I. INTRODUCTION

Wind power has played an important role in the global transition toward sustainable and renewable energy sources. Unlike fossil fuels, wind power is a clean and inexhaustive resource as it is freely available and abundant across the globe. Both in Europe and worldwide, the wind power industry has been growing at a fast pace. In 2021 wind electricity generation increased by a record 273 TWh (up 17%). This was 55% higher growth than that achieved in 2020 and was the highest among all renewable power technologies (IEA, 2022). The development of wind power helps many countries significantly cut down their carbon footprint and achieve their emission reduction targets while reducing their dependence on fossil fuels as well as vulnerability to energy price fluctuations and supply disruption. In Europe, Denmark had the highest share of wind with an impressive 55% in 2022, followed by Ireland with 34%. The UK became the country with the third highest share of wind with 28% (Wind Europe, 2023). In this report, I will focus on exploring, analyzing, and forecasting the wind power industry in Denmark. The report aims to answer the question "How do weather factors and climate change impact wind power production in Denmark?"

In the first part of the report, I will discuss about the background of wind power industry and regulation in Denmark and develop the hypothesis relating to research question based on existing literatures. Then, I will implement exploratory analysis on dataset and forecast the wind power production with different forecasting models including SARIMA, Dynamic Hamonic Regression Model, Linear Regression, Random Forest, and Regression with Arima Error. The relationship between the weather-related factors and wind energy production is also explored using regression model. Lastly, the results and limitations of this project will be discussed.

II. BACKGROUND &THEORY

2.1 Wind Energy in Denmark

Denmark has been a pioneer in developing commercial onshore wind power since 1970s and a substantial share of the wind turbines around the world are produced by Danish manufacturers such as Vestas—the world's largest wind-turbine (Wikipedia, 2023). Wind energy industry in Denmark has grown substantially over the last decades. In 2022, approximately 55% of total electricity production was covered by wind power, which was significantly increased from 19.3% in 2009 (Statista, 2023). With this figure, Denmark became the country with the highest wind power penetration in the world (Ourworldindata, 2022). Following the success of onshore wind, Denmark was also a pioneer in developing offshore wind technology. The first offshore wind farm in the world was installed in Vindeby, Denmark in 2000. One of the critical factors contributing to Denmark's success in wind power is its favorable geographical location. It has more than 7,000 kilometers of coastline which provides ideal conditions for wind power development. The coastal areas experience strong and consistent wind speeds, making them potential for offshore wind farms. Although onshore wind accounts for the largest part of wind energy, the wind power sector in Denmark has moved toward offshore in recent years. Offshore wind farms offer several advantages compared to their onshore counterparts, including higher wind speeds, and reduced visual impact and noise. Winds are more stable and stronger, allowing the turbines to harvest the energy more effectively and generating a higher amount of electricity per unit installed. However, capital investment in offshore wind farms can be very expensive, and the operations and maintenance of offshore turbines are difficult and restricted due to weather conditions. The country has invested in research to optimize offshore wind technologies, aiming at increasing energy production, and reducing costs. Denmark's achievements in wind power have resulted in substantial environmental benefits. By reducing reliance on fossil fuels, wind energy has helped Denmark reduce greenhouse gas emissions and combat the climate change. The Danish government is considering an expansion of its offshore wind capacity by adding an extra 4 GW by the end of the decade, increasing the target for 2030 from 8.9 GW to 12.9 GW. This substantial increase in capacity would enable Denmark to produce approx. 49.5 TWh/year of clean and sustainable energy. The country planned to completely decarbonize the electricity sector and get rid of fossil fuels by 2030 (Enerdata, 2022).

2.2 Regulation

Denmark has implemented a robust regulatory framework to support the development and the growth of its wind power industry. The well-designed government policies have been a key driver to make Denmark a global leader in wind energy. The regulatory framework for wind power in Denmark is primarily governed by the Promotion of Renewable Energy Act issued in 2008 which provides legal basis for promotion of energy production using renewable energy sources. According to the Energy Act, the right to exploit energy from water and wind within the territorial waters and the exclusive economic zone (up to 200 nautical miles) around Denmark belongs to the Danish State (Promotion of Energy Act, 2008). To establish an offshore wind farm in Denmark, the operators need to be granted three licenses by Danish Energy Agency including license to carry out preliminary investigations, license to establish the offshore wind turbines, and license to exploit wind power for a certain number of years. To encourage wind power investments, Denmark has provided various financial support mechanisms. It established a detailed feed-in tariff system for wind power since late 1970, where wind power producers receive fixed price per amount of wind energy generated, instead of receiving the market electricity price (IEA, 2013). Wind power producers were guaranteed a feed-in tariff paid by grid companies that depended on the location of the turbine, plus a carbon-tax refund and a production subsidy from the government. These policies significantly boosted wind power development. Besides, the country also subsided up to 30% of initial capital cost in the early years which was gradually reduced to 20%, and then zero in 1988 when it observed the reliable growth and improved cost-effectiveness of the turbines (Irena, 2013). From 2000 onwards, new support systems were implemented, in which the support scheme transitioned from a feed-in tariff to a variable premium to enhance integration with the liberalized electricity market. Renewable energy plants are subjected to more market-based tariffs. In addition to the spot market price for electricity, new wind power plants are eligible to receive a premium price for each kWh produced. For onshore turbines connected to the grid after February 2008, the premium price was set at 34 Euros/MWh for 22,000 full load hours, equivalent to around 10 years, while the average spot market was approximately 38 Euros/MWh in 2008. On the other hand, most offshore wind plants have been financed by electrical utilities under an agreement with the Danish government. Offshore turbines connected to the grid after January 2000 receive a feed-in-tariff set at 61 Euros/MHh for 42,000 full load hours, i.e. about 12 years. Notably, the additional costs of wind power (compared to conventional power) are passed on to Danish power consumers as support for renewable electricity is paid by all electricity consumers as a public service obligation (PSO) (Jauréguy-Naudin, 2010). In addition to national regulations, Denmark is an active member of the International Energy Agency Wind Technology Collaboration Program and has contributed to the development of international standards and best practices in wind energy.

2.3 The NordPool Spot Market

The Danish grid is directly connected to Sweden, Norway, and Germany, and the electricity generated will be traded in the joint electricity exchange market called Nord Pool Spot. The NordPool Spot operates as a spot market for electricity in the Nordic countries. It plays a vital role in facilitating the efficient trading and allocation of electricity in the Nordic region. The Nord Pool spot market serves as a platform which allows market participants to submit bids and offers indicating the quantity of electricity they are willing to buy or sell, and the price at which they are willing to transact. The market operates on the principle of supply and demand, and the trading platform matches these bids and offers to determine the clearing price for each trading period. The Nord Pool spot market consists of two main markets: the day-ahead market, and the intraday market. The day-ahead market is the most significant market where all suppliers give bids for production in each hour of the next day before 12 o'clock and the consumers decide the amount and prices they will buy for each hour of the next day. The Intraday market operates after the day-ahead market and allows market participants to their positions closer to real-time. Their positions can be adjusted based on updated supply and demand conditions, unexpected changes that occur during the day (Nordpoolgroup, 2023). The Energinet, known as the Danish Transmission System Operator (TSO), is responsible for the management and operation of the electricity transmission grid in Denmark. This includes the authority to curtail wind turbines when necessary for security purposes and responsibility to compensate the owners of wind power plants for any loss of earnings resulting from curtailment (Jauréguy-Naudin, 2010). It plays an important role in ensuring that the operation of wind power plants aligns with the overall electricity market in Denmark.

2.4 Literature Review

Even though wind energy is currently one of the most promising renewable energy sources, wind resources are uncertain and susceptible to weather volatility. The production of wind energy depends on weather conditions such as wind speed, temperature, (e.g. on days with weak wind speed, low amount of energy can be produced and vice versa). The level of wind power generation can fluctuate throughout the day and across different seasons. This unpredictable fluctuations in wind energy production causes economic challenges and makes it difficult to compete with other forms of electricity generation (Thomas Leopold Berg, 2020). Therefore, forecasting accurate wind power production is crucial in helping grid operators and power system planners more efficiently integrate wind power into the electricity grid and make better resource allocation decisions. Weather variables such as wind direction, temperature, pressure, and humidity, among others, influence wind generation, indicating that they can be promising to predict wind power production according to Amir Sharifian (2017). A study done by A Bossavy developed wind power forecasting models using wind speed and direction NWPs as input (A Bossavy, 2013). Their models focus on predicting ramp events which are significant changes in wind production over one or several hours. The model results showed that the forecasts based on NWP ensembles turned out to be reliable with greater accuracy. Another study done by Ekaterina Vladislavleva (2012) also examine the impact of different weather conditions such as wind speed, pressure, temperature on the energy output of wind farms in Australia. They found that wind energy output can be predicted from publicly available weather data with high accuracy up to 80%. The study also identified the variables wind gust, dewpoint, and humidity as the most important factors for accurate wind energy output prediction.

Based on above literature, the hypotheses of this report can be presented as follows:

H1: Weather related factors have a significant relationship with wind energy production.

III. DATA AND MODELING

3.1 Data

The data used in this report is historical Danish electricity production and consumption data, historical weather data.

The Danish electricity production and consumption data was downloaded from Energi Data Service which provides the open access to data about Danish energy system such as electricity consumption, production, CO2 emission (Energidataservice, 2023). The dataset contains electricity consumption and production from renewable energy sources as well as the exchange energy between Denmark and other countries such as Norway, Sweden, Germany. It has 322,030 observations and covers the period of 2004-2023 at hourly level. As I will focus on exploring and predicting total wind power production in Denmark, only suitable variables are selected from data for analysis.

The historical weather data was retrieved from (DMI, 2023) and contains weather related variables such as temperature, wind speed, pressure, etc over period of 2015-2023. As the raw data of meteorological measurement is in .txt format and updated in 10 min frequency, I need to convert it to suitable format and aggregate to daily level for easier analysis. To predict the wind production, I will use two variables from weather data including “temp_dry” (.i.e present air temperature measured 2 m over terrain - °C) and “wind_speed” (.i.e mean wind speed measured 10 m over terrain - m/s).

