IE 360 Statistical Forecasting and Time Series

Solar Power Forecasting Project Report

Instructor: Mustafa Gökçe Baydoğan

Project Group 28:

Melis Tuncer	2019402156
Kaan Ertan	2019402087
Sude Filiz	2021402273

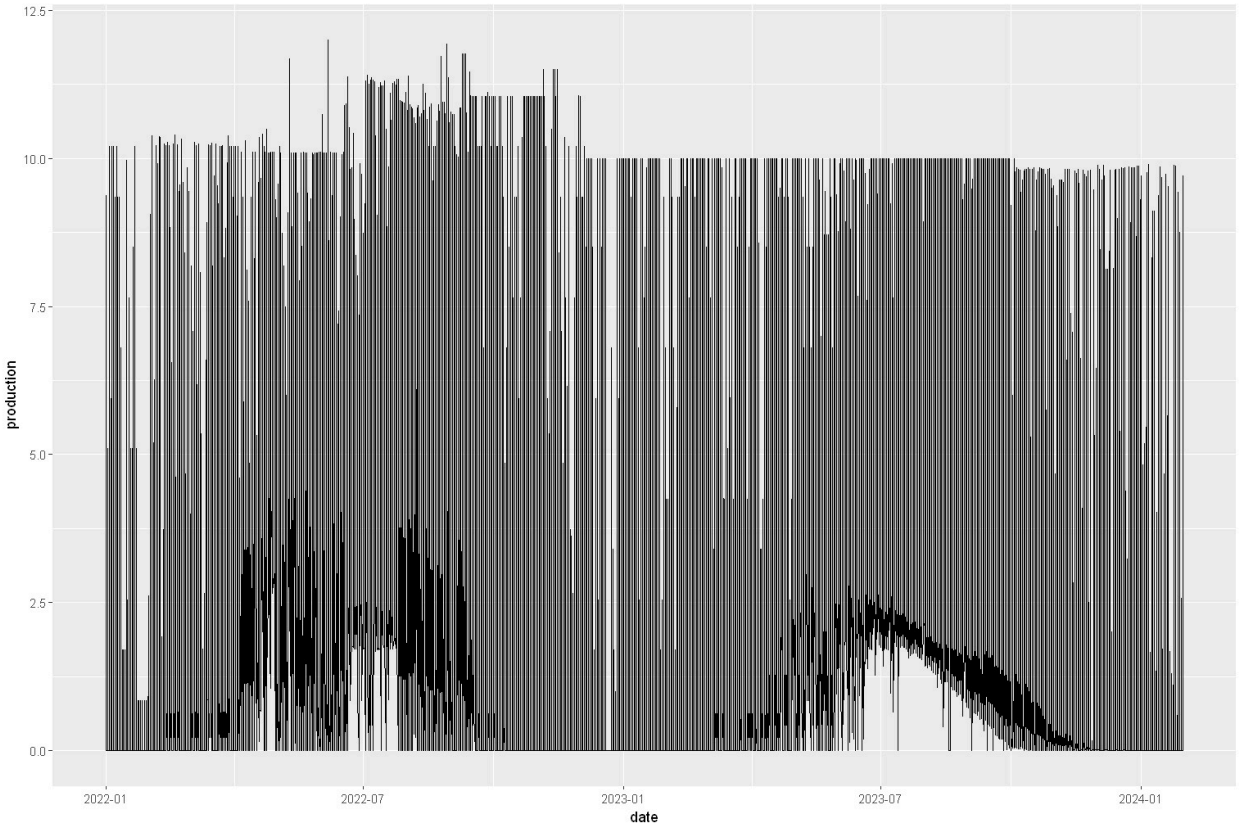
Introduction:

The problem at hand depicts a real-life forecasting problem that involves many collinear variables. The aim is to provide accurate predictions for hourly solar power production of the Edikli Solar Power Plant in Niğde/Türkiye. Providing accurate predictions is essential for real-life market operations since electricity pricing and capacity planning will rely heavily on production forecasts. In the context of solar energy, it is impossible to deliberately increase production without directly increasing the physical solar panel capacity (eg: by improving panel power generation capacity, increasing the number of panels, increasing panel maintenance, etc.). Therefore, the power generated in a day depends completely on the incoming solar energy absorbed by the panels and transformed into electrical energy. Since the solar energy plant has such a naturally determined production amount, it is vital to predict the production beforehand for a more advantageous pricing policy.

For the prediction of solar energy production beforehand, the best approach is to construct a model that takes meteorological variables as inputs. Meteorological variables have a direct effect on the incoming solar energy on the Earth's surface, thereby having an indirect effect on the solar power generated by the plant. This way, the model can predict solar power generation using these observable parameters at hand. Before moving on with our specific approach, it is fruitful to examine our variables for a better understanding of the problem.

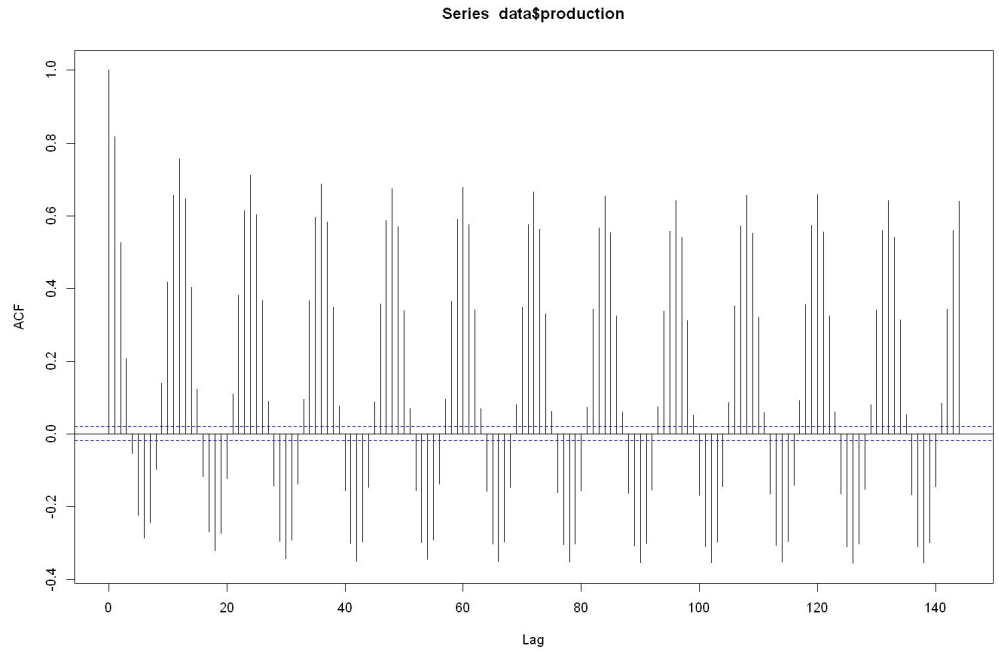
Descriptive Analysis of the Given Data & Variables:

For each variable whose data was given, we first analyzed the variable and its meaning in our framework. Then we moved forward with their data visualization and the analysis of their respective effects on the solar power production. For this, we utilized the ggplot package in R. The following are the results of this analysis:

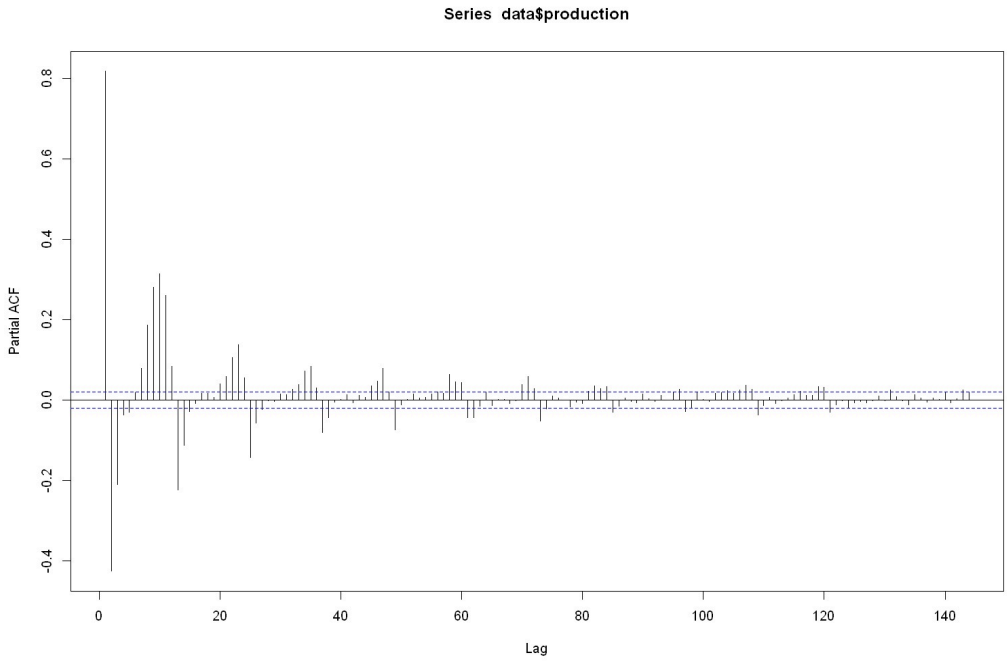


The Overall Plot of Our Production Data

(notice that both the seasonal nature of solar power production and the maximum solar power production are clearly seen)

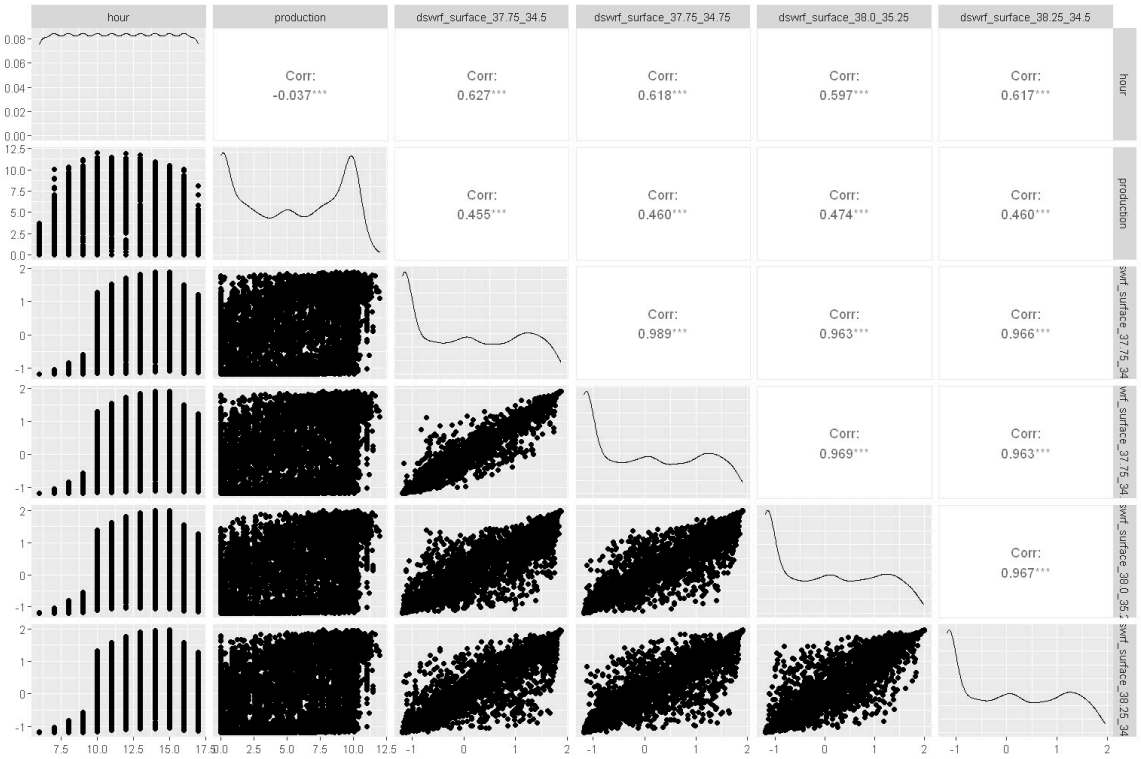


The Autocorrelation Function Plot of Our Production Data



The Partial Autocorrelation Function Plot of Our Production Data

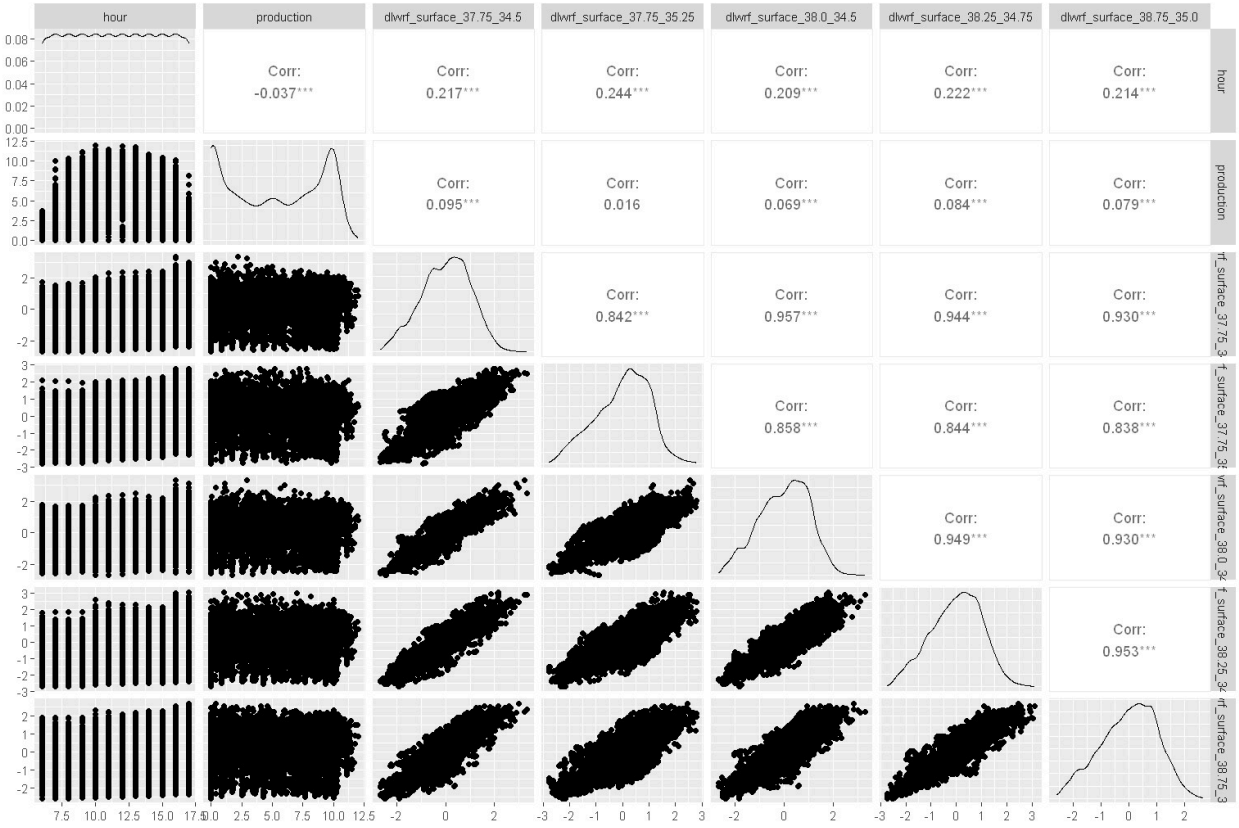
DSWRF_surface: This variable is the “Downward Shortwave Radiation Flux on Surface”, which is an estimate of the total shortwave radiation that reaches the Earth’s surface. Its computation includes both the albedo (the fraction of light that is reflected by Earth) and atmospheric composition of Earth. DSWRF significantly impacts the production output of solar power plants, since when the flux is high solar panels receive more sunlight and generate more electricity. During the days with low flux (due to clouds or atmospheric conditions), plants generate much less electricity. However, it is vital to note that this variable is strongly correlated to the atmospheric variables and therefore there may be multicollinearity between those variables and this.



