplanet.net

12. UNEP-WCMC and IUCN (2021), Protected Planet: The World Database on Protected Areas (WDPA) and World Database on Other Effective Area-based Conservation Measures (WD-OECM), Protected Areas in Ukraine shapefile, Available at: www.protectedplanet.net Accessed on 12.03.2020

13. Voldoire, Aurore (2019). CNRM-CERFACS CNRM-CM6-1-HR model output prepared for CMIP6 ScenarioMIP ssp245. Version 20191202. Earth System Grid Federation. https://doi.org/10.22033/ESGF/CMIP6.4190

14. Voldoire, Aurore (2019). CNRM-CERFACS CNRM-CM6-1-HR model output prepared for CMIP6 ScenarioMIP ssp585. Version 20191202. Earth System Grid Federation. https://doi.org/10.22033/ESGF/CMIP6.4225

15. Voldoire, Aurore (2020). CNRM-CERFACS CNRM-CM6-1-HR model output prepared for CMIP6 ScenarioMIP ssp126. Version 20200127. Earth System Grid Federation. https://doi.org/10.22033/ESGF/CMIP6.4185

9. Appendix

#Python Code

#importing libraries

```
import os
import urllib
import zipfile
import numpy as np
import pysal as ps
import scipy.spatial as spatial
import geopandas as gpd
from geopandas import GeoDataFrame
import pandas as pd
import shapely.geometry
from shapely.geometry import Point
import matplotlib as mpl
import matplotlib.path as path
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.simplefilter('ignore')
from sklearn.datasets import load_iris
from scipy import stats
from scipy.stats import shapiro
from scipy.stats import normaltest
from libpysal.weights.contiguity import Queen
from libpysal import examples
import numpy as np
import os
import splot
from pysal.explore import esda
```

#Extrcting data from netCDF4 is the only difference in code for analysis of wind speed projected by CMIP6 and historical trends. Thus, only climatic analysis notebook is presented here. If JupiterLab notebooks are required, please send an email to olga.kostur@gmail.com

```
ssp126 = Dataset(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\ssp126_201501-210012.nc')

ssp245 = Dataset(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\sfcWind_Amon_CNRM-CM6-1-HR_ssp245_r1i1p1f2_gr_201501-210012.nc')

ssp585 = Dataset(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\ssp585_r1i1p1f2_gr_201501-210012.nc')

print(ssp585.variables.keys())
```

```
for d in ssp585.dimensions.items():

    print(d)

Site1 = [49.18162829, 29.5]

Site2 = [50.18023903, 31.]

Site3 = [51.17884968, 31.5]

#Exracting data per site

sfcWind_ssp126 = ssp126.variables['sfcWind']
sfcWind_ssp245 = ssp245.variables['sfcWind']
sfcWind_ssp585 = ssp585.variables['sfcWind']
lon_ssp126 = ssp126.variables['lon']
lat_ssp126 = ssp126.variables['lat']
time_ssp126 = ssp126.variables['time']

lon_ssp245 = ssp245.variables['lon']
lat_ssp245 = ssp245.variables['lat']
time_ssp245 = ssp245.variables['time']

lon_ssp585 = ssp585.variables['lon']
lat_ssp585 = ssp585.variables['lat']
time_ssp585 = ssp585.variables['time']

lon_array_ssp126= lon_ssp126[:]
lat_array_ssp126 = lat_ssp126[:]
time_array_ssp126 = time_ssp126[:]

lon_array_ssp245= lon_ssp245[:]
lat_array_ssp245 = lat_ssp245[:]
time_array_ssp245 = time_ssp245[:]

lon_array_ssp585= lon_ssp585[:]
lat_array_ssp585 = lat_ssp585[:]
time_array_ssp585 = time_ssp585[:]

lon_s1_ssp126 = np.abs(lon_array_ssp126 - 30.780).argmin()
lat_s1_ssp126 = np.abs(lat_array_ssp126 - 49.47).argmin()
lon_s2_ssp126 = np.abs(lon_array_ssp126 - 30.95).argmin()
lat_s2_ssp126 = np.abs(lat_array_ssp126 - 49.6).argmin()
lon_s3_ssp126 = np.abs(lon_array_ssp126 - 31.29).argmin()
lat_s3_ssp126 = np.abs(lat_array_ssp126 - 49.89).argmin()

lon_s1_ssp245 = np.abs(lon_array_ssp245 - 30.780).argmin()
lat_s1_ssp245 = np.abs(lat_array_ssp245 - 49.47).argmin()
lon_s2_ssp245 = np.abs(lon_array_ssp245 - 30.95).argmin()
lat_s2_ssp245 = np.abs(lat_array_ssp245 - 49.6).argmin()
lon_s3_ssp245 = np.abs(lon_array_ssp245 - 31.29).argmin()
lat_s3_ssp245 = np.abs(lat_array_ssp245 - 49.89).argmin()
```

```
lon_s1_ssp585 = np.abs(lon_array_ssp585 - 30.780).argmin()
lat_s1_ssp585 = np.abs(lat_array_ssp585 - 49.47).argmin()
lon_s2_ssp585 = np.abs(lon_array_ssp585 - 30.95).argmin()
lat_s2_ssp585 = np.abs(lat_array_ssp585 - 49.6).argmin()
lon_s3_ssp585 = np.abs(lon_array_ssp585 - 31.29).argmin()
lat_s3_ssp585 = np.abs(lat_array_ssp585 - 49.89).argmin()

s1_ssp126 = sfcWind_ssp126[:,lat_s1_ssp126, lon_s1_ssp126]
s2_ssp126 = sfcWind_ssp126[:,lat_s2_ssp126, lon_s2_ssp126]
s3_ssp126 = sfcWind_ssp126[:,lat_s3_ssp126, lon_s3_ssp126]

s1_ssp245 = sfcWind_ssp245[:,lat_s1_ssp245, lon_s1_ssp245]
s2_ssp245 = sfcWind_ssp245[:,lat_s2_ssp245, lon_s2_ssp245]
s3_ssp245 = sfcWind_ssp245[:,lat_s3_ssp245, lon_s3_ssp245]

s1_ssp585 = sfcWind_ssp585[:,lat_s1_ssp585, lon_s1_ssp585]
s2_ssp585 = sfcWind_ssp585[:,lat_s2_ssp585, lon_s2_ssp585]
s3_ssp585 = sfcWind_ssp585[:,lat_s3_ssp585, lon_s3_ssp585]


d = {'Site 1 ssp126': [], 'Site 2 ssp126': [], 'Site 3 ssp126' : [],'Site 1 ssp245': [], 'Site 2 ssp245': [], 'Site 3
ssp245' : [], 'Site 1 ssp585': [], 'Site 2 ssp585': [], 'Site 3 ssp585' : []}

df = pd.DataFrame(data=d)

a = s1_ssp126.tolist()
b= s2_ssp126.tolist()
c= s3_ssp126.tolist()

d = s1_ssp245.tolist()
e = s2_ssp245.tolist()
f = s3_ssp245.tolist()

g = s1_ssp585.tolist()
j = s1_ssp585.tolist()
i = s1_ssp585.tolist()
df['Site 1 ssp126'] = a
df['Site 2 ssp126'] = b
df['Site 3 ssp126'] = c

df['Site 1 ssp245'] = d
df['Site 2 ssp245'] = e
df['Site 3 ssp245'] = f

df['Site 1 ssp585'] = g
df['Site 2 ssp585'] = j
df['Site 3 ssp585'] = i
df.head(20)
```

```
#Extrapolating SSP1-2.6 scenario data

s1_extr_100 = []
for i in df['Site 1 ssp126']:
   s1_extr_100.append(((100/10)**0.2) * i)
df['Site 1 ssp126(100m)'] = s1_extr_100

s2_extr_100 = []
for i in df['Site 2 ssp126']:
   s2_extr_100.append(((100/10)**0.2) * i)
df['Site 2 ssp126(100m)'] = s2_extr_100

s3_extr_100 = []
for i in df['Site 3 ssp126']:
   s3_extr_100.append(((100/10)**0.2) * i)
df['Site 3 ssp126(100m)'] = s3_extr_100


#Extrapolating SSP2 -4.5 scenario data

s1_extr_100 = []
for i in df['Site 1 ssp245']:
   s1_extr_100.append(((100/10)**0.2) * i)
df['Site 1 ssp245(100m)'] = s1_extr_100

s2_extr_100 = []
for i in df['Site 2 ssp245']:
   s2_extr_100.append(((100/10)**0.2) * i)
df['Site 2 ssp245(100m)'] = s2_extr_100

s3_extr_100 = []
for i in df['Site 3 ssp245']:
   s3_extr_100.append(((100/10)**0.2) * i)
df['Site 3 ssp245(100m)'] = s3_extr_100


#Extrapolating SSP5 -8.5 scenario data

s1_extr_100 = []
for i in df['Site 1 ssp585']:
   s1_extr_100.append(((100/10)**0.2) * i)
df['Site 1 ssp585(100m)'] = s1_extr_100

s2_extr_100 = []
for i in df['Site 2 ssp585']:
   s2_extr_100.append(((100/10)**0.2) * i)
df['Site 2 ssp585(100m)'] = s2_extr_100

s3_extr_100 = []
for i in df['Site 3 ssp585']:
   s3_extr_100.append(((100/10)**0.2) * i)
df['Site 3 ssp585(100m)'] = s3_extr_100
```

#Adding Years and Months columns

```
import itertools
lst = range(2015,2101)
y = list(itertools.chain.from_iterable(itertools.repeat(x, 12) for x in lst))
df['Year'] = y

df.to_csv(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\Add
months_correctData.csv', index = False)
ddf =
pd.read_csv(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\Correct_climateData_with
yearsandmonths.csv', low_memory=False)
```

#No differences between and within sites at all 3 scenarios, based on whole data record

```
a = ddf['Site 3 ssp126(100m)']
b = ddf['Site 3 ssp245(100m)']
c = ddf['Site 3 ssp585(100m)']

print(stats.kruskal(a, b, c))

a = ddf['Site 1 ssp126(100m)']
b = ddf['Site 2 ssp126(100m)']
c = ddf['Site 3 ssp126(100m)']

print(stats.kruskal(a, b, c))

a = ddf['Site 1 ssp245(100m)']
b = ddf['Site 2 ssp245(100m)']
c = ddf['Site 3 ssp245(100m)']

print(stats.kruskal(a, b, c))

a = ddf['Site 1 ssp585(100m)']
b = ddf['Site 2 ssp585(100m)']
c = ddf['Site 3 ssp585(100m)']

print(stats.kruskal(a, b, c))

#Plotting trend graphs. Only change of climatic scenario (126 to 245 or 585) is require to generate
remaining figures

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=ddf, x='Year', y= 'Site 1 ssp126(100m)', color = 'maroon', label = 'Site 1', estimator=
'mean')
ax = sns.lineplot(data=ddf, x='Year', y= 'Site 2 ssp126(100m)', color = 'gold', label = 'Site 2', estimator=
'mean')
ax = sns.lineplot(data=ddf, x='Year', y= 'Site 3 ssp126(100m)', color = 'forestgreen', label = 'Site 3',
estimator= 'mean')
```

```
ax.axes.set_title(label = "SSP 1-2.6 projections of wind speed at 3 sites(2021-2080)", fontdict={'fontsize':
'20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Year",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Trend graph of ssp126 (100).png", format="png")
plt.show()

#Plotting boxplot. Only change of climatic scenario (126 to 245 or 585) is require to generate remaining
figures

boxplot_data = ddf[['Site 1 ssp126(100m)_exp', 'Site 2 ssp126(100m)_exp','Site 3 ssp126(100m)_exp']]
data=pd.melt(boxplot_data)
data.head()
my_pal = {'Site 1 ssp126(100m)_exp': 'maroon', 'Site 2 ssp126(100m)_exp' : 'gold', 'Site 3
ssp126(100m)_exp': 'forestgreen'}
fig = plt.figure(figsize=(10, 8))
ax = sns.set_style("whitegrid")
ax = sns.boxplot(x='variable', y='value', data = data,  palette=my_pal)
ax.axes.set_title(label = "Distribution of wnd speed at 3 sites at SSP1-2.6 climatic projections",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Site",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Boxplots SSP126 (100).png", format="png")