3.2 Data Preparation and Exploration

3.2.1 Data Preparation

In this section, I will load in the required packages and data necessary for analysis. The data will then be cleaned and prepared for further investigation. Furthermore, the exploratory analysis will be conducted to gain insights into the trends and patterns in the data.

```r
#loading necessary libraries
library(tidyverse)
library(jsonlite)
library(foreach)
library(doParallel)
library(tidyr)
library(dplyr)
library(parallel)
library(zoo)
library(fpp3)
library(dynlm)
library(tseries)
```

```r
setwd("D:/UNI/COURSES/ENE434 - Energy Industry Analytics/final pj/data")
#load the electricity production data
electricity_prod = read.csv("ProductionConsumptionSettlement.csv", sep = ";")
```

As discussed earlier, the primary objective of this project is to explore and forecast total wind power production in Denmark. To achieve this, I will select only relevant variables for analysis including `HourUTC`, `OffshoreWindLt100MW_MWh`, `OffshoreWindGe100MW_MWh`, `OnshoreWindLt50kW_MWh`, `OnshoreWindGe50kW_MWh`, and `GrossConsumptionMWh` from 27 existing variables from the dataset. The wind production in the data is measured in MWh. As wind production data is categorized into different subcategories in the dataset, I will create a new variable called `wind_prod`, which represents total wind production in Denmark. Furthermore, to simplify the analysis, data will be examined at a daily level rather than hourly level. This approach will help identify trends and patterns in wind power production over long time period.

```r
#select wind production columns
wind_prod <- electricity_prod[,c("HourUTC","PriceArea",
                                 "OffshoreWindLt100MW_MWh",
                                 "OffshoreWindGe100MW_MWh",
                                 "OnshoreWindLt50kW_MWh",
                                 "OnshoreWindGe50kW_MWh",
                                 "GrossConsumptionMWh")]
#convert type of variables to suitable one
wind_prod$HourUTC <- as.Date(wind_prod$HourUTC, format = "%Y-%m-%d %H:%M")
wind_prod$OffshoreWindLt100MW_MWh <-
  as.numeric(gsub(",", ".", wind_prod$OffshoreWindLt100MW_MWh))
wind_prod$OffshoreWindGe100MW_MWh <-
  as.numeric(gsub(",", ".", wind_prod$OffshoreWindGe100MW_MWh))
wind_prod$OnshoreWindLt50kW_MWh <-
  as.numeric(gsub(",", ".", wind_prod$OnshoreWindLt50kW_MWh))
wind_prod$OnshoreWindGe50kW_MWh <-
  as.numeric(gsub(",", ".", wind_prod$OnshoreWindGe50kW_MWh))
wind_prod$GrossConsumptionMWh <-
```

```
  as.numeric(gsub(",", ".", wind_prod$GrossConsumptionMWh))

#aggregate data by date
wind_prod <- wind_prod %>% group_by(date = HourUTC) %>%
  summarise(OffshoreWindLt100MW_MWh = sum(OffshoreWindLt100MW_MWh),
            OffshoreWindGe100MW_MWh = sum(OffshoreWindGe100MW_MWh),
            OnshoreWindLt50kW_MWh = sum(OnshoreWindLt50kW_MWh),
            OnshoreWindGe50kW_MWh = sum(OnshoreWindGe50kW_MWh),
            GrossConsumptionGWh = sum(GrossConsumptionMWh)/1000)
wind_prod <- wind_prod %>%
  transmute(date, OffshoreWind_MWh = OffshoreWindLt100MW_MWh
            + OffshoreWindGe100MW_MWh, OnshoreWind_MWh = OnshoreWindLt50kW_MWh +
            OnshoreWindGe50kW_MWh, GrossConsumptionGWh)

#create new variable which is total wind production in GWh
wind_prod$wind_prod_GWh <- (wind_prod$OffshoreWind_MWh +
                              wind_prod$OnshoreWind_MWh)/1000

#percentage of wind production in total electricity production
wind_prod$wind_prod_percentage <-
  wind_prod$wind_prod_GWh/wind_prod$GrossConsumptionGWh*100
```

Next, I will load in weather data. Since the meteorological data is stored in .txt format and updated in 10 min frequency, it is necessary to convert the data to more suitable format and aggregate to daily level for easier analysis.

```
files <- list.files(
  path = "D:/UNI/COURSES/ENE434 - Energy Industry Analytics/final pj/data/weather data/",
  pattern = "*.txt", full.names = TRUE)
df <- data.frame()
# Specify the number of cores to use
maxcores <- 8
Cores <- min(parallel::detectCores(), maxcores)
# Instantiate the cores:
cl <- makeCluster(Cores)

# Register parallel backend
registerDoParallel(cl)

# covert data to suitable type
weather_data <- foreach(i = 1:length(files), .combine = rbind,
    .packages = c('magrittr', 'dplyr', 'jsonlite', 'tidyverse')) %dopar% {
  lines <- readLines(files[i])
  json_data <- lapply(lines, function(line) fromJSON(line))
  values <- sapply(json_data, function(obj) obj$properties$value)
  type <- sapply(json_data, function(obj) obj$properties$parameterId)
  date <- sapply(json_data, function(obj) obj$properties$observed)
  df_temp <- data.frame(type = type, values = values, date = date)
  df_temp <- df_temp[df_temp["type"] == c("wind_speed", "temp_dry"),]
  df_temp <- df_temp[!duplicated(df_temp$date),]
  df_temp <- aggregate(df_temp$values, by=list(df_temp$type), FUN=mean)
  colnames(df_temp) <- c("type", "values")
  df_temp <- pivot_wider(df_temp, names_from = type, values_from = values)
  df_temp$date <- gsub(".*\\/(.*)\\.txt", "\\1", files[i])
  df_temp
}

stopCluster(cl)
```

```
#change date to date format
weather_data$date <- as.Date(weather_data$date, format = "%Y-%m-%d")

#group data by date
weather_data <- weather_data %>% select(date,temperature = temp_dry, wind_speed)
```

```
#merge wind production and weather data by date
df_wind <- merge(wind_prod, weather_data, by = "date")
```

```
#checking missing values
colSums(is.na(df_wind))
```

```
##                 date     OffshoreWind_MWh      OnshoreWind_MWh
##                    0                    0                    0
##  GrossConsumptionGWh        wind_prod_GWh wind_prod_percentage
##                    0                    0                    0
##          temperature           wind_speed
##                    0                    0
```

```
#checking possible outliers
outlier_variables = c('wind_prod_GWh','temperature','wind_speed')

#for (i in outlier_variables){
#  print(i)
#  print(length(boxplot.stats(df_wind[,i])$out))
#  print(unique(boxplot.stats(df_wind[,i])$out))
#}
```

```
#in case of removing all outliers from dataset
df_wind_new = df_wind
for (x in outlier_variables) {
value = df_wind_new[,x][df_wind_new[,x] %in% boxplot.stats(df_wind_new[,x])$out]
df_wind_new[,x][df_wind_new[,x] %in% value] = NA
}
df_wind_new = drop_na(df_wind_new)
#Percentage of outliers removed from data
print(paste(round((nrow(df_wind) - nrow(df_wind_new))/nrow(df_wind)*100,0),
            "% observations are removed"))
```

```
## [1] "2 % observations are removed"
```

The `df_wind` dataset is merged from `electricity_prod` and `weather_data`. The data set contains no missing values; however, it does contain outliers. To identify the outliers, the box plot method is usded, which presents the minimum, median, first quartiles (Q1), third quartiles (Q3), and maximum values of the numeric variables. Outliers are defined as observations that fall above (Q3 + 1.5IQR) or below (Q1 -

1.5IQR), where IQR is the difference between Q1 and Q3. However, the removal of outliers detected by this method can lead to the loss of approx. 2% of observations. As size of daya is relatively small, after thorough examination, I decided to remove the outliers that deviate significantly from other observations.

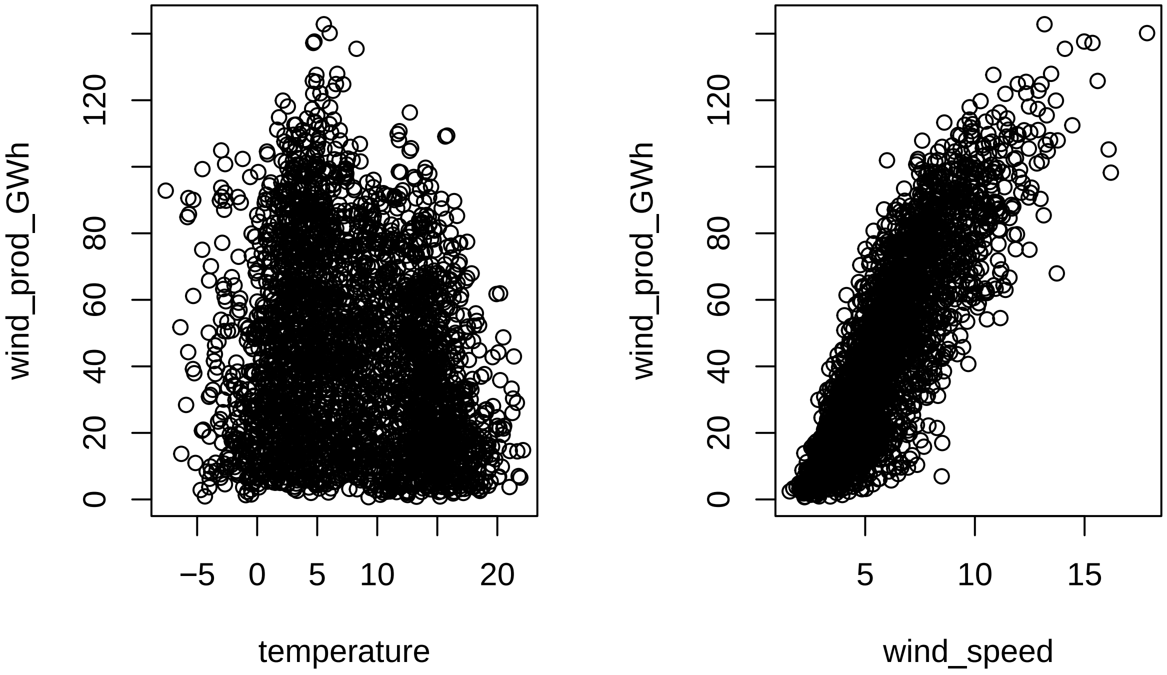
```
# remove  outliers
df_wind[,'wind_speed'][df_wind[,'wind_speed'] == 17.84769] = NA
df_wind = drop_na(df_wind)
```

3.2.2 Data Exploration

In next steps, I will examine the relationship between wind production and other weather variables by using scatter plots and calculating correlation. The correlation is computed using Pearson method. The Pearson correlation coefficient quantifies the strength and direction of the linear relation between the variables. The coefficient ranges from -1 to +1, where value of +1 indicates a strong positive correlation and value of -1 represents a strong negative correlation. A Pearson correlation value of 0 suggests there is no linear correlation between the variables.

```
#plot the relationship between wind production and weather variables
par(mfrow = c(1, 2))
plot(wind_prod_GWh~., data = df_wind[,-c(1:4,6)])
mtext("Figure 1: Scatter plot between wind power production and
      weather variables",
      outer = TRUE, line = -3, cex = 1)
```

Figure 1: Scatter plot between wind power production and weather variables



```
#correlation
weather_var <- c("temperature", "wind_speed")
cor <- cor(df_wind$wind_prod_GWh, df_wind[weather_var])
as.table(cor)
```

```
##   temperature wind_speed
## A  -0.1957869  0.8569431
```

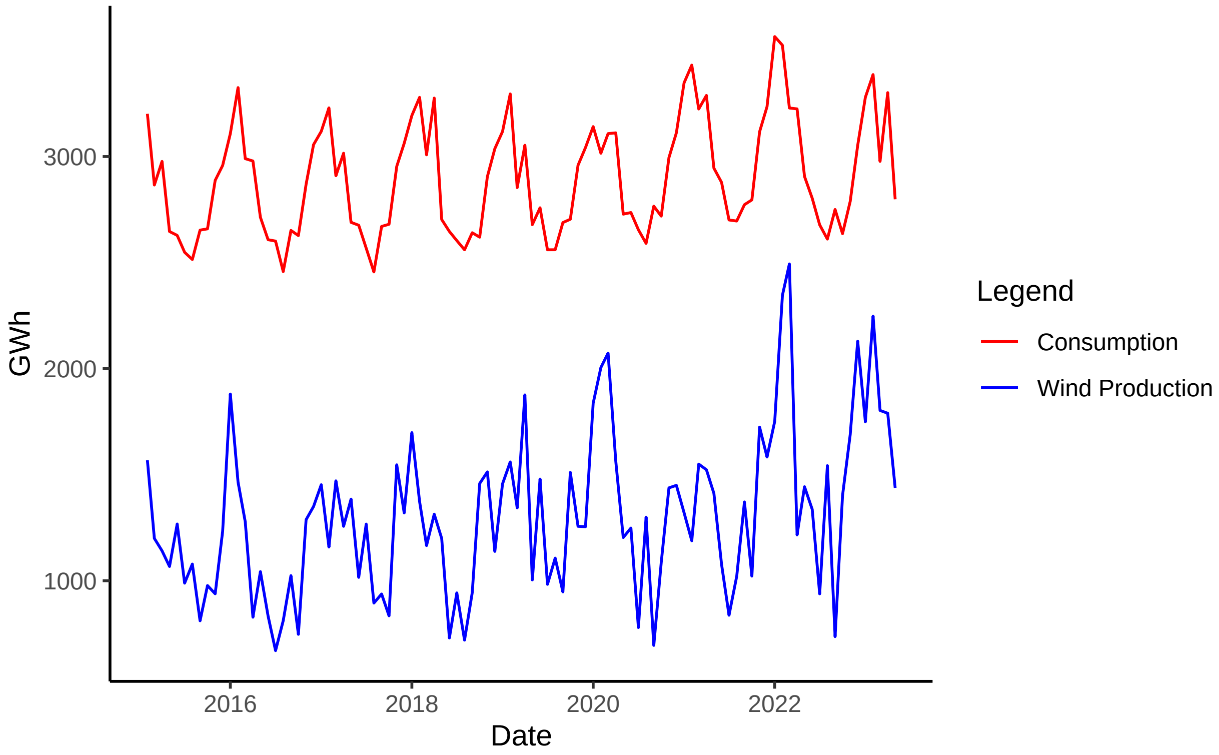
As expected, wind speed demonstrates a strong positive correlation of 0.86 with the produced amount of wind power, indicating that wind speed is a promising predictor of wind production. On the other hand, temperature have negative relationship with wind power production. However, the relationship between wind production and temperature is not quite clear with weak correlation at approximately -0.19. It is necessary to perform further investigation and analysis to better understand potential influence of these variables on wind power generation.

