University of Connecticut

OPIM 5671 – Data Mining and Business Intelligence

Solar Power Generation Forecasting

Project Report

Group 4

Pranay Babu Chinnam

Srihari Madhavan

Moulika Narne

Pavani Bodasakurthi

Priyanka Kankipati

Contents

Introduction

Solar power production and consumption in India have witnessed remarkable growth and transformation in recent years. With a total solar power generation capacity exceeding 35 gigawatts (GW) as of September 2020, India ranks among the world's largest solar power producers. Government initiatives, incentives, and large-scale solar parks have fueled this capacity expansion. India's commitment to solar energy aligns with its sustainable development goals, providing electricity to remote areas and reducing costs for industries. India had numerous solar power plants and installations across the country, with a wide range of capacities and sizes. India's abundant sunshine, low cloud cover, and diverse geography create favorable conditions for solar production year-round. Motivated by this, we have gathered data from a solar power facility located in near Mumbai, India, to conduct an in-depth analysis of solar production based on the weather conditions.

Problem Statement

The objective of this project is to develop an accurate and reliable time series forecasting model for the solar power generation of a solar plant, specifically focusing on the daily power generation. This forecasting model will utilize historical solar power generation data in conjunction with concurrent weather sensor data, including ambient temperature, module temperature, and irradiation. In this project, through the forecasting, we will try to forecast the power generated, analyze components such as trend, seasonality and cycles of occurrence and the occurrence in different times of the week, using models like exponential smoothing, and, ARIMAX. We aim to test the cross correlation and causation of various weather factors on solar power generation.

Dataset Description

The dataset contains information related to approximately 1 month performance and output of a solar power plant captured over 15-minute intervals, including various attributes such as date and time stamps, weather conditions, power generation readings, and possibly other relevant data points. Analyzing this dataset can help users gain insights into the efficiency and reliability of solar power generation under different weather conditions and times of the day.

Data Exploration

To perform detailed exploration and forecasting of the data, we first analyzed the raw dataset. We have extracted the base data from Kaggle, which contains the details about solar power generation from a plant located near Mumbai, India.

The overview of the same is in the snapshot and description below:

a. Plant_2_Generation_Data.csv:

   This dataset contains the solar power generation data for one plant gathered at 15 minutes intervals over a 34 days period, and has the following variables:

   DATE_TIME : Date and time for each observation. Observations recorded at 15 minute intervals.

   PLANT_ID : Plant ID - this will be common for the entire file.

   SOURCE_KEY : Source key in this file stands for the inverter id.

   DC_POWER : Amount of DC power generated by the inverter (source_key) in this 15 minute interval. Units - kW.

   AC_POWER : Amount of AC power generated by the inverter (source_key) in this 15 minute interval. Units - kW.

   DAILY_YIELD : Daily yield is a cumulative sum of power generated on that day, till that point in time.

   TOTAL_YIELD : This is the total yield for the inverter till that point in time.

DATE_TIME	PLANT_ID	SOURCE_KEY	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD
2020-05-15 0:00:00	4136001	4UPUqMRk7TRMgm	0	0	9425	2429011
2020-05-15 0:00:00	4136001	81aHJ1q11NBPMrL	0	0	0	1215278736
2020-05-15 0:00:00	4136001	9kRcWv60rDACzjR	0	0	3075.333333	2247719577
2020-05-15 0:00:00	4136001	Et9kgGMDl729KT4	0	0	269.9333333	1704250
2020-05-15 0:00:00	4136001	IQ2d7wF4YD8zU1Q	0	0	3177	19941526
2020-05-15 0:00:00	4136001	LYwnQax7tkwH5Cb	0	0	1872.5	1794958634
2020-05-15 0:00:00	4136001	LlT2YUhhzqhg5Sw	0	0	1094.357143	282592810
2020-05-15 0:00:00	4136001	Mx2yZCDsyf6DPfv	0	0	5692.2	2453646
2020-05-15 0:00:00	4136001	NgDl19wMapZy17u	0	0	1866.2	111512591
2020-05-15 0:00:00	4136001	PeE6FRyGXUgsRhN	0	0	651.2	1348350801
2020-05-15 0:00:00	4136001	Qf4GUc1pJu5T6c6	0	0	0	838421377
2020-05-15 0:00:00	4136001	Quc1TzYxW2pYoW	0	0	5495	329509085

b. Plant_2_Weather_Sensor_Data.csv

This dataset contains the weather sensor data gathered for one solar plant every 15 minutes over a 34-day period, and has the following column variables:

DATE_TIME: Date and time for each observation. Observations recorded at 15-minute intervals.

PLANT_ID: Plant ID - this will be common for the entire file.

SOURCE_KEY: Stands for the sensor panel id. This will be common for the entire file because there's only one sensor panel for the plant.

AMBIENT_TEMPERATURE: This is the ambient temperature at the plant.

MODULE_TEMPERATURE: There's a module (solar panel) attached to the sensor panel. This is the temperature reading for that module.

IRRADIATION: Amount of irradiation for the 15-minute interval.

DATE_TIME	PLANT_ID	SOURCE_KEY	AMBIENT_TEMPER	MODULE_TEMPERA	IRRADIATION
2020-05-15 0:00:00	4136001	iq8k7ZNt4Mwm3w0	27.0047637	25.0607889	0
2020-05-15 0:15:00	4136001	iq8k7ZNt4Mwm3w0	26.88081143	24.42186883	0
2020-05-15 0:30:00	4136001	iq8k7ZNt4Mwm3w0	26.68205534	24.42729031	0
2020-05-15 0:45:00	4136001	iq8k7ZNt4Mwm3w0	26.5005889	24.4206776	0
2020-05-15 1:00:00	4136001	iq8k7ZNt4Mwm3w0	26.596148	25.08821041	0
2020-05-15 1:15:00	4136001	iq8k7ZNt4Mwm3w0	26.51274003	25.31796967	0
2020-05-15 1:30:00	4136001	iq8k7ZNt4Mwm3w0	26.49433897	25.21719253	0
2020-05-15 1:45:00	4136001	iq8k7ZNt4Mwm3w0	26.42041021	25.06506231	0
2020-05-15 2:00:00	4136001	iq8k7ZNt4Mwm3w0	26.40194613	24.69146937	0
2020-05-15 2:15:00	4136001	iq8k7ZNt4Mwm3w0	26.22607821	24.55948079	0
2020-05-15 2:30:00	4136001	iq8k7ZNt4Mwm3w0	26.2603992	24.48240663	0
2020-05-15 2:45:00	4136001	iq8k7ZNt4Mwm3w0	26.2978261	24.4886982	0
2020-05-15 3:00:00	4136001	iq8k7ZNt4Mwm3w0	26.3282491	24.50628962	0
2020-05-15 3:15:00	4136001	iq8k7ZNt4Mwm3w0	26.15870973	24.59519743	0
2020-05-15 3:30:00	4136001	iq8k7ZNt4Mwm3w0	26.07849933	24.80131437	0
2020-05-15 3:45:00	4136001	iq8k7ZNt4Mwm3w0	26.00130397	24.60781745	0

c. Mumbai_temp.csv

This dataset is a timeseries dataset that we as a team generated to be able to forecast the solar power generation more accurately. We extracted the weather conditions from the daily weather reports at Mumbai over the same period of time to help us with the same. The dataset contains the following variable columns :

Temperature : temperature measure in Fahrenheit

Dew Point : dewpoint measure in Fahrenheit

Humidity : the percent humidity at that point in time

Wind : direction of wind – categorical variable

Wind Speed : the wind speed in mph

Wind Gust : the measure of wind gust in mph

Pressure : the measure of mercury rise in inches while calculating the pressure

Precip. : the measure of precipitation in inches

Condition : the condition definition of that area – which is a categorical variable

DateTime : Date time stamp of when the measurement was taken- data available for every 30 minutes

Temperature	Dew Point	Humidity	Wind	Wind Speed	Wind Gust	Pressure	Precip.	Condition	DateTime
86 °F	79 °F	79 %	NNE	5 mph	0 mph	29.70 in	0.0 in	Haze	5/15/2020 00:00
86 °F	77 °F	74 %	N	5 mph	0 mph	29.70 in	0.0 in	Haze	5/15/2020 00:30
86 °F	77 °F	74 %	NNW	6 mph	0 mph	29.70 in	0.0 in	Haze	5/15/2020 01:00
86 °F	77 °F	74 %	NNW	6 mph	0 mph	29.70 in	0.0 in	Haze	5/15/2020 01:30
86 °F	77 °F	74 %	N	6 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 02:00
88 °F	77 °F	70 %	NNW	5 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 02:30
90 °F	75 °F	62 %	NNW	5 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 03:00
91 °F	75 °F	59 %	VAR	3 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 03:30
91 °F	75 °F	59 %	CALM	0 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 04:00
91 °F	75 °F	59 %	W	9 mph	0 mph	29.73 in	0.0 in	Haze	5/15/2020 04:30
91 °F	75 °F	59 %	WNW	8 mph	0 mph	29.73 in	0.0 in	Smoke	5/15/2020 05:00
91 °F	77 °F	63 %	W	15 mph	0 mph	29.73 in	0.0 in	Smoke	5/15/2020 05:30
91 °F	79 °F	66 %	W	14 mph	0 mph	29.70 in	0.0 in	Partly Cloudy	5/15/2020 06:00
93 °F	79 °F	63 %	W	15 mph	0 mph	29.70 in	0.0 in	Partly Cloudy	5/15/2020 06:30
91 °F	77 °F	63 %	WNW	14 mph	0 mph	29.70 in	0.0 in	Fair	5/15/2020 07:00
91 °F	79 °F	66 %	WNW	15 mph	0 mph	29.67 in	0.0 in	Fair	5/15/2020 07:30
91 °F	79 °F	66 %	WNW	17 mph	0 mph	29.67 in	0.0 in	Fair	5/15/2020 08:00

Data Manipulation and Imputation

Since the data we extracted is in different files, we had to merge this information. We used excel VLOOKUP to merge the same.

However, if we notice, the temperature information is available at every 30 minute intervals, whereas the solar power generation is available at every 15 minute intervals. We had missing values of the weather conditions missing for every alternate row. Hence, we used Python to perform data imputation.

The merged data is put into a pandas dataframe called df. Seeing the overview of this dataframe, we can notice that the variables from Temperature to weather condition, there are about 50% of null values.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3259 entries, 0 to 3258
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   DATE_TIME            3259 non-null   datetime64[ns]
 1   DATE_TIME.1          3259 non-null   object
 2   PLANT_ID             3259 non-null   int64
 3   SOURCE_KEY           3259 non-null   object
 4   AMBIENT_TEMPERATURE  3259 non-null   float64
 5   MODULE_TEMPERATURE   3259 non-null   float64
 6   MODULE_TEMP_F        3259 non-null   float64
 7   IRRADIATION          3259 non-null   float64
 8   Temperature          1624 non-null   float64
 9   Dew Point            1624 non-null   float64
 10  Humidity             1624 non-null   float64
 11  Wind                 1624 non-null   object
 12  Wind Speed           1624 non-null   float64
 13  Wind Gust            1624 non-null   float64
 14  Pressure             1624 non-null   float64
 15  Precip.              1624 non-null   float64
 16  Condition            1624 non-null   object
dtypes: datetime64[ns](1), float64(11), int64(1), object(4)
memory usage: 433.0+ KB
```

To impute the numerical values , of the data, we first subset the dataframe with only the numerical variables.

```
[ ] df_temp = df[['DATE_TIME','Temperature','Dew Point','Humidity','Wind Speed','Wind Gust','Pressure','Precip.']]
```

```
df_temp.head()
```

	DATE_TIME	Temperature	Dew Point	Humidity	Wind Speed	Wind Gust	Pressure	Precip.
0	2020-05-15 00:00:00	86.0	79.0	79.0	5.0	0.0	29.7	0.0
1	2020-05-15 00:15:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2020-05-15 00:30:00	86.0	77.0	74.0	5.0	0.0	29.7	0.0
3	2020-05-15 00:45:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2020-05-15 01:00:00	86.0	77.0	74.0	6.0	0.0	29.7	0.0

The imputation condition we used was interpolate – this will give us the mean value of the neighboring values. Which means, the value for record #2, will be the average of the values of #1 and #3 record measures for the same variable.

```
[ ]  df_temp.interpolate(inplace = True)
```

For the categorical variables, we have used a forward fill technique for the imputation.

```
[ ]  df_temp_2 = df[['DATE_TIME','Wind','Condition']]
     df_temp_2 = df_temp_2.fillna(method='ffill')
```

Now, if we see the combined data frame after all imputed values, we have handled all the null values.

```
[ ]  df_final.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3259 entries, 0 to 3258
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   DATE_TIME            3259 non-null   datetime64[ns]
 1   Formatted_dateTime   3259 non-null   object
 2   PLANT_ID             3259 non-null   int64
 3   SOURCE_KEY           3259 non-null   object
 4   AMBIENT_TEMPERATURE  3259 non-null   float64
 5   MODULE_TEMPERATURE   3259 non-null   float64
 6   MODULE_TEMP_F        3259 non-null   float64
 7   IRRADIATION          3259 non-null   float64
 8   Temperature          3259 non-null   float64
 9   Dew Point            3259 non-null   float64
 10  Humidity             3259 non-null   float64
 11  Wind Speed           3259 non-null   float64
 12  Wind Gust            3259 non-null   float64
 13  Pressure             3259 non-null   float64
 14  Precip.              3259 non-null   float64
 15  Wind                 3259 non-null   object
 16  Condition            3259 non-null   object
dtypes: datetime64[ns](1), float64(11), int64(1), object(4)
memory usage: 433.0+ KB
```

For ease of understanding, we have renamed the columns with the unit value