plt.show()


#Creating data set with yearly means to produce descriptive statistics table

index = range(2015, 2101)

columns = ['Site 1 126 yearly mean', 'Site 2 126 yearly mean', 'Site 3 126 yearly mean', 'Site 1 245 yearly
mean', 'Site 2 245 yearly mean', 'Site 3 245 yearly mean', 'Site 1 585 yearly mean', 'Site 2 585 yearly mean',
'Site 3 585 yearly mean']
yearly_means = pd.DataFrame(index=index, columns=columns)
site1_ssp126_yearly_means = []
site2_ssp126_yearly_means = []
site3_ssp126_yearly_means = []

site1_ssp245_yearly_means = []
site2_ssp245_yearly_means = []
site3_ssp245_yearly_means = []

site1_ssp585_yearly_means = []
site2_ssp585_yearly_means = []
site3_ssp585_yearly_means = []

for i in range(2015, 2101):
  site1_ssp126_yearly_means.append(ddf[ddf['Year'] == i]['Site 1 ssp126(100m)'].mean())
  site2_ssp126_yearly_means.append(ddf[ddf['Year'] == i]['Site 2 ssp126(100m)'].mean())
  site3_ssp126_yearly_means.append(ddf[ddf['Year'] == i]['Site 3 ssp126(100m)'].mean())
  site1_ssp245_yearly_means.append(ddf[ddf['Year'] == i]['Site 1 ssp245(100m)'].mean())
  site2_ssp245_yearly_means.append(ddf[ddf['Year'] == i]['Site 2 ssp245(100m)'].mean())
```

```
    site3_ssp245_yearly_means.append(ddf[ddf['Year'] == i]['Site 3 ssp245(100m)'].mean())
    site1_ssp585_yearly_means.append(ddf[ddf['Year'] == i]['Site 1 ssp585(100m)'].mean())
    site2_ssp585_yearly_means.append(ddf[ddf['Year'] == i]['Site 2 ssp585(100m)'].mean())
    site3_ssp585_yearly_means.append(ddf[ddf['Year'] == i]['Site 3 ssp585(100m)'].mean())

yearly_means['Site 1 126 yearly mean'] = site1_ssp126_yearly_means
yearly_means['Site 2 126 yearly mean'] = site2_ssp126_yearly_means
yearly_means['Site 3 126 yearly mean'] = site3_ssp126_yearly_means
yearly_means['Site 1 245 yearly mean'] = site1_ssp245_yearly_means
yearly_means['Site 2 245 yearly mean'] = site2_ssp245_yearly_means
yearly_means['Site 3 245 yearly mean'] = site3_ssp245_yearly_means
yearly_means['Site 1 585 yearly mean'] = site1_ssp585_yearly_means
yearly_means['Site 2 585 yearly mean'] = site2_ssp585_yearly_means
yearly_means['Site 3 585 yearly mean'] = site3_ssp585_yearly_means

import itertools
lst = range(2015, 2101)
yearly_means['Year'] = lst

yearly_means.to_csv(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\Add_rateCORREC
TDATA.csv', index = False)


#Generating descriptive statistics of annual fluctuations

rate =
pd.read_csv(r'C:\Users\Olga\Desktop\3rdyear\Dissertation\ClimateModelsData\Add_rateCORRECTDATA.cs
v', low_memory=False)
print(rate['CR Site 1 126 yearly mean'].describe())
print(rate['CR Site 3 126 yearly mean'].describe())

print(rate['CR Site 1 245 yearly mean'].describe())
print(rate['CR Site 3 245 yearly mean'].describe())

print(rate['CR Site 3 585 yearly mean'].describe())


#Adding wind speed cubbed columns for weibull distribution

for i in ddf[['Site 1 ssp585(100m)',
       'Site 2 ssp585(100m)', 'Site 3 ssp585(100m)', 'Site 1 ssp126(100m)',
       'Site 2 ssp126(100m)', 'Site 3 ssp126(100m)', 'Site 1 ssp245(100m)',
       'Site 2 ssp245(100m)', 'Site 3 ssp245(100m)']]:
    ddf[i + 'cubbed'] = ddf[i] ** 3


#Creating a dataframe for weibull distribution

dfw = pd.DataFrame( columns=["Site1 f(v) ssp126", "Site2 f(v) ssp126", "Site3 f(v) ssp126", "Site1 F(v)
ssp126", "Site2 F(v) ssp126", "Site3 F(v) ssp126" ])
dfw['Speed'] = range(0, 25)
```

```
Esite1_ssp126 = ddf['Site 1 ssp126(100m)cubbed'].mean()/(ddf['Site 1 ssp126(100m)'].mean()**3)
Esite2_ssp126 = ddf['Site 2 ssp126(100m)cubbed'].mean()/(ddf['Site 2 ssp126(100m)'].mean()**3)
Esite3_ssp126 = ddf['Site 3 ssp126(100m)cubbed'].mean()/(ddf['Site 3 ssp126(100m)'].mean()**3)

k_site1_ssp126 = 1+ (3.69/(Esite1_ssp126**2))
k_site2_ssp126 = 1+ (3.69/(Esite2_ssp126**2))
k_site3_ssp126 = 1+ (3.69/(Esite3_ssp126**2))


c_site1_ssp126 = ddf['Site 1 ssp126(100m)'].mean()/math.gamma(1+1/k_site1_ssp126)
c_site2_ssp126 = ddf['Site 2 ssp126(100m)'].mean()/math.gamma(1+1/k_site2_ssp126)
c_site3_ssp126 = ddf['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_ssp126)


k = k_site1_ssp126
c = c_site1_ssp126
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site1 f(v) ssp126"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site1 F(v) ssp126"] = result2


k = k_site2_ssp126
c = c_site2_ssp126
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site2 f(v) ssp126"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site2 F(v) ssp126"] = result2


k = k_site3_ssp126
c = c_site3_ssp126
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site3 f(v) ssp126"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site3 F(v) ssp126"] = result2
```

```
#Calculting Parameters
Esite1_ssp245 = ddf['Site 1 ssp245(100m)cubbed'].mean()/(ddf['Site 1 ssp245(100m)'].mean()**3)
Esite2_ssp245 = ddf['Site 2 ssp245(100m)cubbed'].mean()/(ddf['Site 2 ssp245(100m)'].mean()**3)
Esite3_ssp245 = ddf['Site 3 ssp245(100m)cubbed'].mean()/(ddf['Site 3 ssp245(100m)'].mean()**3)

k_site1_ssp245 = 1+ (3.69/(Esite1_ssp245**2))
k_site2_ssp245 = 1+ (3.69/(Esite2_ssp245**2))
k_site3_ssp245 = 1+ (3.69/(Esite3_ssp245**2))


c_site1_ssp245 = ddf['Site 1 ssp245(100m)'].mean()/math.gamma(1+1/k_site1_ssp245)
c_site2_ssp245 = ddf['Site 2 ssp245(100m)'].mean()/math.gamma(1+1/k_site2_ssp245)
c_site3_ssp245 = ddf['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_ssp245)


k = k_site1_ssp245
c = c_site1_ssp245
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site1 f(v) ssp245"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site1 F(v) ssp245"] = result2


k = k_site2_ssp245
c = c_site2_ssp245
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site2 f(v) ssp245"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site2 F(v) ssp245"] = result2


k = k_site3_ssp245
c = c_site3_ssp245
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site3 f(v) ssp245"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site3 F(v) ssp245"] = result2


Esite1_ssp585 = ddf['Site 1 ssp585(100m)cubbed'].mean()/(ddf['Site 1 ssp585(100m)'].mean()**3)
Esite2_ssp585 = ddf['Site 2 ssp585(100m)cubbed'].mean()/(ddf['Site 2 ssp585(100m)'].mean()**3)
```

```
Esite3_ssp585 = ddf['Site 3 ssp585(100m)cubbed'].mean()/(ddf['Site 3 ssp585(100m)'].mean()**3)

k_site1_ssp585 = 1+ (3.69/(Esite1_ssp585**2))
k_site2_ssp585 = 1+ (3.69/(Esite2_ssp585**2))
k_site3_ssp585 = 1+ (3.69/(Esite3_ssp585**2))


c_site1_ssp585 = ddf['Site 1 ssp585(100m)'].mean()/math.gamma(1+1/k_site1_ssp585)
c_site2_ssp585 = ddf['Site 2 ssp585(100m)'].mean()/math.gamma(1+1/k_site2_ssp585)
c_site3_ssp585 = ddf['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_ssp585)


k = k_site1_ssp585
c = c_site1_ssp585
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site1 f(v) ssp585"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site1 F(v) ssp585"] = result2


k = k_site2_ssp585
c = c_site2_ssp585
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site2 f(v) ssp585"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site2 F(v) ssp585"] = result2


k = k_site3_ssp585
c = c_site3_ssp585
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
dfw["Site3 f(v) ssp585"] = result1
result2 = []
for i in range (0, 25):
   result2.append(1- math.exp(-(i/c)**k))
dfw["Site3 F(v) ssp585"] = result2

#Plotting Distributions

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site1 f(v) ssp126', color = 'maroon')
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site2 f(v) ssp126', color = 'gold')
```

```
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site3 f(v) ssp126', color = 'forestgreen')
ax2 = ax.twinx()
sns.lineplot(data=dfw, x='Speed', y= 'Site1 F(v) ssp126', color = 'maroon', label = 'Site 1', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site2 F(v) ssp126', color = 'gold', label = 'Site 2', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site3 F(v) ssp126', color = 'forestgreen', label = 'Site 3', ax = ax2)
ax.axes.set_title(label = "Weibull distributions of Wind Speed at 3 sites at SSP1-2.6 scenario ",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability Density f(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax2.set_ylabel("Cummulative Probability Density F(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family':
'Calibri'} )
plt.savefig("Cummulative & Weibull overall (SSP126).png", format="png")
plt.show()

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site1 f(v) ssp245', color = 'maroon')
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site2 f(v) ssp245', color = 'gold')
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site3 f(v) ssp245', color = 'forestgreen')
ax2 = ax.twinx()
sns.lineplot(data=dfw, x='Speed', y= 'Site1 F(v) ssp245', color = 'maroon', label = 'Site 1', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site2 F(v) ssp245', color = 'gold', label = 'Site 2', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site3 F(v) ssp245', color = 'forestgreen', label = 'Site 3', ax = ax2)
ax.axes.set_title(label = "Weibull distributions of Wind Speed at 3 sites at SSP2-4.5 scenario ",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability Density f(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax2.set_ylabel("Cummulative Probability Density F(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family':
'Calibri'} )
plt.savefig("Cummulative & Weibull overall (SSP245).png", format="png")
plt.show()