Then I will plot the wind power production and electricity consumption over time.

```
#plot wind production and consumption over time
df_wind %>% group_by(date = ceiling_date(date, "month")) %>%
  summarise(wind_prod_GWh = sum(wind_prod_GWh),
            GrossConsumptionGWh = sum(GrossConsumptionGWh)) %>%
  ggplot(aes(x = date)) + geom_line(aes(y = GrossConsumptionGWh,
                                        color = "Consumption")) +
  geom_line(aes(y = wind_prod_GWh, color = "Wind Production")) +
  scale_color_manual(name = "Legend", values = c("Consumption" = "red",
                                                 "Wind Production" = "blue")) +
  labs(title = "Figure 2",
       subtitle = "Wind Production and Consumption over time",
       x = "Date", y = "GWh") + theme_classic()
```

Figure 2

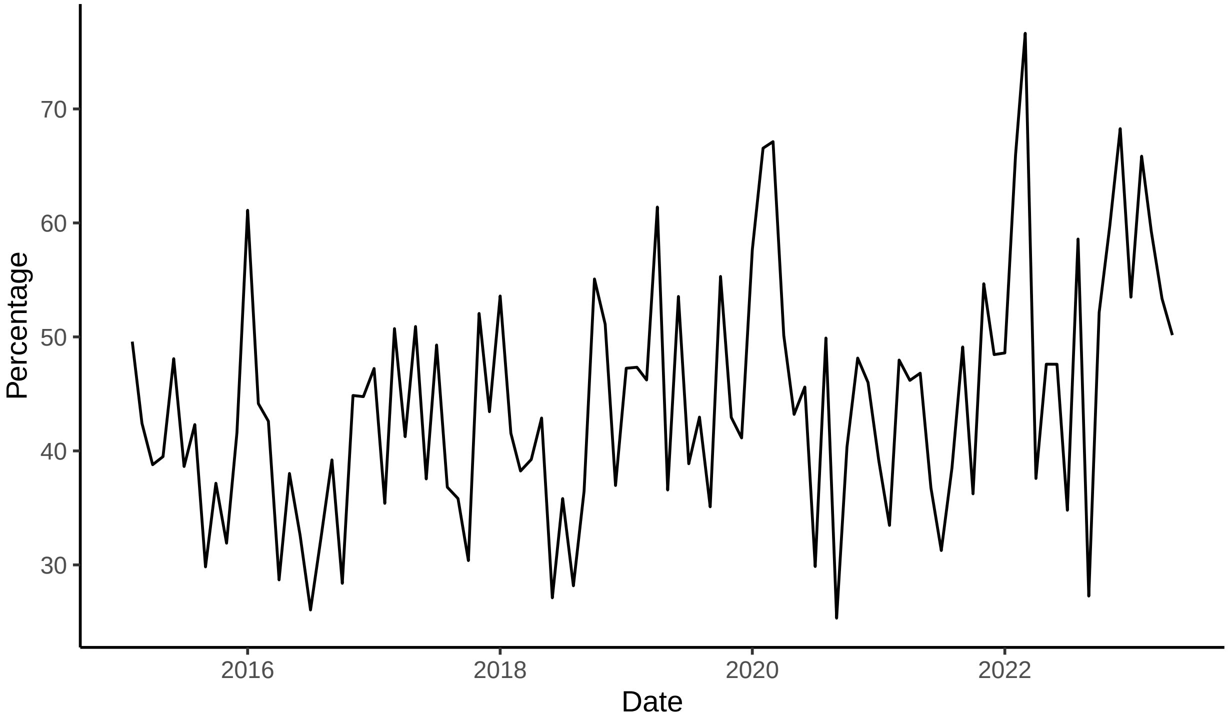
Wind Production and Consumption over time



```
#plot wind production percentage over time
df_wind %>% group_by(date = ceiling_date(date, "month")) %>%
  summarise(wind_prod_percentage = mean(wind_prod_percentage)) %>%
  ggplot(aes(x = date)) +
    geom_line(aes(y = wind_prod_percentage)) +
    labs(title = "Figure 3",
         subtitle = "Share of Wind Production in Electricity
         Consumption by Month", x = "Date", y = "Percentage")+ theme_classic()
```

Figure 3

Share of Wind Production in Electricity Consumption by Month

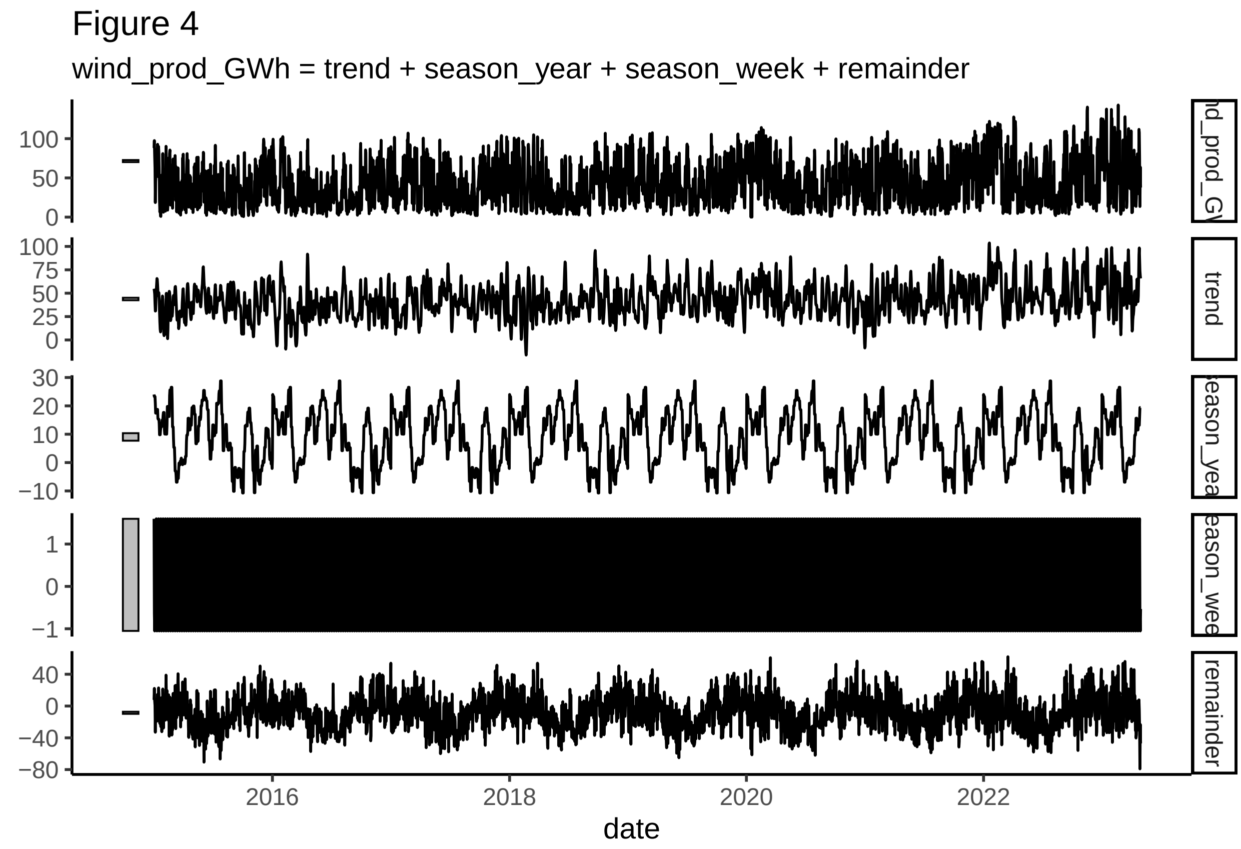


The figure 2 shows that the wind production fluctuated with overall increasing trend over the time. Besides, there are clear seasonal patterns throughout the year where the wind power production tends to be high in winter and low in summer. From figure 3, it can be observed that the share of wind production in total electricity consumption exhibits a consistent upward trend from 2015-2023. Notably, there are some periods where wind production share exceeds 70%. This trend reflects the Danish government's commitment to expanding investment in wind industry, aiming to increase share of electricity production from wind to 84% by 2035 (Wikipedia,2023).

In general, the wind production and total electricity consumption for the period 2015 to 2023 were highly correlated from 2015 to 2023. During the period of low consumption, wind production also tended to be low, and vice versa. This relationship can be attributed to the dependence of wind production on weather. On days with high wind activity (.i.e, bad weather), people tend to stay indoor and consume more electricity. Besides, it is worth noting that both total consumption and wind production experience a slight decline in 2016 and 2019/2020. The reduction in wind production in 2016 might be attributed to low wind speeds and unusual weather patters. Furthermore, there were a slowdown in the installation of new wind turbines in 2016 compared to previous years, leading to decrease in wind production. A possible reason for decline in 2019/2020 might be due to Covid-19 pandemic in which halted commercial and industrial activities lead to reduction in electricity demand and the utilization of wind power. However, The wind industry quickly recovered and there were peak production during the first half of 2022. One possible reason for this could be the effect of the war in Ukraine, which prompted high production in wind power to offset the rising cost of natural gas, aiming to alleviate the energy crisis (Fortune, 2022).

From figure 2, it can be observed that there are a clear trend and seasonal pattern in wind power production. To estimate the trend and seasonal components of the data, I will use STL method. STL is an acronym for "Seasonal and Trend decomposition using Loess", while loess is a method for estimating nonlinear relationships. The STL method was developed by R. B. Cleveland et al (1990). The STL() function decomposes a time series into three components: trend, seasonal, and remainder. The trend component represents the long-term direction while the seasonal component illustrates the periodic patterns of the data. The remainder component represents the residuals after removing the trend and seasonal patterns (Hyndman et al., 2021).

```r
#convert data frame to time series object
df_wind_ts = as_tsibble(df_wind, index = date) %>% fill_gaps()
#fill na values with 0
df_wind_ts[is.na(df_wind_ts)] <- 0

#Decomposing data using STL function
df_wind_ts %>% select(date, wind_prod_GWh) %>% model(
  STL(wind_prod_GWh ~ trend(window = 7) + season(window="periodic"))) %>%
  components %>% autoplot()+
  labs(title = "Figure 4") + theme_classic()
```



According to the decomposition components above, it can be observed that there was a downward trend in wind production in 2016 and in period of 2019-2020 as analyzed earlier. In addition, one can observe the peak in wind production in 2022. The seasonality based on year and week patter looks quite even over the period of time. There is no strong structure in the remainder.