```
[ ]  df_final.rename(columns={'Temperature': 'Temperature(°F)','Dew Point':'Dew Point(°F)', 'Humidity':'Humidity(%)',
                            'Wind Speed':'Wind Speed(mph)','Wind Gust':'Wind Gust(mph)','Pressure':'Pressure(in)',
                            'Precip.':'Precip.(in)'}, inplace=True)
```

To handle working with categorical variables within time series forecasting, we have created dummy variables for the Condition categorical variable as follows.

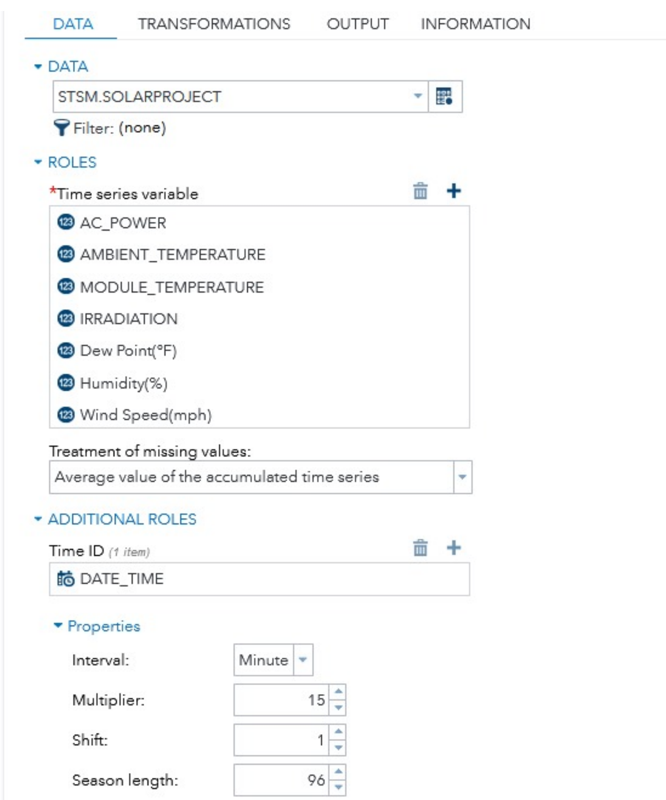
```
[ ]  df_final['Condition_num'] = np.where(df_final['Condition'] == "Fair", 0,
                                      np.where((df_final['Condition'] == "Haze")
                                     | (df_final['Condition'] == "Smoke")
                                     | (df_final['Condition'] == "Ha"), 1,
                                     np.where((df_final['Condition'] == "Partly Cloudy")
                                     | (df_final['Condition'] == "Drizzle")
                                     | (df_final['Condition'] == "Showers in the Vicinity")
                                     | (df_final['Condition'] == "Light Rain")
                                     | (df_final['Condition'] == "Light Drizzle")
                                     | (df_final['Condition'] == "Light Rain / Windy")
                                     | (df_final['Condition'] == "Drizzle / Windy"), 2, 3)))
```

We grouped the conditions based on similarity of the dataset and assigned numerical values. We finally assigned this manipulated value to a new variable called Condition_num.

The final dataset after all imputation and data manipulation, we have 67,699 rows and 22 columns.

Data Preparation in SAS

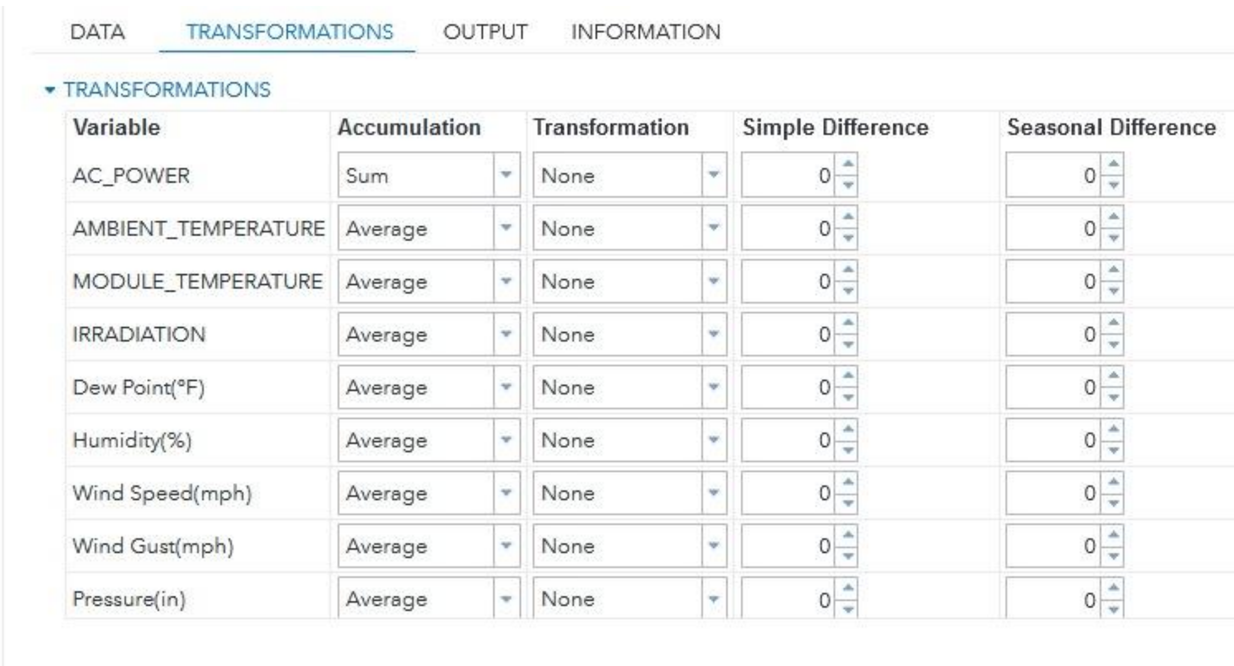
We performed data aggregation in SAS using the timeseries data preparation module.



During data preparation, we removed the insignificant variables like, Wind Direction, Plant ID, Source_key and Precipitation. We termed these variables insignificant because, wind direction is a categorical variable and is very random. Plant_ID and Source_key are constant values which are same across the dataset, and Precipitation, which is a constant of value 0.

The interval is in Minutes, with a multiplier of 15. This is because the data is plotted at 15-minute intervals. Also, the season length is 96. (4 measures every hour and 24 hours per day).

We then performed aggregation to find out the power generated from all panels at the single point in time.



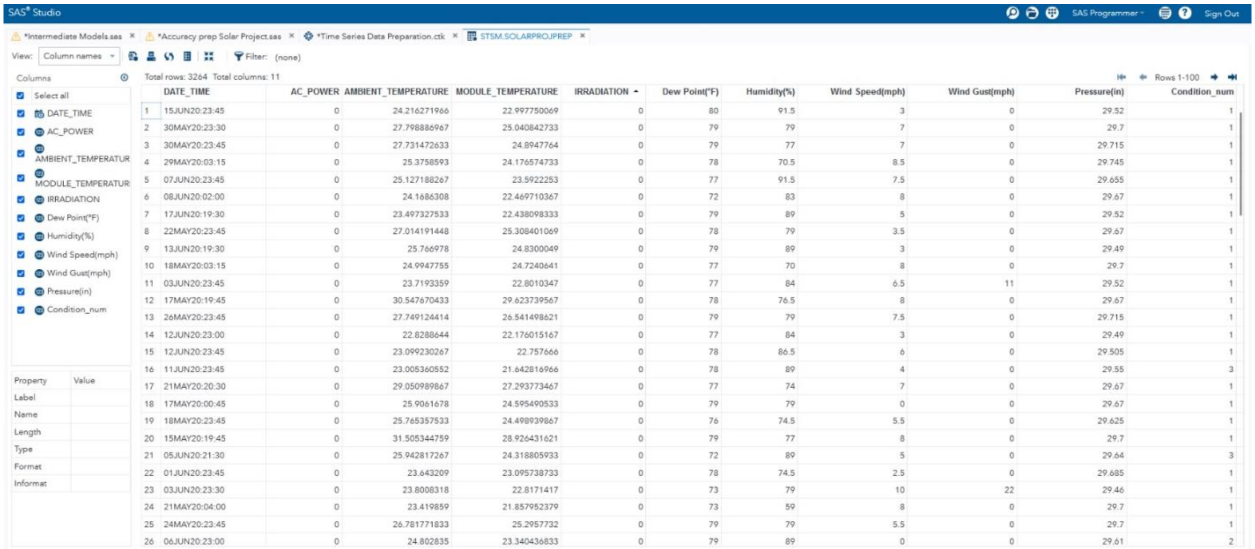
DATA TRANSFORMATIONS OUTPUT INFORMATION

▾ TRANSFORMATIONS

Variable	Accumulation	Transformation	Simple Difference	Seasonal Difference
AC_POWER	Sum	None	0	0
AMBIENT_TEMPERATURE	Average	None	0	0
MODULE_TEMPERATURE	Average	None	0	0
IRRADIATION	Average	None	0	0
Dew Point(°F)	Average	None	0	0
Humidity(%)	Average	None	0	0
Wind Speed(mph)	Average	None	0	0
Wind Gust(mph)	Average	None	0	0
Pressure(in)	Average	None	0	0

We did the sum of the AC_Power value, while we averaged the other fields.

The final dataset for exploration and modelling is as follows :

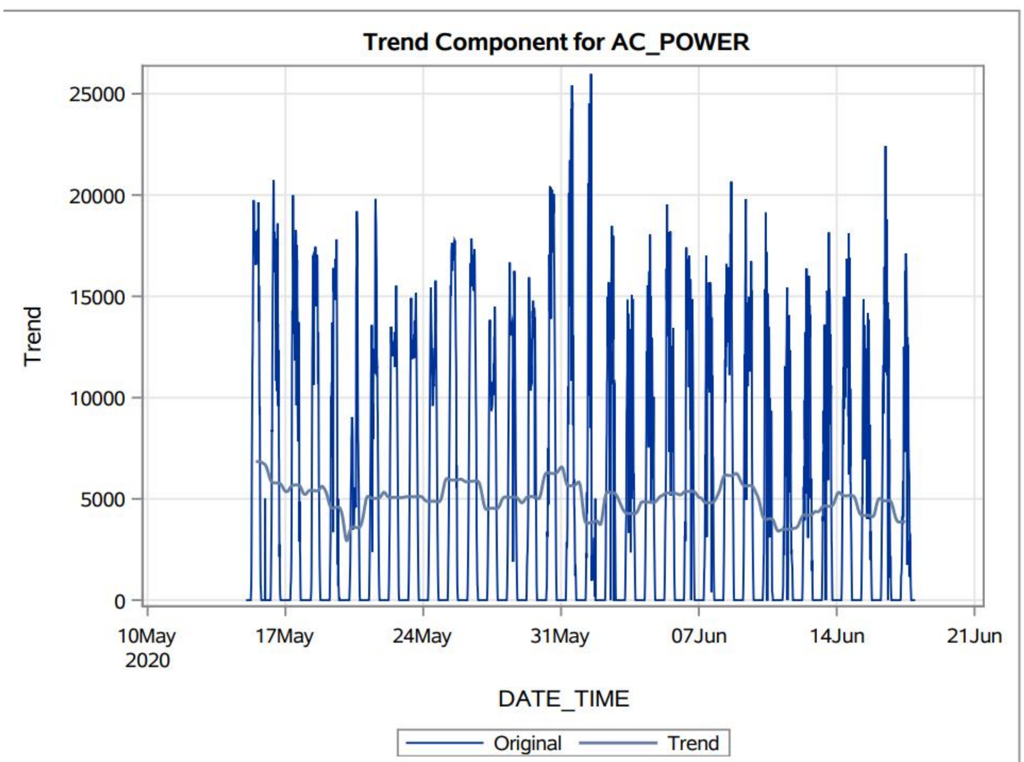


Total rows: 3264 Total columns: 11

	DATE_TIME	AC_POWER	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	Dew Point(°F)	Humidity(%)	Wind Speed(mph)	Wind Gust(mph)	Pressure(in)	Condition_num
1	15JUN20:23:45	0	24.216271966	22.997750069	0	80	91.5	3	0	29.52	1
2	30MAY20:23:30	0	27.798886967	25.040842733	0	79	79	7	0	29.7	1
3	30MAY20:23:45	0	27.731472633	24.8947764	0	79	77	7	0	29.715	1
4	29MAY20:03:15	0	25.3758593	24.176574733	0	78	70.5	8.5	0	29.745	1
5	07JUN20:23:45	0	25.127188267	23.5922253	0	77	91.5	7.5	0	29.655	1
6	08JUN20:02:00	0	24.1686308	22.469710367	0	72	83	8	0	29.67	1
7	17JUN20:19:30	0	23.497327533	22.438098333	0	79	89	5	0	29.52	1
8	22MAY20:23:45	0	27.014191448	25.308401069	0	78	79	3.5	0	29.67	1
9	13JUN20:19:30	0	25.766978	24.8300049	0	79	89	3	0	29.49	1
10	18MAY20:03:15	0	24.9947755	24.7240641	0	77	70	8	0	29.7	1
11	03JUN20:23:45	0	23.7193359	22.8010347	0	77	84	6.5	11	29.52	1
12	17MAY20:19:45	0	30.547670433	29.623739567	0	78	76.5	8	0	29.67	1
13	26MAY20:23:45	0	27.749124414	26.541498621	0	79	79	7.5	0	29.715	1
14	12JUN20:23:00	0	22.8288644	22.176015167	0	77	84	3	0	29.49	1
15	12JUN20:23:45	0	23.099230267	22.757666	0	78	86.5	6	0	29.505	1
16	11JUN20:23:45	0	23.005360552	21.642816966	0	78	89	4	0	29.55	3
17	21MAY20:20:30	0	29.050989867	27.293773467	0	77	74	7	0	29.67	1
18	17MAY20:00:45	0	25.9061678	24.595490533	0	79	79	0	0	29.67	1
19	18MAY20:23:45	0	25.765357533	24.498939867	0	76	74.5	5.5	0	29.625	1
20	15MAY20:19:45	0	31.505344759	28.926431621	0	79	77	8	0	29.7	1
21	05JUN20:21:30	0	25.942817267	24.318805933	0	72	89	5	0	29.64	3
22	01JUN20:23:45	0	23.643209	23.095738733	0	78	74.5	2.5	0	29.685	1
23	03JUN20:23:30	0	23.8008318	22.8171417	0	73	79	10	22	29.46	1
24	21MAY20:04:00	0	23.419859	21.857952379	0	73	59	8	0	29.7	1
25	24MAY20:23:45	0	26.781771833	25.2957732	0	79	79	5.5	0	29.7	1
26	06JUN20:23:00	0	24.802835	23.340436833	0	79	89	0	0	29.61	2

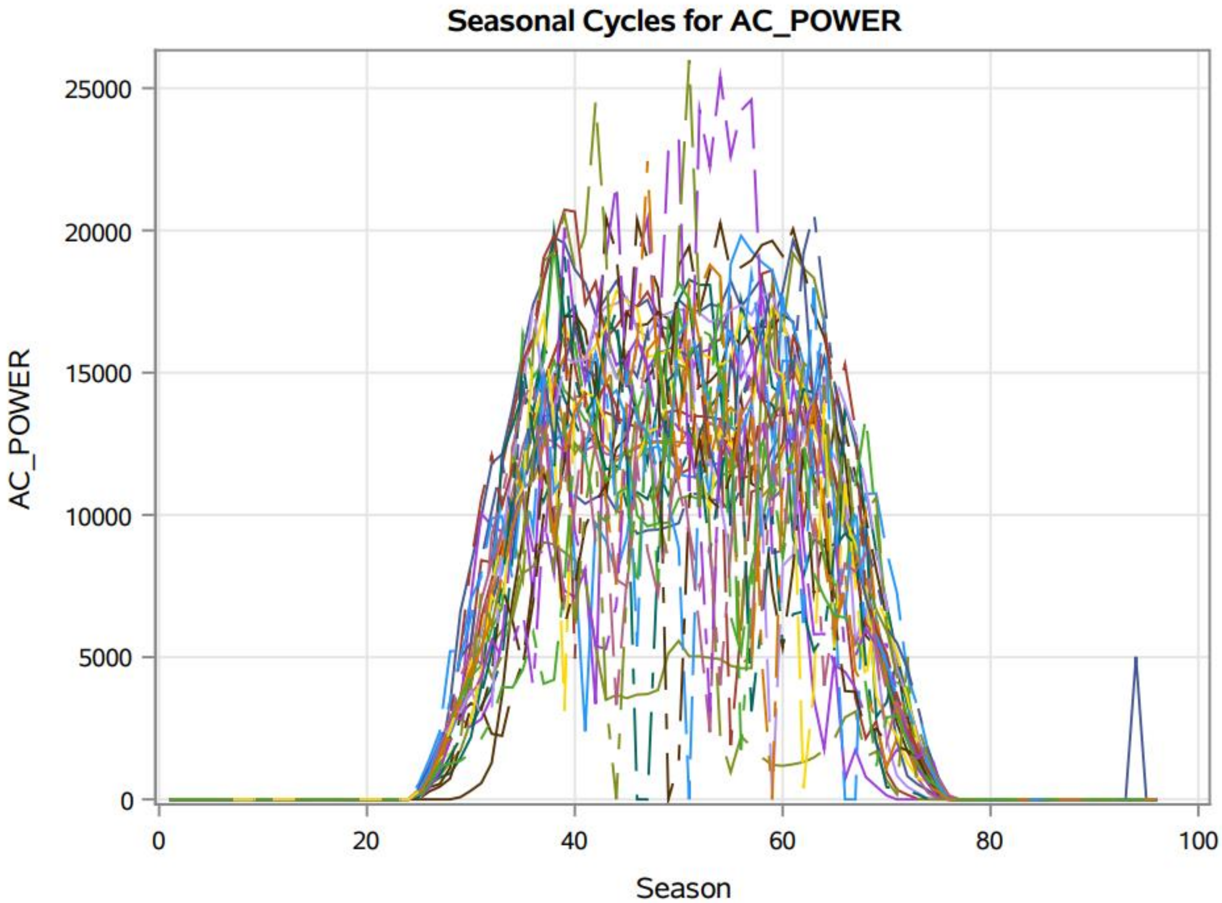
<u>Time Series Exploration</u>

Below are the insights after performing the timeseries exploration, with the y-variable as 'AC_POWER' , Dependent time variable as DATE_TIME and all other variables are the independent variables.



There is no significant trend component based on the above plot shown for AC_POWER.

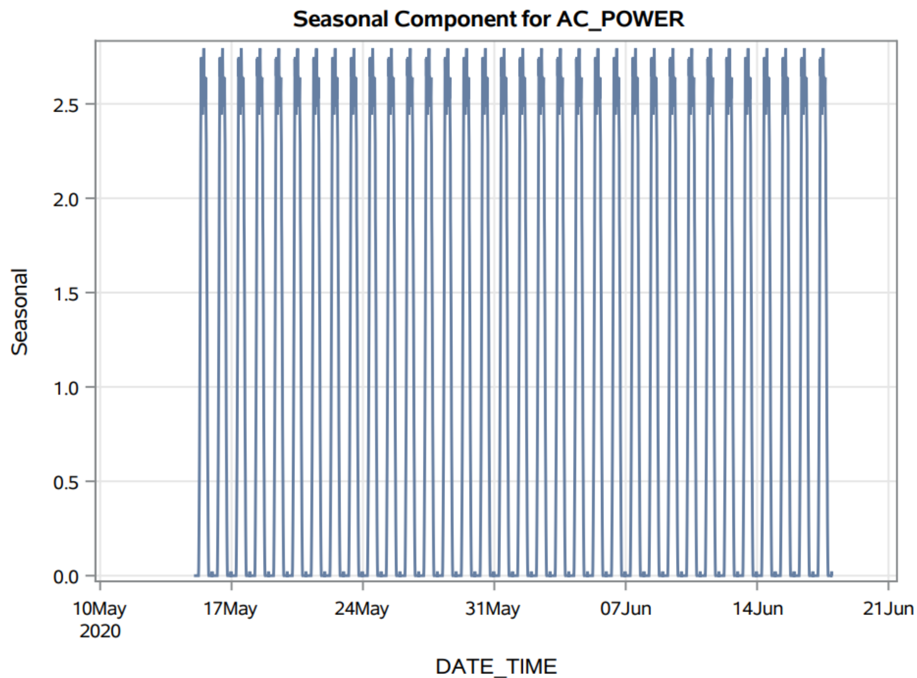
However, we have found a seasonality component.

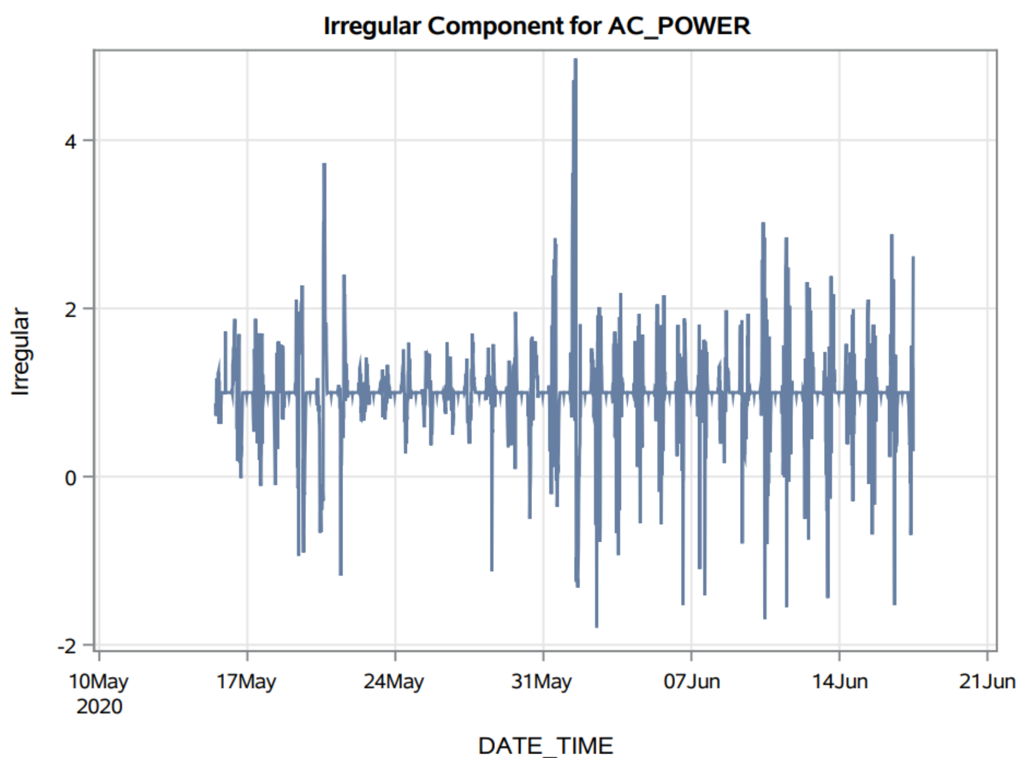


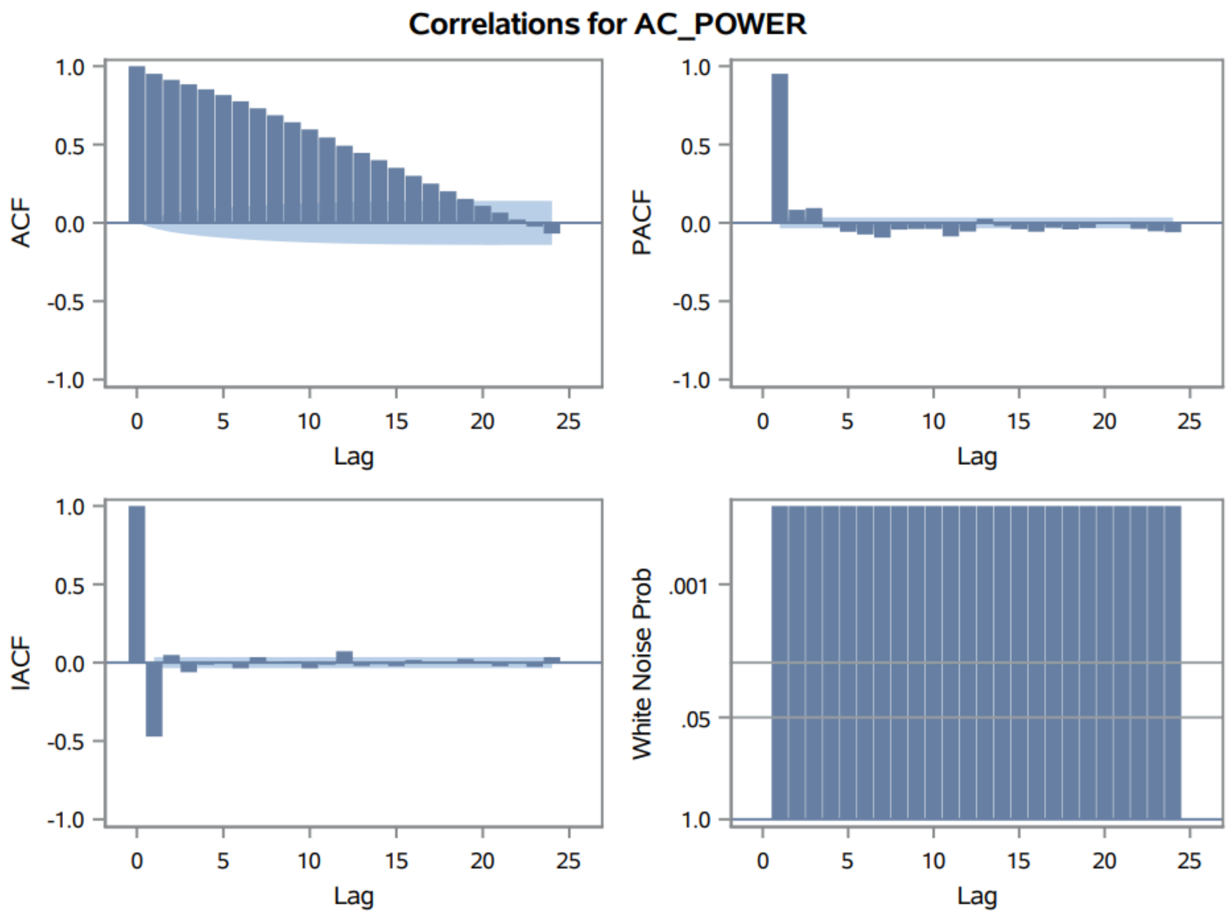
This seasonality can be explained with a seasonal cycle of 96. The graph can be interpreted easily as that there is no solar power generated before sunrise and after sunset. It is higher during the afternoon with peaks at noon and slowly decreasing as the evening sets.

We can notice a few dips in the values, which can be explained with external events. Maybe there was a cloudy day which has caused low power generation.

The below graphs about the seasonal component and the irregular component also show that there is seasonality in the dataset. We can confirm this because the irregular component is less significant than the seasonal component.


Seasonal Component for AC_POWER
Seasonal
2.5
2.0
1.5
1.0
0.5
0.0
10May
2020
17May
24May
31May
07Jun
14Jun
21Jun
DATE_TIME


Irregular Component for AC_POWER
Irregular
4
2
0
-2
10May
2020
17May
24May
31May
07Jun
14Jun
21Jun
DATE_TIME



The above correlations graph for AC_Power shows that there is signal which can be modelled, as the white noise test fails. The ACF graph shows a decreasing trend and there are significant peaks at different lags within the PACF and IACF graphs. This explains that this time series data can be modeled using AR and MA.

Augmented Dickey-Fuller tests, are statistical tests used to assess whether a time series data set is stationary or non-stationary. Stationarity is an important concept in time series analysis because many time series models assume that the data is stationary, meaning that statistical properties like the mean and variance do not change over time.

Augmented Dickey-Fuller Unit Root Tests							
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-97.1343	<.0001	-7.02	<.0001		
	1	-80.6302	<.0001	-6.35	<.0001		
	2	-65.4746	<.0001	-5.69	<.0001		
Single Mean	0	-158.407	0.0001	-9.01	<.0001	40.56	0.0010
	1	-133.853	0.0001	-8.18	<.0001	33.43	0.0010
	2	-110.177	0.0001	-7.36	<.0001	27.05	0.0010
Trend	0	-158.896	0.0001	-9.02	<.0001	40.72	0.0010
	1	-134.308	0.0001	-8.19	<.0001	33.59	0.0010
	2	-110.589	0.0001	-7.37	<.0001	27.20	0.0010

We notice that the p values for all the types are significant, and our null hypothesis fails, so the dataset can be determined as stationary.