It is seen in the correlation table that DSWRF_surface variables of different location groups all have positive correlation (seen in the 2. row, around 0.45) with production,

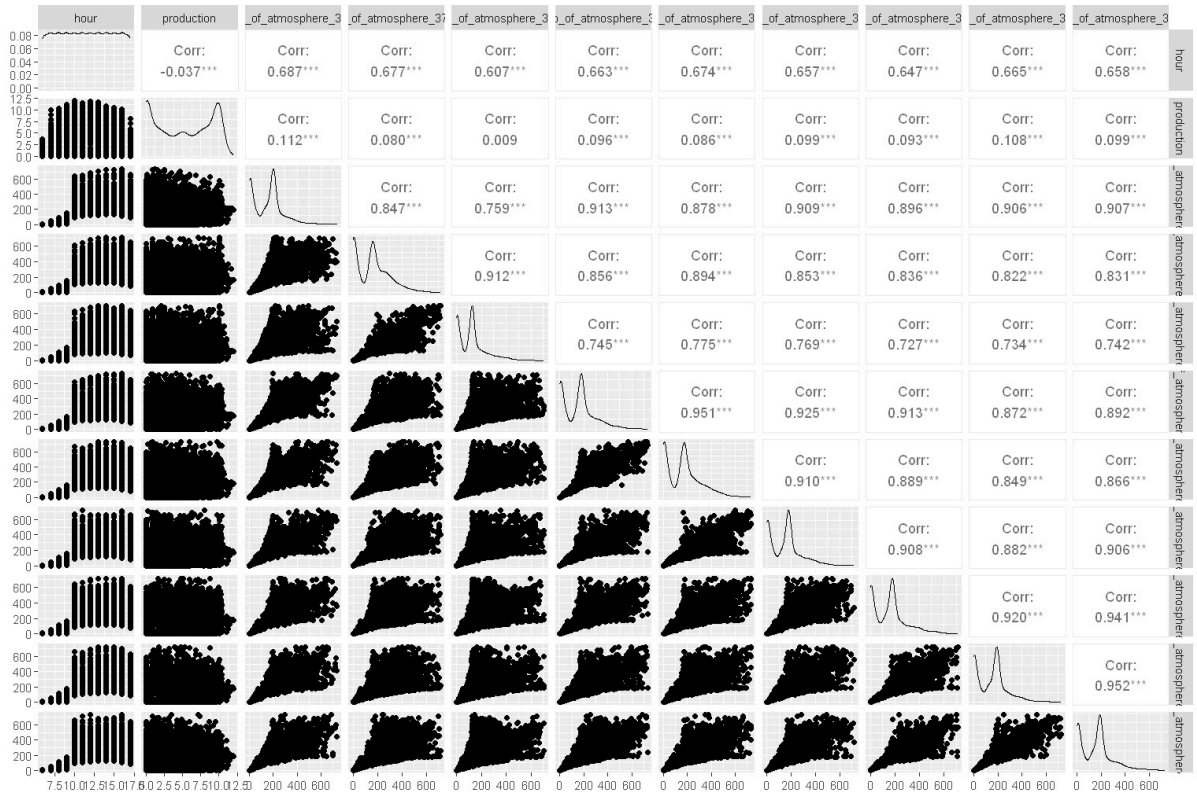
DLWRF_surface: This variable is the "Downward Longwave Radiation Flux on Surface". Its difference from DSWRF is that DLWRF has a longer wavelength and it is the solar-based radiation that is emitted back from Earth’s surface. In this sense, it doesn’t contribute too much to the solar panels’ energy production directly. However,

since solar panels lose heat by this means, a higher DLWRF means that solar panels will be colder early in the morning and it will take longer for them to heat up and this slightly effects their efficiency.



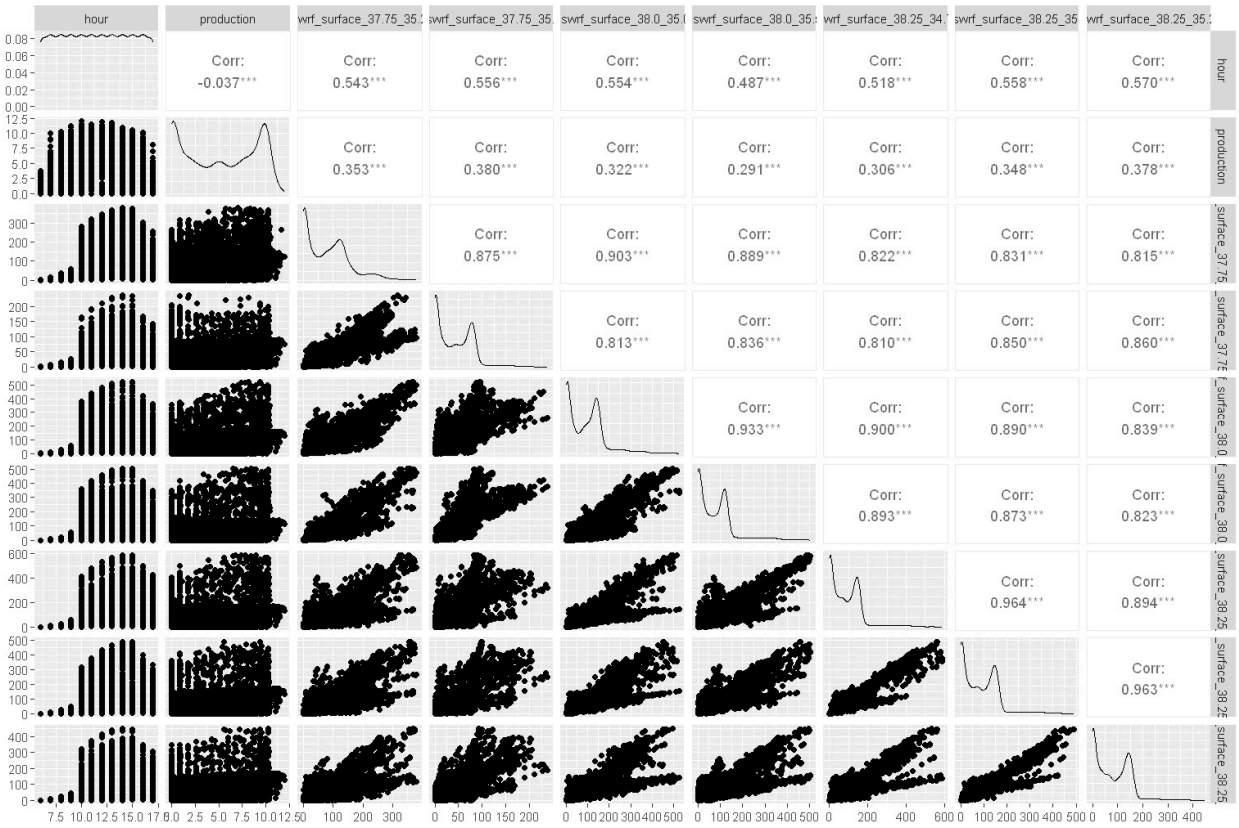
In the correlation table we see that the DLWRF_surface variables of different location groups have slightly positive correlation (around 0.08) with solar power production

USWRF_top_of_atmosphere: This variable is the “Upward Shortwave Radiation Flux at the Top of the Atmosphere”, and represents the amount of shortwave solar radiation reflected back into space from the top of the atmosphere. This flux is influenced by factors such as cloud cover and atmospheric conditions. Changes in this can affect the amount of solar radiation reaching the Earth's surface, and therefore the solar energy production.