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site1 f(v) ssp585', color = 'maroon')
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site2 f(v) ssp585', color = 'gold')
ax = sns.lineplot(data=dfw, x='Speed', y= 'Site3 f(v) ssp585', color = 'forestgreen')
ax2 = ax.twinx()
sns.lineplot(data=dfw, x='Speed', y= 'Site1 F(v) ssp585', color = 'maroon', label = 'Site 1', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site2 F(v) ssp585', color = 'gold', label = 'Site 2', ax = ax2)
sns.lineplot(data=dfw, x='Speed', y= 'Site3 F(v) ssp585', color = 'forestgreen', label = 'Site 3', ax = ax2)
ax.axes.set_title(label = "Weibull distributions of Wind Speed at 3 sites at SSP5-8.5 scenario ",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability Density f(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax2.set_ylabel("Cummulative Probability Density F(v)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family':
'Calibri'} )
plt.savefig("Cummulative & Weibull overall (SSP585).png", format="png")
plt.show()


#Calculating Power output
```

```
power1 = []
for i in range (0,25):
   power1.append(0.5 * 1.225 * (i**3) * dfw['Site1 f(v) ssp126'][i] )
print(sum(power1))

power2 = []
for i in range (0, 25):
   power2.append(0.5 * 1.225 * (i**3)  * dfw['Site3 f(v) ssp126'][i])
print(sum(power2[4:]))

power3 = []
for i in range (0, 25):
   power3.append(0.5 * 1.225 * (i**3) * dfw['Site1 f(v) ssp245'][i])
print(sum(power3[4:]))

power4 = []
for i in range (0, 25):
   power4.append(0.5 * 1.225 * (i**3) * dfw['Site3 f(v) ssp245'][i] )
print(sum(power4[4:]))

power5 = []
for i in range (0, 25):
   power5.append(0.5 * 1.225 * (i**3)  * dfw['Site3 f(v) ssp585'][i])
print(sum(power5[4:]))

#Comparing monthly wind speeds for all sites and all scenarios

d = {'kw monthly between sites': [0.37, 0.26, 0.0, 0.42, 0.85, 0.98, 0.53, 0.01, 0.23, 0.66, 0.13 ], 'Month' :
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], 'p value' : [0.05,
0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05,0.05 ]}
seasonality = pd.DataFrame(data=d

a = ddf[ddf['Months'] == 'Dec']['Site 1 ssp126(100m)']
b = ddf[ddf['Months'] == 'Dec']['Site 3 ssp126(100m)']
c = ddf[ddf['Months'] == 'Dec']['Site 1 ssp245(100m)']
d = ddf[ddf['Months'] == 'Dec']['Site 3 ssp245(100m)']
e = ddf[ddf['Months'] == 'Dec']['Site 2 ssp585(100m)']

print(stats.kruskal(a, b, c, d, e))

a = ddf[ddf['Months'] == 'Nov']['Site 1 ssp126(100m)']
b = ddf[ddf['Months'] == 'Nov']['Site 3 ssp126(100m)']
c = ddf[ddf['Months'] == 'Nov']['Site 1 ssp245(100m)']
d = ddf[ddf['Months'] == 'Nov']['Site 3 ssp245(100m)']
e = ddf[ddf['Months'] == 'Nov']['Site 2 ssp585(100m)']

data = [a, b, c, d, e]

#perform Dunn's test using a Bonferonni correction for the p-values
import scikit_posthocs as sp
sp.posthoc_dunn(data, p_adjust = 'bonferroni')
```

```
# Records of p-values. This cell shou only be executed when all values are recoded

seasonality['Site1&2(ssp126) & Site3(ssp126)'] = [1, 1, 1, 1, 1, 1, 1, 1,1, 1, 1,   1]
seasonality['Site1&2(ssp126) & Site1&2(ssp245)'] = [1, 1,  0.02, 1, 1,1, 1, 0.13, 1, 1, 1, 1, ]
seasonality['Site3(ssp126) & Site1&2(ssp245)'] = [1, 1, 0.65, 0.53, 1, 1,1, 0.80, 1, 1,  1, 0.14 ]
seasonality['Site1&2(ssp126) & Site3 (ssp245)'] = [1, 0.25, 0.000265,1, 1, 1,1, 0.009, 0.19, 1,   1,1 ]
seasonality['Site3(ssp126) & Site3 (ssp245)'] = [1,  1,  0.025029, 1, 1, 1,  1, 0.102, 1, 1,   1, 1]
seasonality['Site1&2(ssp245) & Site3 (ssp245)'] = [1, 1, 1, 1, 1,  1, 1, 1, 1, 1,   1,  1 ]
seasonality['Site1&2(ssp126) & Site1,2&3 (ssp585)'] = [1, 1, 0.16, 1, 1, 1, 1, 1, 1, 1, 1,  1 ]
seasonality['Site3(ssp126) & Site1,2&3 (ssp585)'] = [1, 1,  1, 1, 1,  1, 0.95, 1, 1, 1, 0.19, 0.52]
seasonality['Site1 &2(ssp245) & Site1,2&3 (ssp585)'] = [1, 1, 1, 1, 1,1, 1, 1, 1, 1,  1, 1 ]
seasonality['Site3(ssp245) & Site1,2&3 (ssp585)'] = [1, 1, 0.73, 1, 1, 1, 1, 0.67, 1, 1, 0.25, 1 ]
seasonality['p value'] = [0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05]

fig = plt.figure(figsize=(16, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1&2(ssp126) & Site3(ssp126)', label =
'Site1&2(ssp126) & Site3(ssp126)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1&2(ssp126) & Site1&2(ssp245)', label =
'Site1&2(ssp126) & Site1&2(ssp245)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site3(ssp126) & Site1&2(ssp245)', label = 'Site3(ssp126) &
Site1&2(ssp245)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1&2(ssp126) & Site3 (ssp245)', label =
'Site1&2(ssp126) & Site3 (ssp245)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site3(ssp126) & Site3 (ssp245)', label = 'Site3(ssp126) &
Site3 (ssp245)' )
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1&2(ssp245) & Site3 (ssp245)', label =
'Site1&2(ssp245) & Site3 (ssp245)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1&2(ssp126) & Site1,2&3 (ssp585)', label =
'Site1&2(ssp126) & Site1,2&3 (ssp585)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site3(ssp126) & Site1,2&3 (ssp585)', label = 'Site3(ssp126)
& Site1,2&3 (ssp585)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site1 &2(ssp245) & Site1,2&3 (ssp585)', label = 'Site1
&2(ssp245) & Site1,2&3 (ssp585)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'Site3(ssp245) & Site1,2&3 (ssp585)', label = 'Site3(ssp245)
& Site1,2&3 (ssp585)')
ax = sns.lineplot(data=seasonality, x='Month', y= 'p value', color = 'black', label = 'Significance level - 0.05')
ax.lines[0].set_linestyle("--")
ax.lines[1].set_linestyle("--")
ax.lines[2].set_linestyle("--")
ax.lines[3].set_linestyle("--")
ax.lines[4].set_linestyle("--")
ax.lines[5].set_linestyle("--")
ax.lines[6].set_linestyle("--")
ax.lines[7].set_linestyle("--")
ax.lines[8].set_linestyle("--")
ax.lines[9].set_linestyle("--")

leg = ax.legend()
leg_lines = leg.get_lines()
leg_lines[2].set_linestyle("--")
```

```
ax.axes.set_title(label = "P-values of Dunn-Bonferroni test comaring each variale pair's for monthly
differences", fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Months",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("p-values",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )

plt.savefig("Dunn-Bonferroni test.png", format="png")
plt.show()

#Generting descriptive statistics

print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp126(100m)'].mean())
print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp126(100m)'].std())
print(ddf[ddf['Months'] == 'Dec']['Site 3 ssp126(100m)'].mean())
print(ddf[ddf['Months'] == 'Dec']['Site 3 ssp126(100m)'].std())
print('SSP245')
print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp245(100m)'].mean())
print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp245(100m)'].std())
print(ddf[ddf['Months'] == 'Dec']['Site 3 ssp245(100m)'].mean())
print(ddf[ddf['Months'] == 'Dec']['Site 3 ssp245(100m)'].std())
print('SSP585')
print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp585(100m)'].mean())
print(ddf[ddf['Months'] == 'Dec']['Site 1 ssp585(100m)'].std())

import numpy as np
import matplotlib.pyplot as plt

# set width of bars
barWidth = 0.15

# set heights of bars
bars1 = seasonality['Site 1&2 monthly means(ssp126)']
bars2 = seasonality['Site 3 monthly means(ssp126)']
bars3 = seasonality['Site 1&2 monthly means(ssp245)']
bars4 = seasonality['Site 3 monthly means(ssp245)']
bars5 = seasonality['Site 1,2 &3 monthly means(ssp585)']


# Set position of bar on X axis
r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]
r4 = [x + barWidth for x in r3]
r5 = [x + barWidth for x in r4]

# Make the plot
fig = plt.figure(figsize=(17, 13))
sns.set_style("whitegrid")
ax = plt.bar(r1, bars1, color='maroon', width=barWidth, edgecolor='white', label='Site 1&2(ssp126)', alpha
= 0.3)
ax = plt.bar(r2, bars2, color='forestgreen', width=barWidth, edgecolor='white', label='Site 3(ssp126)', alpha
= 0.3)
ax = plt.bar(r3, bars3, color='gold', width=barWidth, edgecolor='white', label='Site 1&2(ssp245)', alpha =
0.3)
```

```
ax = plt.bar(r4, bars4, color='red', width=barWidth, edgecolor='white', label='Site 3(ssp245)', alpha = 0.3)
ax = plt.bar(r5, bars5, color='blue', width=barWidth, edgecolor='white', label='Site 1,2,3(ssp585)', alpha =
0.3)



axes2 = plt.twinx()
axes2.plot(seasonality['Month'], seasonality['Site 1&2 monthly std(ssp126)'], color = 'maroon', label='Site
1&2(ssp126)')
axes2.plot(  seasonality['Month'], seasonality['Site 3 monthly std(ssp126)'], color = 'forestgreen', label='Site
3(ssp126)')
axes2.plot( seasonality['Month'], seasonality['Site 1&2 monthly std(ssp245)'], color = 'gold', label='Site
1&2(ssp245)')
axes2.plot(  seasonality['Month'], seasonality['Site 3 monthly std(ssp245)'], color = 'red', label='Site
3(ssp245)')
axes2.plot( seasonality['Month'], seasonality['Site 1,2 &3 monthly std(ssp585)'], color = 'blue', label='Site
1,2 & 3(ssp585)')
axes2.legend(loc='upper center')

plt.title(label = "Monthly means and standard deviation variation at 3 SSPs", fontdict={'fontsize': '20',
'fontweight' : '3', 'family': 'Calibri'} )
axes2.set_ylabel("Standard Deviation",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly mean and standard deviation per scenario.png", format="png")
plt.show()


#Calculation parameters for monthly Weibull distributions at site 1&2 and 3 at SSP 1-2.6


Esite1_2_jan = ddf[ddf['Months'] == 'Jan']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jan']['Site 1 ssp126(100m)'].mean()**3)