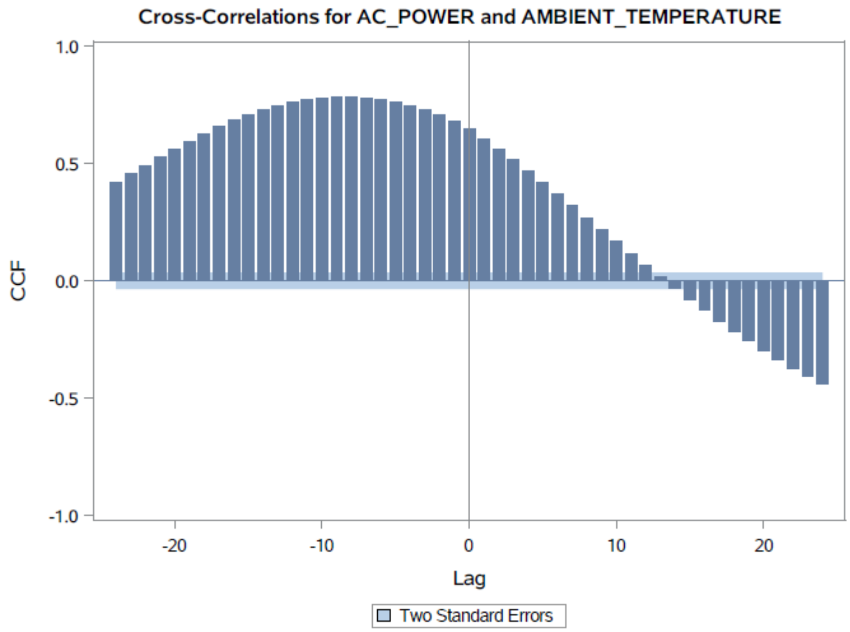
Also, through observation we notice that all our independent variables are time series. Also, the cross-correlation plots for all the variables against our Dependent variable shows that there is significance at multiple lags.

Hence, we need to perform pre-whitening to identify if the independent variables are significant or not , and if they are significant, we will have to identify at which lag(s) they are significant.

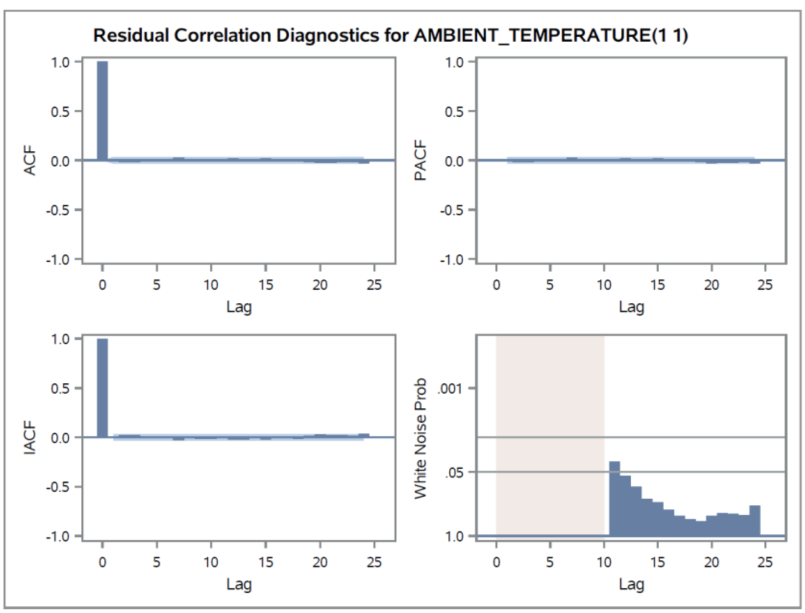
Independent Variable Pre-whitening

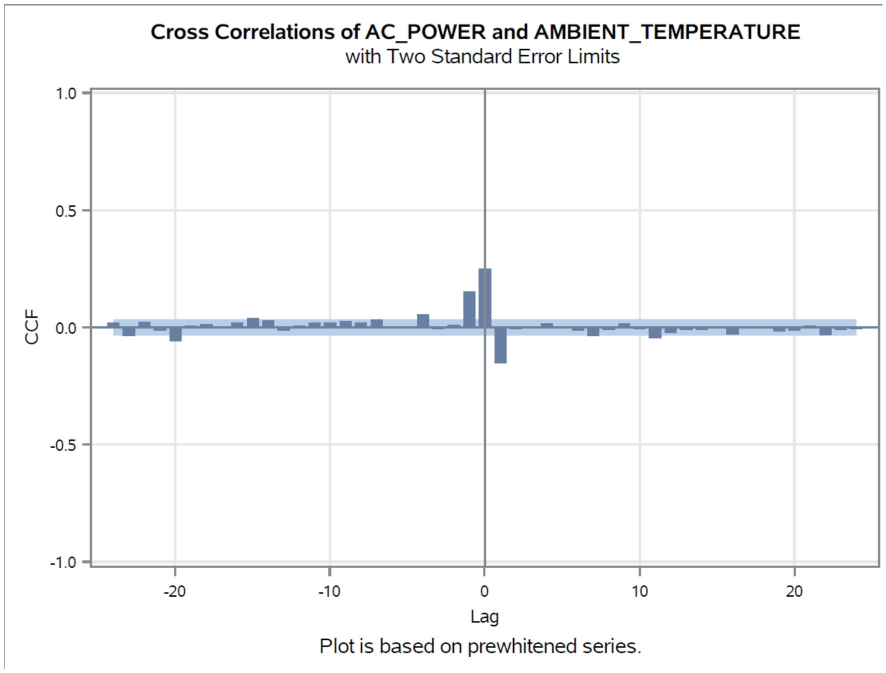
AMBIENT_TEMPERATURE Pre-whitening

We observed a cross-correlation between AC_POWER and AMBIENT_TEMPERATURE, but it was challenging to determine whether this relationship was a true association or if the autocorrelation influenced it in both time series. The correlation value obtained at this stage may not accurately represent the underlying relationship.



After applying whitening techniques to remove autocorrelation, we reevaluated the cross-correlation between AC_POWER and AMBIENT_TEMPERATURE. Any observed cross-correlations at this stage are more likely to be genuine, as they are not influenced by autocorrelation in the individual series.

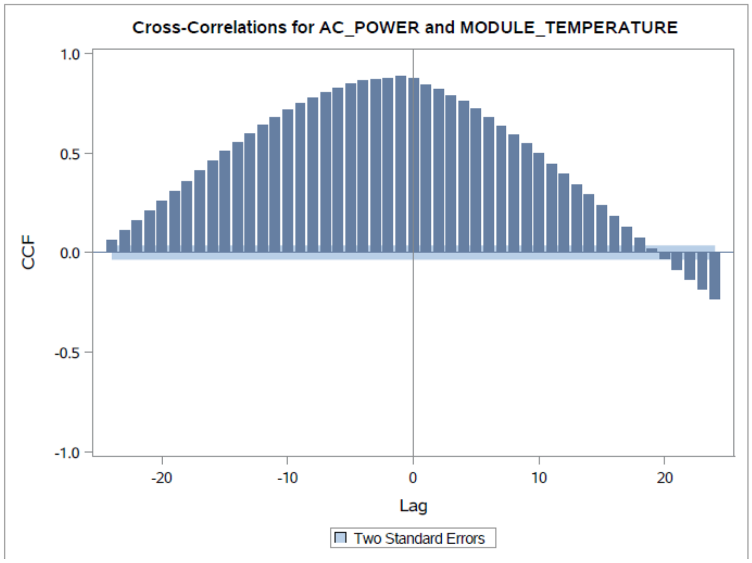




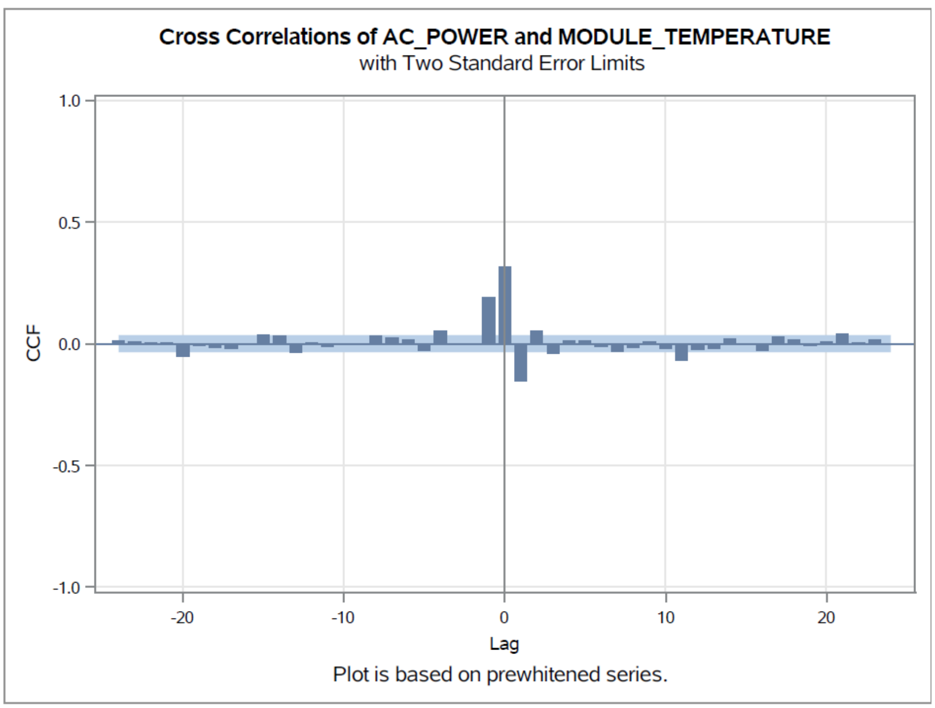
In conclusion, our analysis, after Pre-whitening, identified significant cross-correlations at lag 0 and lag 1 between AC_POWER and AMBIENT_TEMPERATURE. These findings suggest both an immediate and short-term relationship between temperature changes and solar power generation, with practical implications for the optimization of renewable energy systems.

MODULE_TEMPERATURE pre-whitening

Initially, we observed a cross-correlation between AC_POWER and MODULE_TEMPERATURE. However, this relationship may be influenced by the autocorrelation within each of the individual time series, making it challenging to draw definitive conclusions.



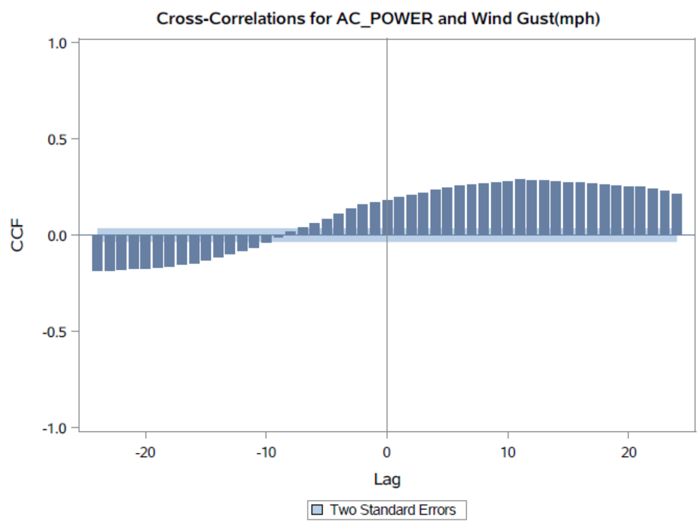
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



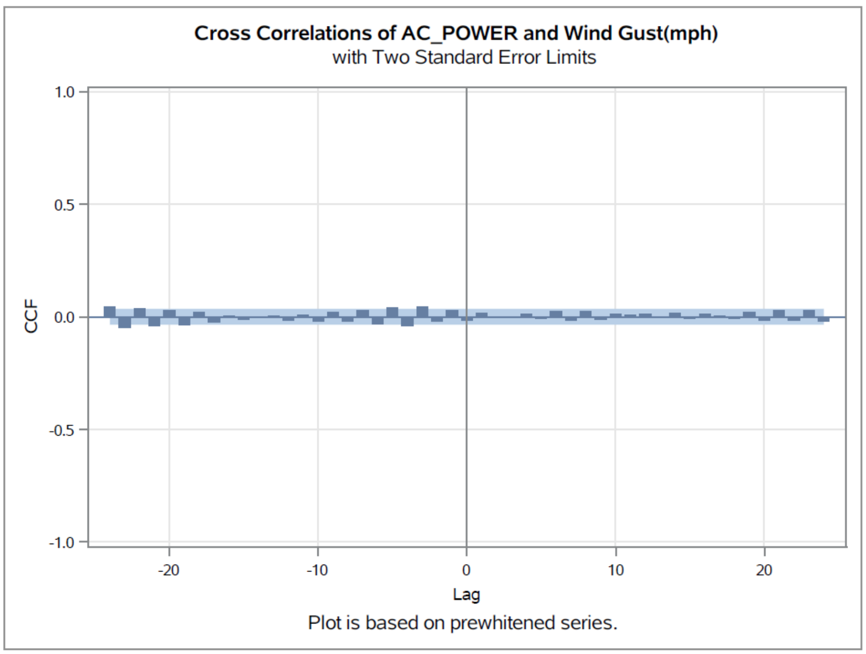
Our Pre-whitening analysis revealed significant cross-correlations at lag 0 and lag 1 between AC_POWER and MODULE_TEMPERATURE, indicating immediate and short-term relationships. These findings have practical implications for optimizing solar panel systems by responding to temperature fluctuations in real time.

Wind Gust Pre-whitening

Initially, we compute the cross-correlation between AC_POWER and Wind Gust without Pre-whitening. This helps us understand the raw relationship between the two variables, considering the effects of inherent autocorrelation in each time series.



Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.

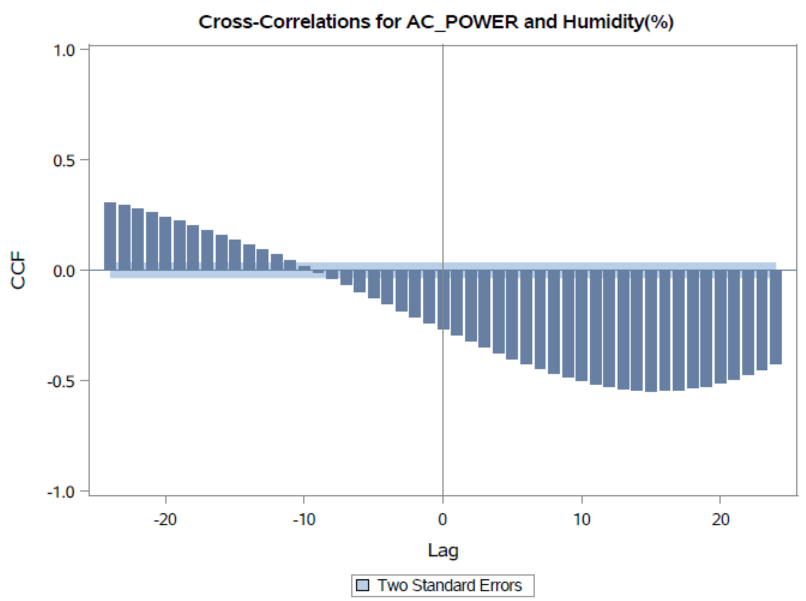


After applying Pre-whitening, did not reveal significant cross-correlations between AC_POWER and Wind Gust. This suggests that there may not be a significant relationship between solar power generation and sudden wind gusts within the analyzed time period.

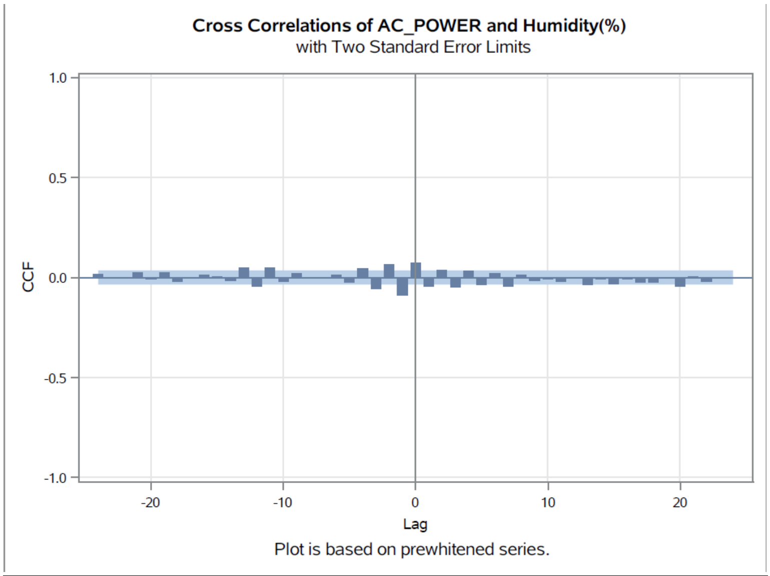
While the absence of significant cross-correlations may indicate that wind gusts do not have a substantial impact on solar power generation during the observed time frame.

Humidity Pre-whitening

Initially, we observed a cross-correlation between AC_POWER and Humidity. However, this relationship may be influenced by the autocorrelation within each of the individual time series, making it challenging to draw definitive conclusions.



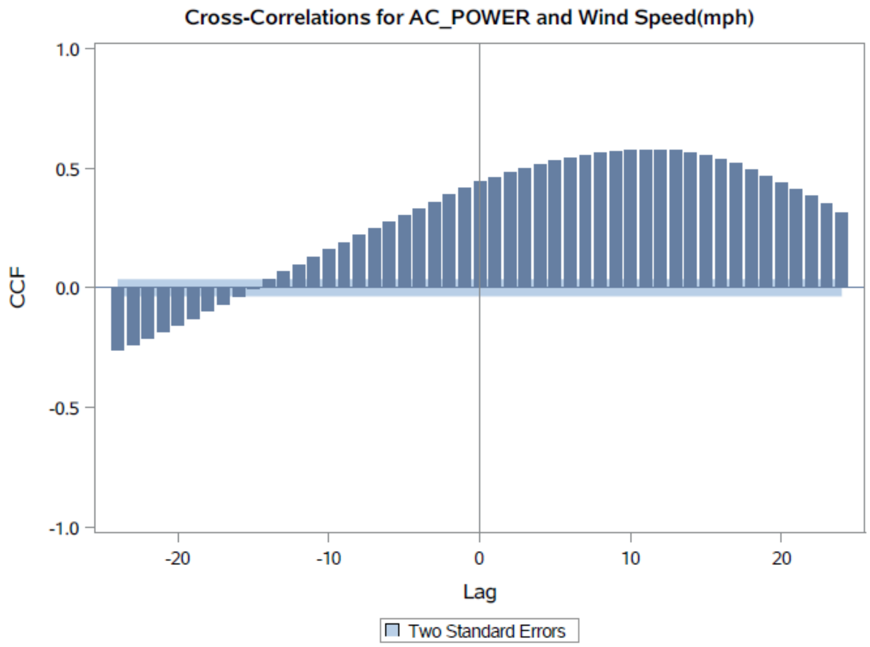
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



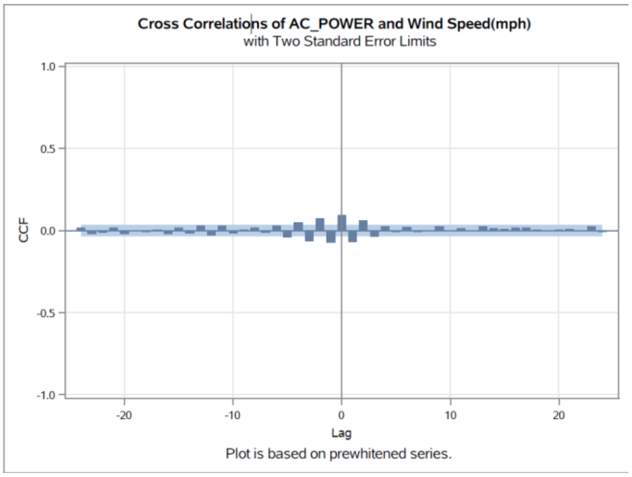
Our analysis, after Pre-whitening, revealed a significant cross-correlation at lag 0 between AC_POWER and Humidity. This suggests an immediate relationship between solar power generation and humidity levels within the observed time period, which may have practical implications for renewable energy systems. Further investigation is needed to understand the full extent of this relationship.

Wind Speed Pre-whitening

We compute the cross-correlation between AC_POWER and Wind Speed without pre-whitening. This helps us understand the raw relationship between the two variables, considering the effects of inherent autocorrelation in each time series.



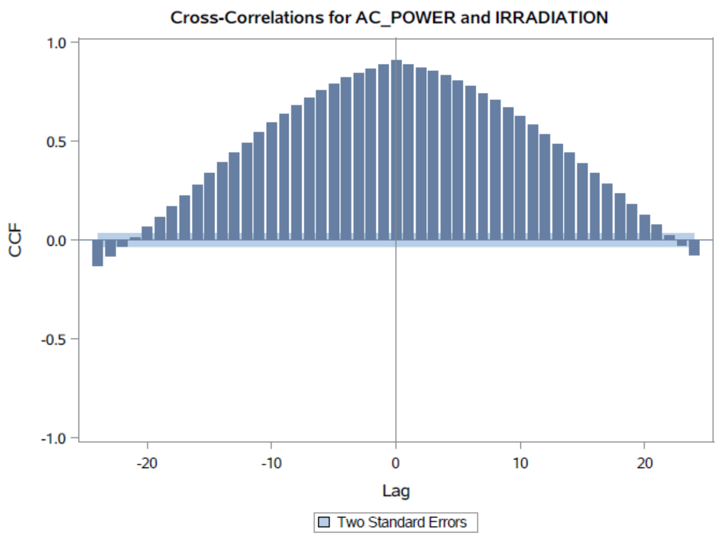
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



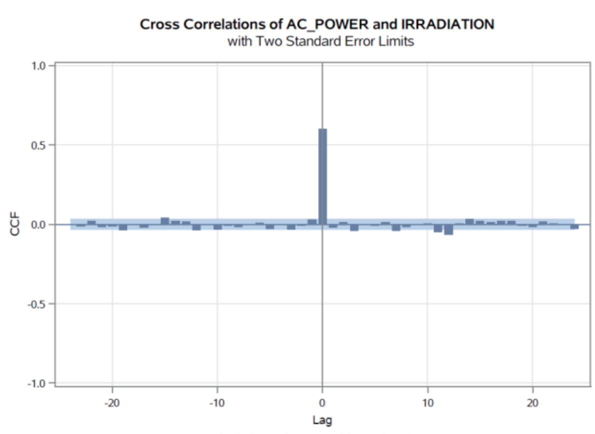
After Pre-whitening, has uncovered significant cross-correlations at lag 0, lag 1, and lag 2 between AC_POWER and Wind Speed. This suggests immediate, short-term, and slightly longer-term relationships within the observed time period, with implications for renewable energy system management. Further research may be needed to understand the full extent of these relationships.

IRRADIATION Pre-whitening

We compute the cross-correlation between AC_POWER and IRRADIATION without Pre-whitening. This helps us understand the raw relationship between the two variables, considering the effects of inherent autocorrelation in each time series.



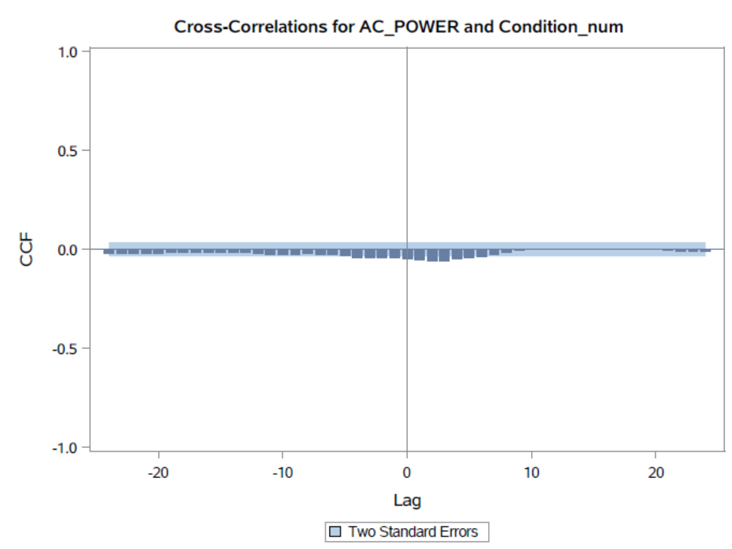
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



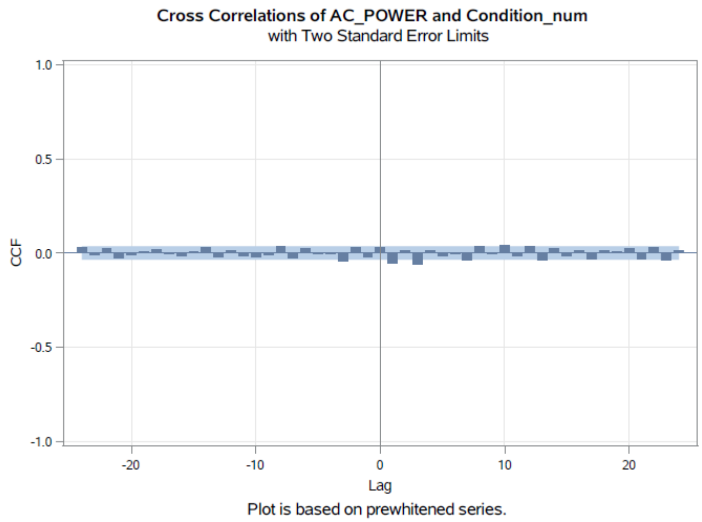
After Pre-whitening, has revealed a significant cross-correlation at lag 0 between AC_POWER and IRRADIATION. This highlights an immediate and direct relationship between solar power generation and solar irradiance levels, emphasizing the importance of sunlight for efficient power generation in renewable energy systems.

Condition_num Pre-whitening

Initially, we observed a cross-correlation between AC_POWER and Condition_num. However, this relationship may be influenced by the autocorrelation within each of the individual time series, making it challenging to draw definitive conclusions.



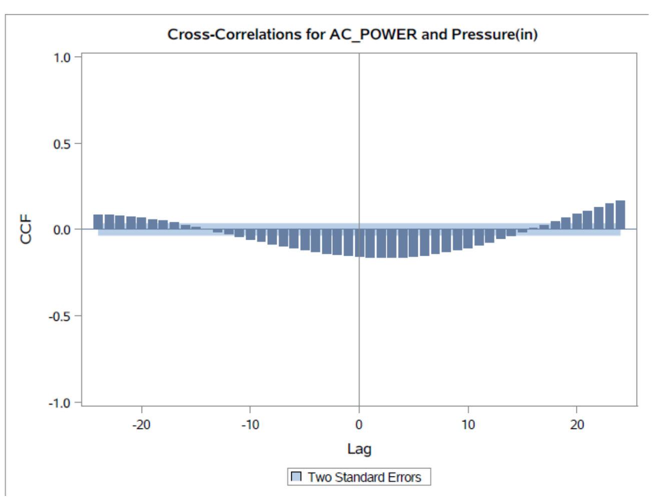
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



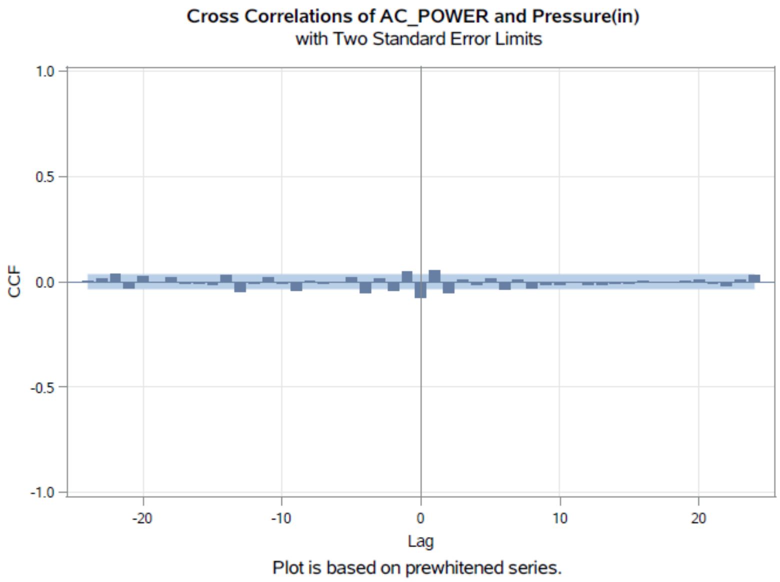
Our analysis, after Pre-whitening, did not reveal significant cross-correlations between AC_POWER and Condition_num. This suggests that, within the observed time period, there is no significant relationship between solar power generation and the numerical indicator of environmental conditions.

Pressure Pre-whitening

Initially, we compute the cross-correlation between AC_POWER and Pressure without Pre-whitening. This helps us understand the raw relationship between the two variables, considering the effects of inherent autocorrelation in each time series.



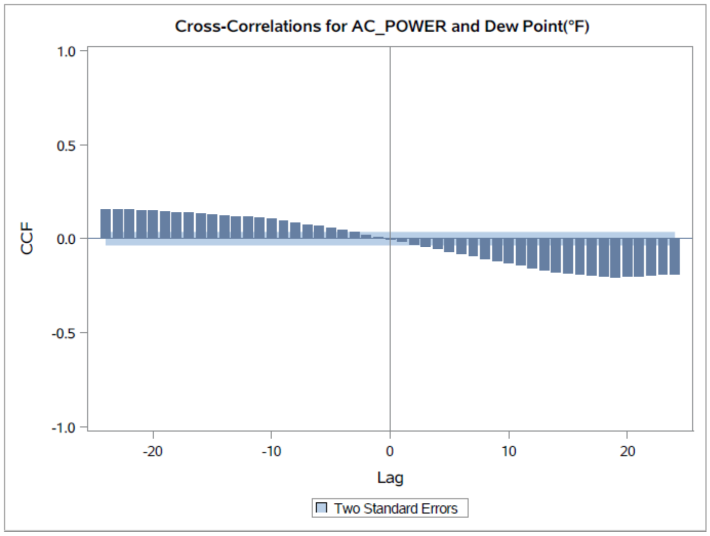
Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



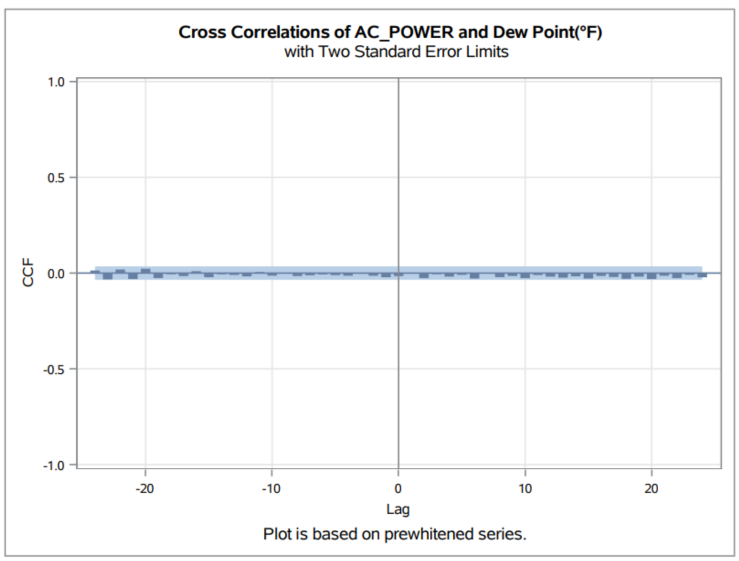
Our analysis, after Pre-whitening, did not reveal significant cross-correlations between AC_POWER and Pressure. This suggests that, within the observed period, there is no significant relationship between solar power generation and atmospheric pressure.

Dew Point Pre-whitening

Initially, we compute the cross-correlation between AC_POWER and Dew Point without Pre-whitening. This helps us understand the raw relationship between the two variables, considering the effects of inherent autocorrelation in each time series.



Following the Pre-whitening process, which removes autocorrelation, we reevaluate the cross-correlation. Any cross-correlations detected at this stage are more likely to be genuine, as they are not confounded by autocorrelation.



In conclusion, our analysis, after Pre-whitening, did not reveal significant cross-correlations between AC_POWER and Dew Point. This suggests that, within the observed time period, there is no significant relationship between solar power generation and Dew Point.