The trend and seasonal pattern in wind production data indicates that the data might not be stationary and contain trend and seasonality. In the case of a time-series dataset, it is crucial to determine whether the time series is stationary or not. Stationarity refers to a time series that does not contain any trend or seasonal patterns, meaning that the distribution of the time series remains constant over time. Models applied to stationary time series generally yield more accurate predictions (Quantinsti, 2021). To assess whether the time series is stationary, I will use the statistical test called ADF (.i.e, The Augmented Dickey Fuller Test). The null hypothesis for this test is that there is a unit root, and the alternative hypothesis is that there is no unit root, meaning that data is stationary (Statisticshowto, 2022). Besides the ADF test, I also test for dynamics and serial correlation in time series by looking at autocorrelation functions (ACF) and partial autocorrelation functions (PACF).

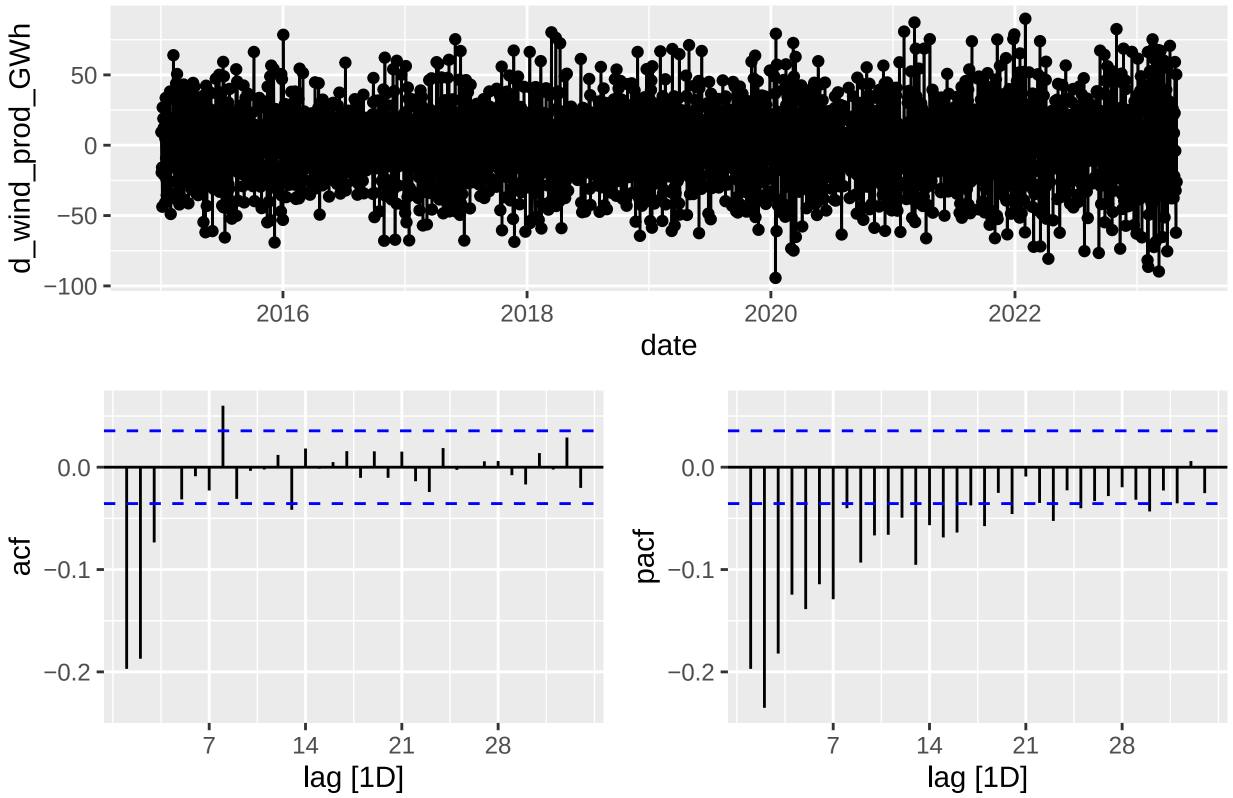
```
adf.test(df_wind_ts$wind_prod_GWh)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df_wind_ts$wind_prod_GWh
## Dickey-Fuller = -11.111, Lag order = 14, p-value = 0.01
## alternative hypothesis: stationary
```

With the p-value of 0.01, I can reject the null hypothesis that the data is not stationary with 99% confidence. However, to ensure better prediction, I still proceed by taking difference on data.

```
#difference the timeseries data
df_wind_ts = df_wind_ts%>% mutate(
  d_wind_prod_GWh = difference(wind_prod_GWh)
)

df_wind_ts = df_wind_ts %>% filter(!is.na(d_wind_prod_GWh))

#testing autocorrelation
df_wind_ts %>% fill_gaps() %>% gg_tsdisplay(d_wind_prod_GWh, plot_type = 'partial') +
  labs(title = "Figure 5")
```

Figure 5



Both the ACF (Auto Correlation Function) and PACF (Partial Auto Correlation Function) plots start with a lag of 1, which is the correlation of the time series with itself. The plots below show the decay pattern in time series data. The PACF shown significant spikes at lag 2, suggesting non-seasonal AR(2) component. The significant spike at lag 7 suggests a seasonal AR(1) component. On the other hand, the ACF is suggestive of significant spikes at lag 1, meaning that MA(1) component can be selected for SARIMA model.

3.2.3 Modeling

In this section, I will build several forecasting models to predict wind power production. The models include SARIMA, Dynamic Harmonic Regression Model, Linear Regression, Random Forest, and Regression with Arima Error. The models will be trained on train set and then evaluated using test set. To assess the performance of the models, two key metrics will be utilized: Mean Squared Error (MSE), Mean Absolute Error (MAE).

- MSE: Mean Squared Error (MSE) is a measure of the difference between a model's predicted and observed values. It is a metric for determining how close the model's predictions are to the observed values (Pishro-Nik, 2022).
- MAE is a metric computed as the mean of absolute errors (the difference between predicted and actual values) (Sammut & Webb, 2010).

Splitting data

As the dataset is a time series, it is necessary to split it into training set and test set with respect to time. The training set is from 01/01/2015 to 2021/04/30 and the test set is from 2021/04/30 to 07/05/2023. By splitting data into training and test set, models can yield better prediction and avoid overfitting. The ratio of train and test set is 3:1.

```
train <- df_wind_ts[df_wind_ts$date < "2021-04-30",]
test <- df_wind_ts[df_wind_ts$date >= "2021-04-30",]
```

SARIMA Model

Seasonal ARIMA (SARIMA) model is an extension of ARIMA model that can be used to forecast time series containing trends and seasonality. The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF (Hyndman et al., 2021). Based on the observation from both PACF and ACF in figure 5, I will model the time series data using seasonal ARIMA model with seasonality parameters specified as follows.

```
arima_mod <- train %>% model(
  arima210110 = ARIMA(wind_prod_GWh ~ pdq(2,1,0) + PDQ(1,1,0)),
  arima011110 = ARIMA(wind_prod_GWh ~ pdq(0,1,1) + PDQ(1,1,0)),
  arima211110 = ARIMA(wind_prod_GWh ~ pdq(2,1,1) + PDQ(1,1,0)),
  auto  = ARIMA(wind_prod_GWh),
  search = ARIMA(wind_prod_GWh, stepwise=FALSE)
)

arima_mod
```

```
## # A mable: 1 x 5
##                 arima210110              arima011110              arima211110
##                     <model>                  <model>                  <model>
## 1 <ARIMA(2,1,0)(1,1,0)[7]> <ARIMA(0,1,1)(1,1,0)[7]> <ARIMA(2,1,1)(1,1,0)[7]>
## # i 2 more variables: auto <model>, search <model>
```

```
# check the best ARIMA model in terms of AICc
glance(arima_mod) %>%
arrange(AICc) %>% select(.model:BIC)
```

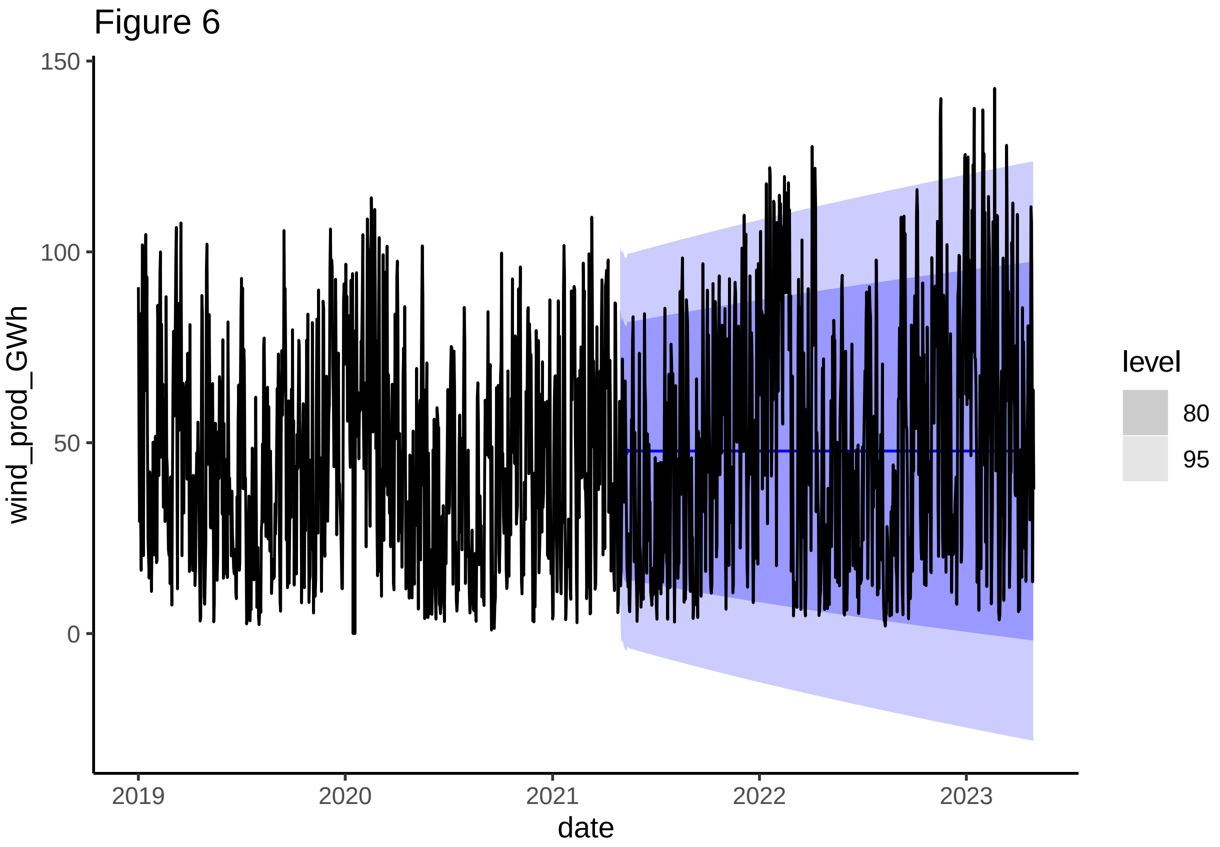
```
## # A tibble: 5 x 6
##   .model      sigma2 log_lik    AIC   AICc    BIC
##   <chr>        <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
## 1 search        484. -10412. 20837. 20837. 20871.
## 2 auto          484. -10414. 20837. 20837. 20866.
## 3 arima211110   707. -10821. 21652. 21652. 21681.
## 4 arima210110   844. -11022. 22051. 22051. 22074.
## 5 arima011110   861. -11046. 22097. 22097. 22114.
```

```
arima_mod %>% select(search) %>% report()
```

```
## Series: wind_prod_GWh
## Model: ARIMA(1,1,2)(2,0,0)[7]
##
## Coefficients:
##          ar1      ma1      ma2     sar1    sar2
##       0.3845  -0.8154  -0.1547  -0.0303  0.0217
## s.e.  0.0407   0.0433   0.0404   0.0214  0.0213
##
## sigma^2 estimated as 484.1:  log likelihood=-10412.39
## AIC=20836.79   AICc=20836.82   BIC=20871.26
```