Esite3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jan']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_jan = 1+ (3.69/(Esite1_2_jan**2))
k_site3_jan = 1+ (3.69/(Esite3_jan**2))
c_site1_2_jan = ddf[ddf['Months'] == 'Jan']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_jan)
c_site3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_jan)

Esite1_2_feb = ddf[ddf['Months'] == 'Feb']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Feb']['Site 1 ssp126(100m)'].mean()**3)
Esite3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Feb']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_feb = 1+ (3.69/(Esite1_2_feb**2))
k_site3_feb = 1+ (3.69/(Esite3_feb**2))
c_site1_2_feb = ddf[ddf['Months'] == 'Feb']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_feb)
c_site3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_feb)

Esite1_2_mar = ddf[ddf['Months'] == 'Mar']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Mar']['Site 1 ssp126(100m)'].mean()**3)
```

```
Esite3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Mar']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_mar = 1+ (3.69/(Esite1_2_mar**2))
k_site3_mar = 1+ (3.69/(Esite3_mar**2))
c_site1_2_mar = ddf[ddf['Months'] == 'Mar']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_mar)
c_site3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_mar)

Esite1_2_apr = ddf[ddf['Months'] == 'Apr']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Apr']['Site 1 ssp126(100m)'].mean()**3)
Esite3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Apr']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_apr = 1+ (3.69/(Esite1_2_apr**2))
k_site3_apr = 1+ (3.69/(Esite3_apr**2))
c_site1_2_apr = ddf[ddf['Months'] == 'Apr']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_apr)
c_site3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_apr)

Esite1_2_may = ddf[ddf['Months'] == 'May']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'May']['Site 1 ssp126(100m)'].mean()**3)
Esite3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'May']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_may = 1+ (3.69/(Esite1_2_may**2))
k_site3_may = 1+ (3.69/(Esite3_may**2))
c_site1_2_may = ddf[ddf['Months'] == 'May']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_may)
c_site3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_may)

Esite1_2_jun = ddf[ddf['Months'] == 'Jun']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jun']['Site 1 ssp126(100m)'].mean()**3)
Esite3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jun']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_jun = 1+ (3.69/(Esite1_2_jun**2))
k_site3_jun = 1+ (3.69/(Esite3_jun**2))
c_site1_2_jun = ddf[ddf['Months'] == 'Jun']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_jun)
c_site3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_jun)

Esite1_2_jul = ddf[ddf['Months'] == 'Jul']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jul']['Site 1 ssp126(100m)'].mean()**3)
Esite3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jul']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_jul = 1+ (3.69/(Esite1_2_jul**2))
k_site3_jul = 1+ (3.69/(Esite3_jul**2))
c_site1_2_jul = ddf[ddf['Months'] == 'Jul']['Site 1 ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_jul)
c_site3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_jul)

Esite1_2_aug = ddf[ddf['Months'] == 'Aug']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Aug']['Site 1 ssp126(100m)'].mean()**3)
Esite3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Aug']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_aug = 1+ (3.69/(Esite1_2_aug**2))
k_site3_aug = 1+ (3.69/(Esite3_aug**2))
```

```
c_site1_2_aug = ddf[ddf['Months'] == 'Aug']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_aug)
c_site3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_aug)

Esite1_2_sep = ddf[ddf['Months'] == 'Sep']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Sep']['Site 1 ssp126(100m)'].mean()**3)
Esite3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Sep']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_sep = 1+ (3.69/(Esite1_2_sep**2))
k_site3_sep = 1+ (3.69/(Esite3_sep**2))
c_site1_2_sep = ddf[ddf['Months'] == 'Sep']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_sep)
c_site3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_sep)

Esite1_2_oct = ddf[ddf['Months'] == 'Oct']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Oct']['Site 1 ssp126(100m)'].mean()**3)
Esite3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Oct']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_oct = 1+ (3.69/(Esite1_2_oct**2))
k_site3_oct = 1+ (3.69/(Esite3_oct**2))
c_site1_2_oct = ddf[ddf['Months'] == 'Oct']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_oct)
c_site3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_oct)

Esite1_2_nov = ddf[ddf['Months'] == 'Nov']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Nov']['Site 1 ssp126(100m)'].mean()**3)
Esite3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Nov']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_nov = 1+ (3.69/(Esite1_2_nov**2))
k_site3_nov = 1+ (3.69/(Esite3_nov**2))
c_site1_2_nov = ddf[ddf['Months'] == 'Nov']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_nov)
c_site3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_nov)

Esite1_2_dec = ddf[ddf['Months'] == 'Dec']['Site 1 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Dec']['Site 1 ssp126(100m)'].mean()**3)
Esite3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp126(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Dec']['Site 3 ssp126(100m)'].mean()**3)
k_site1_2_dec = 1+ (3.69/(Esite1_2_dec**2))
k_site3_dec = 1+ (3.69/(Esite3_dec**2))
c_site1_2_dec = ddf[ddf['Months'] == 'Dec']['Site 1
ssp126(100m)'].mean()/math.gamma(1+1/k_site1_2_dec)
c_site3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp126(100m)'].mean()/math.gamma(1+1/k_site3_dec)

df_season = pd.DataFrame( columns=["Site1_2 f(v) - January(ssp126)", "Site1_2 f(v) - February(ssp126)",
"Site1_2 f(v) - March(ssp126)", "Site1_2 f(v) - April(ssp126)", "Site1_2 f(v) - May(ssp126)",
                          "Site1_2 f(v) - June(ssp126)", "Site1_2 f(v) - July(ssp126)", "Site1_2 f(v) -
August(ssp126)", "Site1_2 f(v) - September(ssp126)", "Site1_2 f(v) - October(ssp126)",
                          "Site1_2 f(v) - November(ssp126)", "Site1_2 f(v) - December(ssp126)", "Site3 f(v) -
January(ssp126)", "Site3 f(v) - February(ssp126)", "Site3 f(v) - March(ssp126)", "Site3 f(v) - April(ssp126)",
"Site3 f(v) - May(ssp126)",
                          "Site3 f(v) - June(ssp126)", "Site3 f(v) - July(ssp126)", "Site3 f(v) - August(ssp126)", "Site3
f(v) - September(ssp126)", "Site3 f(v) - October(ssp126)",
```

```
                    "Site3 f(v) - November(ssp126)", "Site3 f(v) - December(ssp126)"])
df_season['Speed'] = range(0, 25)


#Constructing Weibull probability columns for site 1&2 at SSP 1-2.6 scenario

k = k_site1_2_jan
c = c_site1_2_jan
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - January(ssp126)"] = result1

k = k_site1_2_feb
c = c_site1_2_feb
result2 = []
for i in range (0, 25):
   result2.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - February(ssp126)"] = result2

k = k_site1_2_mar
c = c_site1_2_mar
result3 = []
for i in range (0, 25):
   result3.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - March(ssp126)"] = result3

k = k_site1_2_apr
c = c_site1_2_apr
result4 = []
for i in range (0, 25):
   result4.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - April(ssp126)"] = result4

k = k_site1_2_may
c = c_site1_2_may
result5 = []
for i in range (0, 25):
   result5.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - May(ssp126)"] = result5

k = k_site1_2_jun
c = c_site1_2_jun
result6 = []
for i in range (0, 25):
   result6.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - June(ssp126)"] = result6

k = k_site1_2_jul
c = c_site1_2_jul
result7 = []
for i in range (0, 25):
```

```
    result7.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - July(ssp126)"] = result7

k = k_site1_2_aug
c = c_site1_2_aug
result8 = []
for i in range (0, 25):
    result8.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - August(ssp126)"] = result8

k = k_site1_2_sep
c = c_site1_2_sep
result9 = []
for i in range (0, 25):
    result9.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - September(ssp126)"] = result9

k = k_site1_2_oct
c = c_site1_2_oct
result10 = []
for i in range (0, 25):
    result10.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - October(ssp126)"] = result10

k = k_site1_2_nov
c = c_site1_2_nov
result11 = []
for i in range (0, 25):
    result11.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - November(ssp126)"] = result11

k = k_site1_2_dec
c = c_site1_2_dec
result12 = []
for i in range (0, 25):
    result12.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - December(ssp126)"] = result12


#Plotting Weibull distribution of wind speed at site 1&2 for SSP 1-2.6 scenario