Best Model

In this phase of our analysis, we aim to choose the best forecast model for solar power generation after Pre-whitening the time series data for various environmental variables, including Wind Gust, Humidity, Wind Speed, IRRADIATION, Condition_num, Pressure, and Dew Point. We also consider the summary of our findings from the Pre-whitening analysis that revealed no significant relationships in some cases.

After a thorough evaluation of various forecasting models for solar power generation, we have identified the ARIMAX (8,0,3;0,1,0) model as the most suitable for our analysis. This model leverages Autoregressive Integrated Moving Average (ARIMA) techniques in conjunction with exogenous variables (X) to make accurate predictions of solar power generation.

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	29.13068	136.60167	0.21	0.8311	0	AC_POWER	0
MA1,1	-0.13802	0.10878	-1.27	0.2045	1	AC_POWER	0
MA1,2	0.25634	0.22673	1.13	0.2582	2	AC_POWER	0
MA1,3	-0.60564	0.10811	-5.60	<.0001	3	AC_POWER	0
AR1,1	0.55294	0.13638	4.05	<.0001	1	AC_POWER	0
AR1,2	0.21373	0.11689	1.83	0.0675	2	AC_POWER	0
AR1,3	-0.68358	0.09833	-6.95	<.0001	3	AC_POWER	0
AR1,4	0.46063	0.07420	6.21	<.0001	4	AC_POWER	0
AR1,5	0.02278	0.03264	0.70	0.4852	5	AC_POWER	0
AR1,6	0.0039269	0.03083	0.13	0.8986	6	AC_POWER	0
AR1,7	-0.04705	0.02463	-1.91	0.0560	7	AC_POWER	0
AR1,8	0.11216	0.01959	5.73	<.0001	8	AC_POWER	0
NUM1	897.94873	73.15394	12.27	<.0001	0	AMBIENT_TEMPERATURE	0
NUM2	-419.51931	71.02468	-5.91	<.0001	0	AMBIENT_TEMPERATURE	1
NUM3	-398.52552	25.03008	-15.92	<.0001	0	MODULE_TEMPERATURE	0
NUM4	137.53570	20.26441	6.79	<.0001	0	MODULE_TEMPERATURE	1
NUM5	20359.2	486.69765	41.83	<.0001	0	IRRADIATION	0
NUM6	5.13861	10.72729	0.48	0.6319	0	Humidity(%)	0
NUM7	25.55819	22.09966	1.16	0.2475	0	Wind Speed(mph)	0

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
NUM8	-2.03864	26.81765	-0.08	0.9394	0	Wind Speed(mph)	1
NUM9	-24.70306	22.07191	-1.12	0.2631	0	Wind Speed(mph)	2

Constant Estimate	10.61728
Variance Estimate	3543620
Std Error Estimate	1882.451
AIC	56754.72
SBC	56881.98
Number of Residuals	3166

Best Fit Model Description

1. **ARIMA Components (8,0,3):**
   - **p (Autoregressive Order):** 8
   - **d (Integration Order):** 0
   - **q (Moving Average Order):** 3

The ARIMA component of our model accounts for the temporal dependencies within the solar power generation time series data. The autoregressive (AR) factors capture the past values of AC_POWER, while the moving average (MA) factors consider the influence of past forecast errors.

The autoregressive factors are given by:
Factor 1: 1 - 0.55294 B**(1) - 0.21373 B**(2) + 0.68358 B**(3) - 0.46063 B**(4) - 0.02278 B**(5) - 0.00393 B**(6) + 0.04705 B**(7) - 0.11216 B**(8).

These factors account for the autocorrelation in the solar power generation data, capturing the relationships between past and current values.
The moving average factors are given by:
Factor 1: 1 + 0.13802 B**(1) - 0.25634 B**(2) + 0.60564 B**(3)

These factors account for the influence of past forecast errors on the current solar power generation.

2. **Seasonal Differential Component (0,1,0):**
   - **P (Seasonal Autoregressive Order):** 0
   - **D (Seasonal Integration Order):** 1
   - **Q (Seasonal Moving Average Order):** 0

The exogenous variable component of our model accounts for the impact of external factors on solar power generation. In this case, we use a seasonal integration order (D=1) to incorporate the effect of past differences in the exogenous variable, which is likely to be related to solar irradiance (IRRADIATION).

Model for variable AC_POWER	
Estimated Intercept	29.13068
Period(s) of Differencing	96

Autoregressive Factors	
Factor 1:	1 - 0.55294 B**(1) - 0.21373 B**(2) + 0.68358 B**(3) - 0.46063 B**(4) - 0.02278 B**(5) - 0.00393 B**(6) + 0.04705 B**(7) - 0.11216 B**(8)

Moving Average Factors	
Factor 1:	1 + 0.13802 B**(1) - 0.25634 B**(2) + 0.60564 B**(3)

The independent variables (X) would contribute to the forecast with the following coefficients.

Input Number 1	
Input Variable	AMBIENT_TEMPERATURE
Period(s) of Differencing	96
Overall Regression Factor	897.9487

Input Number 2	
Input Variable	AMBIENT_TEMPERATURE
Shift	1
Period(s) of Differencing	96
Overall Regression Factor	-419.519

Input Number 3	
Input Variable	MODULE_TEMPERATURE
Period(s) of Differencing	96
Overall Regression Factor	-398.526

Input Number 4	
Input Variable	MODULE_TEMPERATURE
Shift	1
Period(s) of Differencing	96
Overall Regression Factor	137.5357

Input Number 5	
Input Variable	IRRADIATION
Period(s) of Differencing	96
Overall Regression Factor	20359.21

Input Number 6	
Input Variable	Humidity(%)
Period(s) of Differencing	96
Overall Regression Factor	5.138613

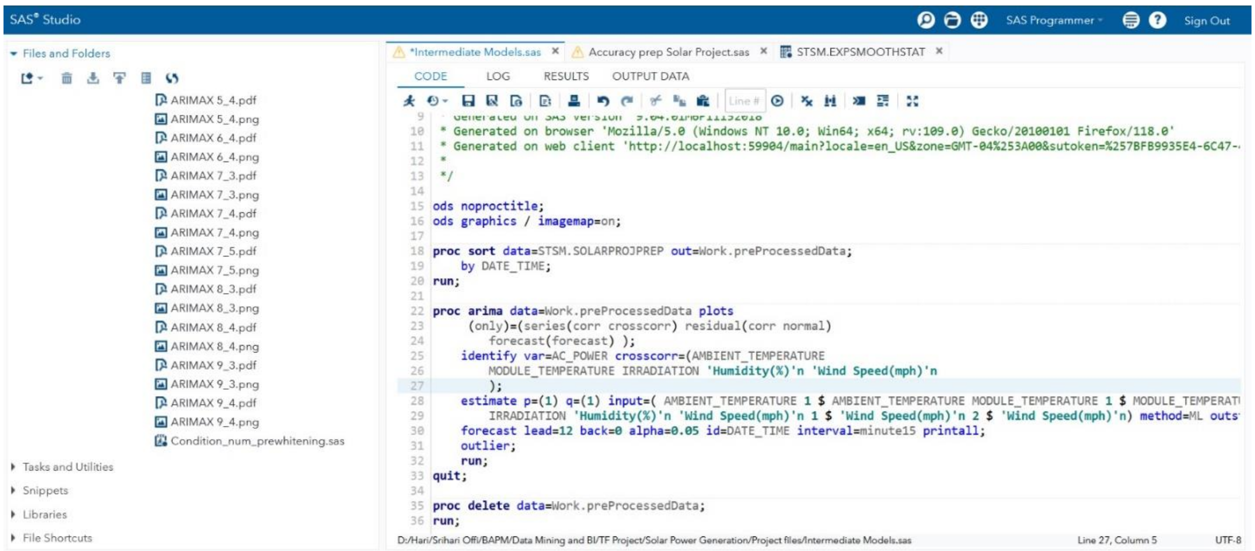
Input Number 7	
Input Variable	Wind Speed(mph)
Period(s) of Differencing	96
Overall Regression Factor	25.55819

Input Number 8	
Input Variable	Wind Speed(mph)
Shift	1
Period(s) of Differencing	96
Overall Regression Factor	-2.03864

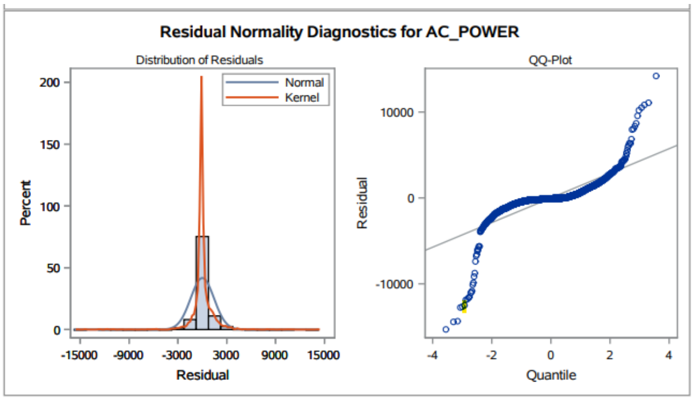
Input Number 9	
Input Variable	Wind Speed(mph)
Shift	2
Period(s) of Differencing	96
Overall Regression Factor	-24.7031

Best Fit Model Generation Process – After Pre-whitening

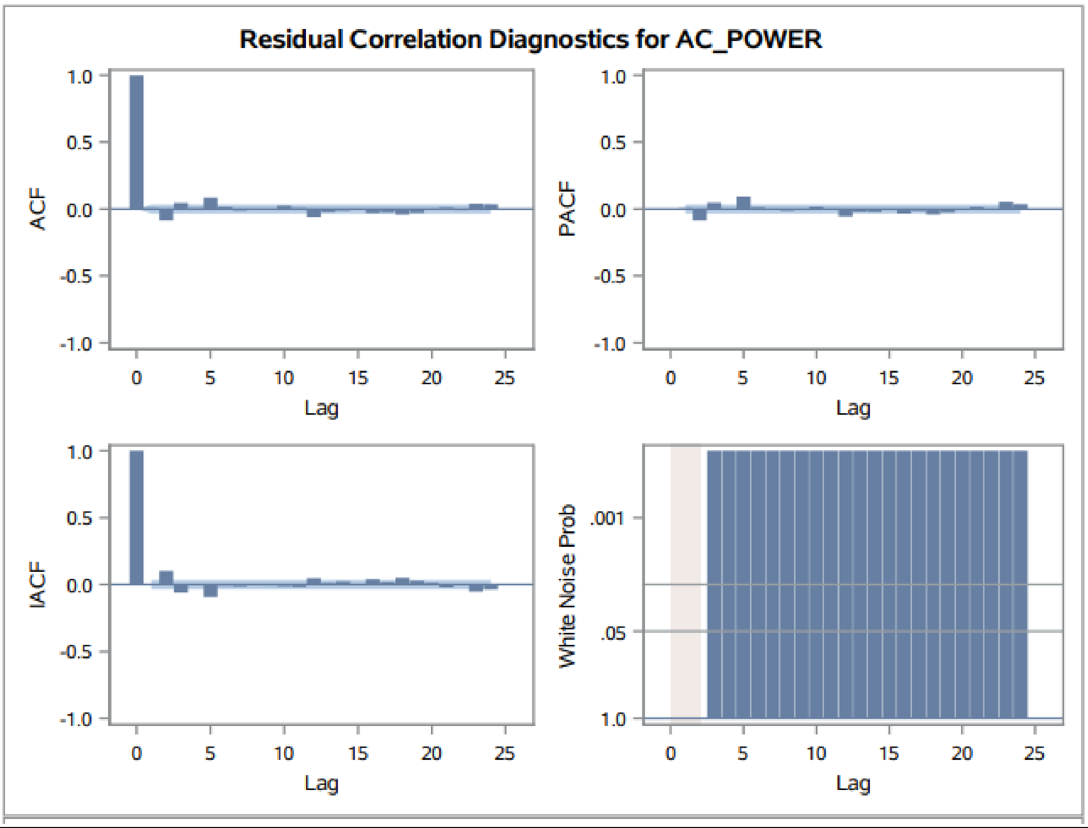
Initial Model with ARMAX (1,1)



The independent variables are considered with the lags that were identified during Pre-whitening which we can identify in the code from lines 28-30.

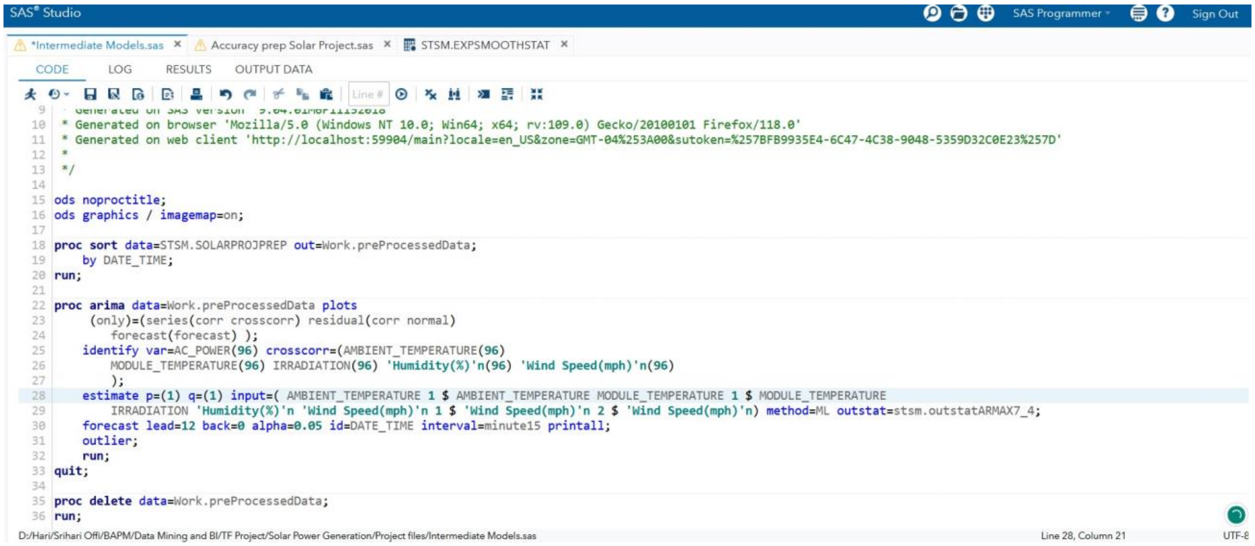


Monday, October 9, 2023 12:45:50 PM



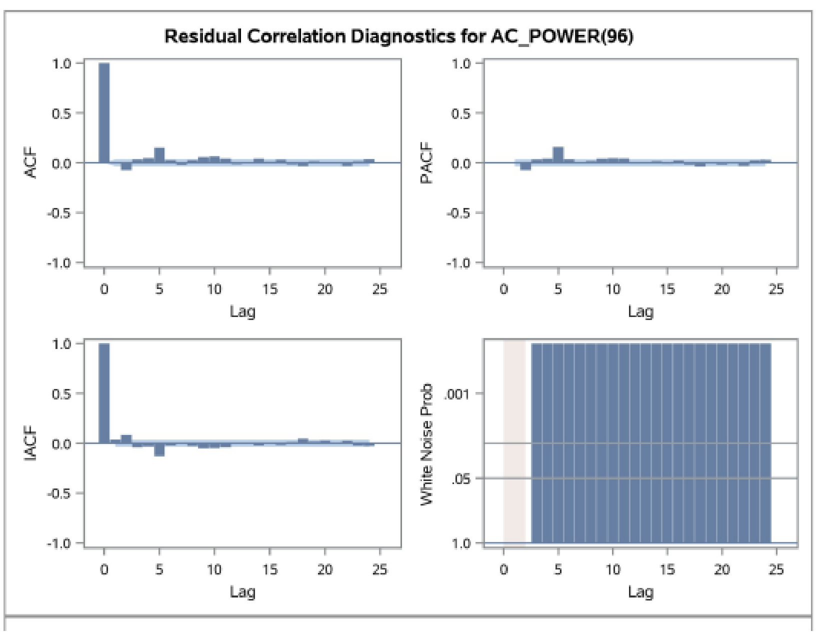
When we first employed the ARMAX (p=1, q=1) model, we observed a sinusoidal pattern in the Residual normality graph of AC_POWER. Consequently, we decided to incorporate a seasonality difference into our models. Also, this model has also failed the white noise test. Initially, we implemented a seasonality of 1 for our model, but when we increased it to 2, we encountered significant computational resource consumption. As a result, we opted to maintain a seasonality of 1 throughout our model generation process.

<u>ARIMAX (p=1,d=0,q=1 ; P=0,D=1,Q=0)</u>



Output

Sunday, October 8, 2023 11:44:26 AM

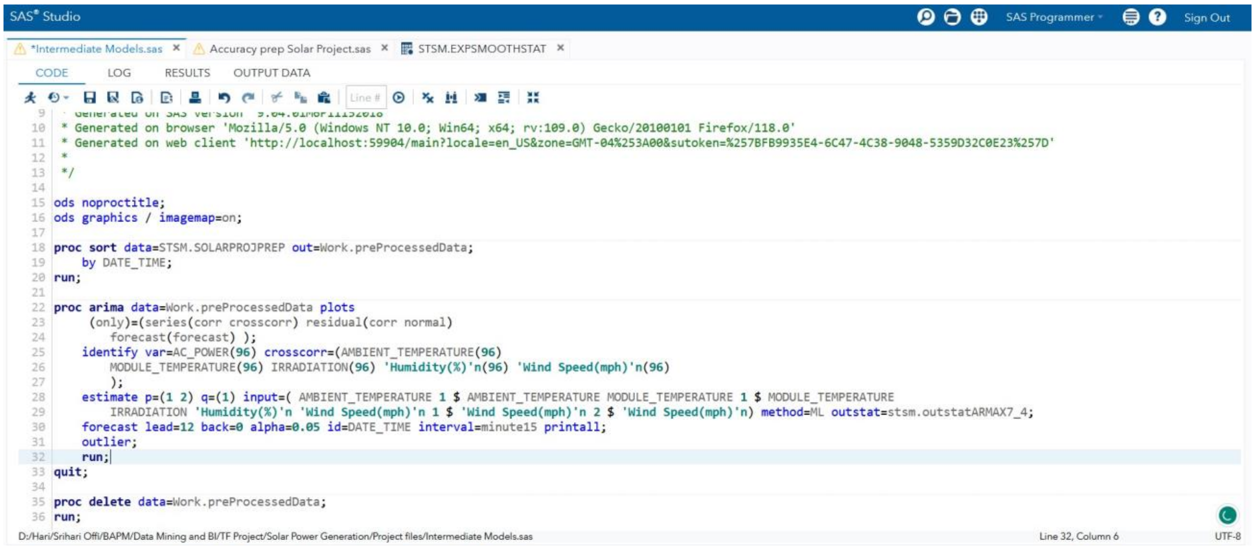


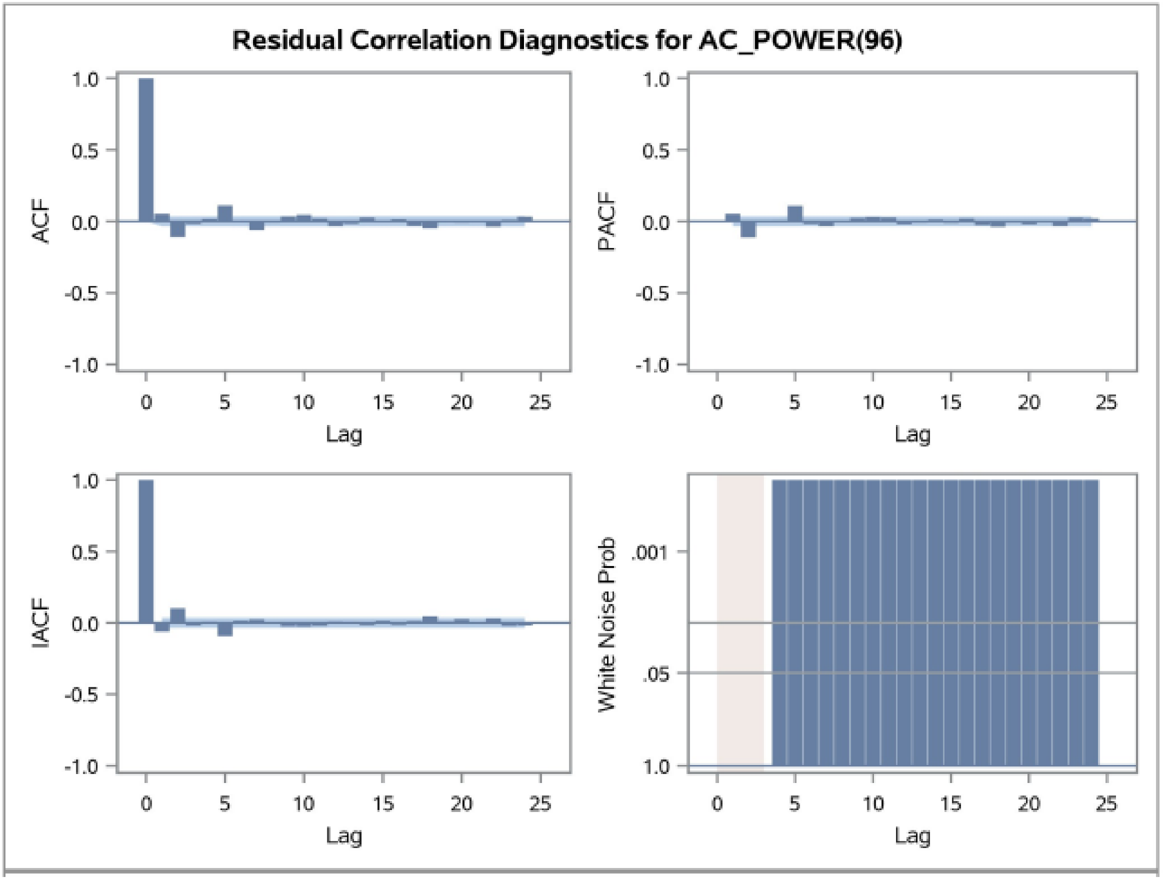
<u>Observation</u>

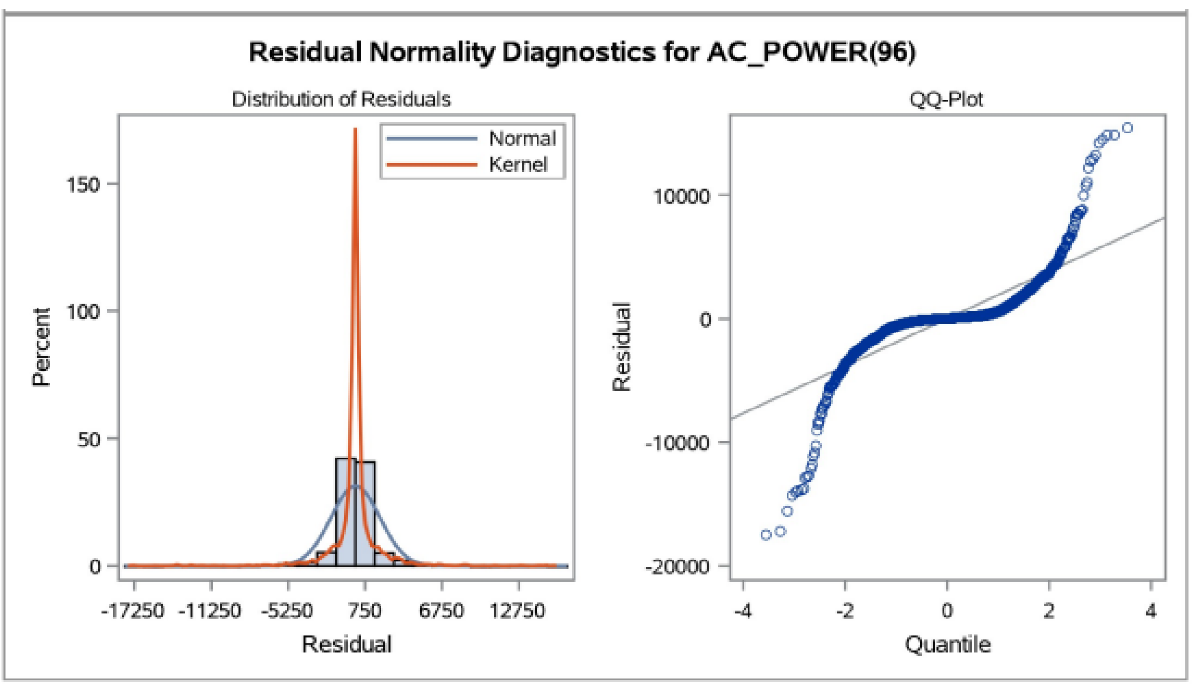
The white noise probability test reveals that there are remaining attributes to be modeled in the series, indicating the need to explore alternative models for further examination and there is significance in the IACF graph at lag 0 and an exponential decrease in the significance at further lags. Also, we notice a sinusoidal lagged effect in the PACF graph as well.

Hence, we opted to try other parameter tuning options.

<u>ARIMAX (p=2,d=0, q=1 ; P=0,D=1,Q=0)</u>

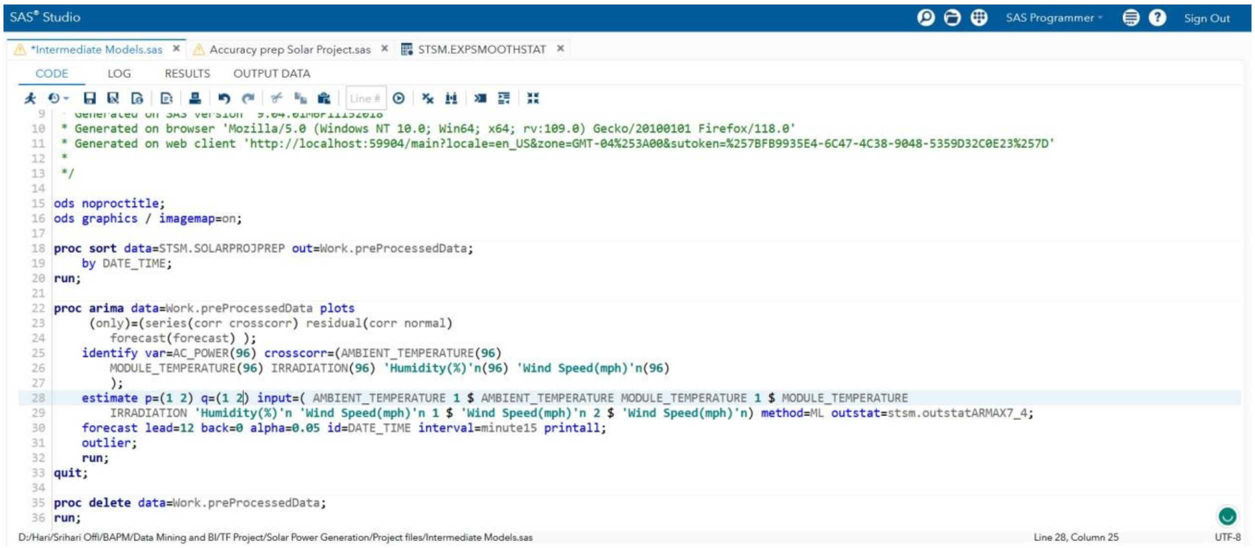



Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

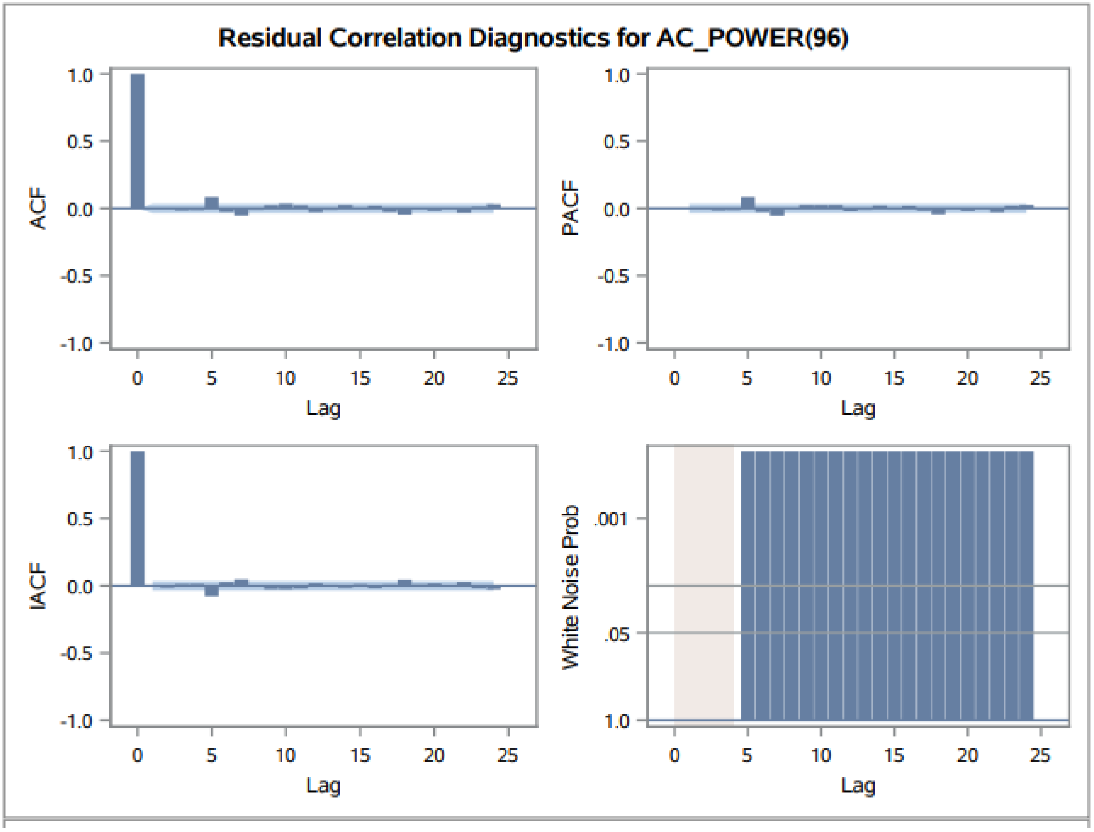

Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
Residual
Quantile
150
100
50
0
-17250
-11250
-5250
750
6750
12750
10000
-10000
-20000
-4
-2
2
4

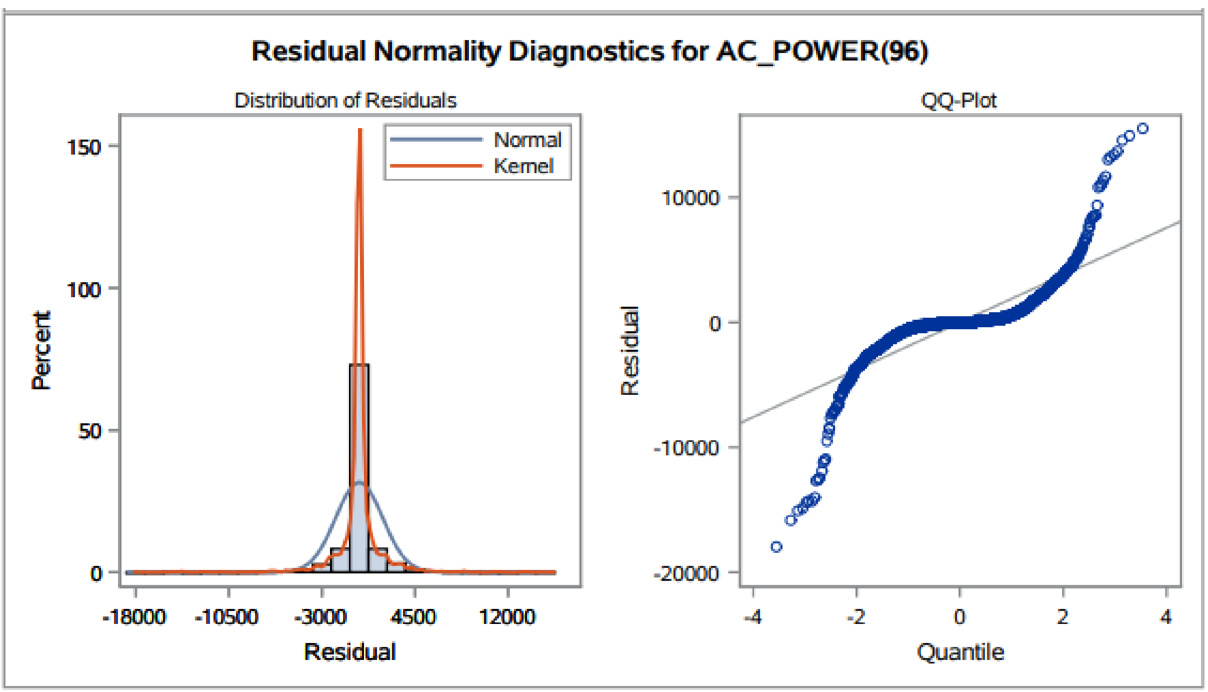
Even in this case, the white noise probability test fails. There is still signal that can be modelled. With these parameters, we still find a significant sinusoidal wave within the PACF and an exponential decrease with later lag significances. Hence, we can perform further parameter tuning in the model.

<u>ARIMAX (p=2,d=0,q=2 ; P=0,D=1,Q=0)</u>



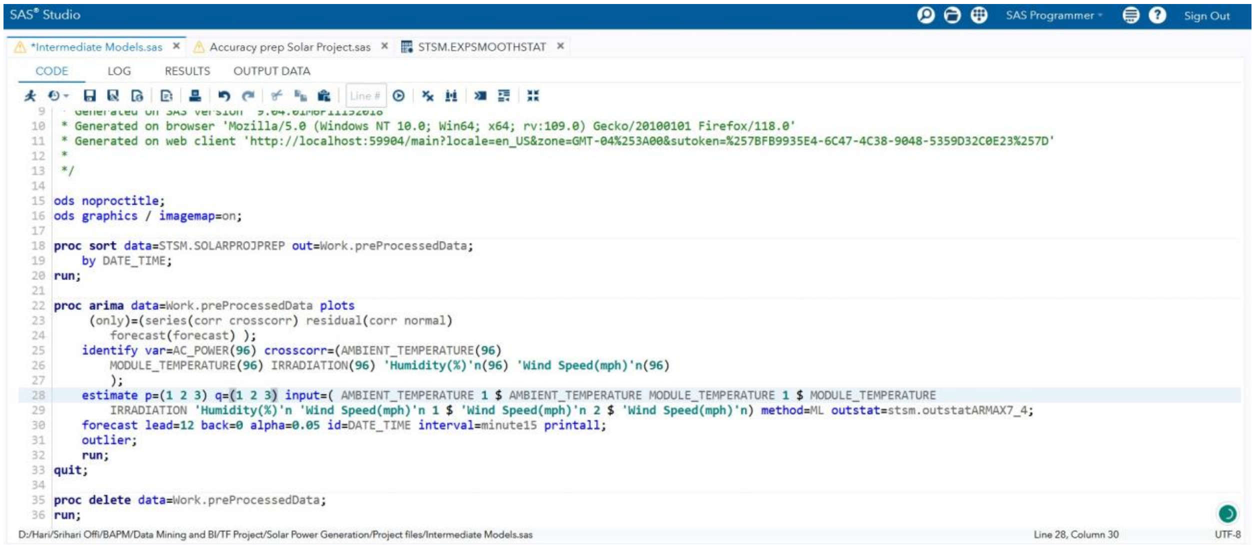
```
* Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0'
* Generated on web client 'http://localhost:59904/main?locale=en_US&zone=GMT-04%253A00&sutoken=%257BFB9935E4-6C47-4C38-9048-5359D32C0E23%257D'
*
*/