According to table above, all five SARIMA models have similar AICc values. Among these models, the ARIMA(1,1,2)(0,0,1) which was suggested by the full search demonstrates the best fit based on AICc. Conversely, the third guess model, ARIMA(0,1,1)(1,1,0), was the worst fit. Therefore, I will choose the `ARIMA(1,1,2)(0,0,1)` model to forecast on the test set.

```
forecast_arima <- arima_mod %>% select(search) %>% forecast(h = "2 years")

forecast_arima %>% autoplot(df_wind_ts %>% filter(date >= "2019-01-01")) +
labs(title = "Figure 6") +
theme_classic()
```



```
pred1 <- forecast_arima$.mean
mse1 <- mean((pred1 - test$wind_prod_GWh)^2)
mae1 <- mean(abs(pred1 - test$wind_prod_GWh))
cat("Mean Absolute Error (MAE):", mae1, "\n")
```

```
## Mean Absolute Error (MAE): 27.56636
```

```
cat("Mean Squared Error (MSE):", mse1, "\n")
```

```
## Mean Squared Error (MSE): 1075.218
```

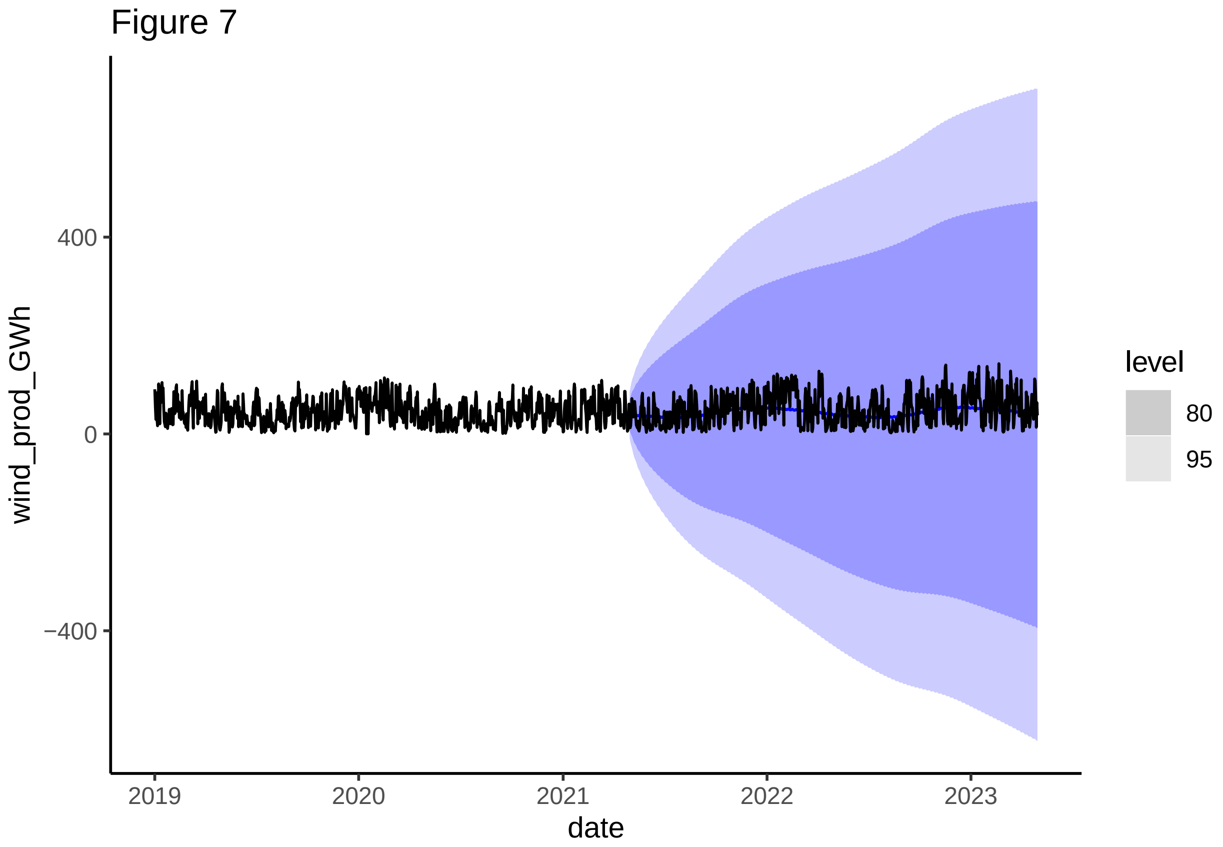
The SARIMA model does not seem to capture the variation in wind power production. The model's mean absolute error and mean square error are 27.57 and 1075.22 respectively.

Dynamic harmonic regression model

The Dynamic harmonic regression model is a time series forecasting model that incorporates harmonic components to capture periodic patterns in the data. The harmonic structure is built of Fourier terms that consist of sine of cosine terms to form a periodic function. This approach is often better than other forecasting models when there are long seasonal periods (Hyndman et al., 2021)

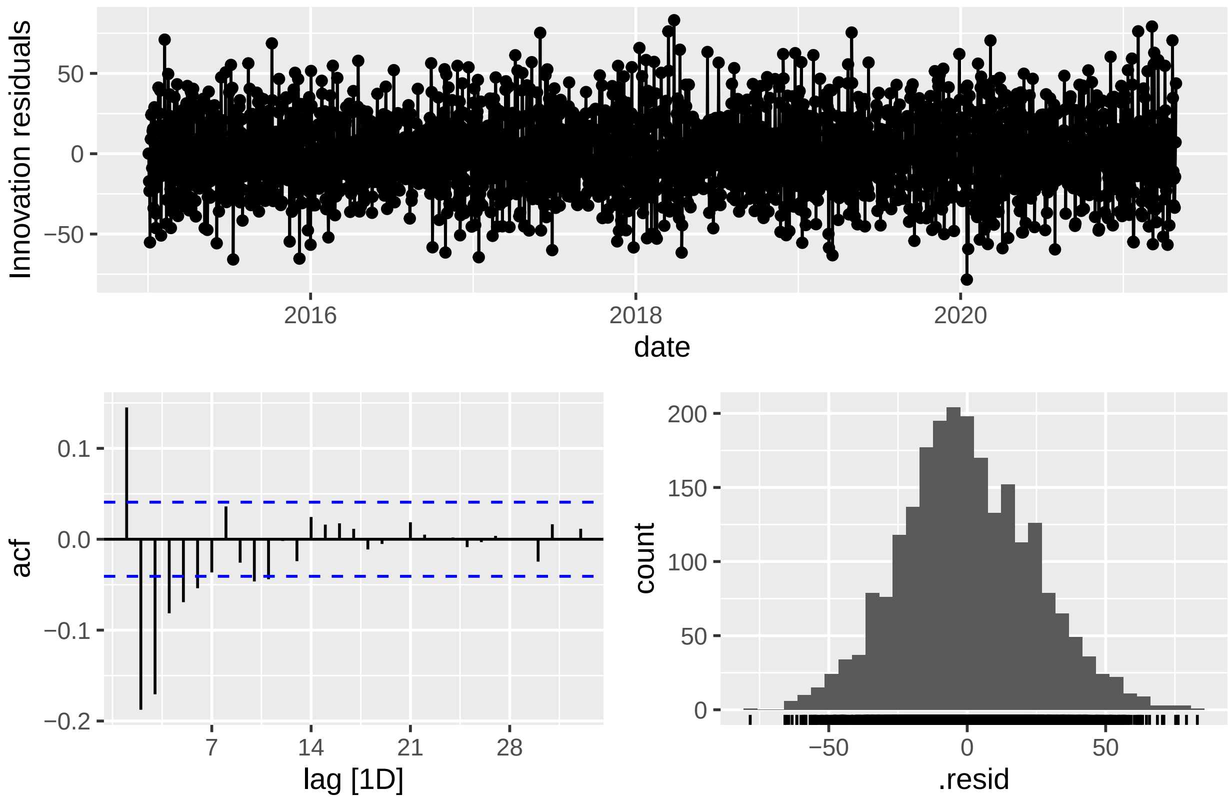
Based on the observation from Figure 4, it is evident that wind production has two distinct seasonality: weekly and yearly. To capture for these seasonal patterns, I will incorporate Fourier terms for capturing seasonality with ARIMA errors.

```
fourier_mod <- train %>% model(fourier = ARIMA(wind_prod_GWh ~ fourier(period = "week", K = 1) +
                                              fourier(period = "year", K = 3) + PDQ(0,0,0)))

forecast2 = fourier_mod %>% select(fourier) %>% forecast(new_data = test)
forecast2 %>% autoplot(df_wind_ts %>% filter(date >= "2019-01-01")) +
labs(title = "Figure 7") +
theme_classic()
```



```
fourier_mod %>% gg_tsresiduals() + labs(title = "Figure 8")
```

Figure 8



Although the short-term forecasts appear to be reasonable, it is important to note that there are still several spikes in ACF plot. These spikes indicate that the model may not fully capture the all the underlying patterns and dynamics of the data. Besides, the residual plot is slight skewed, indicating that there is many information that have not been captured by this model.

```
pred2 <- forecast2$.mean
mse2 <- mean((pred2 - test$wind_prod_GWh)^2)
mae2 <- mean(abs(pred2 - test$wind_prod_GWh))
cat("Mean Absolute Error (MAE):", mae2, "\n")
```

```
## Mean Absolute Error (MAE): 26.09309
```

```
cat("Mean Squared Error (MSE):", mse2, "\n")
```