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - January(ssp126)', label = 'January')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - February(ssp126)', label = 'February')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - March(ssp126)', label = 'March')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - April(ssp126)', label = 'April')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - May(ssp126)',  label = 'May')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - June(ssp126)',  label = 'June')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - July(ssp126)', label = 'July')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - August(ssp126)', label = 'August')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - September(ssp126)',  label = 'September')
```

```
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - October(ssp126)',  label = 'October')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - November(ssp126)',  label = 'November', color = 'black')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - December(ssp126)',  label = 'December', color = 'aqua')

ax.axes.set_title(label = "Monthly Weibull distribution at Site 1&2 at SSP1-2.6 scenario",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability of occurrence",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly weibull at Site 1&2(ssp126.png", format="png")
plt.show()

#Calculating operational ranges

print(sum(df_season['Site1_2 f(v) - June(ssp126)'][5:]))
print(sum(df_season['Site1_2 f(v) - July(ssp126)'][5:]))
print(sum(df_season['Site1_2 f(v) - August(ssp126)'][5:]))


#Constructing Weibull probability columns for site 3 at SSP 1-2.6 scenario


k = k_site3_jan
c = c_site3_jan
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - January(ssp126)"] = result1

k = k_site3_feb
c = c_site3_feb
result2 = []
for i in range (0, 25):
   result2.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - February(ssp126)"] = result2

k = k_site3_mar
c = c_site3_mar
result3 = []
for i in range (0, 25):
   result3.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - March(ssp126)"] = result3

k = k_site3_apr
c = c_site3_apr
result4 = []
for i in range (0, 25):
   result4.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - April(ssp126)"] = result4

k = k_site3_may
```

```
c = c_site3_may
result5 = []
for i in range (0, 25):
   result5.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - May(ssp126)"] = result5

k = k_site3_jun
c = c_site3_jun
result6 = []
for i in range (0, 25):
   result6.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - June(ssp126)"] = result6

k = k_site3_jul
c = c_site3_jul
result7 = []
for i in range (0, 25):
   result7.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - July(ssp126)"] = result7

k = k_site3_aug
c = c_site3_aug
result8 = []
for i in range (0, 25):
   result8.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - August(ssp126)"] = result8

k = k_site3_sep
c = c_site3_sep
result9 = []
for i in range (0, 25):
   result9.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - September(ssp126)"] = result9

k = k_site3_oct
c = c_site3_oct
result10 = []
for i in range (0, 25):
   result10.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - October(ssp126)"] = result10

k = k_site3_nov
c = c_site3_nov
result11 = []
for i in range (0, 25):
   result11.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - November(ssp126)"] = result11

k = k_site3_dec
c = c_site3_dec
result12 = []
for i in range (0, 25):
   result12.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
```

```
df_season["Site3 f(v) - December(ssp126)"] = result12


#Plotting Weibull distribution of wind speed at site 3 for SSP 1-2.6 scenario

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - January(ssp126)', label = 'January')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - February(ssp126)', label = 'February')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - March(ssp126)', label = 'March')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - April(ssp126)', label = 'April')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - May(ssp126)',  label = 'May')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - June(ssp126)',  label = 'June')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - July(ssp126)', label = 'July')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - August(ssp126)', label = 'August')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - September(ssp126)',  label = 'September')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - October(ssp126)',  label = 'October')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - November(ssp126)',  label = 'November', color =
'black')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - December(ssp126)',  label = 'December', color =
'aqua')

ax.axes.set_title(label = "Monthly Weibull distribution at Site 3 at SSP1-2.6 scenario", fontdict={'fontsize':
'20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability of occurrence",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly weibull at Site 3(ssp126.png", format="png")
plt.show()

#Constructing Average probability values for ssp126
df_season["Average f(v) - January(ssp126)"] = (df_season['Site1_2 f(v) - January(ssp126)'] +
df_season['Site3 f(v) - January(ssp126)'] )/2
df_season["Average f(v) - February(ssp126)"] = (df_season['Site1_2 f(v) - February(ssp126)'] +
df_season['Site3 f(v) - February(ssp126)'] )/2
df_season["Average f(v) - March(ssp126)"] = (df_season['Site1_2 f(v) - March(ssp126)'] + df_season['Site3
f(v) - March(ssp126)'] )/2
df_season["Average f(v) - April(ssp126)"] = (df_season['Site1_2 f(v) - April(ssp126)'] + df_season['Site3 f(v) -
April(ssp126)'] )/2
df_season["Average f(v) - May(ssp126)"] = (df_season['Site1_2 f(v) - May(ssp126)'] + df_season['Site3 f(v) -
May(ssp126)'] )/2
df_season["Average f(v) - June(ssp126)"] = (df_season['Site1_2 f(v) - June(ssp126)'] + df_season['Site3 f(v) -
June(ssp126)'] )/2
df_season["Average f(v) - July(ssp126)"] = (df_season['Site1_2 f(v) - July(ssp126)'] + df_season['Site3 f(v) -
July(ssp126)'] )/2
df_season["Average f(v) - August(ssp126)"] = (df_season['Site1_2 f(v) - August(ssp126)'] + df_season['Site3
f(v) - August(ssp126)'] )/2
df_season["Average f(v) - September(ssp126)"] = (df_season['Site1_2 f(v) - September(ssp126)'] +
df_season['Site3 f(v) - September(ssp126)'] )/2
df_season["Average f(v) - October(ssp126)"] = (df_season['Site1_2 f(v) - October(ssp126)'] +
df_season['Site3 f(v) - October(ssp126)'])/2
```

```
df_season["Average f(v) - November(ssp126)"] = (df_season['Site1_2 f(v) - November(ssp126)'] +
df_season['Site3 f(v) - November(ssp126)'] )/2

df_season["Average f(v) - December(ssp126)"] = (df_season['Site1_2 f(v) - December(ssp126)'] +
df_season['Site3 f(v) - December(ssp126)'])/2


#Calculating Power output

power1 = []
for i in range (0, 25):
   power1.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - January(ssp126)"][i] )
df_season["Average Power Output  - January(ssp126)"] = power1

power2 = []
for i in range (0, 25):
   power2.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - February(ssp126)"][i])
df_season["Average Power Output  - February(ssp126)"] = power2

power3 = []
for i in range (0, 25):
   power3.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - March(ssp126)"][i])
df_season["Average Power Output  - March(ssp126)"] = power3

power4 = []
for i in range (0, 25):
   power4.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - April(ssp126)"][i] )
df_season["Average Power Output  - April(ssp126)"] = power4

power5 = []
for i in range (0, 25):
   power5.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - May(ssp126)"][i])
df_season["Average Power Output  - May(ssp126)"] = power5

power6 = []
for i in range (0, 25):
   power6.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - June(ssp126)"][i])
df_season["Average Power Output  - June(ssp126)"] = power6

power7 = []
for i in range (0, 25):
   power7.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - July(ssp126)"][i] )
df_season["Average Power Output  - July(ssp126)"] = power7

power8 = []
for i in range (0, 25):
   power8.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - August(ssp126)"][i])
df_season["Average Power Output  - August(ssp126)"] = power8

power9 = []
for i in range (0,25):
   power9.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - September(ssp126)"][i])
```

```
df_season["Average Power Output  - September(ssp126)"] = power9

power10 = []
for i in range (0, 25):
   power10.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - October(ssp126)"][i])
df_season["Average Power Output  - October(ssp126)"] = power10

power11 = []
for i in range (0, 25):
   power11.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - November(ssp126)"][i] )
df_season["Average Power Output  - November(ssp126)"] = power11

power12 = []
for i in range (0, 25):
   power12.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - December(ssp126)"][i])
df_season["Average Power Output  - December(ssp126)"] = power12

print(df_season["Average Power Output  - December(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - January(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - February(ssp126)"][4:].sum())
print(df_season["Average Power Output  - March(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - April(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - May(ssp126)"][4:].sum())
print(df_season["Average Power Output  - June(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - July(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - August(ssp126)"][4:].sum())