ods noproctitle;
ods graphics / imagemap=on;

proc sort data=STSM.SOLARPROJPREP out=Work.preProcessedData;
    by DATE_TIME;
run;

proc arima data=Work.preProcessedData plots
     (only)=(series(corr crosscorr) residual(corr normal)
        forecast(forecast) );
    identify var=AC_POWER(96) crosscorr=(AMBIENT_TEMPERATURE(96)
        MODULE_TEMPERATURE(96) IRRADIATION(96) 'Humidity(%)'n(96) 'Wind Speed(mph)'n(96)
        );
    estimate p=(1 2) q=(1 2) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML outstat=stsm.outstatARMAX7_4;
    forecast lead=12 back=0 alpha=0.05 id=DATE_TIME interval=minute15 printall;
    outlier;
    run;
quit;

proc delete data=Work.preProcessedData;
run;
```

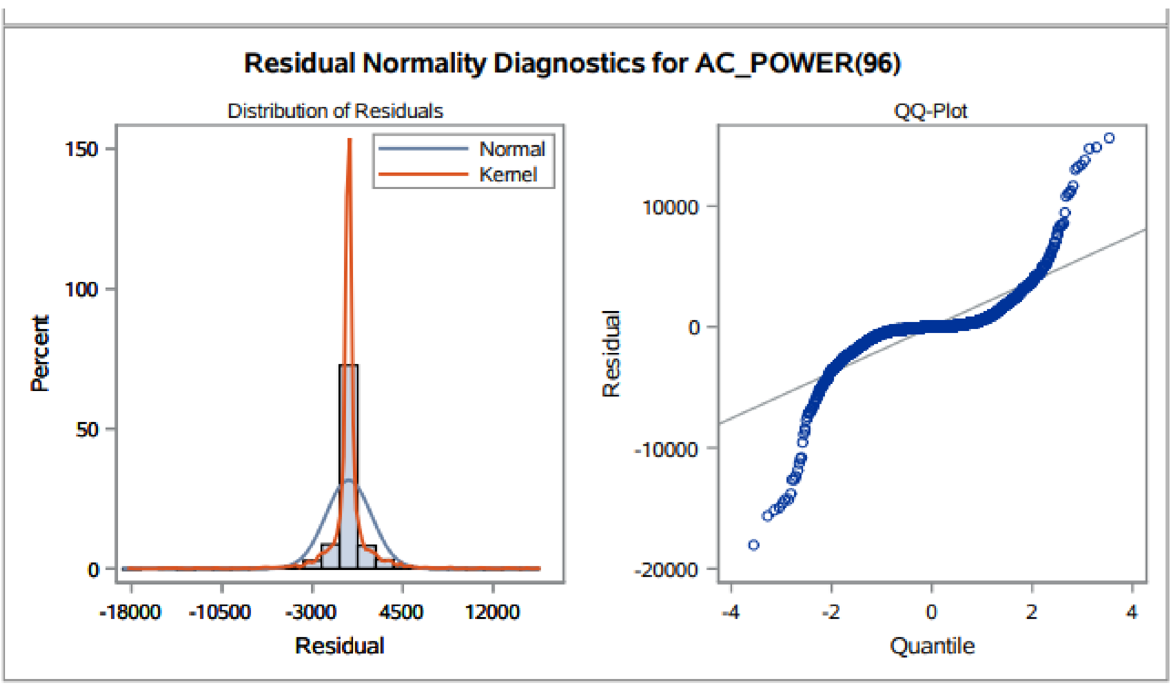


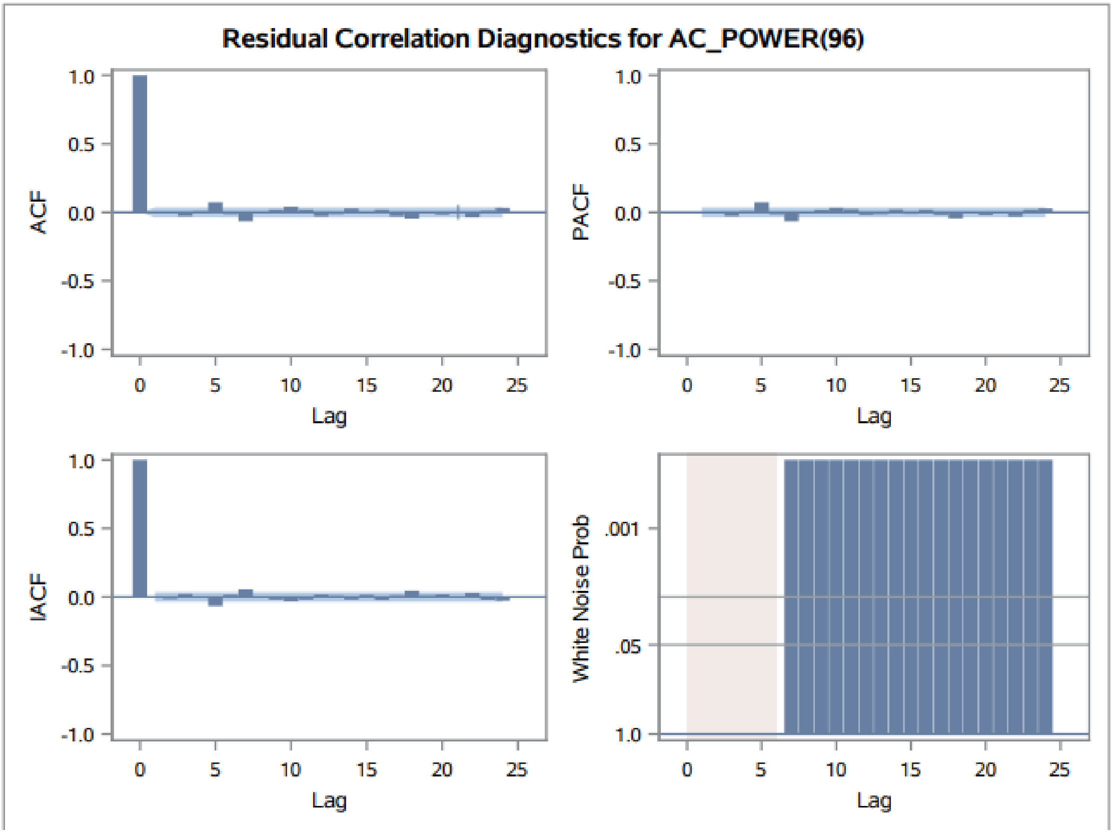


The white noise probability test still fails and there are peaks indicating significance at different lags within the PACF and IACF graphs. Hence, we continue with further modelling.

ARIMAX (p=3,d=0,q=3 ; P=0,D=1,Q=0)

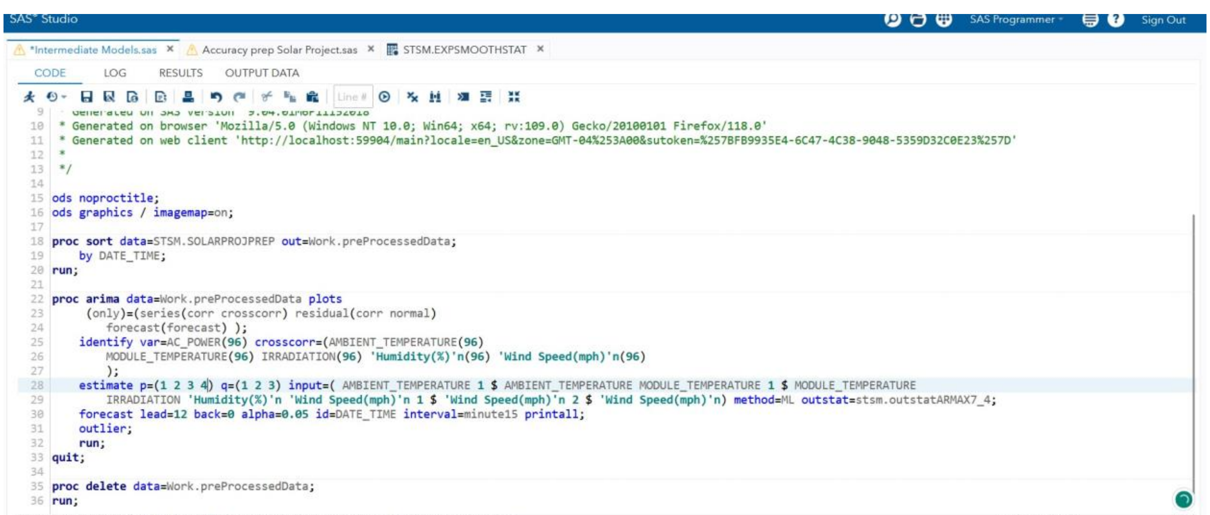


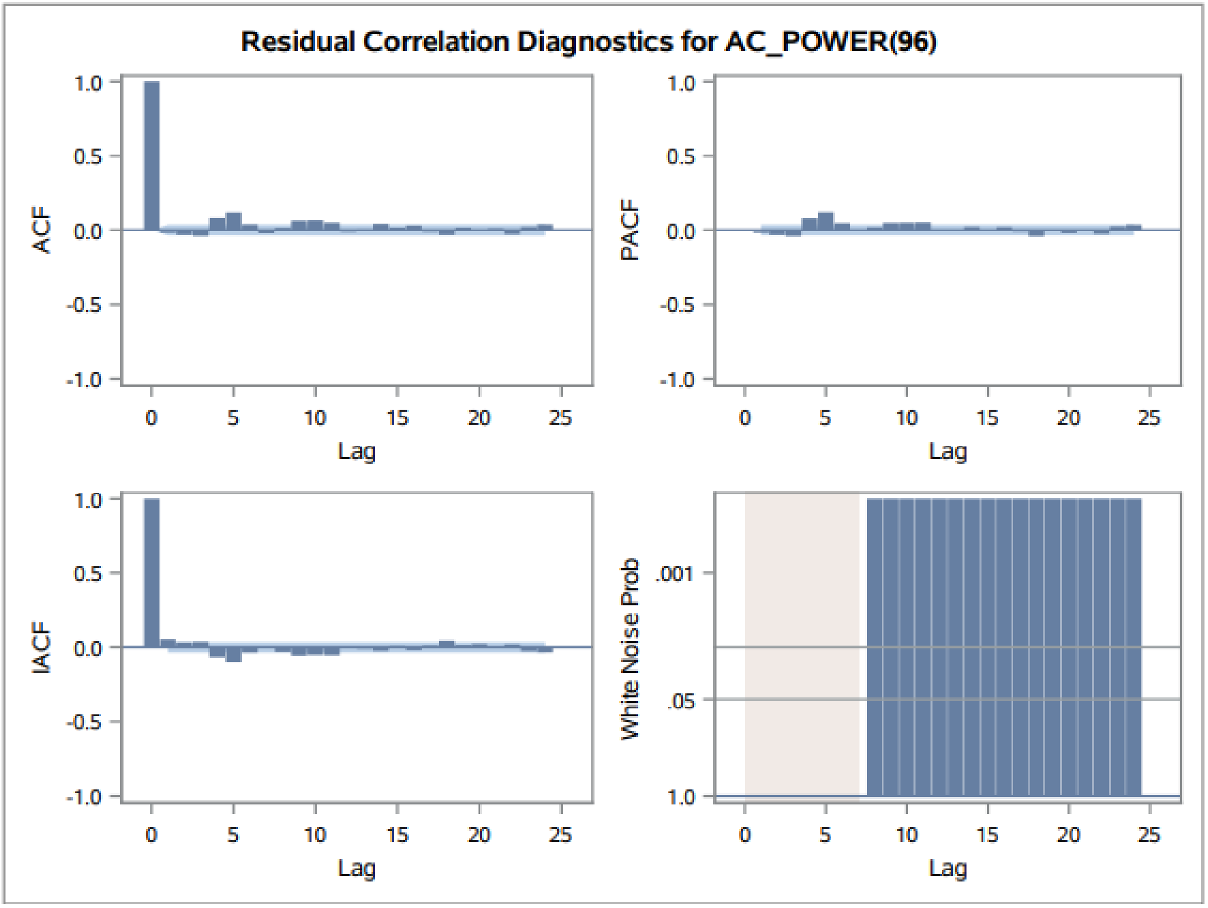


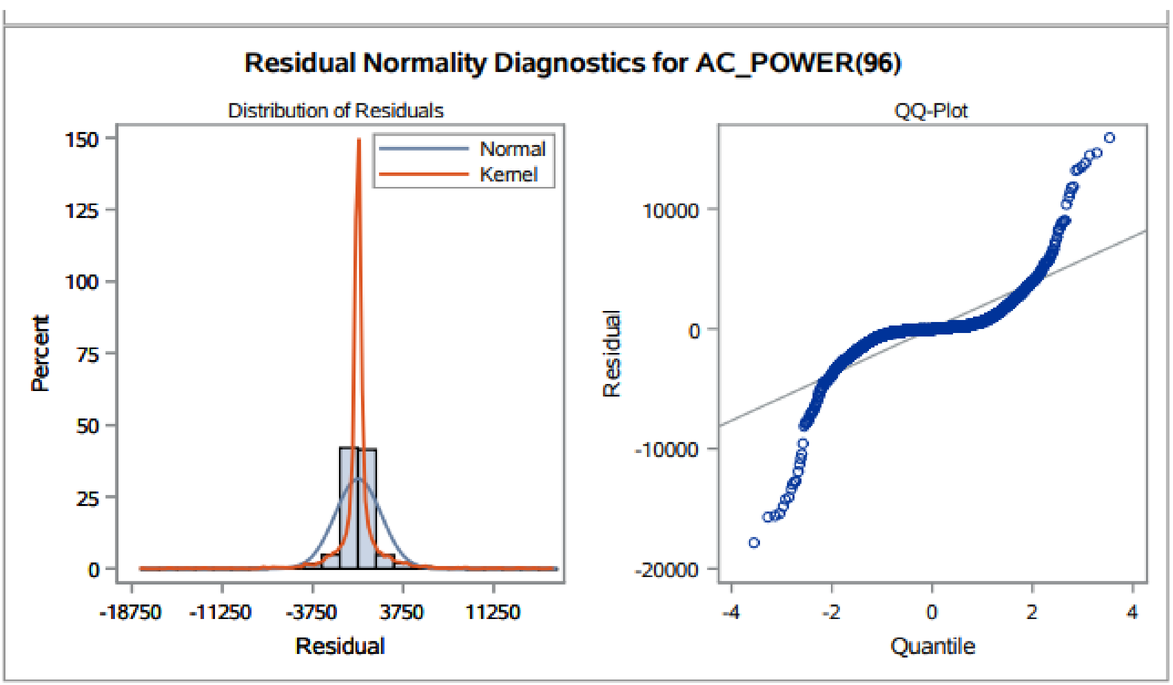


The white noise probability test still fails and reveals that there are remaining attributes to be modeled in the series. We will proceed with further modelling.

<u>ARIMAX (p=4, d=0, q=3 ; P=0,D=1,Q=0)</u>



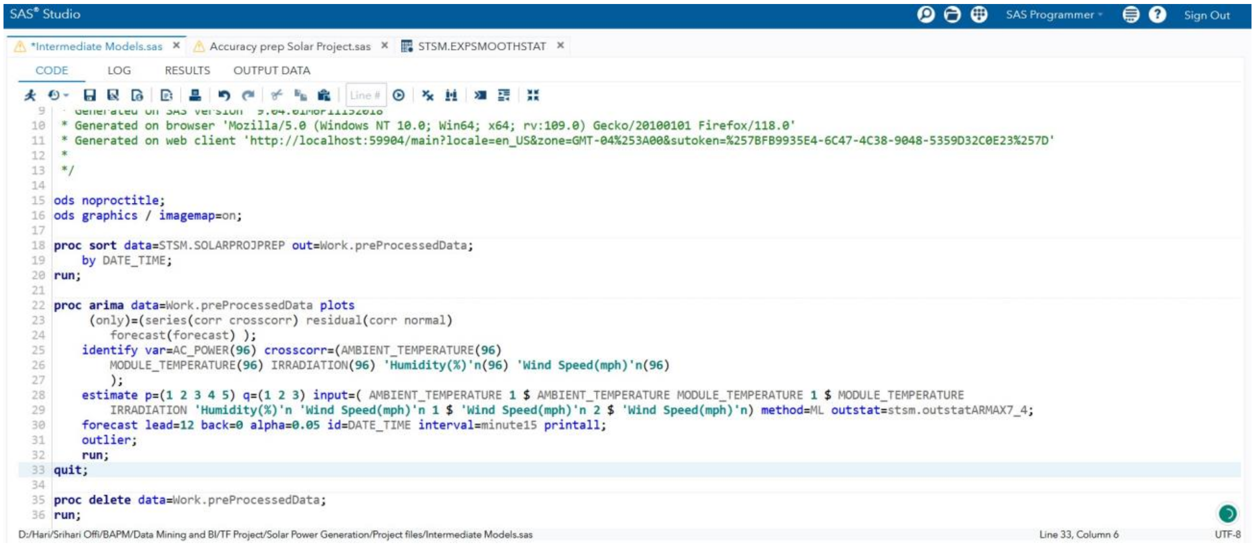




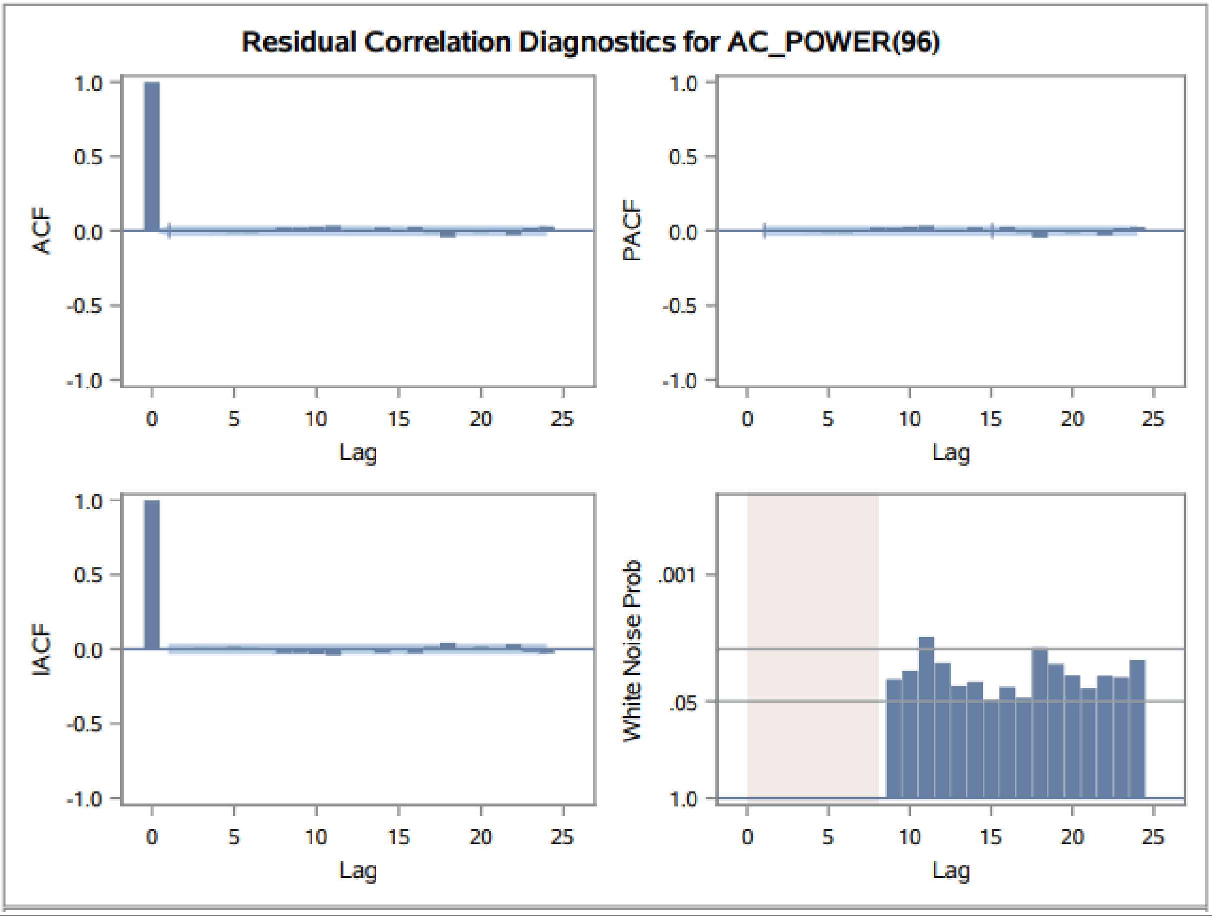
<u>Observation</u>

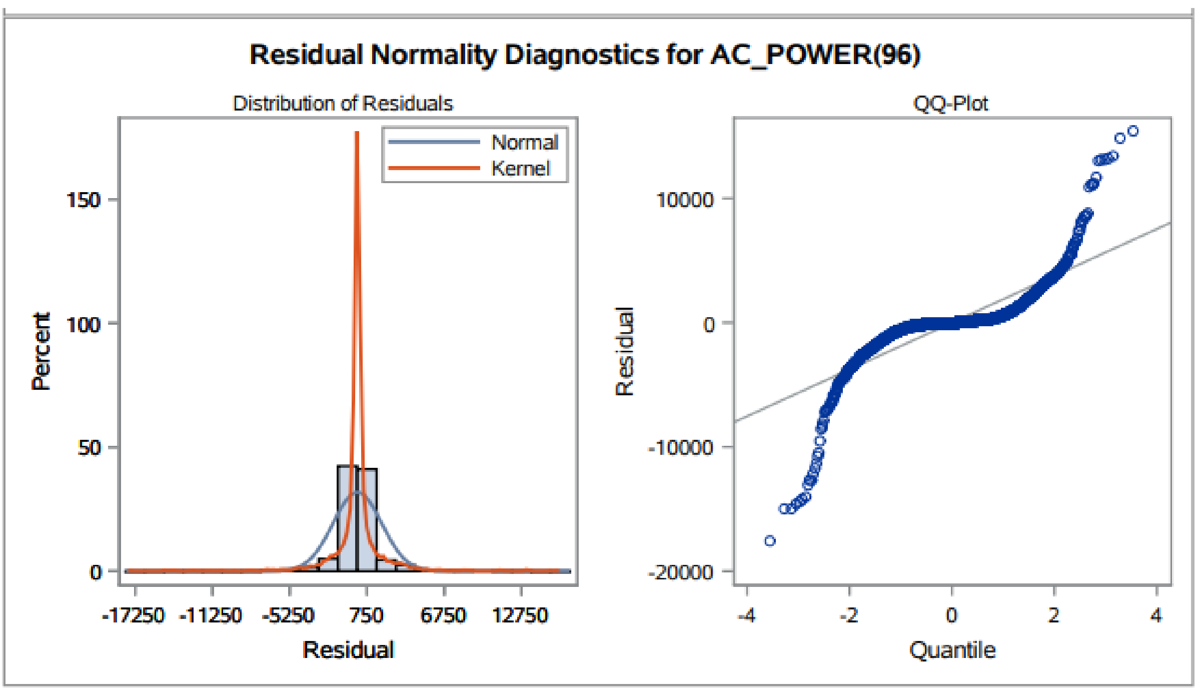
The white noise probability test reveals that there are remaining attributes to be modeled in the series with some less noise, indicating the need to explore alternative models for further examination and there is a significant peak in IACF at lag 0 and there some significant lags in the PACF. Hence, we opted to go for the next models.

ARIMAX (p=5,d=0,q=3 ; P=0,D=1,Q=0)




```
 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0'
 * Generated on web client 'http://localhost:59904/main?locale=en_US&zone=GMT-04%253A00&sutoken=%257BFB9935E4-6C47-4C38-9048-5359D32C0E23%257D'
 *
 */

ods noproctitle;
ods graphics / imagemap=on;

proc sort data=STSM.SOLARPROJPREP out=Work.preProcessedData;
    by DATE_TIME;
run;

proc arima data=Work.preProcessedData plots
     (only)=(series(corr crosscorr) residual(corr normal)
        forecast(forecast) );
    identify var=AC_POWER(96) crosscorr=(AMBIENT_TEMPERATURE(96)
        MODULE_TEMPERATURE(96) IRRADIATION(96) 'Humidity(%)'n(96) 'Wind Speed(mph)'n(96)
        );
    estimate p=(1 2 3 4 5) q=(1 2 3) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML outstat=stsm.outstatARMAX7_4;
    forecast lead=12 back=0 alpha=0.05 id=DATE_TIME interval=minute15 printall;
    outlier;
    run;
quit;