In the correlation table we see that the USWRF_top_of_atmosphere variables of different location groups have slightly positive correlation (around 0.1) with production

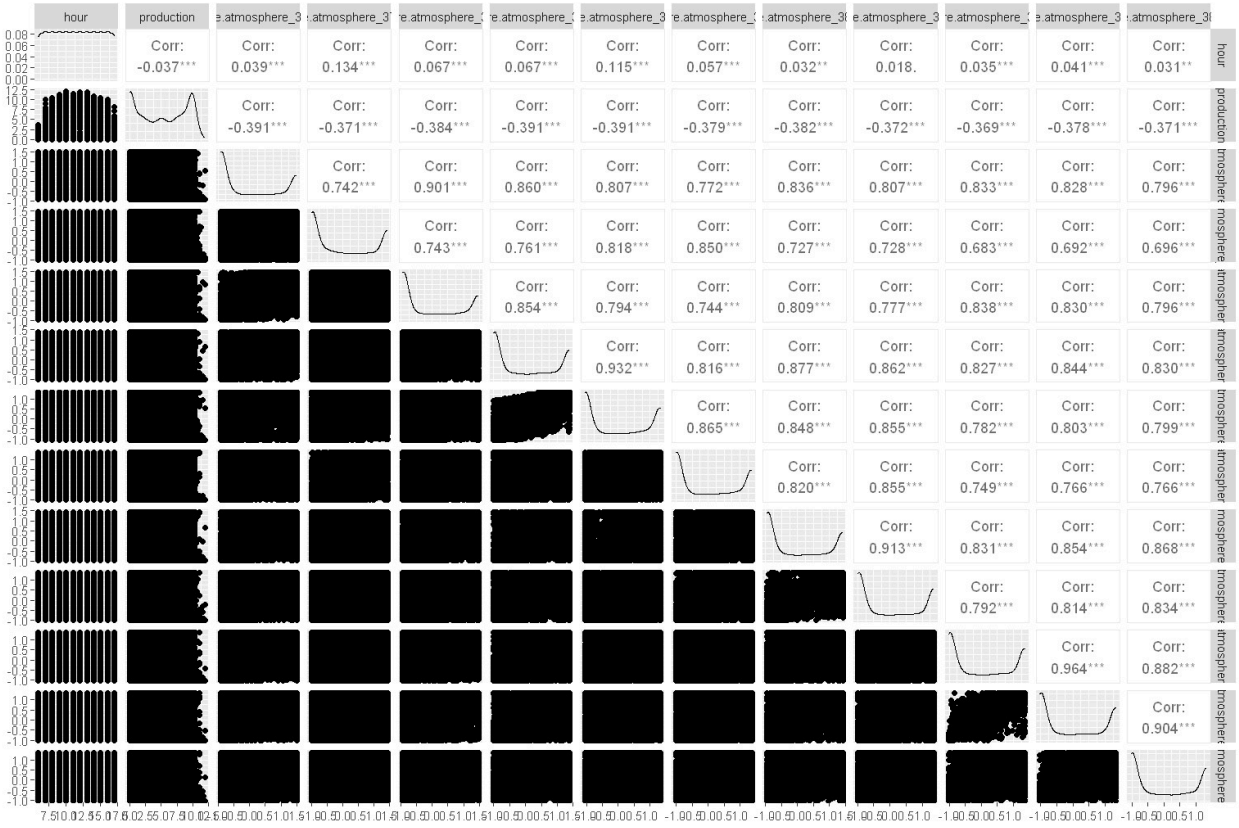
USWRF_surface: This is the “Upward Shortwave Radiation Flux at the Earth's surface”, and this variable shows the amount of shortwave solar radiation reflected by the Earth's surface. Surface conditions such as the surface being land or water, vegetation, and surface color all affect this variable. But since we are focusing on solar panel surfaces, we can only use this variable to understand the flux intensity and therefore the incoming solar energy. High values of USWRF_surface indicate more solar radiation hitting and being reflected by the surface.



It is clear that the USWRF_surface variables of different location groups have positive correlation (around 0.35) with production

TCDC_low.cloud.layer, TCDC_middle.cloud.layer, TCDC_high.cloud.layer, TCDC_entire.atmosphere: These are the total cloud cover percentages for different types of clouds. Although evaluating the entire atmosphere as a whole is easier and yields less collinear results, it is important to note the differences of cloud types. Clouds are classified according to their altitude in the atmosphere: low, middle, high. Low clouds can significantly reduce solar radiation reaching the Earth's surface by blocking direct sunlight. Moreover, they diffuse sunlight but do so unevenly over the surface. So they may lead to a less uniform flux on solar panels, which may decrease efficiency. Middle clouds cannot completely block sunlight like low clouds, but they still can reduce solar flux by scattering, reflecting, and absorbing sunlight. Additionally, the presence of middle clouds often indicate changing weather conditions, which can lead to further changes in solar power generation either for the better or for worse. Lastly, high clouds

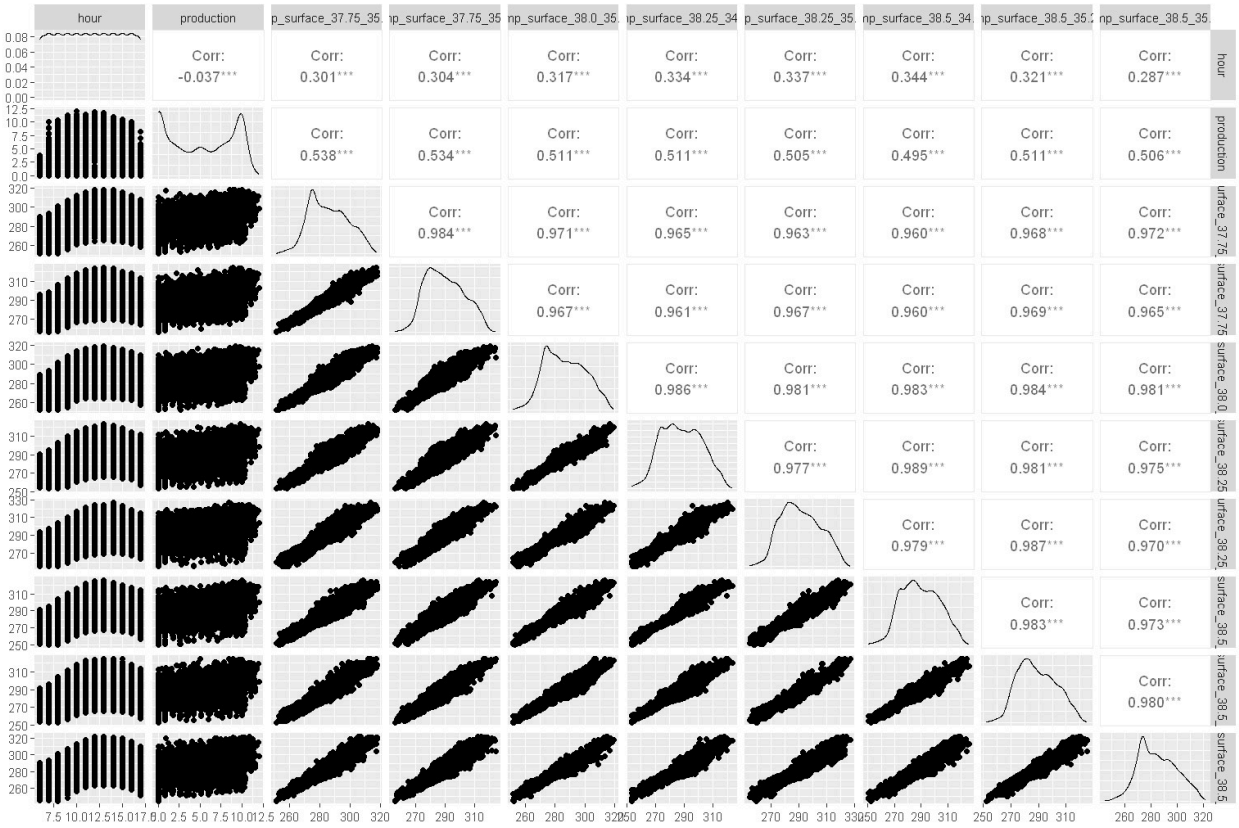
are thin clouds that can only block a limited amount of sunlight. However, they can still reduce solar flux by scattering and diffusing sunlight. Similar to middle clouds, high clouds may also indicate approaching weather systems. Their difference is that high clouds often indicate weather changes in a longer term, and can be useful for long term weather forecasting.