print(df_season["Average Power Output  - September(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - October(ssp126)"][4:].sum() )
print(df_season["Average Power Output  - November(ssp126)"][4:].sum())


#Calculation parameters for Weibull distributions at site 1&2 and 3 at SSP 2-4.5

Esite1_2_jan = ddf[ddf['Months'] == 'Jan']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jan']['Site 1 ssp245(100m)'].mean()**3)
Esite3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jan']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_jan = 1+ (3.69/(Esite1_2_jan**2))
k_site3_jan = 1+ (3.69/(Esite3_jan**2))
c_site1_2_jan = ddf[ddf['Months'] == 'Jan']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_jan)
c_site3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_jan)

Esite1_2_feb = ddf[ddf['Months'] == 'Feb']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Feb']['Site 1 ssp245(100m)'].mean()**3)
Esite3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Feb']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_feb = 1+ (3.69/(Esite1_2_feb**2))
k_site3_feb = 1+ (3.69/(Esite3_feb**2))
c_site1_2_feb = ddf[ddf['Months'] == 'Feb']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_feb)
c_site3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_feb)
```

```
Esite1_2_mar = ddf[ddf['Months'] == 'Mar']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Mar']['Site 1 ssp245(100m)'].mean()**3)
Esite3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Mar']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_mar = 1+ (3.69/(Esite1_2_mar**2))
k_site3_mar = 1+ (3.69/(Esite3_mar**2))
c_site1_2_mar = ddf[ddf['Months'] == 'Mar']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_mar)
c_site3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_mar)

Esite1_2_apr = ddf[ddf['Months'] == 'Apr']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Apr']['Site 1 ssp245(100m)'].mean()**3)
Esite3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Apr']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_apr = 1+ (3.69/(Esite1_2_apr**2))
k_site3_apr = 1+ (3.69/(Esite3_apr**2))
c_site1_2_apr = ddf[ddf['Months'] == 'Apr']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_apr)
c_site3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_apr)

Esite1_2_may = ddf[ddf['Months'] == 'May']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'May']['Site 1 ssp245(100m)'].mean()**3)
Esite3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'May']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_may = 1+ (3.69/(Esite1_2_may**2))
k_site3_may = 1+ (3.69/(Esite3_may**2))
c_site1_2_may = ddf[ddf['Months'] == 'May']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_may)
c_site3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_may)

Esite1_2_jun = ddf[ddf['Months'] == 'Jun']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jun']['Site 1 ssp245(100m)'].mean()**3)
Esite3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jun']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_jun = 1+ (3.69/(Esite1_2_jun**2))
k_site3_jun = 1+ (3.69/(Esite3_jun**2))
c_site1_2_jun = ddf[ddf['Months'] == 'Jun']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_jun)
c_site3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_jun)

Esite1_2_jul = ddf[ddf['Months'] == 'Jul']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jul']['Site 1 ssp245(100m)'].mean()**3)
Esite3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jul']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_jul = 1+ (3.69/(Esite1_2_jul**2))
k_site3_jul = 1+ (3.69/(Esite3_jul**2))
c_site1_2_jul = ddf[ddf['Months'] == 'Jul']['Site 1 ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_jul)
c_site3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_jul)

Esite1_2_aug = ddf[ddf['Months'] == 'Aug']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Aug']['Site 1 ssp245(100m)'].mean()**3)
```

```
Esite3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Aug']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_aug = 1+ (3.69/(Esite1_2_aug**2))
k_site3_aug = 1+ (3.69/(Esite3_aug**2))
c_site1_2_aug = ddf[ddf['Months'] == 'Aug']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_aug)
c_site3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_aug)

Esite1_2_sep = ddf[ddf['Months'] == 'Sep']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Sep']['Site 1 ssp245(100m)'].mean()**3)
Esite3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Sep']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_sep = 1+ (3.69/(Esite1_2_sep**2))
k_site3_sep = 1+ (3.69/(Esite3_sep**2))
c_site1_2_sep = ddf[ddf['Months'] == 'Sep']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_sep)
c_site3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_sep)

Esite1_2_oct = ddf[ddf['Months'] == 'Oct']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Oct']['Site 1 ssp245(100m)'].mean()**3)
Esite3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Oct']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_oct = 1+ (3.69/(Esite1_2_oct**2))
k_site3_oct = 1+ (3.69/(Esite3_oct**2))
c_site1_2_oct = ddf[ddf['Months'] == 'Oct']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_oct)
c_site3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_oct)

Esite1_2_nov = ddf[ddf['Months'] == 'Nov']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Nov']['Site 1 ssp245(100m)'].mean()**3)
Esite3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Nov']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_nov = 1+ (3.69/(Esite1_2_nov**2))
k_site3_nov = 1+ (3.69/(Esite3_nov**2))
c_site1_2_nov = ddf[ddf['Months'] == 'Nov']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_nov)
c_site3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_nov)

Esite1_2_dec = ddf[ddf['Months'] == 'Dec']['Site 1 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Dec']['Site 1 ssp245(100m)'].mean()**3)
Esite3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp245(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Dec']['Site 3 ssp245(100m)'].mean()**3)
k_site1_2_dec = 1+ (3.69/(Esite1_2_dec**2))
k_site3_dec = 1+ (3.69/(Esite3_dec**2))
c_site1_2_dec = ddf[ddf['Months'] == 'Dec']['Site 1
ssp245(100m)'].mean()/math.gamma(1+1/k_site1_2_dec)
c_site3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp245(100m)'].mean()/math.gamma(1+1/k_site3_dec)


#Constructing Weibull probability columns for site 1&2 at SSP 2_4.5 scenario

k = k_site1_2_jan
```

```
c = c_site1_2_jan
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - January(ssp245)"] = result1

k = k_site1_2_feb
c = c_site1_2_feb
result2 = []
for i in range (0, 25):
   result2.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - February(ssp245)"] = result2

k = k_site1_2_mar
c = c_site1_2_mar
result3 = []
for i in range (0, 25):
   result3.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - March(ssp245)"] = result3

k = k_site1_2_apr
c = c_site1_2_apr
result4 = []
for i in range (0, 25):
   result4.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - April(ssp245)"] = result4

k = k_site1_2_may
c = c_site1_2_may
result5 = []
for i in range (0, 25):
   result5.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - May(ssp245)"] = result5

k = k_site1_2_jun
c = c_site1_2_jun
result6 = []
for i in range (0, 25):
   result6.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - June(ssp245)"] = result6

k = k_site1_2_jul
c = c_site1_2_jul
result7 = []
for i in range (0, 25):
   result7.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - July(ssp245)"] = result7

k = k_site1_2_aug
c = c_site1_2_aug
result8 = []
for i in range (0, 25):
   result8.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
```

```
df_season["Site1_2 f(v) - August(ssp245)"] = result8

k = k_site1_2_sep
c = c_site1_2_sep
result9 = []
for i in range (0, 25):
   result9.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - September(ssp245)"] = result9

k = k_site1_2_oct
c = c_site1_2_oct
result10 = []
for i in range (0, 25):
   result10.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - October(ssp245)"] = result10

k = k_site1_2_nov
c = c_site1_2_nov
result11 = []
for i in range (0, 25):
   result11.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - November(ssp245)"] = result11

k = k_site1_2_dec
c = c_site1_2_dec
result12 = []
for i in range (0, 25):
   result12.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site1_2 f(v) - December(ssp245)"] = result12


#Plotting Weibull distribution of wind speed at site 1&2 for SSP 1-2.6 scenario

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - January(ssp245)', label = 'January')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - February(ssp245)', label = 'February')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - March(ssp245)', label = 'March')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - April(ssp245)', label = 'April')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - May(ssp245)',  label = 'May')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - June(ssp245)',  label = 'June')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - July(ssp245)', label = 'July')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - August(ssp245)', label = 'August')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - September(ssp245)',  label = 'September')