proc delete data=Work.preProcessedData;
run;
```

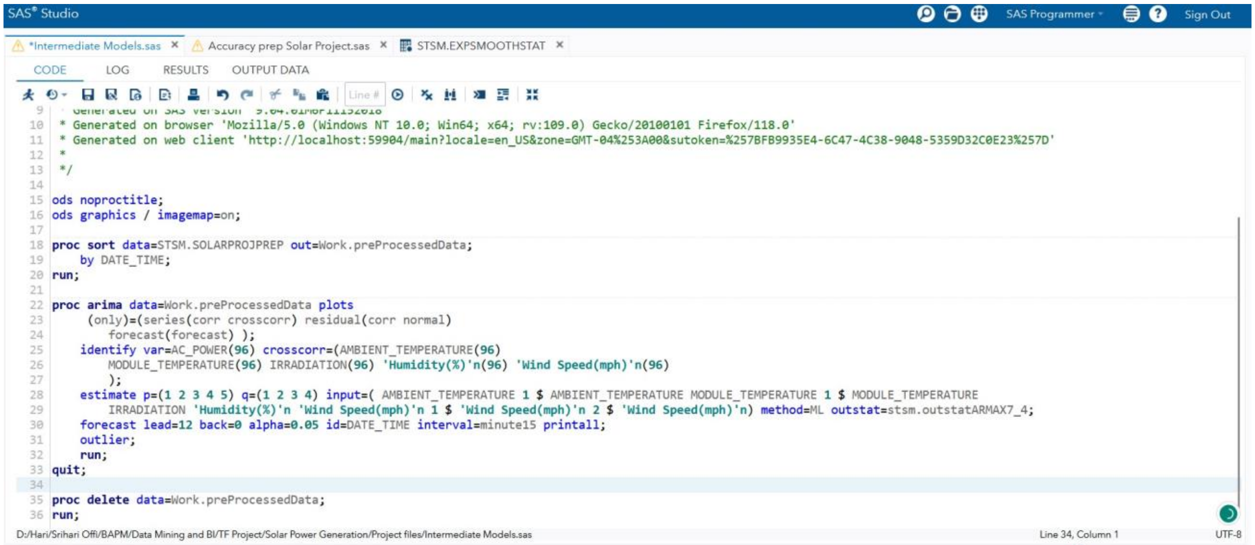

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

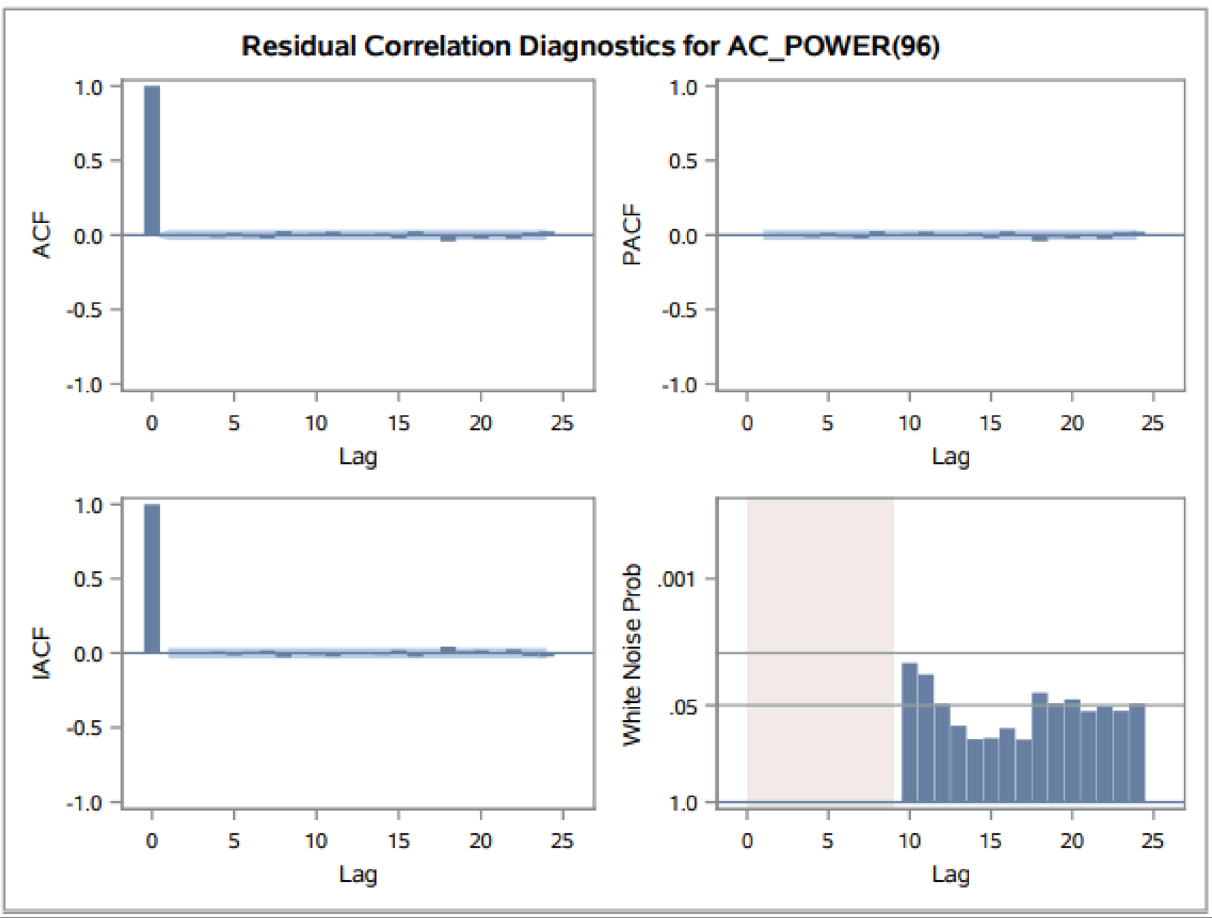

Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
150
100
50
0
-17250
-11250
-5250
750
6750
12750
Residual
10000
0
-10000
-20000
-4
-2
0
2
4
Quantile

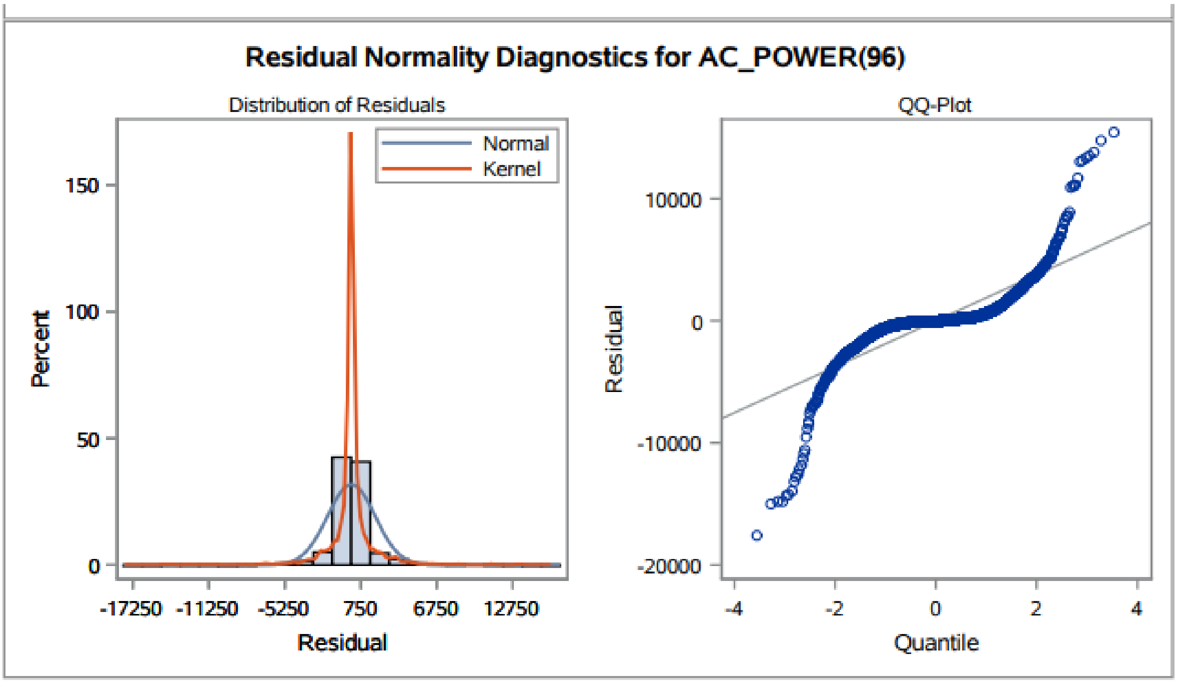
Observation

The white noise probability test is better than the previous models, but it still fails. Also, there is significance shown in the IACF and PACF plots. Hence, doing further modelling.

ARIMAX1 (p=5,d=0,q=4 ; P=0,D=1,Q=0)



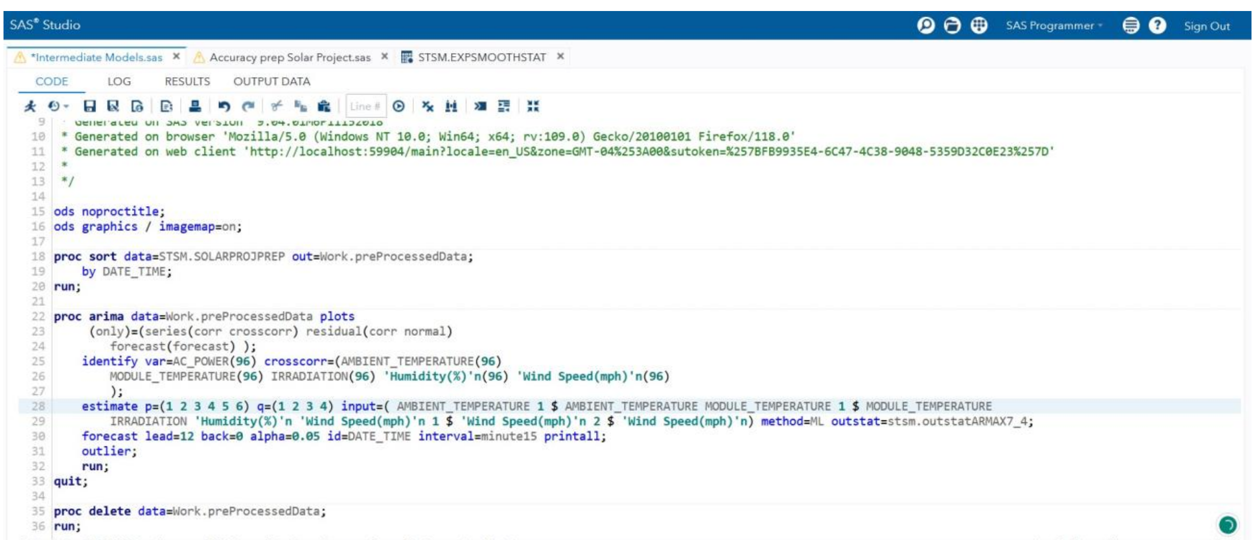


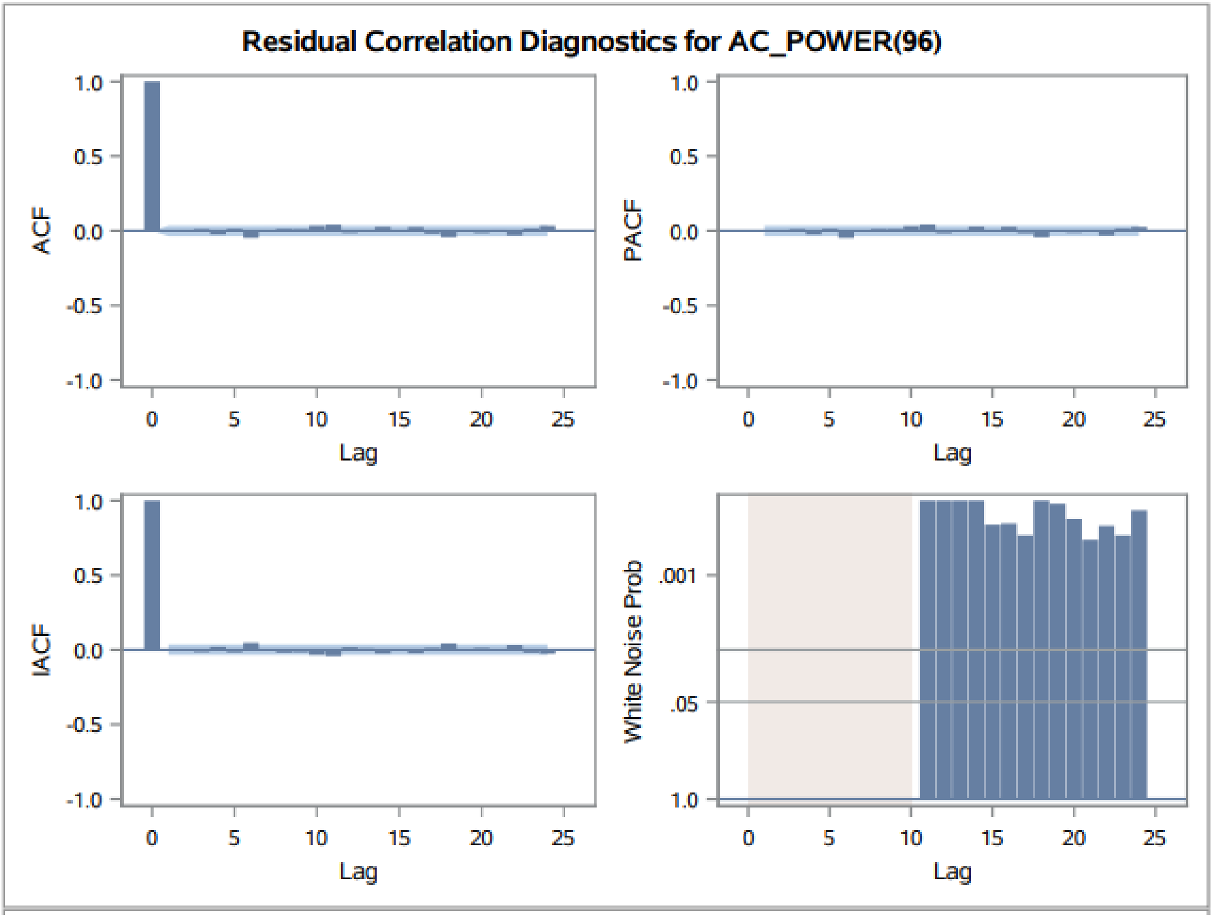


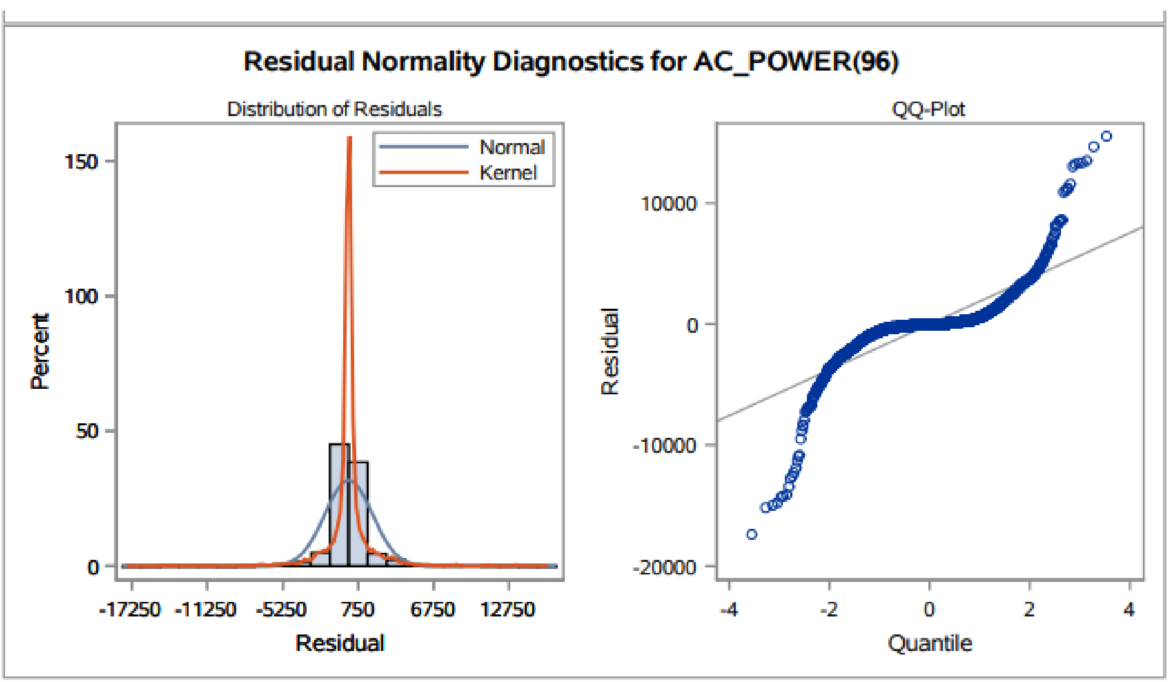
<u>Observation</u>

The white noise probability test has improved, but it is still not good and fails. Hence, we proceed to do further modelling to do proper forecasting so that all the signal is captured.

<u>ARIMAX1 (p=6,d=0,q=4 ; P=0,D=1,Q=0)</u>



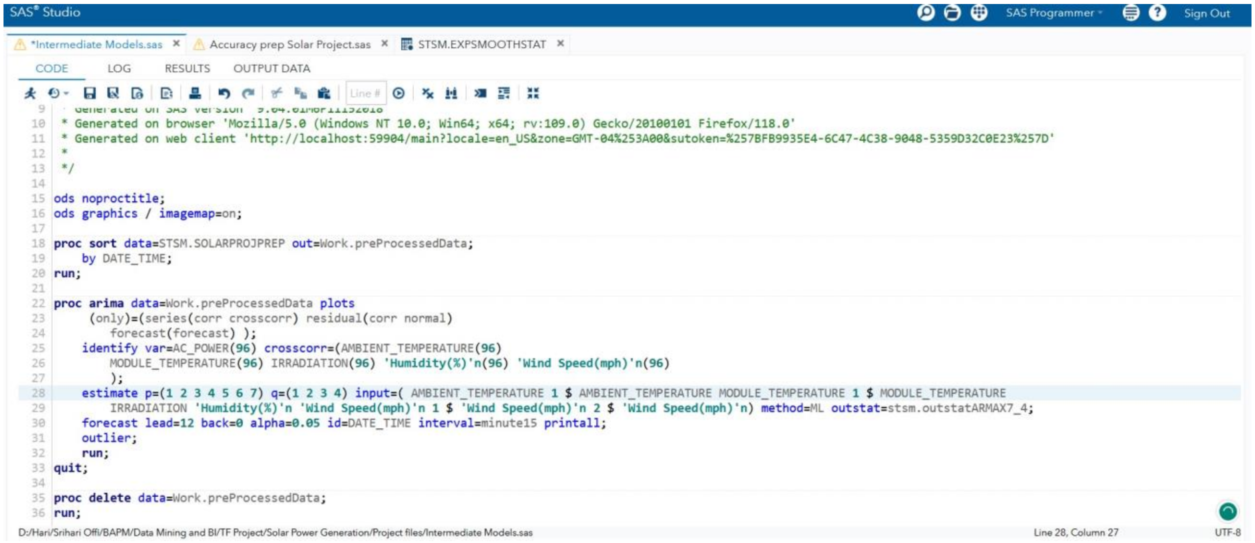

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

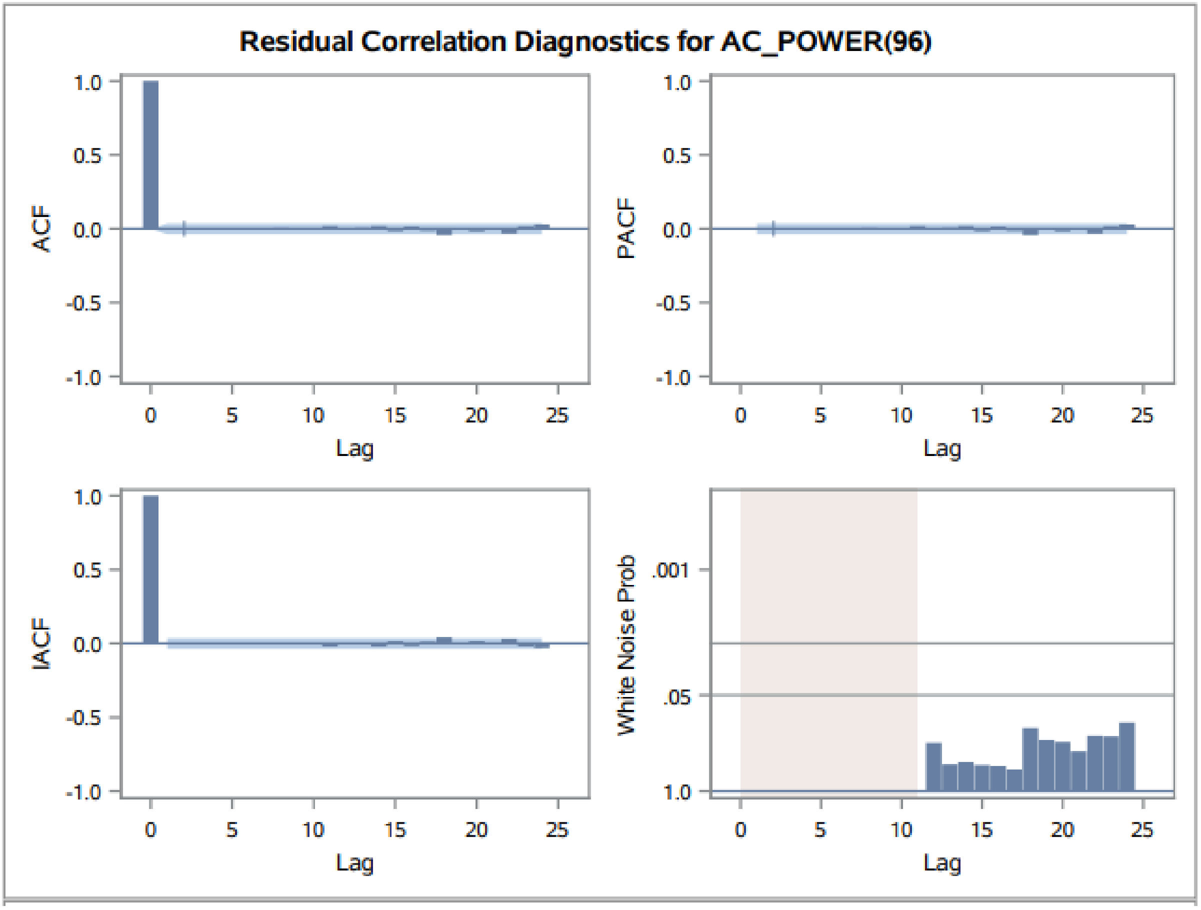

Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
150
100
50
0
-17250
-11250
-5250
750
6750
12750
Residual
10000
-10000
-20000
-4
-2
2
4
Quantile

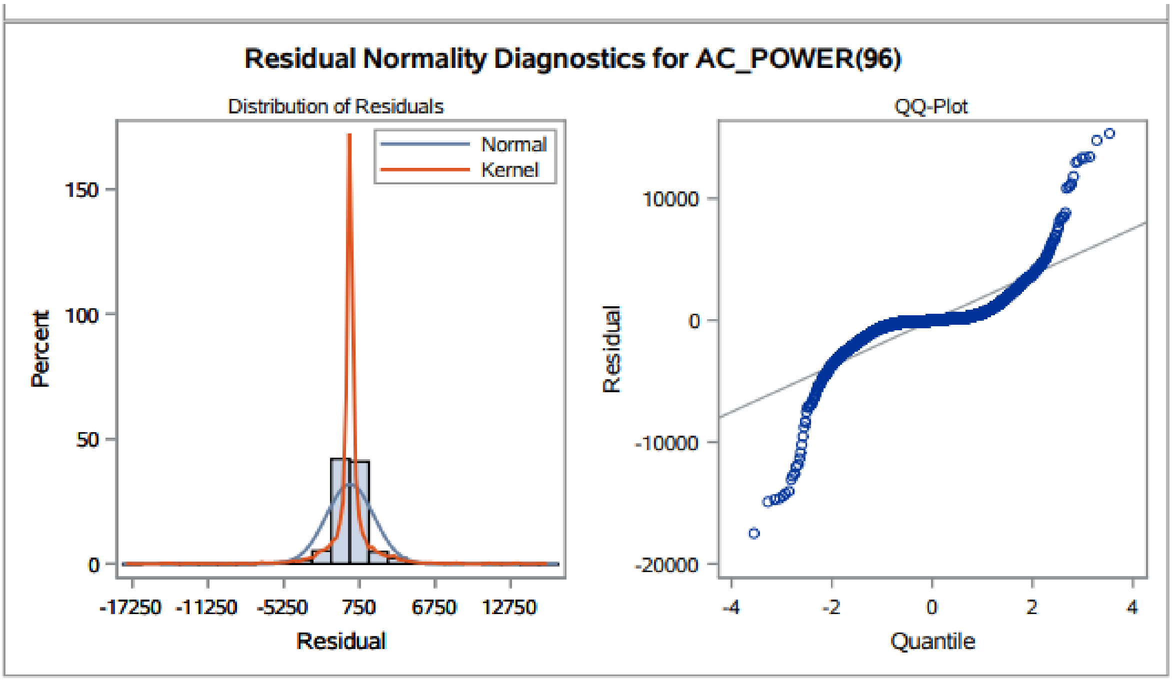
Observation

Similar to the previous conditions, the white noise probability test fails in this setup as well. We can do further modelling by changing the parameters.

ARIMAX (p=7,d=0,q=4 ; P=0,D=1,Q=0)



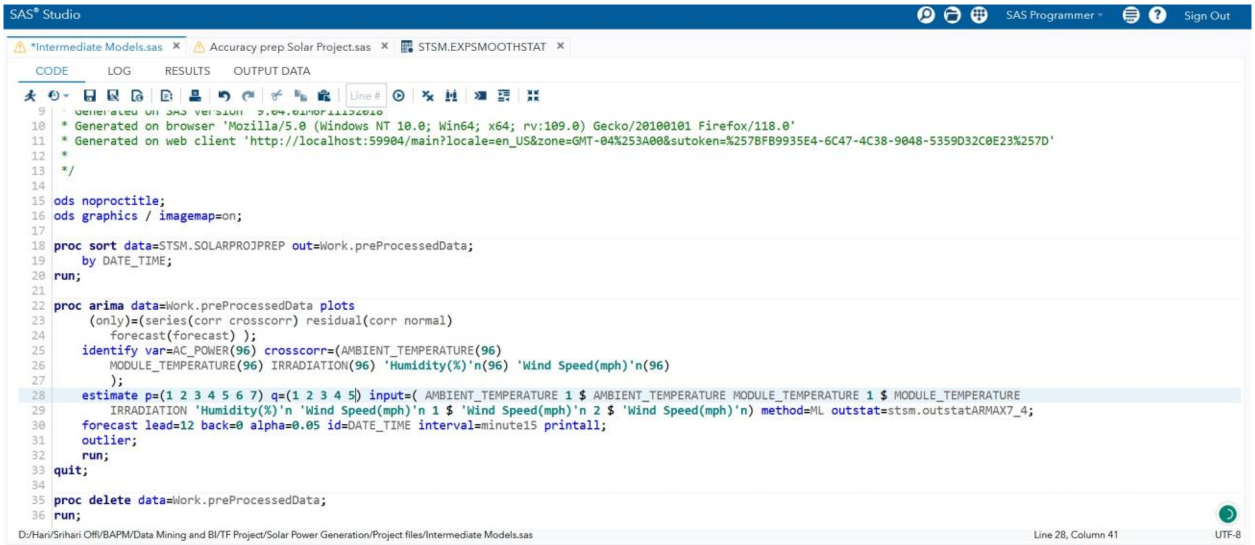

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25


Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
Residual
Quantile
150
100
50
0
-17250
-11250
-5250
750
6750
12750
10000
-10000
-20000
-4
-2
2
4

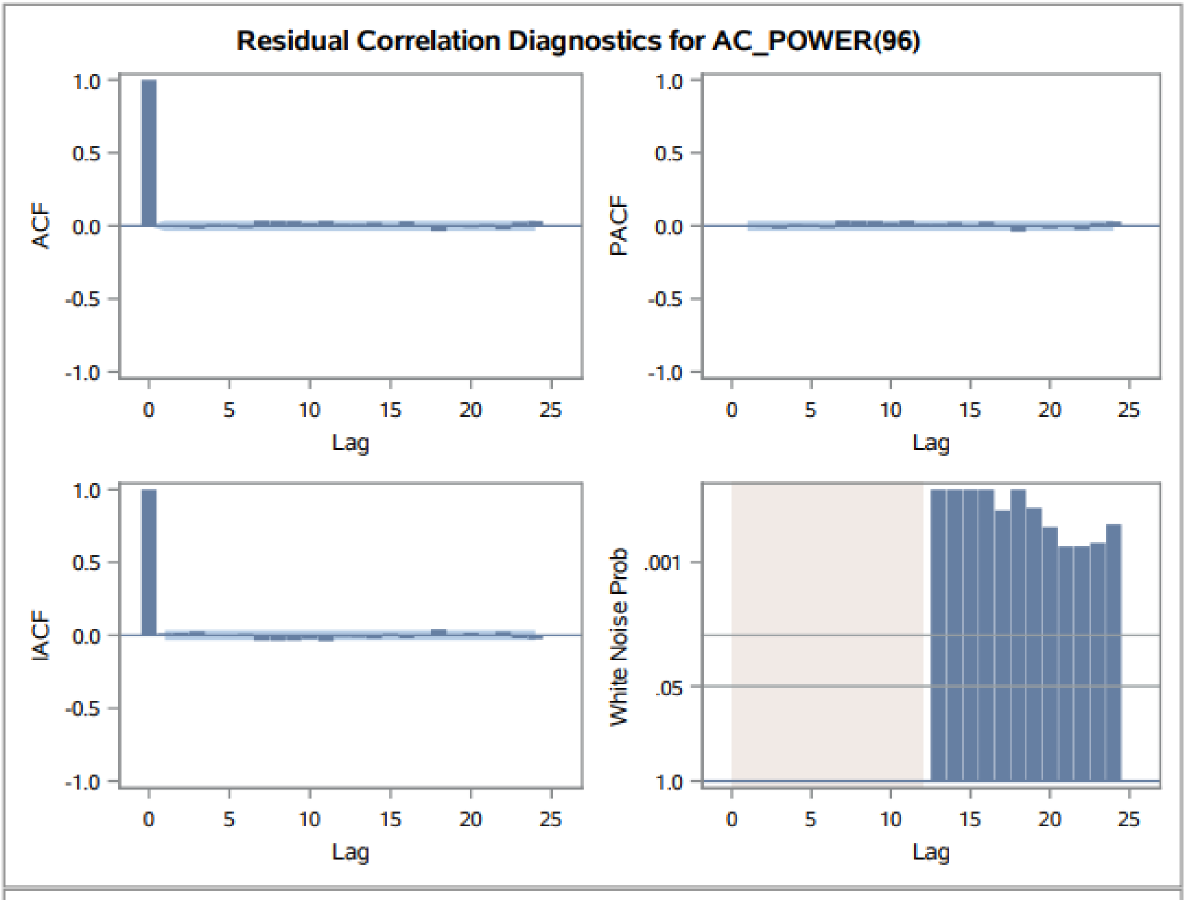
<u>Observation</u>

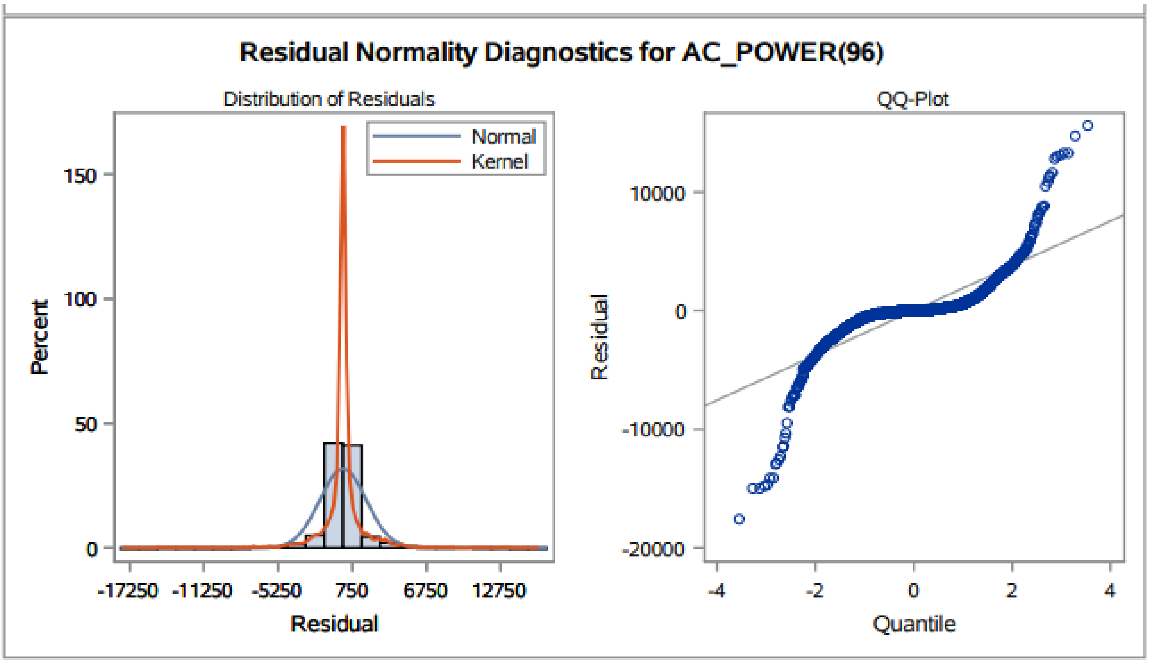
This model seems to be a good fit, because this has passed the white noise test. However, we will not the adjacent models with tuning to explore the best model for this time series data.

<u>ARIMAX (p=7,d=0,q=5 ; P=0,D=1,Q=0)</u>



Sunday, October 8, 2023 11:59:29

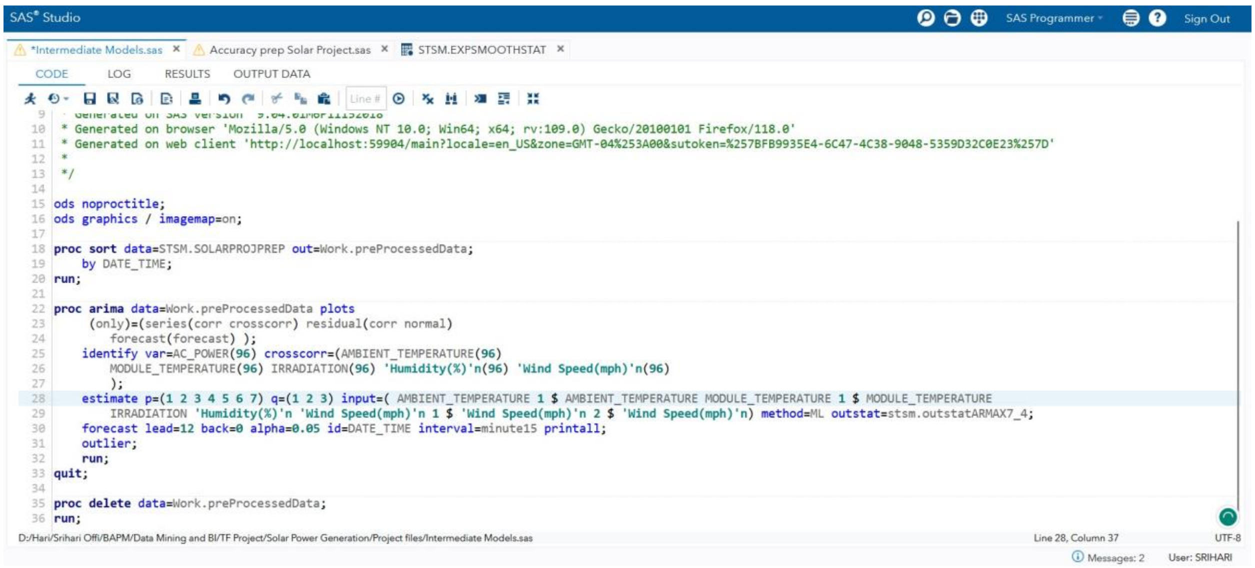


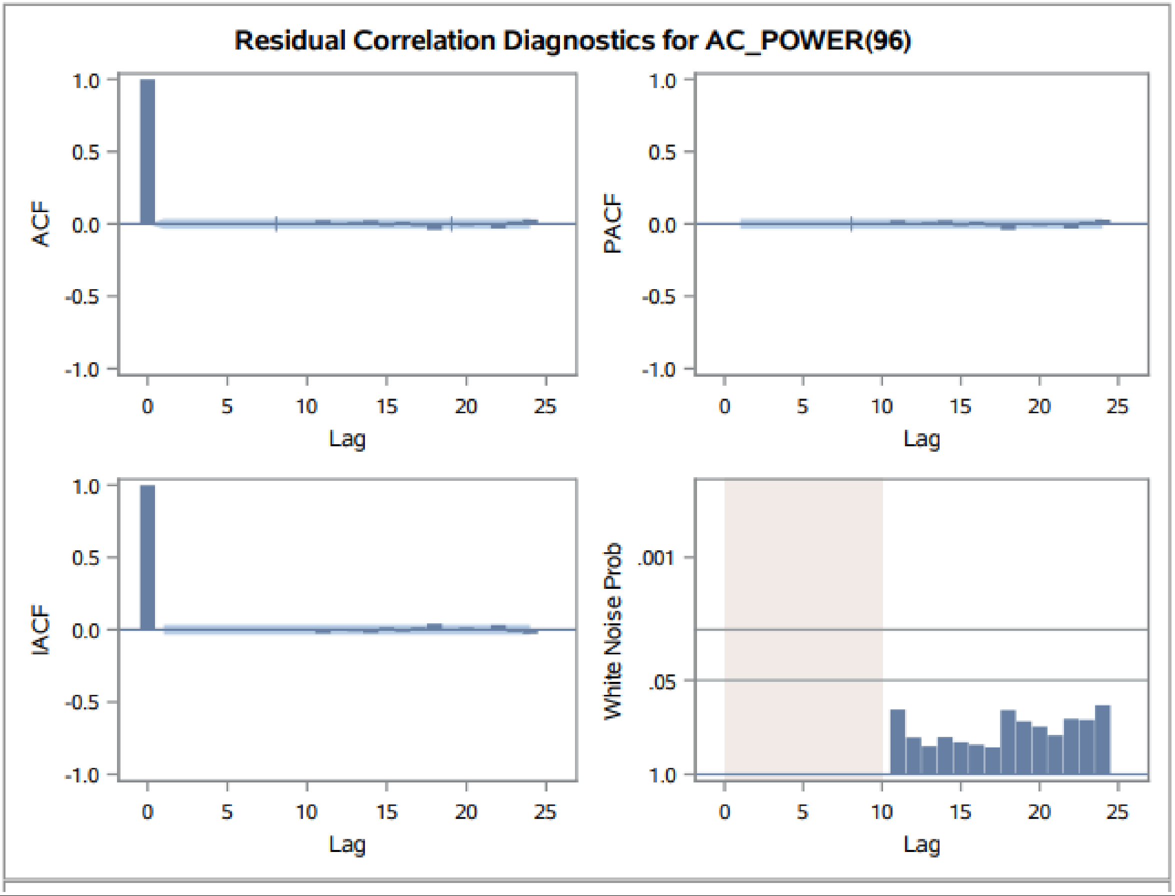


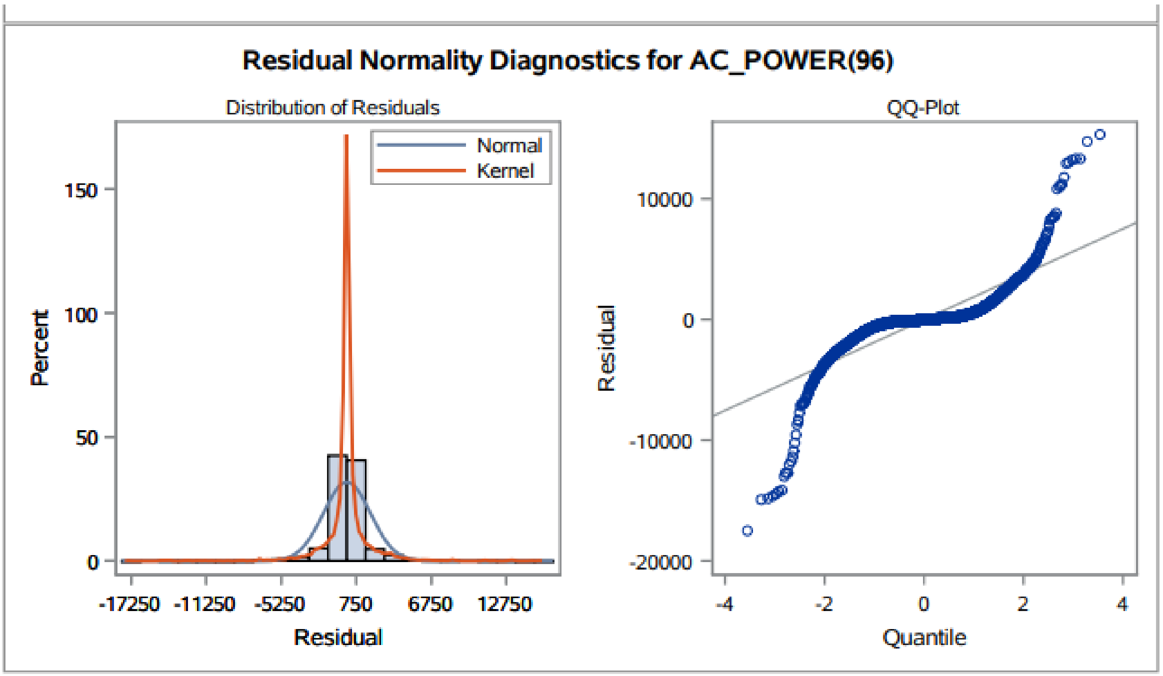
Observation

By changing the MA to 5 and running this model, it again fails the white noise test. Hence, we know that the q value is not 5. We will still continue our search for the best fit model with the help of other parameter tuning.

ARIMAX (p=7,d=0,q=3 ; P=0,D=1,Q=0)



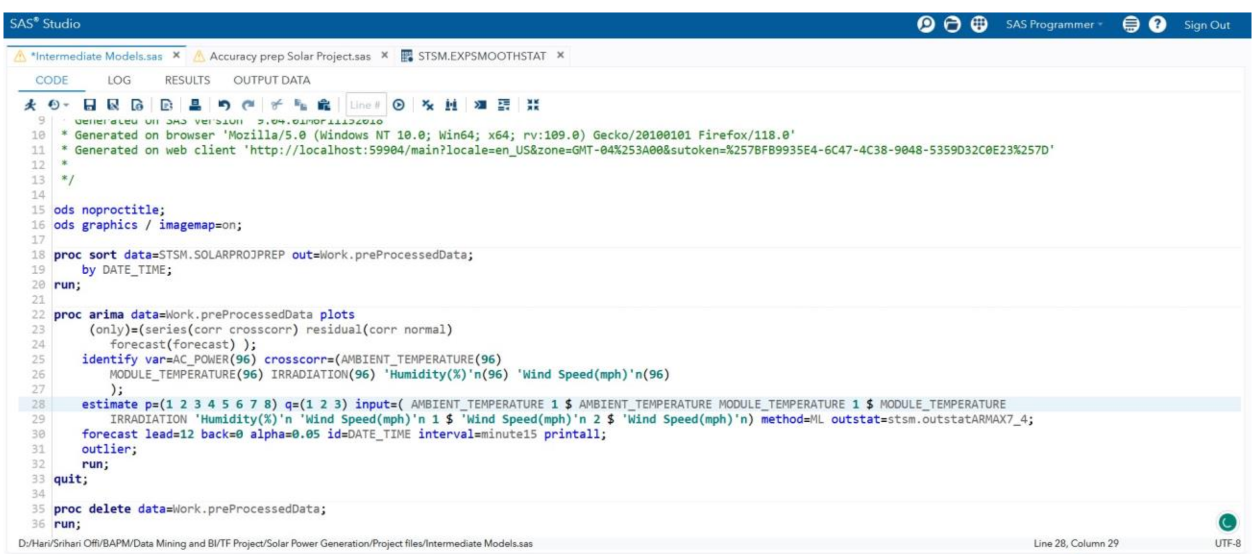

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
.001
.05
1.0

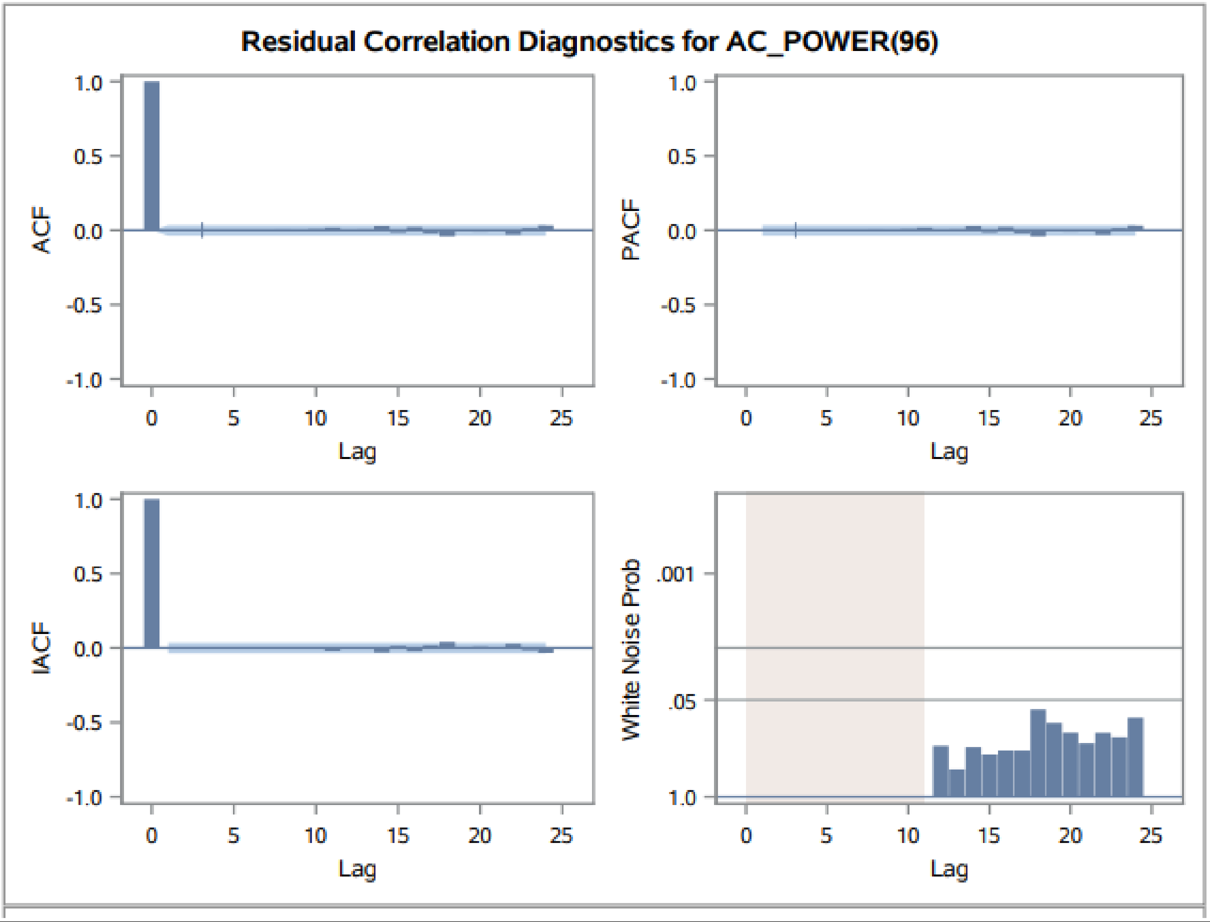