We see that the TCDC_entire.atmosphere variables of different location groups have negative correlation (around -0.37) with production

CSNOW_surface: This is a categorical variable for the existence of snow. When the solar panels are covered with snow, they cannot function until they are cleared off the snow. Since the solar panels might get covered by snow in winter, it is important to include this scenario in a model.

TMP_surface: This is the surface level temperature. Temperatures do not directly affect solar power production, but are useful for representing seasonality. Since both the surface temperatures and solar power generation are positively correlated to the sun light exposure, this may be beneficial to model the seasonal differences in the sun light exposure (due to both weather conditions and the Earth's tilted axis resulting in different day lengths). Moreover it is known that high temperatures affect the solar panels and decrease their efficiency.



TMP_surface variables of different location groups have positive correlation (around 0.53) with solar power production

Summary of Our Approach:

Our approach involved several steps and methodologies to ensure the reliability and accuracy of our predictions.

We began by preparing the data in Python, transforming it into a wide format and removing zero-production hours. Initial regression analyses indicated room for improvement, leading us to switch to R for further analysis, where we examined ACF and PACF plots to identify seasonal patterns and long-term dependencies. To enhance model interpretability and accuracy, we applied Lasso regressions to select significant variables.First, we performed Lasso regression on the original set of variables. Next, we expanded the model by incorporating both the original and weighted variables. Lastly, we employed a more stringent Lasso regression with a higher penalty parameter to further refine the variable selection. Consequently, we transitioned to ARIMA models, grouping the data by hour and updating the models daily with the most recent information. Differencing techniques were used to remove trends and achieve stationarity. After developing ARIMA models for each hour, we combined the results for overall performance evaluation and attempted to create a unified ARIMA model using the entire dataset.

Literature:

Our literature research was focused on 3 fields: understanding the general relationship between atmospheric events and solar power generation, understanding the provided variables and their meteorological foundations, and understanding methods to convert this relationship into a forecasting model.

Solar power depends directly on factors such as the intensity and angle of the incoming sunlight. The atmospheric conditions have a critical direct effect on the incoming sunlight, so the provided variables have an indirect relationship with the solar power

generation since they work as an indicator of the incoming sunlight rather than a di rect measure of it. For instance, the cloud cover parameters are definitely important for our model, yet they do not have a direct impact on the power generation. Instead, they affect the solar power production via their effect of filtering the incoming sunlight to the Earth's surface. This concept applies to the other parameters as well, in the sense that our parameters merely reflect the changes in the incoming solar energy level, thus helping us to understand and predict the solar power generation. This makes it important to understand these parameters.

Of course, at the end of the day, we construct our model by examining and learning from the past data. But understanding the climatological background is a must in order to ensure that our model makes sense not only in reference to the data at hand, but also in reference to the meteorological phenomena in the nature. For this, we examined some key literature for meteorology and climatology by Adams & Lambert (2006) as well as by Desonie (2012).

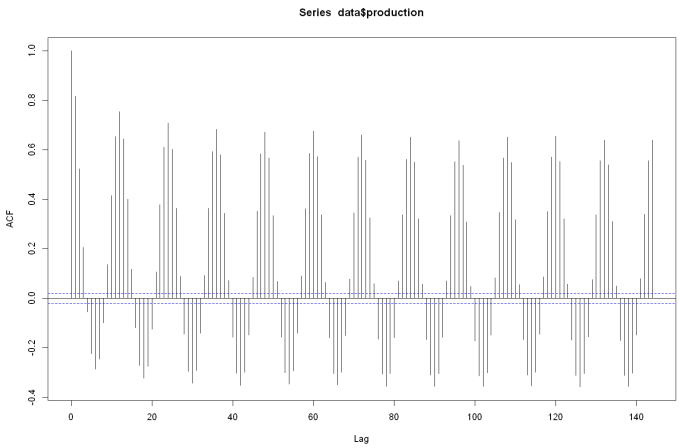
In order to construct a good model that learns from the data and provides accurate forecasts, we benefitted mostly from the methods and frameworks that we have studied in our course. In addition to these, we benefitted from time series modelling resources such as Montgomery et al. (2008) and Hyndman & Athanasopoulos (2018). These helped us to research for the best possible data cleaning and forecast making strategies, and helped us to troubleshoot our mistakes on the way to the best solution.

Approach

Data Preparation Using Python:

At first, we used Python to transform the data into a wide format and eliminate duplicate entries. Subsequently, we identified the hours when the production value was zero and eliminated them from our model because they were unnecessary to our analysis. Hours with zero production values were 0, 1, 2, 3, 4, 19, 20, 21, 22, and 23. We performed regression analysis on the filtered_data that we obtained. The resulting R-squared value was 0.629. Given that this value was exceedingly high, we continued to weight the variables based on their 1/p values. We then conducted two regression analyses: one with only the 10 weighted variables and one with 260 variables which includes both the original and weighted variables. In both cases, the R-squared value was greater than desirable. In our fourth approach, we conducted a series of individual regressions for each variable, resulting ten different regression models. The primary goal of this approach was to find out which location would be significant if we regression with that seperate variable. However, the outcomes from this approach were suboptimal, yielding no significant discoveries.

Following the implementation of these approaches in Python, we switched to R for further analysis. After importing the necessary libraries in R, we examined the ACF plot of our data.