ax = sns.lineplot(data=df_season, x='Speed', y= "Site1_2 f(v) - October(ssp245)",  label = 'October')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - November(ssp245)',  label = 'November', color
= 'black')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site1_2 f(v) - December(ssp245)',  label = 'December', color
= 'aqua')

ax.axes.set_title(label = "Monthly Weibull distribution at Site 1&2 at SSP2-4.5 scenario",
fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
```

```
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability of occurrence",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly weibull at Site 1&2(ssp245).png", format="png")
plt.show()
k = k_site3_jan
c = c_site3_jan
result1 = []
for i in range (0, 25):
  result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - January(ssp245)"] = result1

k = k_site3_feb
c = c_site3_feb
result2 = []
for i in range (0, 25):
  result2.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - February(ssp245)"] = result2

k = k_site3_mar
c = c_site3_mar
result3 = []
for i in range (0, 25):
  result3.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - March(ssp245)"] = result3

k = k_site3_apr
c = c_site3_apr
result4 = []
for i in range (0, 25):
  result4.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - April(ssp245)"] = result4

k = k_site3_may
c = c_site3_may
result5 = []
for i in range (0, 25):
  result5.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - May(ssp245)"] = result5

k = k_site3_jun
c = c_site3_jun
result6 = []
for i in range (0, 25):
  result6.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - June(ssp245)"] = result6

k = k_site3_jul
c = c_site3_jul
result7 = []
for i in range (0, 25):
  result7.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - July(ssp245)"] = result7
```

```
k = k_site3_aug
c = c_site3_aug
result8 = []
for i in range (0, 25):
   result8.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - August(ssp245)"] = result8

k = k_site3_sep
c = c_site3_sep
result9 = []
for i in range (0, 25):
   result9.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - September(ssp245)"] = result9

k = k_site3_oct
c = c_site3_oct
result10 = []
for i in range (0, 25):
   result10.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - October(ssp245)"] = result10

k = k_site3_nov
c = c_site3_nov
result11 = []
for i in range (0, 25):
   result11.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - November(ssp245)"] = result11

k = k_site3_dec
c = c_site3_dec
result12 = []
for i in range (0, 25):
   result12.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - December(ssp245)"] = result12


#Plotting Weibull distribution of wind speed at site 3 for SSP 2_4.5 scenario

fig = plt.figure(figsize=(10, 8))
sns.set_style("whitegrid")
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - January(ssp245)', label = 'January')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - February(ssp245)', label = 'February')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - March(ssp245)', label = 'March')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - April(ssp245)', label = 'April')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - May(ssp245)',  label = 'May')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - June(ssp245)',  label = 'June')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - July(ssp245)', label = 'July')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - August(ssp245)', label = 'August')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - September(ssp245)',  label = 'September')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - October(ssp245)',  label = 'October')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - November(ssp245)',  label = 'November', color =
'black')
```

```
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - December(ssp245)',  label = 'December', color = 'aqua')

ax.axes.set_title(label = "Monthly Weibull distribution at Site 3 at SSP2-4.5 scenario", fontdict={'fontsize': '20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability of occurrence",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly weibull at Site 3(ssp245).png", format="png")
plt.show()

df_season["Average f(v) - January(ssp245)"] = (df_season['Site1_2 f(v) - January(ssp245)'] + df_season['Site3 f(v) - January(ssp245)'] )/2
df_season["Average f(v) - February(ssp245)"] = (df_season['Site1_2 f(v) - February(ssp245)'] + df_season['Site3 f(v) - February(ssp245)'] )/2
df_season["Average f(v) - March(ssp245)"] = (df_season['Site1_2 f(v) - March(ssp245)'] + df_season['Site3 f(v) - March(ssp245)'] )/2
df_season["Average f(v) - April(ssp245)"] = (df_season['Site1_2 f(v) - April(ssp245)'] + df_season['Site3 f(v) - April(ssp245)'] )/2
df_season["Average f(v) - May(ssp245)"] = (df_season['Site1_2 f(v) - May(ssp245)'] + df_season['Site3 f(v) - May(ssp245)'] )/2
df_season["Average f(v) - June(ssp245)"] = (df_season['Site1_2 f(v) - June(ssp245)'] + df_season['Site3 f(v) - June(ssp245)'] )/2
df_season["Average f(v) - July(ssp245)"] = (df_season['Site1_2 f(v) - July(ssp245)'] + df_season['Site3 f(v) - July(ssp245)'] )/2
df_season["Average f(v) - August(ssp245)"] = (df_season['Site1_2 f(v) - August(ssp245)'] + df_season['Site3 f(v) - August(ssp245)'] )/2
df_season["Average f(v) - September(ssp245)"] = (df_season['Site1_2 f(v) - September(ssp245)'] + df_season['Site3 f(v) - September(ssp245)'] )/2
df_season["Average f(v) - October(ssp245)"] = (df_season['Site1_2 f(v) - October(ssp245)'] + df_season['Site3 f(v) - October(ssp245)'])/2
df_season["Average f(v) - November(ssp245)"] = (df_season['Site1_2 f(v) - November(ssp245)'] + df_season['Site3 f(v) - November(ssp245)'] )/2
df_season["Average f(v) - December(ssp245)"] = (df_season['Site1_2 f(v) - December(ssp245)'] + df_season['Site3 f(v) - December(ssp245)'])/2


#Calculating Power output

power1 = []
for i in range (0, 25):
   power1.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - January(ssp245)"][i] )
df_season["Average Power Output  - January(ssp245)"] = power1

power2 = []
for i in range (0, 25):
   power2.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - February(ssp245)"][i])
df_season["Average Power Output  - February(ssp245)"] = power2

power3 = []
for i in range (0, 25):
   power3.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - March(ssp245)"][i])
df_season["Average Power Output  - March(ssp245)"] = power3
```

```
power4 = []
for i in range (0, 25):
   power4.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - April(ssp245)"][i] )
df_season["Average Power Output  - April(ssp245)"] = power4

power5 = []
for i in range (0, 25):
   power5.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - May(ssp245)"][i])
df_season["Average Power Output  - May(ssp245)"] = power5

power6 = []
for i in range (0, 25):
   power6.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - June(ssp245)"][i])
df_season["Average Power Output  - June(ssp245)"] = power6

power7 = []
for i in range (0, 25):
   power7.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - July(ssp245)"][i] )
df_season["Average Power Output  - July(ssp245)"] = power7

power8 = []
for i in range (0, 25):
   power8.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - August(ssp245)"][i])
df_season["Average Power Output  - August(ssp245)"] = power8

power9 = []
for i in range (0,25):
   power9.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - September(ssp245)"][i])
df_season["Average Power Output  - September(ssp245)"] = power9

power10 = []
for i in range (0, 25):
   power10.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - October(ssp245)"][i])
df_season["Average Power Output  - October(ssp245)"] = power10

power11 = []
for i in range (0, 25):
   power11.append(0.5 * 1.225 * (i**3) * df_season["Average f(v) - November(ssp245)"][i] )
df_season["Average Power Output  - November(ssp245)"] = power11

power12 = []
for i in range (0, 25):
   power12.append(0.5 * 1.225 * (i**3)  * df_season["Average f(v) - December(ssp245)"][i])
df_season["Average Power Output  - December(ssp245)"] = power12

print(df_season["Average Power Output  - December(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - January(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - February(ssp245)"][4:].sum())
print(df_season["Average Power Output  - March(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - April(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - May(ssp245)"][4:].sum())
print(df_season["Average Power Output  - June(ssp245)"][4:].sum() )
```

```
print(df_season["Average Power Output  - July(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - August(ssp245)"][4:].sum())
print(df_season["Average Power Output  - September(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - October(ssp245)"][4:].sum() )
print(df_season["Average Power Output  - November(ssp245)"][4:].sum())

#Calculation parameters for Weibull distributions at site 1,2 and 3 at SSP 5-8.5

Esite3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jan']['Site 3 ssp585(100m)'].mean()**3)
k_site3_jan = 1+ (3.69/(Esite3_jan**2))
c_site3_jan = ddf[ddf['Months'] == 'Jan']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_jan)

Esite3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Feb']['Site 3 ssp585(100m)'].mean()**3)