<u>Observation</u>

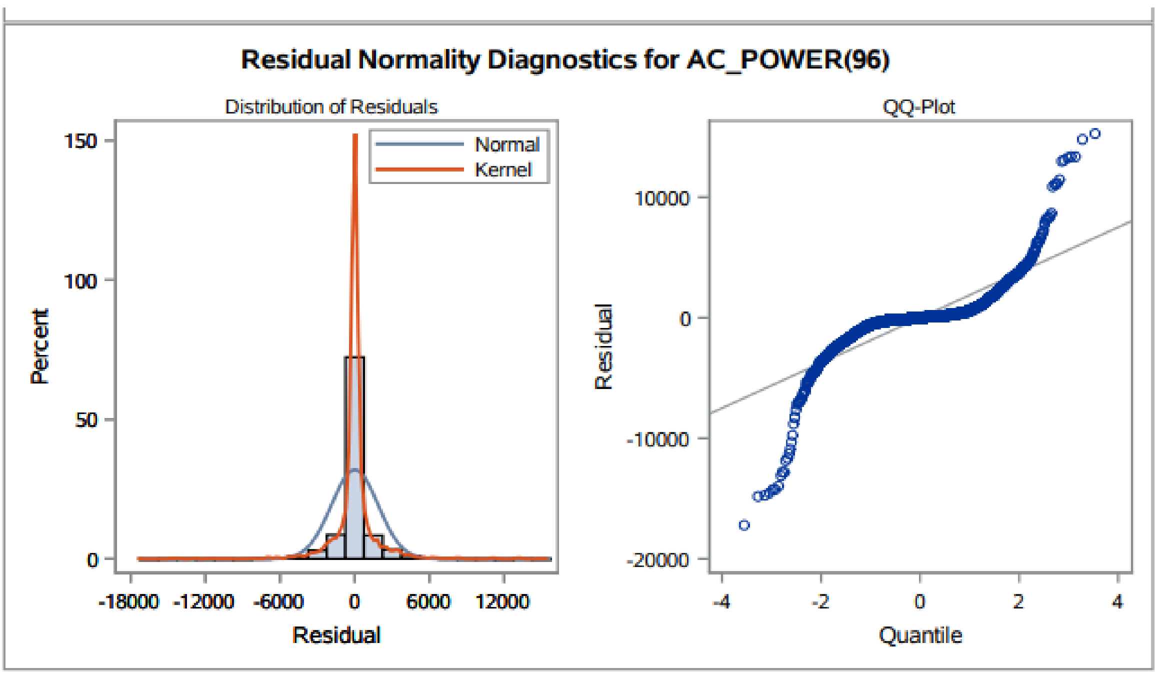
This model has also passed the white noise test. However, we will still do further parameter tuning to find the best fit model.

<u>ARIMAX (p=8,d=0,q=3 ; P=0,D=1,Q=0)</u>





Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

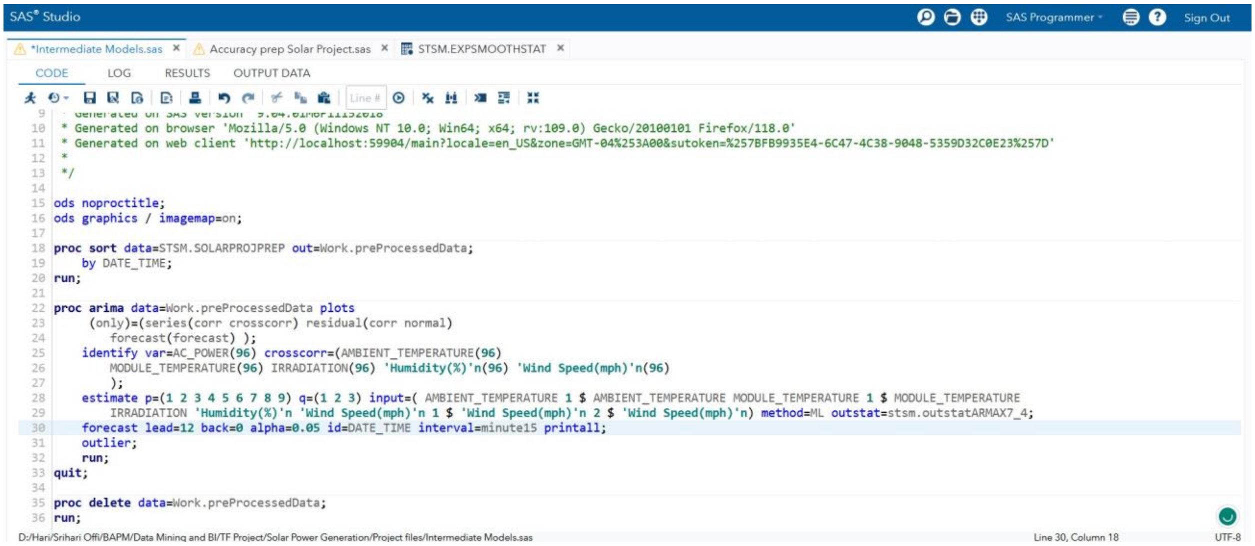


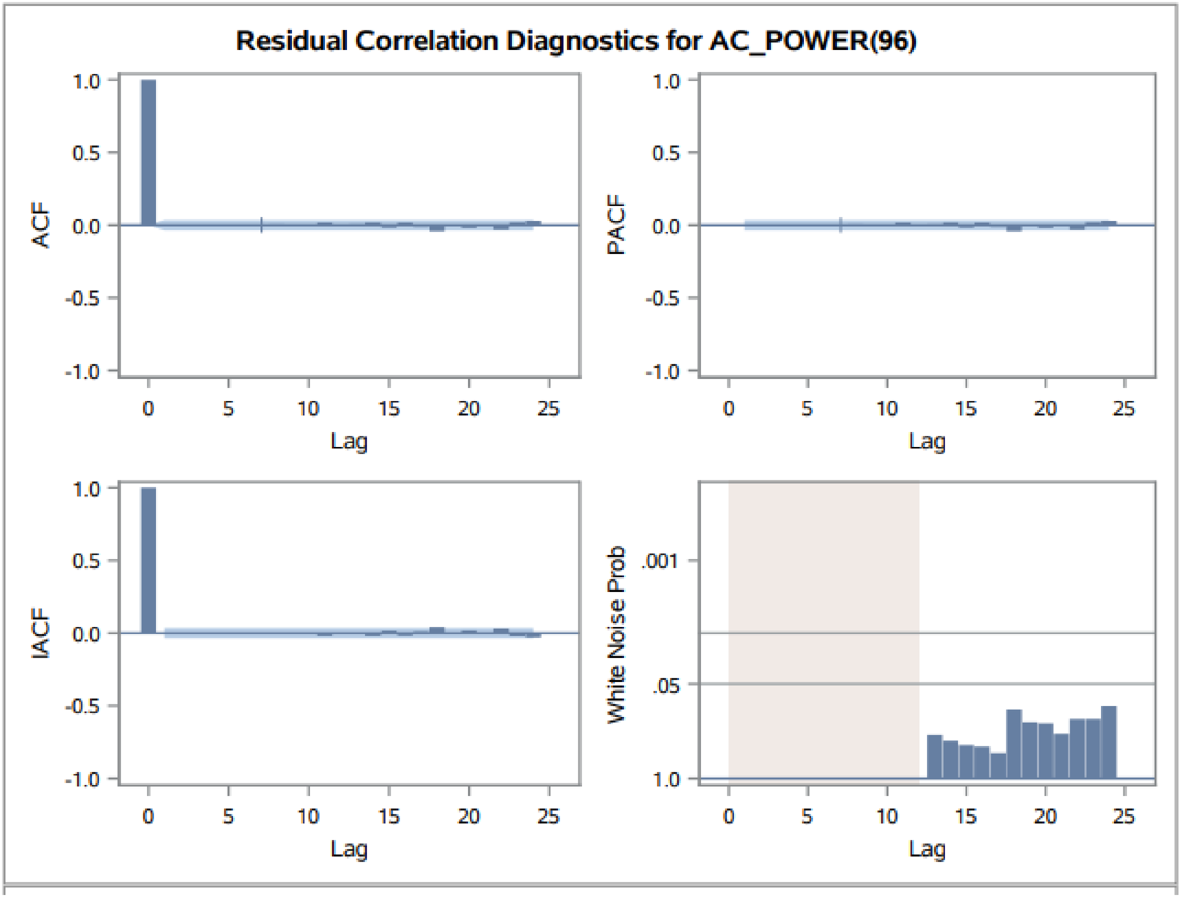
Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
Normal
Kernel
Percent
150
100
50
0
-18000
-12000
-6000
6000
12000
Residual
QQ-Plot
Residual
10000
-10000
-20000
Quantile
-4
-2
2
4

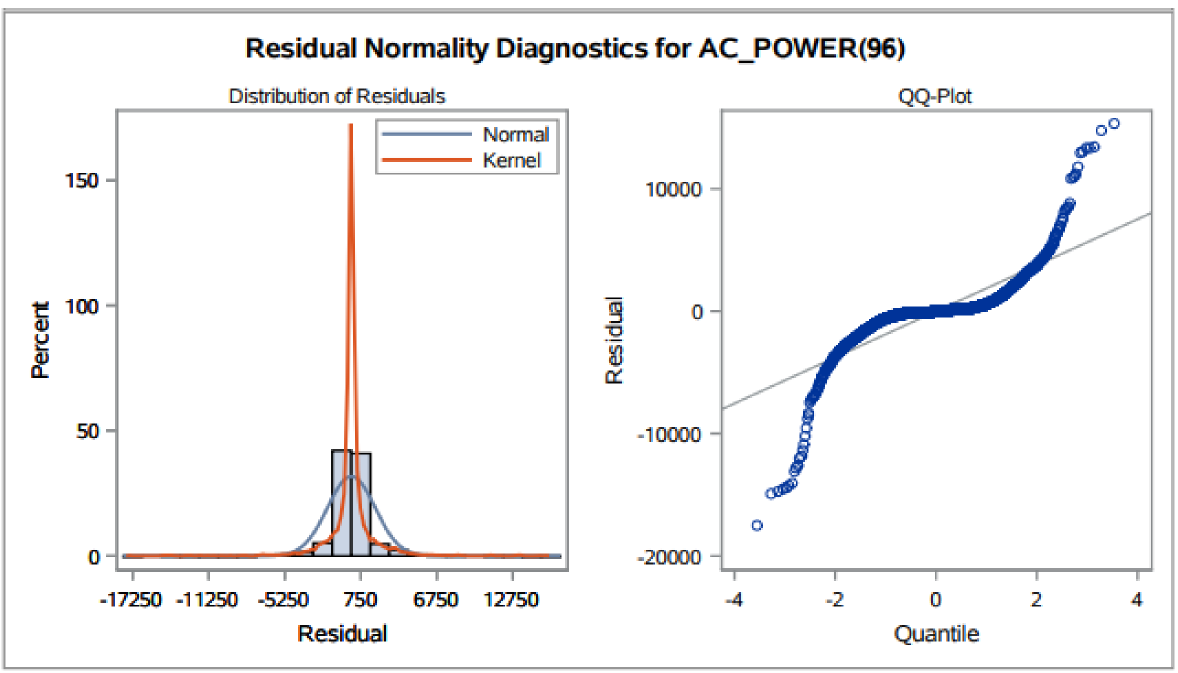
Observation

The white noise probability test we could see that there is very less signal and that it has passed. Hence this is one of the good models. We will do further modelling to see for the best fit model.

ARIMAX1 (p=9,d=0,q=3 ; P=0,D=1,Q=0)



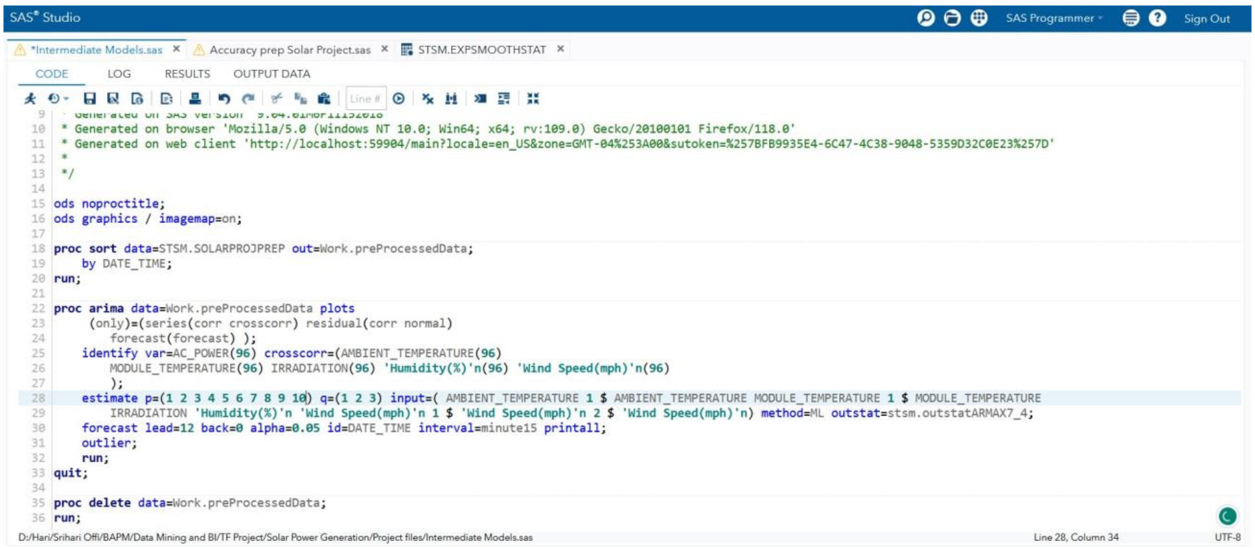

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

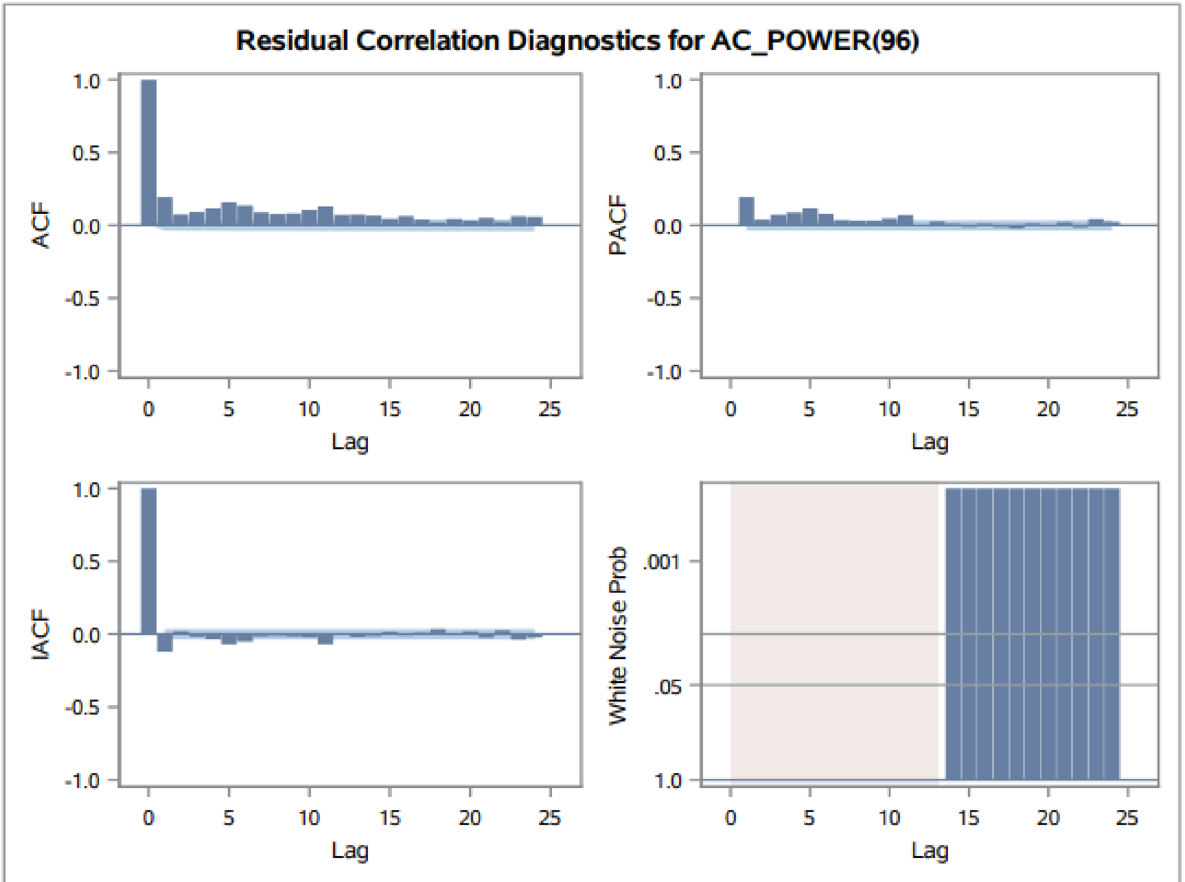

Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
150
100
50
0
-17250
-11250
-5250
750
6750
12750
Residual
10000
0
-10000
-20000
-4
-2
0
2
4
Quantile

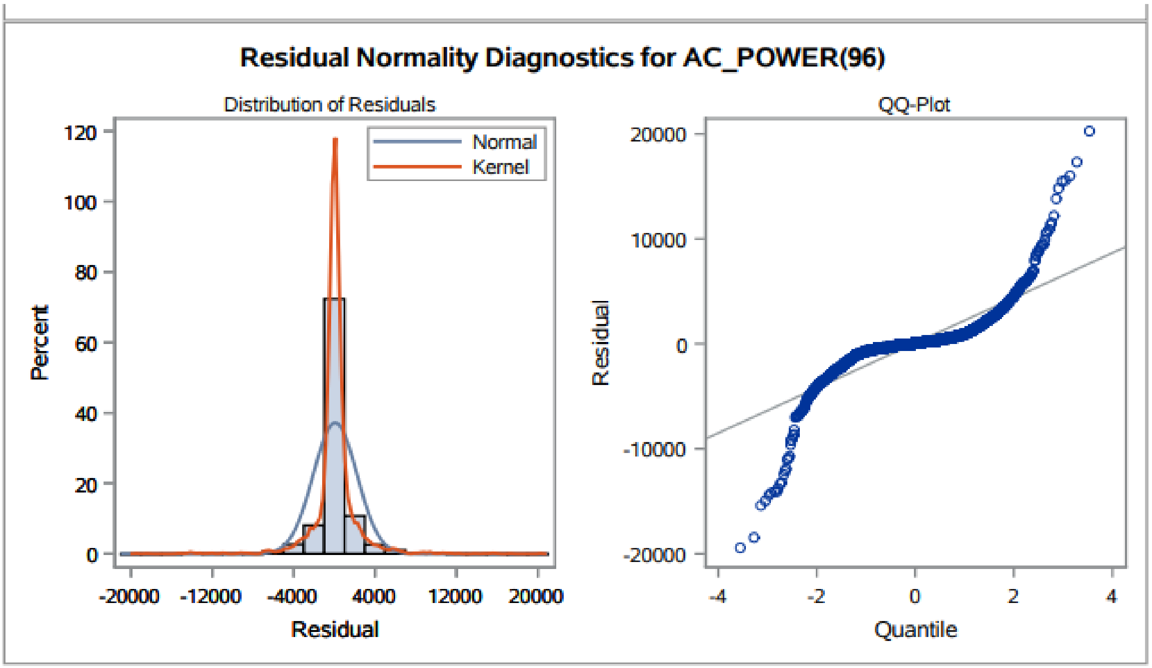
Observation

The white noise probability test passed for this scenario as well. So, this is also one of the models which we can consider. However, let us do some more parameter tuning to understand the best model fit.

ARIMAX (p=10,d=0,q=3 ; P=0,D=1,Q=0)



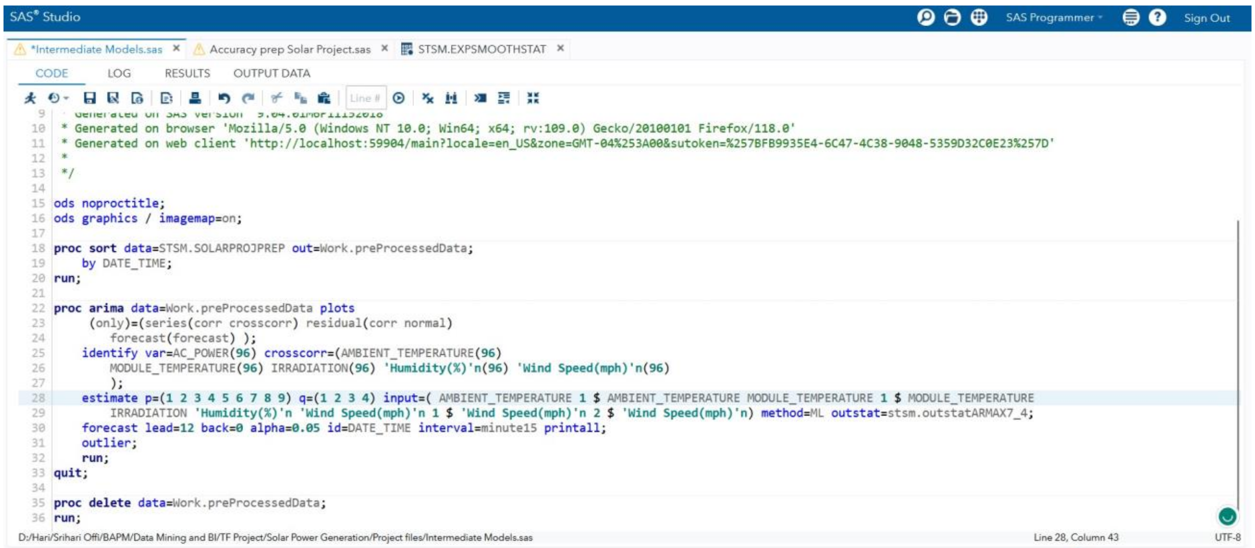

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

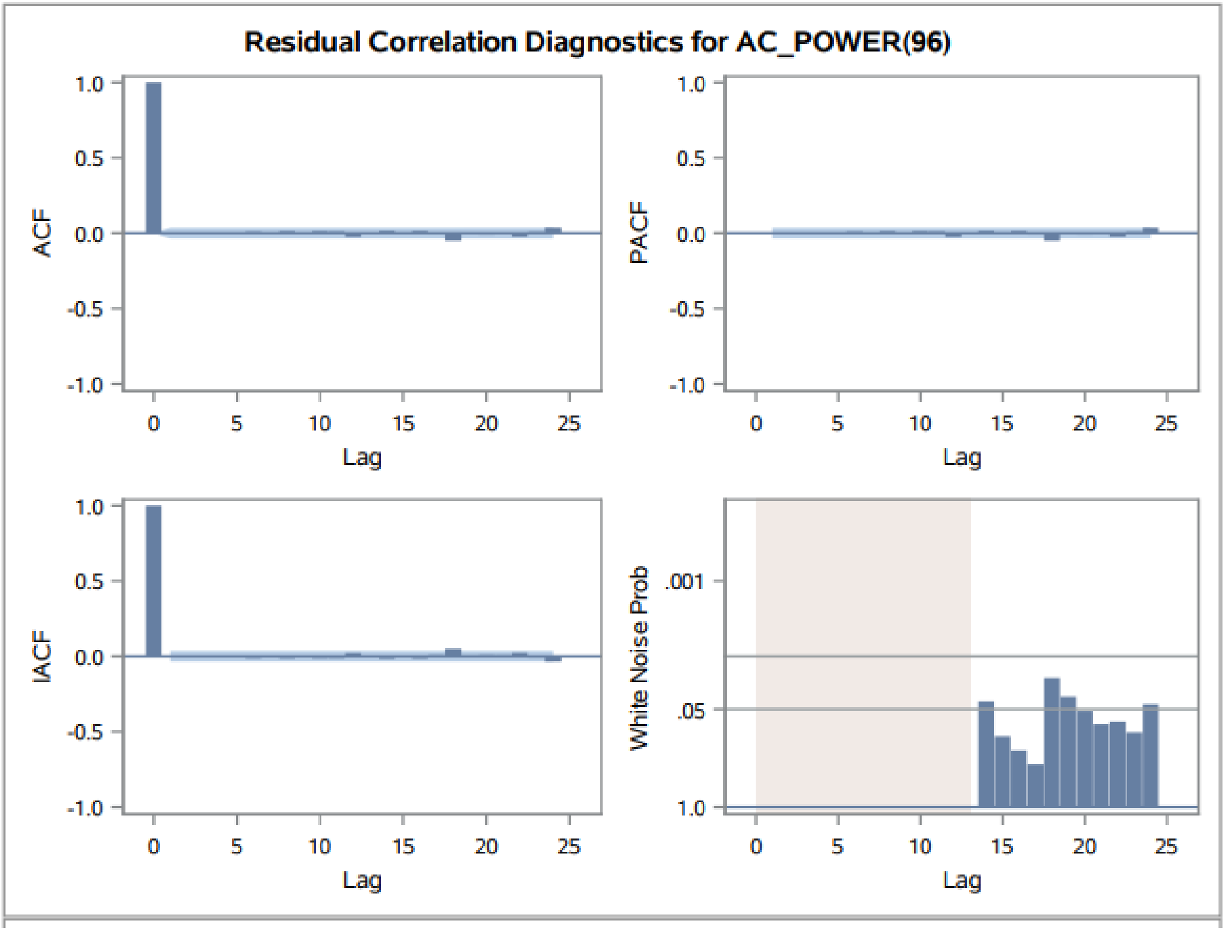

Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
Normal
Kernel
Percent
Residual
-20000
-12000
-4000
4000
12000
20000
0
20
40
60
80
100
120
QQ-Plot
Quantile
-4
-2
0
2
4
-20000
-10000
10000
20000

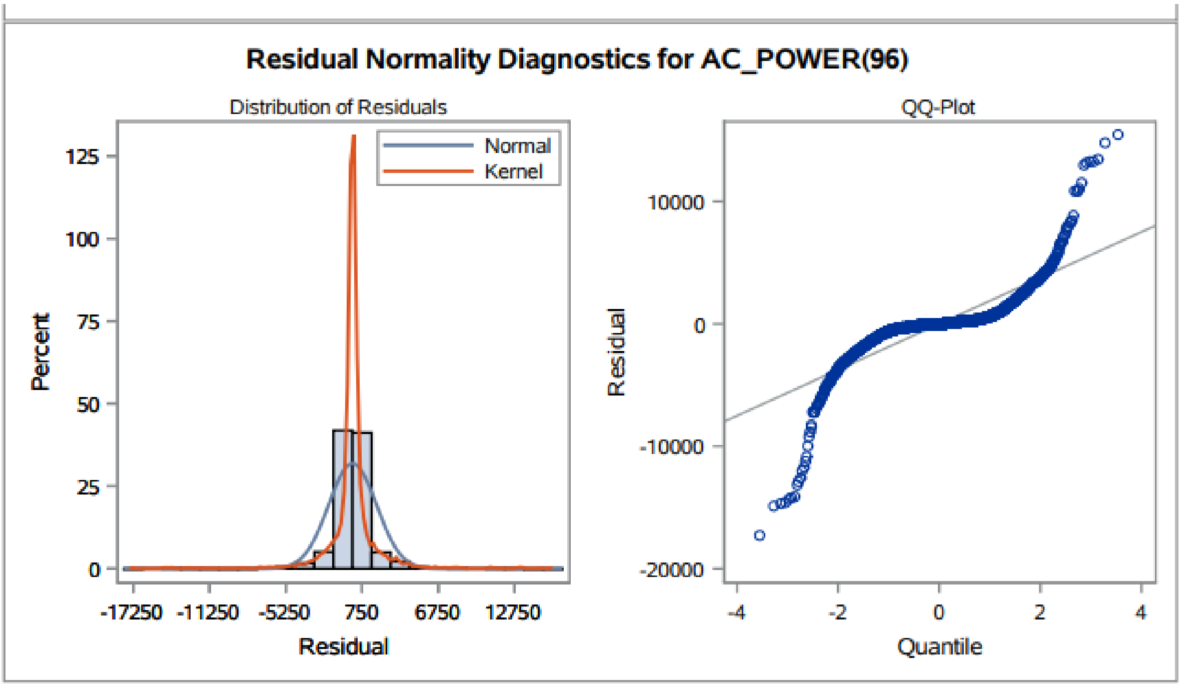
Observation

The white noise probability test fails here again. Which brings us to the conclusion that the parameters chosen for this is not right and we will have to adjust the parameters. This model was AR 10 , so let us down that AR component and try with MA 4.

ARIMAX1 (p=9,d=0,q=4 ; P=0,D=1,Q=0)



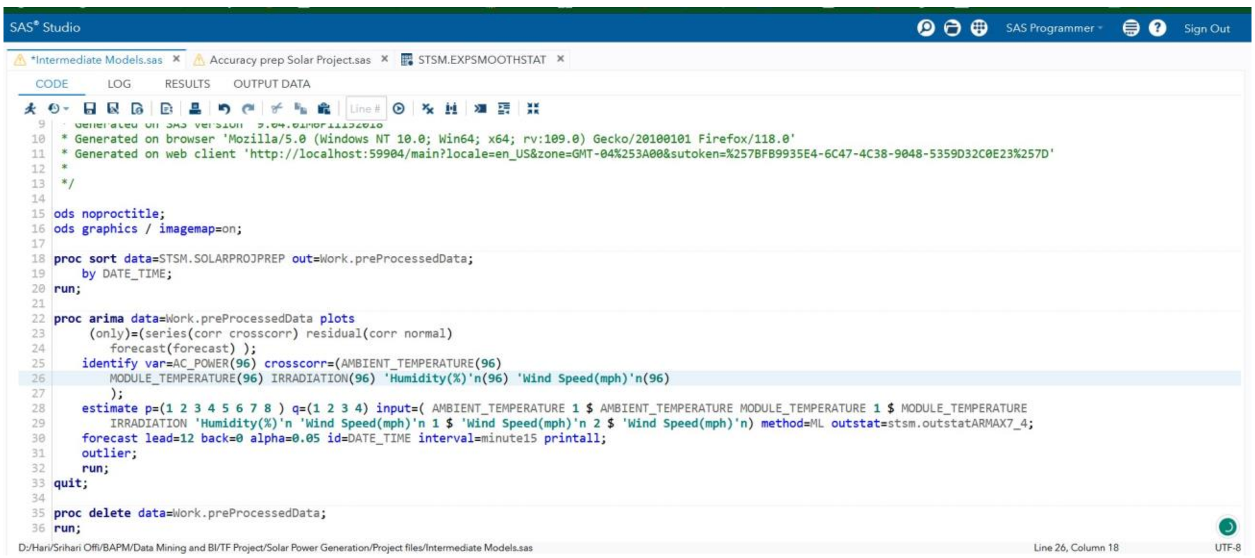

Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25



Observation

The white noise probability test fails here too, meaning that we should perform further parameter tuning. Let us reduce the AR component by one more value.

ARIMAX1 (p=8,d=0,q=4 ; P=0,D=1,Q=0)



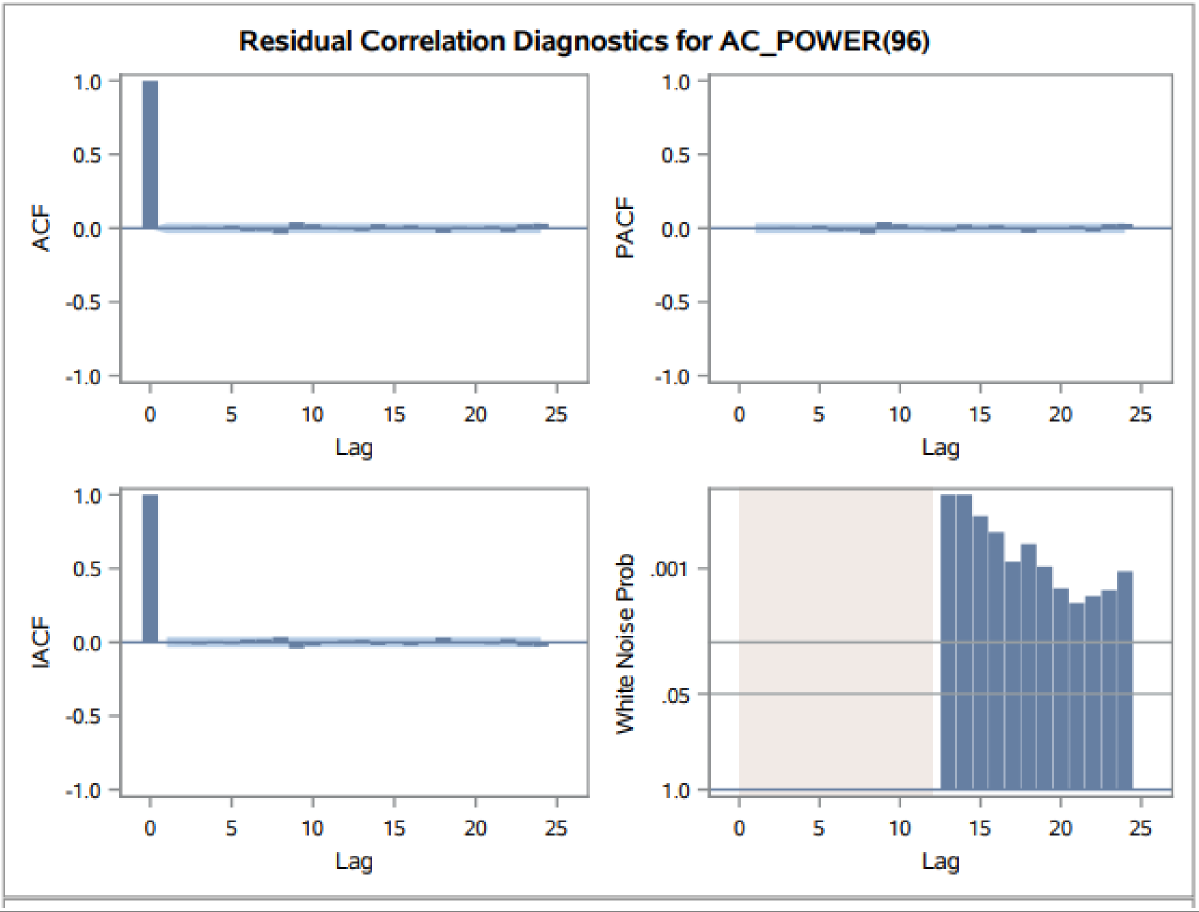
SAS® Studio | SAS Programmer | Sign Out

*Intermediate Models.sas | Accuracy prep Solar Project.sas | STSM.EXPSMOOTHSTAT

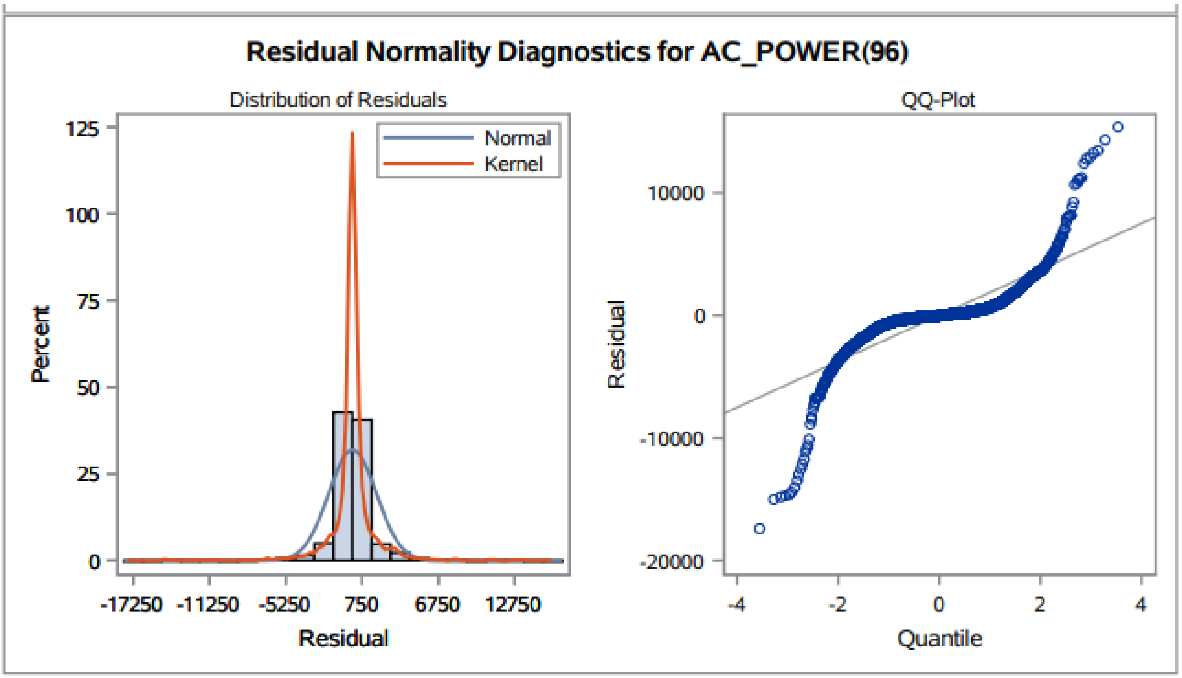
CODE | LOG | RESULTS | OUTPUT DATA