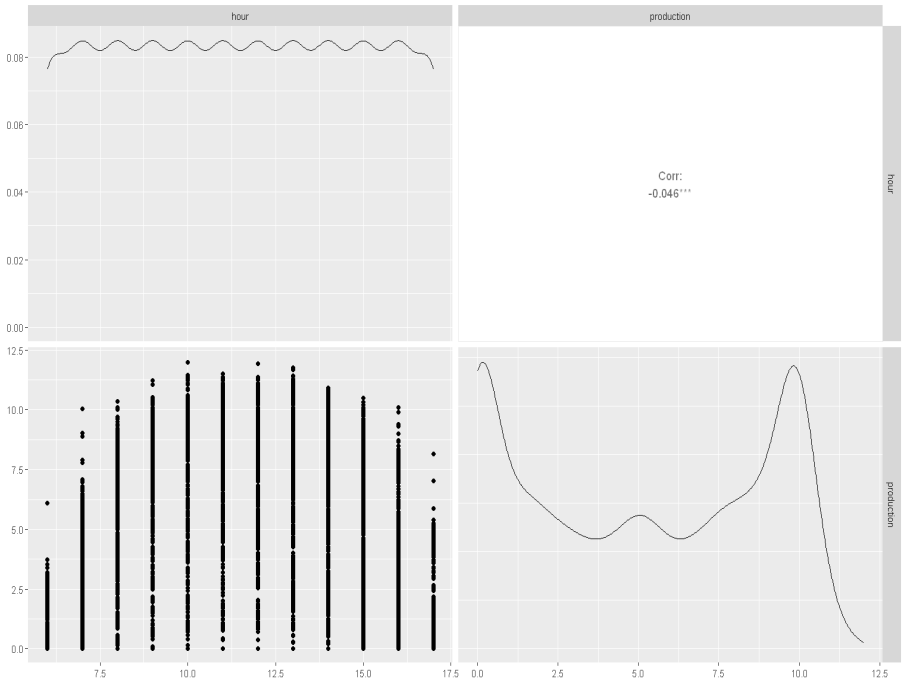
Autocorrelation plot of time series data 'production'

The ACF plot of the time series shows strong seasonal patterns with significant autocorrelations at regular intervals, approximately every 12 lags. Furthermore the gradual decay in autocorrelation over several lags indicates long-term dependencies in the series, implying that implying that production levels fluctuate seasonally as well as follow stable trends.

Then, we used the ggpairs function to examine the relationship between hour and production.

```
hour_and_prod <- c("hour", "production")
filtered_data_h_vs_prod <- train_data[, ..hour_and_prod, drop = FALSE]

ggpairs(filtered_data_h_vs_prod)
```

R code snippet used to generate the plot depicting the relationship between hour and production.



Pairwise plot examining the relationship between hour and production

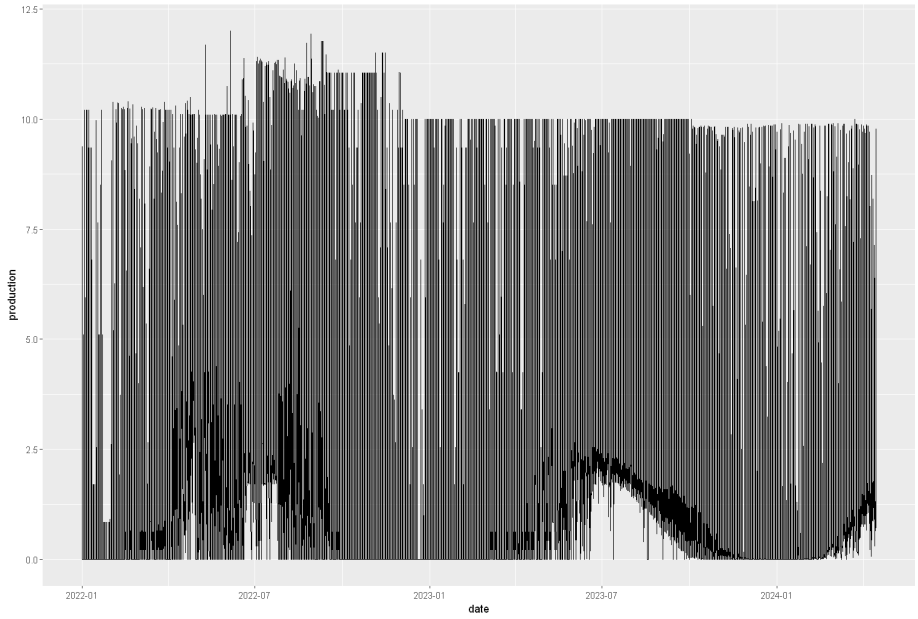
In the lower left plot, we looked at the hours when production was highest. Since we do not observe a linear relationship between time and production, we cannot take time as a

predictor. We transformed hour to -(hour-12)^2. By doing so, we ensure that the new hour column has a linear relationship with production and it will have higher values when it is close to 12, as expected.

After that, we utilized the ggplot function to visualize the relationship between date and production.

```
require(ggplot2)

ggplot(train_data, aes(x = date, y = production)) + geom_line()
```

Code snippet used to generate the plot depicting the relationship between date and production at all hours



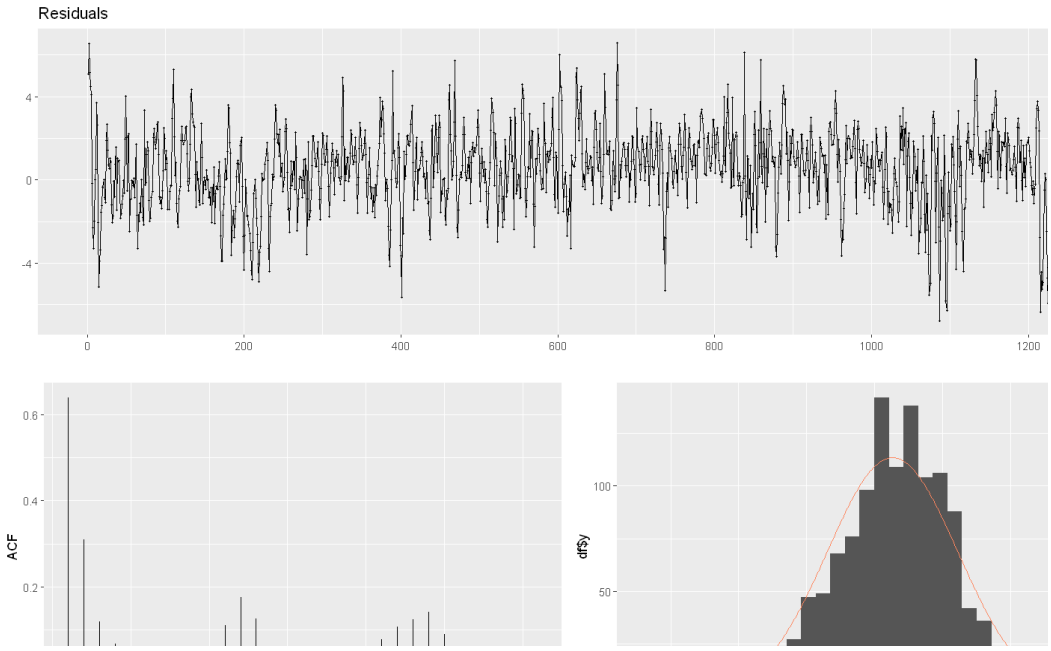
ggplot examining the relationship between hour and production

We notice that there is a maximum production level of approximately 10 in general based on the ggplot. Consequently, we included the maximum production level for the previous three days as a second variable for maximum production.

Regression with all variables:

Subsequently, we import and prepare the test data covering the period from February 1st to May 15th. For testing, we prepared the test data similarly, removed any missing values, and used the model to predict solar power production. Negative predictions were replaced with zero since production cannot be negative. We developed a time series linear regression model utilizing all available variables, establishing this as our baseline. As a result of our test, we found the adjusted R-squared to be 0.7393, which is quite high.

We calculated the residuals (differences between actual and predicted values) and plotted the actual and predicted values over time using ggplot, showing how well the model's predictions matched the actual production values. Also, we plot the autocorrelation function.