k_site3_feb = 1+ (3.69/(Esite3_feb**2))
c_site3_feb = ddf[ddf['Months'] == 'Feb']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_feb)

Esite3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Mar']['Site 3 ssp585(100m)'].mean()**3)
k_site3_mar = 1+ (3.69/(Esite3_mar**2))
c_site3_mar = ddf[ddf['Months'] == 'Mar']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_mar)

Esite3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Apr']['Site 3 ssp585(100m)'].mean()**3)
k_site3_apr = 1+ (3.69/(Esite3_apr**2))
c_site3_apr = ddf[ddf['Months'] == 'Apr']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_apr)

Esite3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'May']['Site 3 ssp585(100m)'].mean()**3)
k_site3_may = 1+ (3.69/(Esite3_may**2))
c_site3_may = ddf[ddf['Months'] == 'May']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_may)

Esite3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jun']['Site 3 ssp585(100m)'].mean()**3)
k_site3_jun = 1+ (3.69/(Esite3_jun**2))
c_site3_jun = ddf[ddf['Months'] == 'Jun']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_jun)

Esite3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Jul']['Site 3 ssp585(100m)'].mean()**3)
k_site3_jul = 1+ (3.69/(Esite3_jul**2))
c_site3_jul = ddf[ddf['Months'] == 'Jul']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_jul)

Esite3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Aug']['Site 3 ssp585(100m)'].mean()**3)
k_site3_aug = 1+ (3.69/(Esite3_aug**2))
c_site3_aug = ddf[ddf['Months'] == 'Aug']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_aug)

Esite3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Sep']['Site 3 ssp585(100m)'].mean()**3)
k_site3_sep = 1+ (3.69/(Esite3_sep**2))
c_site3_sep = ddf[ddf['Months'] == 'Sep']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_sep)
```

```
Esite3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Oct']['Site 3 ssp585(100m)'].mean()**3)
k_site3_oct = 1+ (3.69/(Esite3_oct**2))
c_site3_oct = ddf[ddf['Months'] == 'Oct']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_oct)

Esite3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Nov']['Site 3 ssp585(100m)'].mean()**3)
k_site3_nov = 1+ (3.69/(Esite3_nov**2))
c_site3_nov = ddf[ddf['Months'] == 'Nov']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_nov)

Esite3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp585(100m)cubbed'].mean()/(ddf[ddf['Months'] ==
'Dec']['Site 3 ssp585(100m)'].mean()**3)
k_site3_dec = 1+ (3.69/(Esite3_dec**2))
c_site3_dec = ddf[ddf['Months'] == 'Dec']['Site 3 ssp585(100m)'].mean()/math.gamma(1+1/k_site3_dec)


k = k_site3_jan
c = c_site3_jan
result1 = []
for i in range (0, 25):
   result1.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - January(ssp585)"] = result1

k = k_site3_feb
c = c_site3_feb
result2 = []
for i in range (0, 25):
   result2.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - February(ssp585)"] = result2

k = k_site3_mar
c = c_site3_mar
result3 = []
for i in range (0, 25):
   result3.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - March(ssp585)"] = result3

k = k_site3_apr
c = c_site3_apr
result4 = []
for i in range (0, 25):
   result4.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - April(ssp585)"] = result4

k = k_site3_may
c = c_site3_may
result5 = []
for i in range (0, 25):
   result5.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - May(ssp585)"] = result5

k = k_site3_jun
```

```
c = c_site3_jun
result6 = []
for i in range (0, 25):
   result6.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - June(ssp585)"] = result6

k = k_site3_jul
c = c_site3_jul
result7 = []
for i in range (0, 25):
   result7.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - July(ssp585)"] = result7

k = k_site3_aug
c = c_site3_aug
result8 = []
for i in range (0, 25):
   result8.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - August(ssp585)"] = result8

k = k_site3_sep
c = c_site3_sep
result9 = []
for i in range (0, 25):
   result9.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - September(ssp585)"] = result9

k = k_site3_oct
c = c_site3_oct
result10 = []
for i in range (0, 25):
   result10.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - October(ssp585)"] = result10

k = k_site3_nov
c = c_site3_nov
result11 = []
for i in range (0, 25):
   result11.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - November(ssp585)"] = result11

k = k_site3_dec
c = c_site3_dec
result12 = []
for i in range (0, 25):
   result12.append((k/c)*((i/c)**(k-1)) *math.exp(-1* ((i/c)**k)))
df_season["Site3 f(v) - December(ssp585)"] = result12


#Plotting Weibull distribution of wind speed at site 1, 2 and 3 for SSP 5_8.5 scenario

fig = plt.figure(figsize=(10, 8))
```

```
sns.set_style("whitegrid")
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - January(ssp585)', label = 'January')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - February(ssp585)', label = 'February')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - March(ssp585)', label = 'March')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - April(ssp585)', label = 'April')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - May(ssp585)',  label = 'May')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - June(ssp585)',  label = 'June')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - July(ssp585)', label = 'July')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - August(ssp585)', label = 'August')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - September(ssp585)',  label = 'September')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - October(ssp585)',  label = 'October')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - November(ssp585)',  label = 'November', color =
'black')
ax = sns.lineplot(data=df_season, x='Speed', y= 'Site3 f(v) - December(ssp585)',  label = 'December', color =
'aqua')

ax.axes.set_title(label = "Monthly Weibull distribution at all sites at SSP5-8.5 scenario", fontdict={'fontsize':
'20', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_xlabel("Wind Speed (m/s)",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
ax.set_ylabel("Probability of occurrence",fontdict={'fontsize': '15', 'fontweight' : '3', 'family': 'Calibri'} )
plt.savefig("Monthly weibull at Site 1, 2,3(ssp585).png", format="png")
plt.show()

power1 = []
for i in range (0, 25):
   power1.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - January(ssp585)'][i] )
df_season["Average Power Output  - January(ssp585)"] = power1

power2 = []
for i in range (0, 25):
   power2.append(0.5 * 1.225 * (i**3)  * df_season['Site3 f(v) - February(ssp585)'][i])
df_season["Average Power Output  - February(ssp585)"] = power2

power3 = []
for i in range (0, 25):
   power3.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - March(ssp585)'][i])
df_season["Average Power Output  - March(ssp585)"] = power3

power4 = []
for i in range (0, 25):
   power4.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - April(ssp585)'][i] )
df_season["Average Power Output  - April(ssp585)"] = power4

power5 = []
for i in range (0, 25):
   power5.append(0.5 * 1.225 * (i**3)  * df_season['Site3 f(v) - May(ssp585)'][i])
df_season["Average Power Output  - May(ssp585)"] = power5

power6 = []
for i in range (0, 25):
   power6.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - June(ssp585)'][i])
df_season["Average Power Output  - June(ssp585)"] = power6
```

```
power7 = []
for i in range (0, 25):
   power7.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - July(ssp585)'][i] )
df_season["Average Power Output  - July(ssp585)"] = power7

power8 = []
for i in range (0, 25):
   power8.append(0.5 * 1.225 * (i**3)  * df_season['Site3 f(v) - August(ssp585)'][i])
df_season["Average Power Output  - August(ssp585)"] = power8

power9 = []
for i in range (0,25):
   power9.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - September(ssp585)'][i])
df_season["Average Power Output  - September(ssp585)"] = power9

power10 = []
for i in range (0, 25):
   power10.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - October(ssp585)'][i])
df_season["Average Power Output  - October(ssp585)"] = power10

power11 = []
for i in range (0, 25):
   power11.append(0.5 * 1.225 * (i**3) * df_season['Site3 f(v) - November(ssp585)'][i] )
df_season["Average Power Output  - November(ssp585)"] = power11

power12 = []
for i in range (0, 25):
   power12.append(0.5 * 1.225 * (i**3)  * df_season['Site3 f(v) - December(ssp585)'][i])
df_season["Average Power Output  - December(ssp585)"] = power12

print(df_season["Average Power Output  - December(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - January(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - February(ssp585)"][4:].sum())
print(df_season["Average Power Output  - March(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - April(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - May(ssp585)"][4:].sum())
print(df_season["Average Power Output  - June(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - July(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - August(ssp585)"][4:].sum())
print(df_season["Average Power Output  - September(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - October(ssp585)"][4:].sum() )
print(df_season["Average Power Output  - November(ssp585)"][4:].sum())
```