```
 * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0'
 * Generated on web client 'http://localhost:59904/main?locale=en_US&zone=GMT-04%253A00&sutoken=%257BFB9935E4-6C47-4C38-9048-5359D32C0E23%257D'
 *
 */

ods noproctitle;
ods graphics / imagemap=on;

proc sort data=STSM.SOLARPROJPREP out=Work.preProcessedData;
    by DATE_TIME;
run;

proc arima data=Work.preProcessedData plots
     (only)=(series(corr crosscorr) residual(corr normal)
        forecast(forecast) );
    identify var=AC_POWER(96) crosscorr=(AMBIENT_TEMPERATURE(96)
        MODULE_TEMPERATURE(96) IRRADIATION(96) 'Humidity(%)'n(96) 'Wind Speed(mph)'n(96)
        );
    estimate p=(1 2 3 4 5 6 7 8 ) q=(1 2 3 4) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML outstat=stsm.outstatARMAX7_4;
    forecast lead=12 back=0 alpha=0.05 id=DATE_TIME interval=minute15 printall;
    outlier;
    run;
quit;

proc delete data=Work.preProcessedData;
run;
```

D:/Hari/Srihari Offi/BAPM/Data Mining and BI/TF Project/Solar Power Generation/Project files/Intermediate Models.sas | Line 26, Column 18 | UTF-8



Residual Correlation Diagnostics for AC_POWER(96)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25



Residual Normality Diagnostics for AC_POWER(96)
Distribution of Residuals
QQ-Plot
Normal
Kernel
Percent
125
100
75
50
25
0
-17250
-11250
-5250
750
6750
12750
Residual
10000
0
-10000
-20000
-4
-2
0
2
4
Quantile

<u>Observation</u>

The white noise probability test fails here as well. We will stop the parameter tuning here and list down all the parameters where the white noise test has passed.

Model Fit Comparison

We will now perform a comparison of those models which passed the white noise test. We have combined all the models into one code.

Below is the snapshot of the code.

SAS® Studio

*Intermediate Models.sas | *Accuracy prep Solar Project.sas | *Modeling and Forecasting

CODE LOG RESULTS OUTPUT DATA

```

/* STSM03s04d.sas */
ods select none;

proc arima data=WORK._TEMP plots=none;
    identify var=_y_fit(96) crosscorr=(AMBIENT_TEMPERATURE(96)
        MODULE_TEMPERATURE(96) IRRADIATION(96) 'Humidity(%)'n(96) 'Wind Speed(mph)'n(96)
        );
    estimate p=(1 2 3 4 5 6 7) q=(1 2 3) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML;
    forecast lead=&nhold id=DATE_TIME interval=minute15 out=WORK.ARIMAX7_3_forecast nooutall;
    run;
    estimate p=(1 2 3 4 5 6 7) q=(1 2 3 4) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML;
    forecast lead=&nhold id=DATE_TIME interval=minute15 out=WORK.ARIMAX7_4_forecast nooutall;
    run;
    estimate p=(1 2 3 4 5 6 7 8) q=(1 2 3) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML;
    forecast lead=&nhold id=DATE_TIME interval=minute15 out=WORK.ARIMAX8_3_forecast nooutall;
    run;
    estimate p=(1 2 3 4 5 6 7 8 9) q=(1 2 3) input=( AMBIENT_TEMPERATURE 1 $ AMBIENT_TEMPERATURE MODULE_TEMPERATURE 1 $ MODULE_TEMPERATURE
        IRRADIATION 'Humidity(%)'n 'Wind Speed(mph)'n 1 $ 'Wind Speed(mph)'n 2 $ 'Wind Speed(mph)'n) method=ML;
    forecast lead=&nhold id=DATE_TIME interval=minute15 out=WORK.ARIMAX9_3_forecast nooutall;
    run;
quit;

ods select all;

/* STSM03s04e.sas */


%accuracy(indsn=WORK.ARIMAX7_3_forecast, timeid=DATE_TIME, series=AC_POWER,
    numholdback=&nhold);
%accuracy(indsn=WORK.ARIMAX7_4_forecast, timeid=DATE_TIME, series=AC_POWER,
    numholdback=&nhold);
%accuracy(indsn=WORK.ARIMAX8_3_forecast, timeid=DATE_TIME, series=AC_POWER,
    numholdback=&nhold);
%accuracy(indsn=WORK.ARIMAX9_3_forecast, timeid=DATE_TIME, series=AC_POWER,
    numholdback=&nhold);
```

D:/Hari/Srihari Offi/BAPM/Data Mining and BI/TF Project/Solar Power Generation/Project files/Accuracy prep Solar Project.sas

The output values of the error metrics is as below.

Series	Model	Holdback Periods	MAPE	MAE	MSE	RMSE
AC_POWER	WORK.ARIMAX7_3_forecast	384	23.63%	1583.25	9479327.23	3078.85

Series	Model	Holdback Periods	MAPE	MAE	MSE	RMSE
AC_POWER	WORK.ARIMAX7_4_forecast	384	23.66%	1583.84	9494810.05	3081.36

Series	Model	Holdback Periods	MAPE	MAE	MSE	RMSE
AC_POWER	WORK.ARIMAX8_3_forecast	384	23.55%	1584.96	9559091.50	3091.78

Series	Model	Holdback Periods	MAPE	MAE	MSE	RMSE
AC_POWER	WORK.ARIMAX9_3_forecast	384	23.35%	1574.06	9467328.37	3076.90

We notice a very slight difference in the error metrics within these models, but the best one being ARIMAX (p=9,d=0,q=3 ; P=0,D=1,Q=0).

We will also compare the AIC and SBC values.

Model	**p**	**d**	**q**	**P**	**D**	**Q**	**AIC**	**SBC**	**MAPE**
ARIMAX	7	0	3	0	1	0	56784.1043	56881.0729	23.63
ARIMAX	7	0	4	0	1	0	56782.7517	56885.7808	23.66
ARIMAX	8	0	3	0	1	0	56767.4537	56870.4829	23.55
ARIMAX	9	0	3	0	1	0	56784.4338	56893.5235	23.35
Additive seasonal Exponential smoothing	-	-	-	-	-	-	43068.8352	43080.7663	44.8

We got best AIC and SBC at model ARIMAX (p=8,d=0,q=3 ; P=0,D=1,Q=0).

Since the MAPE values are only slightly different, we confirm that the best fit model is at ARIMAX (p=8,d=0,q=3 ; P=0,D=1,Q=0).

Summary and Inferences

The data exploration revealed seasonality in the AC_POWER time series as since the power generation follows the daily solar cycle. Despite this seasonality, the Augmented Dickey-Fuller Test rejected the null hypothesis and thus showed that it can be modeled with ARIMAX, but we had to deal with seasonality by using seasonal differentiation. We prewhitened the independent variables, etc. to see the causal significance as well as which lags provide significant effects and we found Dew point and pressure to be insignificant. This conveys that the dew point and pressure factors do not affect power generation which can be agreed upon. In the final best model we find that IRRADIATION, AMBIENT_TEMPERATURE and MODULE_TEMPERATURE and significant predictors of AC_POWER. This effect of IR Radiation on AC_POWER can be obvious as Solar panels convert radiation into power and the ambient and module temperature are dependent on the solar emission which in turn affects the power generation. The independent factors like Wind Speed and Humidity were not significant in predicting AC_POWER ($p > 0.05$) but when these factors are removed the MAPE of the models increases showing that they have some importance in prediction. This may be attributed to the wind blowing away dust from the panels and the humid air is less dense, hence increasing its transmissivity of radiation.

References

Data Sources:
https://www.kaggle.com/datasets/anikannal/solar-power-generation-data?select=Plant_1_Generation_Data.csv
https://www.wunderground.com/history/daily/in/mumbai/VABB/date/2020-5-15