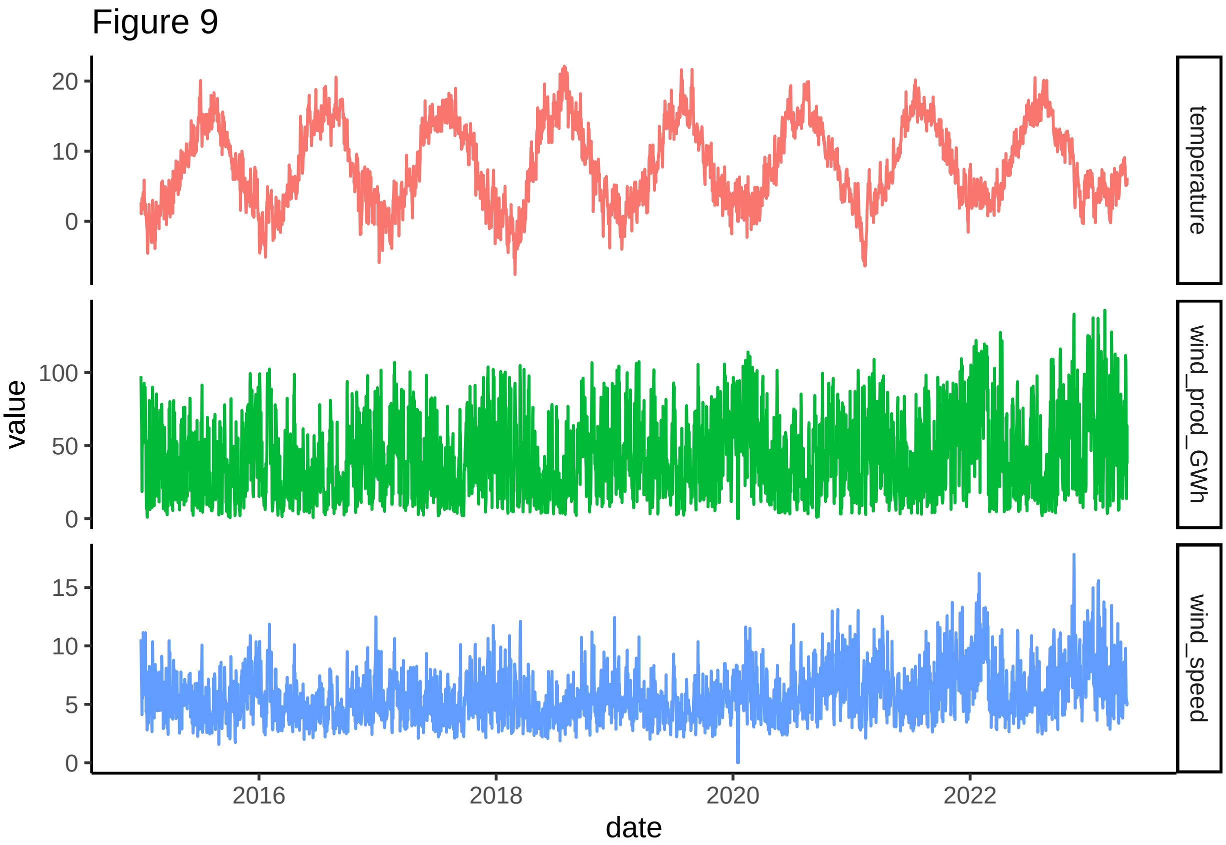
```
## Mean Squared Error (MSE): 993.9125
```

It can be seen that the dynamic harmonic regression model has better performance than the SARIMA model with mean absolute error and mean square error are 26.09 and 993.91 respectively.

Linear Regression Model

A linear regression model is a statistical approach used to establish relationship between a dependent variable and other independent variables. It assumes a linear relationship between the variables. As discussed earlier, weather variables including wind speed and temperature might have significant relationship with wind power production. Therefore, I will model these relationships using linear regression.

```
df_wind_ts %>%
select(date, wind_prod_GWh, temperature, wind_speed) %>%
pivot_longer(-date, names_to = "variable", values_to = "value") %>%
ggplot(aes(x = date, y = value, color = variable)) +
geom_line(show.legend = FALSE) +
facet_grid(vars(variable), scales = "free_y") +
labs(title = "Figure 9") + theme_classic()
```



According to Figure 9, it can be observed that the wind production has similar pattern with wind speed, showing an increase when wind speed increases. Conversely, it appears to have an opposite pattern with temperature. When wind production increases, the temperature decreases. Additionally, wind power production tends to peak in the winter season. This might be attributed to high demand for home and business heating during this period as well as stronger wind resource. On the other hand, wind power production seems to be lower in the summer, and wind power production fluctuates during the week. To further investigate the relationship of these season variables with wind power production, I will create new dummy variables named `winter`, `summer`, and `weekday`. Besides, I also incorporate lagged weather variables, lagged wind production variables, interaction terms between winter/summer and temperature/wind speed in the model to explore the relationship between these variables and wind power production in more detail.

```
#Create dummy variables for winter, summer
df_wind_ts$winter <- as.factor(ifelse(month(df_wind_ts$date) %in% c(12,1,2), 1, 0))
df_wind_ts$summer <- as.factor(ifelse(month(df_wind_ts$date) %in% c(6,7,8), 1, 0))
df_wind_ts$weekday <- as.factor(weekdays(df_wind_ts$date))

#create lagged weather variables
df_wind_ts$wind_prod_GWh_lag1 <- lag(df_wind_ts$wind_prod_GWh, 1)
```

```
df_wind_ts$temperature_lag1 <- lag(df_wind_ts$temperature, 1)
df_wind_ts$wind_speed_lag1 <- lag(df_wind_ts$wind_speed, 1)

df_wind_ts$wind_prod_GWh_lag2 <- lag(df_wind_ts$wind_prod_GWh, 2)
df_wind_ts$temperature_lag2 <- lag(df_wind_ts$temperature, 2)
df_wind_ts$wind_speed_lag2 <- lag(df_wind_ts$wind_speed, 2)

df_wind_ts$wind_prod_GWh_lag3 <- lag(df_wind_ts$wind_prod_GWh,3)
df_wind_ts$temperature_lag3 <- lag(df_wind_ts$temperature, 3)
df_wind_ts$wind_speed_lag3 <- lag(df_wind_ts$wind_speed, 3)


#remove na values
df_wind_ts <- df_wind_ts %>% na.omit()
```

```
#split data in train and test set
train <- df_wind_ts[df_wind_ts$date < "2021-04-30",]
test <- df_wind_ts[df_wind_ts$date >= "2021-04-30",]
```

```
lin_mod1 <- lm(wind_prod_GWh~ temperature + wind_speed + winter+summer+weekday
               + wind_prod_GWh_lag1 + wind_prod_GWh_lag2 + wind_prod_GWh_lag3
               + temperature_lag1 + temperature_lag2 + temperature_lag3
               + wind_speed_lag1 +  wind_speed_lag2 + wind_speed_lag3
               + winter*wind_speed + summer*wind_speed + winter*temperature
               + summer*temperature, data = train)
summary(lin_mod1)
```

```
##
## Call:
## lm(formula = wind_prod_GWh ~ temperature + wind_speed + winter +
##     summer + weekday + wind_prod_GWh_lag1 + wind_prod_GWh_lag2 +
##     wind_prod_GWh_lag3 + temperature_lag1 + temperature_lag2 +
##     temperature_lag3 + wind_speed_lag1 + wind_speed_lag2 + wind_speed_lag3 +
##     winter * wind_speed + summer * wind_speed + winter * temperature +
##     summer * temperature, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.903  -6.876  -0.265   7.346  45.187
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -7.08620    1.59002  -4.457 8.73e-06 ***
## temperature            0.42190    0.16122   2.617  0.00893 **
## wind_speed            11.36655    0.19478  58.356  < 2e-16 ***
## winter1                3.72325    1.89422   1.966  0.04947 *
## summer1                9.55555    4.24072   2.253  0.02434 *
## weekdayMonday          0.71966    0.90444   0.796  0.42629
## weekdaySaturday        1.41798    0.90406   1.568  0.11691
## weekdaySunday          0.39066    0.90481   0.432  0.66595
## weekdayThursday        0.50140    0.90316   0.555  0.57884
## weekdayTuesday         0.64720    0.90369   0.716  0.47395
## weekdayWednesday       1.28652    0.90302   1.425  0.15438
## wind_prod_GWh_lag1   0.34041    0.02084  16.335  < 2e-16 ***
## wind_prod_GWh_lag2   0.09891    0.02203   4.490 7.48e-06 ***
## wind_prod_GWh_lag3   0.15542    0.02083   7.462 1.20e-13 ***
## temperature_lag1    -0.13056    0.17518  -0.745  0.45617    
## temperature_lag2    -0.02253    0.17541  -0.128  0.89782    
## temperature_lag3    -0.29829    0.14509  -2.056  0.03991 *  
## wind_speed_lag1     -3.31363    0.27638 -11.989  < 2e-16 ***
## wind_speed_lag2     -1.79105    0.28254  -6.339 2.78e-10 ***
## wind_speed_lag3     -2.04439    0.27362  -7.472 1.12e-13 ***
## wind_speed:winter1  -0.57380    0.29702  -1.932  0.05351 .  
## wind_speed:summer1  -0.27672    0.36405  -0.760  0.44726    
## temperature:winter1  0.34742    0.20501   1.695  0.09028 .  
## temperature:summer1 -0.61346    0.23729  -2.585  0.00979 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 11.58 on 2283 degrees of freedom
## Multiple R-squared:  0.8092, Adjusted R-squared:  0.8072 
## F-statistic: 420.9 on 23 and 2283 DF,  p-value: < 2.2e-16
```

From the summary table above, it can be seen that both weather variables (.i.e, temperature and wind speed) and season variables (.i.e, winter and summer) have a significant relationship with wind power production, with p-value below 0.05. The interaction terms between winter/summer and wind speed/temperature appear to have significant relationship with wind power production, with the exception for interaction between summer and wind speed. Furthermore, the historical wind power production, wind speed, and temperature demonstrate significant effects on wind power production, except for `temperature_lag1` and `temperature_lag2`. On the other hand, the days of week are not statistically significant on wind power generation with p-value above 0.05.

Temperature has positive effects on wind power at 99% level. An increase of 1 degree celsius on temperature corresponds to a 0.42 GWh increase in wind production. The coefficent of wind speed is notably larger than that of temperature with value of 11.37 and p-value below 0.001, indicating that wind speed is significant predictor of wind power production. An 1 m/s increase in wind speed leads to an increase of 11.23 GWh on wind power production. The effect of variable `winter` on wind power production is less than that of variable `summer`. The wind power production is expected to have an additional 3.73 GWh on the winter days compared to non-winter days. Similarly, wind power is expected to increase by 9.55 GWh on summer day than non-summer day. For the interaction terms between wind speed and winter, the coefficient can be interpreted that the effect on wind power production of 1 m/s increase in wind speed is 0.57 GWh less on winter days than on non-winter days. The coefficient of the interaction terms between between summer and temperature indicates that, for summer days versus non-summer days, the effect on wind power generation of 1 celsius degree increase in temperature is reduced by 0.61 GWh. All histotical wind power production variables show positive relationships with current wind power production while lag variables of temperature and wind speed have negative impact. For example, an increase in wind speed on past 1 day lead to a decrease of 3.29 GWh on wind power production. The linear model has R-square above 81%, indicating that the predictors explain 81% the variation of wind power production.

Next, I will eliminate these insignificant variables and re-estimate the model to make prediction. The performance of model will be assess based on evaluation metric.

```
lin_mod2 <- lm(wind_prod_GWh~ temperature + wind_speed + winter + summer
               + wind_prod_GWh_lag1 + wind_prod_GWh_lag2+wind_prod_GWh_lag3
               + temperature_lag3+ wind_speed_lag1 + wind_speed_lag2
               + winter*wind_speed + winter*temperature
               + wind_speed_lag3 + summer*temperature, data = train)
```

```
pred3 <- predict(lin_mod2, newdata = test)
mse3 <- mean((pred3 - test$wind_prod_GWh)^2)
mae3 <- mean(abs(pred3 - test$wind_prod_GWh))
cat("Mean Absolute Error (MAE):", mae3, "\n")
```

```
## Mean Absolute Error (MAE): 12.75022
```

```
cat("Mean Squared Error (MSE):", mse3, "\n")
```

```
## Mean Squared Error (MSE): 267.2978
```

It is evident that the linear regression have much better performance compared to previous models with Mean Absolute Error of 12.75 and Mean Squared Error of 267.29.