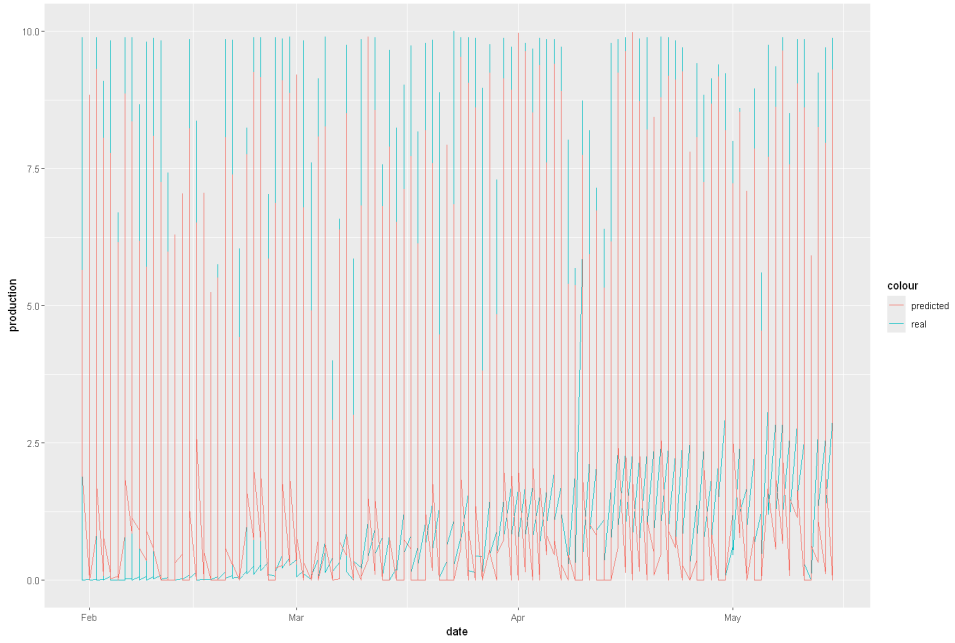
Residuals plot (top), ACF of residuals (bottom left), and histogram with density plot of residuals (bottom right) for the regression model with all variables

Residual is the difference between an observed value and the value predicted by a model.In our residuals plot there are periods where residuals are consistently above or

below zero, it might indicate that the model has not fully captured some underlying patterns or seasonality in the data.

Also the ACF plot should ideally show no significant autocorrelations if the residuals are purely random. Significant spikes at certain lags indicate that there is still some autocorrelation left in the residuals, suggesting that the model might be missing some temporal dependencies.

Although the residuals' histogram suggests they are roughly normally distributed, further statistical tests could provide a more definitive conclusion. Additionally, we applied the Ljung-Box test, a statistical test used to determine whether there is significant autocorrelation in a time series' residuals, indicating whether the residuals are independently distributed.



Actual vs. Predicted Production Values

In the Ljung-Box Test, the null hypothesis (H0) is that the residuals are independently distributed (no autocorrelation), while the alternative hypothesis (H1) is that the

residuals are not independently distributed (there is autocorrelation). Since our p-value is much smaller than 0.05, we reject the null hypothesis. This indicates that there is significant autocorrelation in the residuals of our model, suggesting that the model has not fully captured all the temporal dependencies in the data.

Using the 'accu' function, we evaluated the performance metrics of the baseline linear model.

A data.frame: 1 × 11

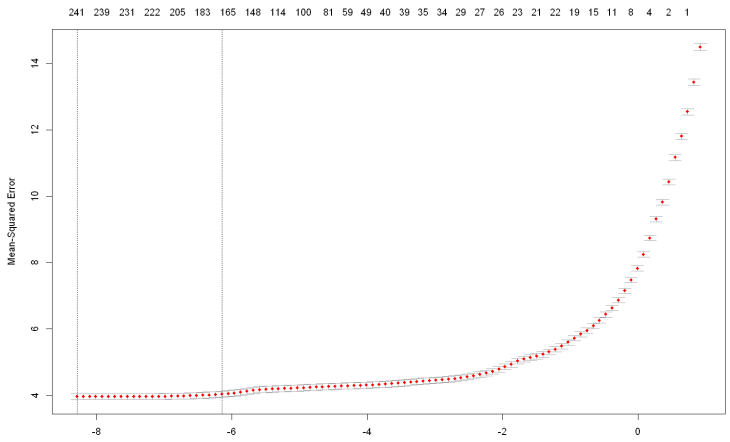
model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.38983	0.6699049	0.1048255	4469.314	1.993499	1.564115	0.3091036	0.3091036

Summary of performance metrics for the baseline linear model

This baseline model provided a reasonable starting point, but results suggested room for improvement. So we continued with lasso regression.

Lasso Regression:

Lasso regression is an analysis technique that selects variables and applies regularization to improve the statistical model's interpretability and prediction accuracy. We tried to eliminate least effective variables using lasso regression. We used 'cv.glmnet' function to perform cross-validation and identify the optimal lambda value.



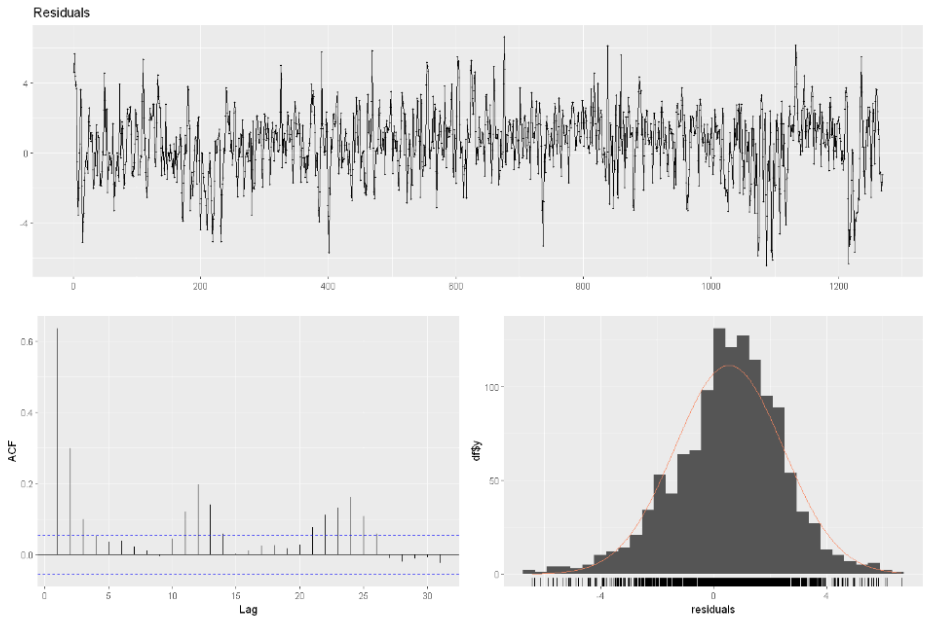
Cross-validated Mean Squared Error (MSE) vs. Log(lambda)

The minimum mean squared error (MSE) is typically found when the model complexity is at its highest, meaning more predictors are included. To avoid overfitting, we use a lambda value within one standard deviation of the minimum MSE.
Afterwards, we continued with the 178 variables we selected with lasso regression.