Random Forest

The Random Forest algorithm is a supervise machine learning model comprised of multiple trees. By growing a number of decision trees on different subsets of the dataset, and then averaging their predictions, the Random Forest model can enhance prediction accuracy and mitigate the risk of overfitting. In this project, I will built random forest model to predict wind power production.

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

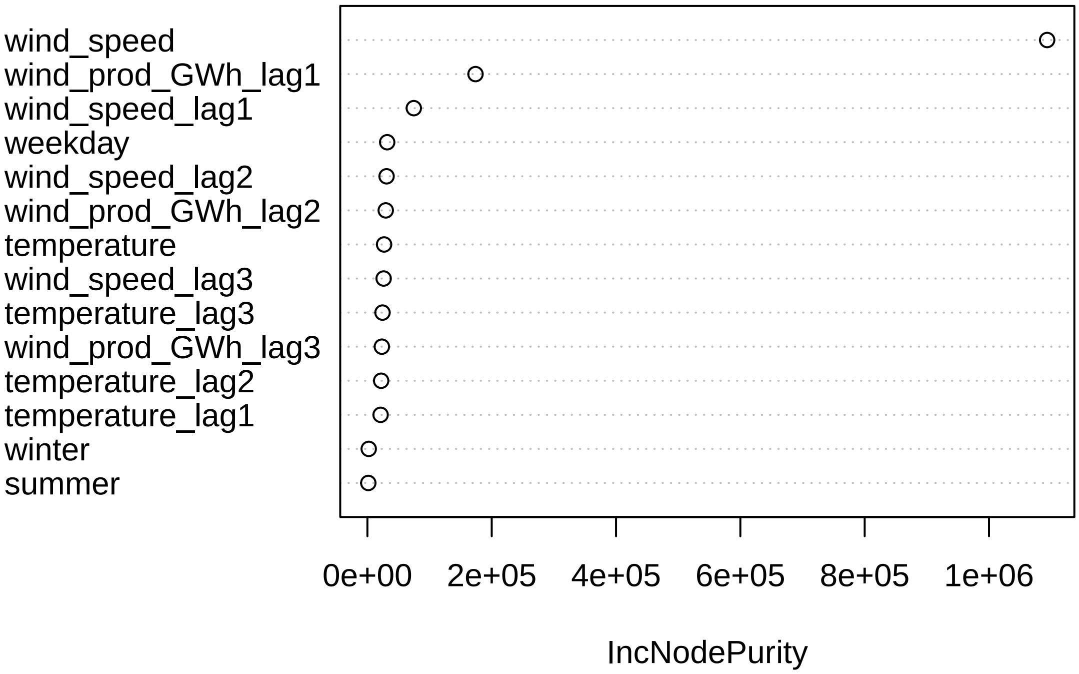
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
RF_mod = randomForest(wind_prod_GWh~ temperature + wind_speed + winter+summer+weekday
               + wind_prod_GWh_lag1 + wind_prod_GWh_lag2 + wind_prod_GWh_lag3
               + temperature_lag1 + temperature_lag2 + temperature_lag3
               + wind_speed_lag1 +  wind_speed_lag2 + wind_speed_lag3,
               data = train, mtry = 8, ntree = 100)
```

```
varImpPlot(RF_mod, main = "Figure 10")
```

Figure 10



In Random Forest model, the variable importance can be measured. One way to measure it by using the incNodepurity which quantifies the total decrease in node impurity that results from splits over that variable. The metric is measured by using training RSS (residual sum of squares). As can be seen, the variables `wind_speed` and `wind_prod_GWh_lag1` appear to be two most important predictors of wind power production accross all of the trees considered in the random forest model.

```
pred4 = predict(RF_mod, newdata = test)
mse4 <- mean((pred4 - test$wind_prod_GWh)^2)
mae4 <- mean(abs(pred4 - test$wind_prod_GWh))
cat("Mean Absolute Error (MAE):", mae4, "\n")
```

```
## Mean Absolute Error (MAE): 14.3203
```

```
cat("Mean Squared Error (MSE):", mse4, "\n")
```

```
## Mean Squared Error (MSE): 329.8428
```

It can be observed that the Random Forest performs worse than the linear regression model with mean absolute error of 14.22 and mean square error of 325.23.

Regression with ARIMA errors

Regression with ARIMA errors is a statistical modeling approach that combines the regression analysis and ARIMA modeling. In a regression with ARIMA errors model, the dependent variable is modeled as a function of both exogenous predictors and error of ARIMA model. The advantage of this model with

ARIMAX model is that coefficients of predictors can be interpreted the same way as normal regression. The ARIMA() function can also be used to select the best ARIMA model for the errors (Hyndman et al., 2021).

```
arima_er <- train %>% model(
  ARIMA(wind_prod_GWh ~ temperature + wind_speed
               + temperature_lag1 + temperature_lag2 + temperature_lag3
               + wind_speed_lag1 +  wind_speed_lag2 + wind_speed_lag3
                   )
)
arima_er
```

```
## # A mable: 1 x 1
##   ARIMA(wind_prod_GWh ~ temperature + wind_speed + temperature_lag1 + \n    te~1
##                                                                        <model>
## 1                                          <LM w/ ARIMA(1,1,1)(2,0,0)[7] errors>
## # i abbreviated name:
## #   1: ‘ARIMA(wind_prod_GWh ~ temperature + wind_speed + temperature_lag1 + \n    temperature_lag2 +
```

The fitted model has an ARIMA (2,1,1)(1,0,2)[7] error. I will predict the wind power production using test set.

```
pred5 = arima_er %>% forecast(new_data = test)
pred5 %>% autoplot(df_wind_ts %>% filter(date >= "2019-01-01")) +
  labs(title = "Figure 11") + theme_classic()
```

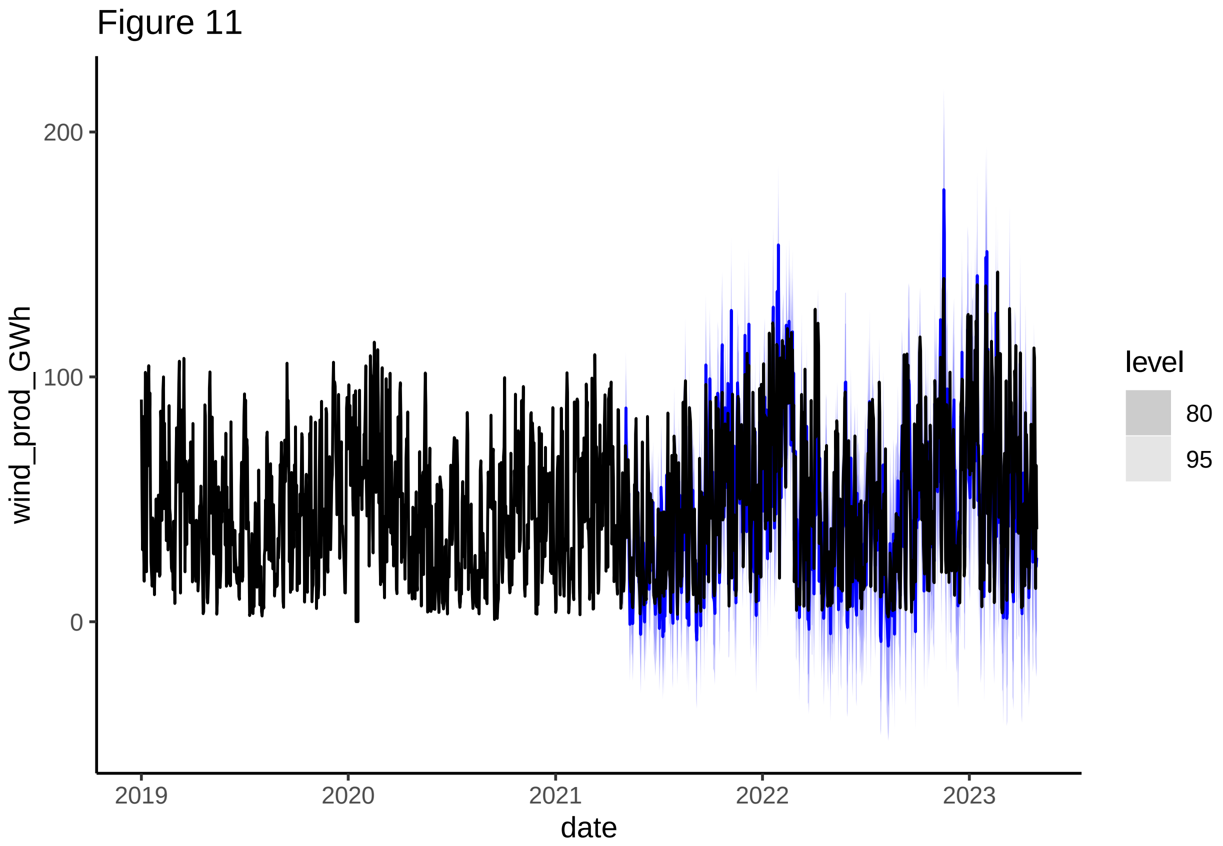


Figure 11

```
pred5 <- pred5$.mean
mse5 <- mean((pred5 - test$wind_prod_GWh)^2)
mae5 <- mean(abs(pred5 - test$wind_prod_GWh))
cat("Mean Absolute Error (MAE):", mae5, "\n")
```

```
## Mean Absolute Error (MAE): 13.54279
```

```
cat("Mean Squared Error (MSE):", mse5, "\n")
```

```
## Mean Squared Error (MSE): 299.0722
```

From Figure 11, it appears that the regression with ARIMA errors model performs quite well in predicting wind power production. However, its performance based on evaluation metric is still a bit worse than that of linear regression model. The regression with ARIMA errors model yields Mean Absolute Error of 13.54 and Mean Squared Error of 299.07.

```
data.frame(
  Model = c("SARIMA", "Dynamic harmonic regression", "Linear Regression", "Random Forest", "Regression w
  MAE = c(mae1, mae2, mae3, mae4, mae5),
  MSE = c(mse1, mse2, mse3, mse4, mse5)
)
```

```
##                           Model      MAE       MSE
## 1                        SARIMA 27.56636 1075.2182
## 2  Dynamic harmonic regression 26.09309  993.9125
## 3            Linear Regression 12.75022  267.2978
## 4                Random Forest 14.32030  329.8428
## 5 Regression with ARIMA errors 13.54279  299.0722
```

The table above clearly shows that linear regression model demonstrates the best performance in predicting wind power production among 5 five models. Following closely behind is the Regression with ARIMA errors model. This indicates that both of these models are suitable for predicting wind power production in the future. In the next section, I will utilize the Regression with ARIMA errors model to forecast the wind power production based on Representative Concentration Pathways (RCP) scenarios as this model has proven to be robust for long-term prediction.

Forecasting wind power production in 2040

Representative Concentration Pathways (RCP) scenarios are a set of potential future greenhouse gas concentration trajectories used in climate change modeling and research. The pathways describe different climate futures, all of which are considered possible depending on the volume of greenhouse gases emitted (Wikipedia, 2023). In this part, I will forecast wind power production in Denmark until 2040 under three of the climate change scenarios selected from IPCC's work including a low (RCP2.6), a medium-high (RCP4.5) and a high (RCP8.5) discharge scenario (DMI, 2022). The RCP 2.6 is the very stringent pathway which requires that carbon dioxide emissions need to go to zero by 2100. RCP 4.5 can be considered as an intermediate scenario where emissions in RCP4.5 peak around 2040, then decline. In RCP 8.5, it is assumed that the greenhouse gas emissions continue to rise until 2100, resulting in worst case climate change scenarios (Wikipedia, 2023)

According to Danish Meteorological Institute (DMI), median temperature is expected to be 9.15 degrees Celsius, 9.37 degrees Celsius, 9.36 degrees Celsius during 2011 - 2040 under RCP2.6, RCP4.5, and RCP8.5

respectively. On the other hand, the median wind speed is anticipated to be 4.85 m/s, 4.87 m/s, and 4.88 m/s in the years 2011 - 2040 under RCP2.6, RCP4.5, and RCP8.5 scenarios, respectively.

Due to the absence of data on the predicted value of temperature and wind speed at the daily level under RCP scenarios, I decided to utilize median yearly value to present the daily temperature and wind speed in the future. This approach, however, might introduce bias into prediction, as the use of median values might not capture the full variability and potential extreme events associated with temperature and wind speed. Although the use of median values may not capture the fluctuations in temperature and wind speed, it might provide a general overview of the expected trends in wind power production under the impact of climate change.

In the next steps, I will simulate the future data of the temperature and wind speed based on the median value provided by these RCP scenarios and forecast wind power production as follows:

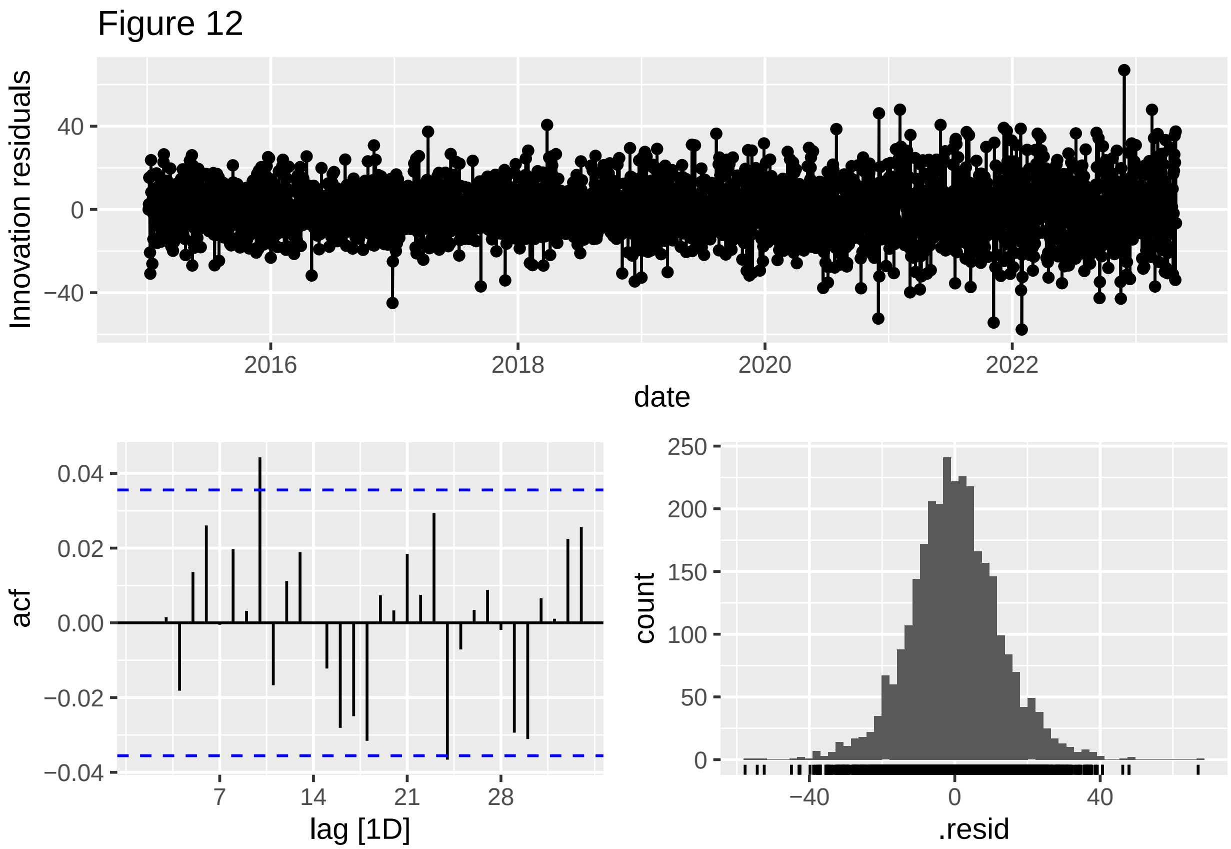
```r
#scenario 1
RCP2_6 = new_data(df_wind_ts, 6455) %>% mutate(
  wind_speed = rep(4.85,6455),
  temperature = rep(9.15,6455) ,
  temperature_lag1 =lag(temperature, 1),
  temperature_lag2 =lag(temperature, 2),
  temperature_lag3 =lag(temperature, 3),
  wind_speed_lag1  = lag(wind_speed, 1),
  wind_speed_lag2  = lag(wind_speed, 2),
  wind_speed_lag3  = lag(wind_speed, 3),
)

#scenario 2
RCP4_5 = new_data(df_wind_ts, 6455) %>% mutate(
  wind_speed = rep(4.87,6455),
  temperature = rep(9.37,6455),
  temperature_lag1 =lag(temperature, 1),
  temperature_lag2 =lag(temperature, 2),
  temperature_lag3 =lag(temperature, 3),
  wind_speed_lag1  = lag(wind_speed, 1),
  wind_speed_lag2  = lag(wind_speed, 2),
  wind_speed_lag3  = lag(wind_speed, 3),
)

#scenario 3
RCP8_5 = new_data(df_wind_ts, 6455) %>% mutate(
  wind_speed = rep(4.88,6455),
  temperature = rep(9.36,6455),
  temperature_lag1 =lag(temperature, 1),
  temperature_lag2 =lag(temperature, 2),
  temperature_lag3 =lag(temperature, 3),
  wind_speed_lag1  = lag(wind_speed, 1),
  wind_speed_lag2  = lag(wind_speed, 2),
  wind_speed_lag3  = lag(wind_speed, 3),
)
```

The Regression with ARIMA errors is fitted with the whole data set, so it can captures the full range of dynamics such as patterns, trends, and seasonality that may exist within the data.

```
#fitting Regression with ARIMA errors with the whole dataset
arima_er <- df_wind_ts %>% model(
  ARIMA(wind_prod_GWh ~ temperature + wind_speed
               + temperature_lag1 + temperature_lag2 + temperature_lag3
               + wind_speed_lag1 +  wind_speed_lag2 + wind_speed_lag3
                   )
)
```

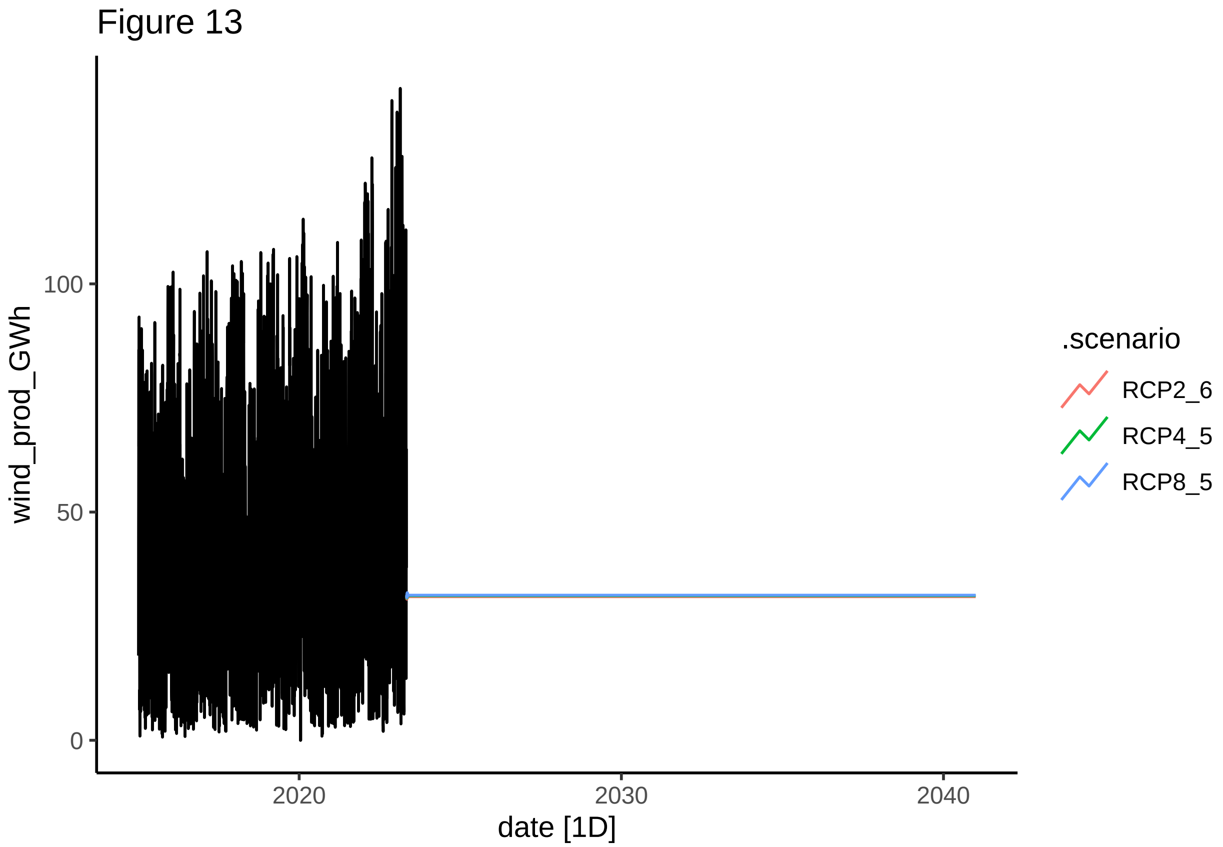
The residuals of Regression with ARIMA errors model can be checked by using gg_tsresiduals() function.

```
arima_er %>% gg_tsresiduals() + labs(title = "Figure 12")
```



From Figure 12, it can be observed that the estimated ARIMA errors are not significantly different from white noise. The residual plot shows the predictions are quite normally distributed. Then, I will proceed to forecast wind power production based on each scenario of RCP.

```
#forecast the wind power production with each scenario
RCP_scenario <- scenarios(RCP2_6 = RCP2_6, RCP4_5 = RCP4_5, RCP8_5 = RCP8_5)
pred6 = arima_er %>% forecast(new_data = RCP_scenario)
df_wind_ts %>% autoplot(wind_prod_GWh) + autolayer(pred6, level = NULL) +
labs(title = "Figure 13") + theme_classic()
```



The Figure 13 shows that the decreasing trend in wind power production toward 2040. However, due to the data being at daily level, it becomes challenging to examine the trend for each scenario.

IV. CONCLUSION & LIMITATIONS

After conducting analysis and implementing several forecasting models, I identified that the weather variables, specifically wind speed and temperature, have significant relationship with the wind power production. This finding aligns with my initial hypothesis. Additionally, the climate change under RCP scenarios is expected to a have negative impact on production of wind power in the future.

However, it is important to acknowledge several limitations in this report. Firstly, the data utilized for the analysis only covers the time period from 2015 to 2023. The lack of historical data poses challenges for the forecasting models to fully capture all dynamics and complexity of the time series. Secondly, analyzing data at daily level might introduce some complications related to seasonality. Aggregating the data to weekly or monthly level could provide better understanding of the underlying seasonality and trend. Furthermore, the prediction mainly relied on weather variables and historical wind power production variables. It is worth noting that there might be other variables that could serve as promising predictors of wind power production such as wind capacity investment, electricity price, etc. Inclusion of other promising variables might enhance the accuracy and robustness of forecasting models.