<u>First Approach : Linear Regression with Lasso Variables + Transformed Hour:</u>

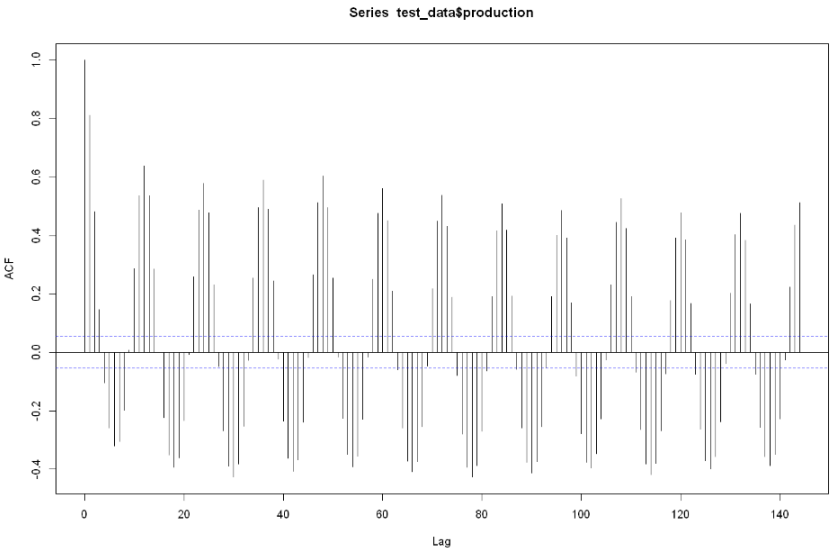
In this approach, we began by preparing our data, which involved transforming the 'hour' variable.
We applied Lasso regression to select the most significant variables from our dataset, and then included the transformed hour variable in our feature set. Using these features, we fitted a linear regression model which we called Lasso(1sd)-LM.

To evaluate the model, we analyzed the residuals through various plots, such as residuals over time, the autocorrelation function (ACF) of residuals, and a histogram of residuals with a density plot.



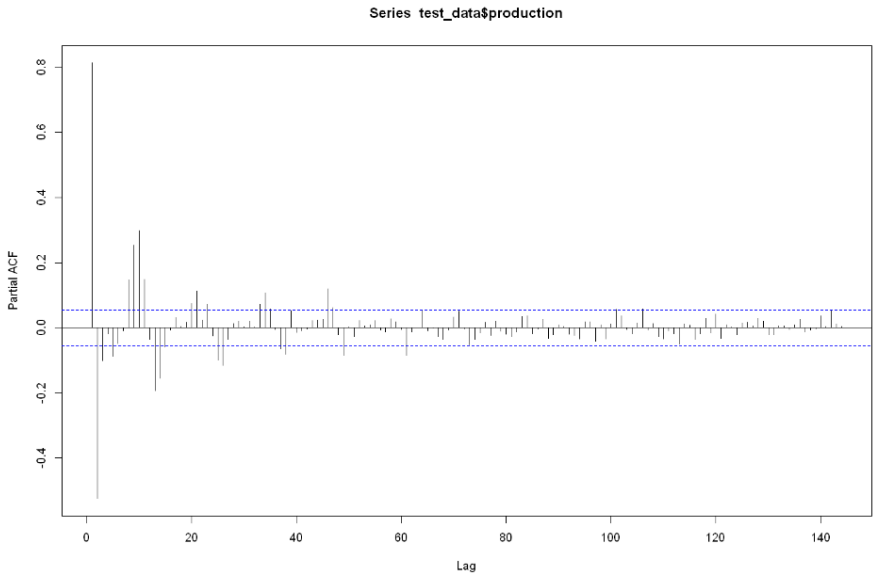
Residuals plot , ACF of residuals , and histogram with density plot of residuals for the first approach.

```
acf(test_data$production, lag.max=144)
```



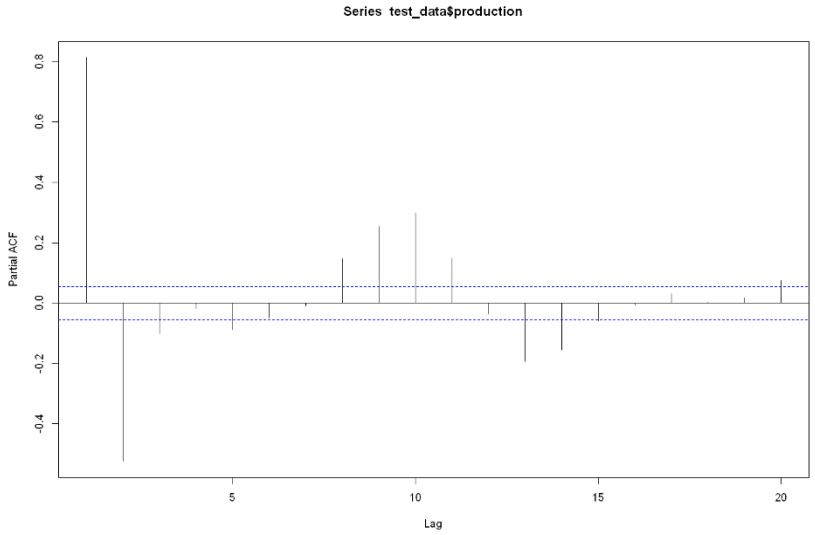
ACF plot examining the relationship between hour and production

```
pacf(test_data$production, lag.max=144)
```



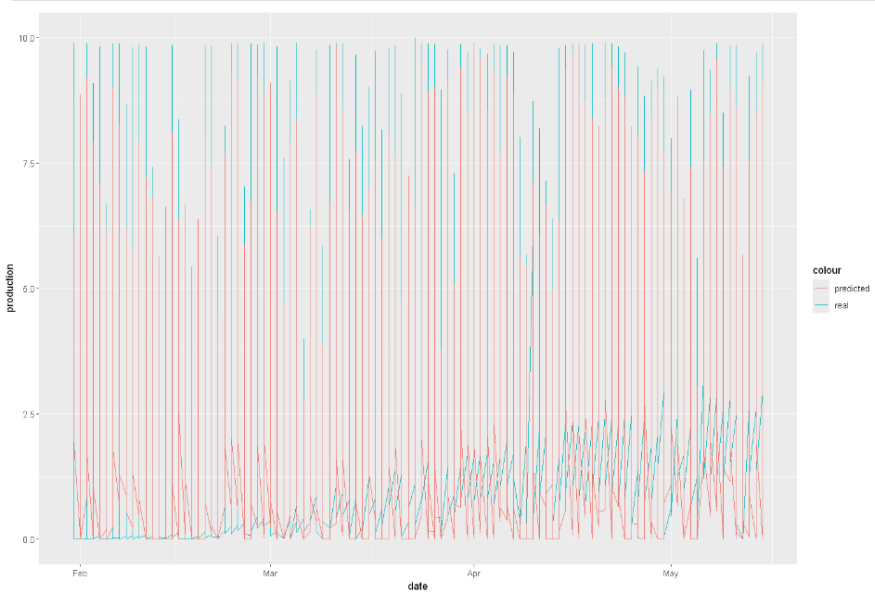
PACF plot with a maximum lag of 144 examining the relationship between hour and production

```
pacf(test_data$production, lag.max=20)
```



PACF plot with a maximum lag of 20, examining relationship between hour - production

We found that while the residuals appeared roughly normally distributed, there were signs of autocorrelation, indicating some temporal dependencies were not fully captured. Additionally, we performed the Ljung-Box test, which confirmed significant autocorrelation in the residuals.




```
        Ljung-Box test

data:  Residuals
Q* = 650.1, df = 10, p-value < 2.2e-16

Model df: 0.   Total lags used: 10
```

Comparison of predicted and real production values over time for first approach, indicating the presence of significant autocorrelation in the residuals as confirmed by the Ljung-Box test

The model's performance was quantitatively assessed using ‘accu’ function, which were compared to a baseline linear model. We will now include lagged values at 10 intervals, as indicated by the peak in the PACF at lag 10. Although this model is impractical for our forecasting task, we aim to evaluate its performance. The reason for its impracticality is that we lack the necessary data for the lagged 10 value when predicting future values (for example, the lagged 10 value for 03.01.2024 12:00 would be 02.01.2024 14:00, which we cannot know in advance). Nonetheless, we construct this model to evaluate its performance.

```
accu1 <- accu("1. Lasso(1sd)-LM", first_model_evaluation$production, first_model_evaluation$predicted)
comparisons <- rbind(comparisons, accu1)
comparisons
```

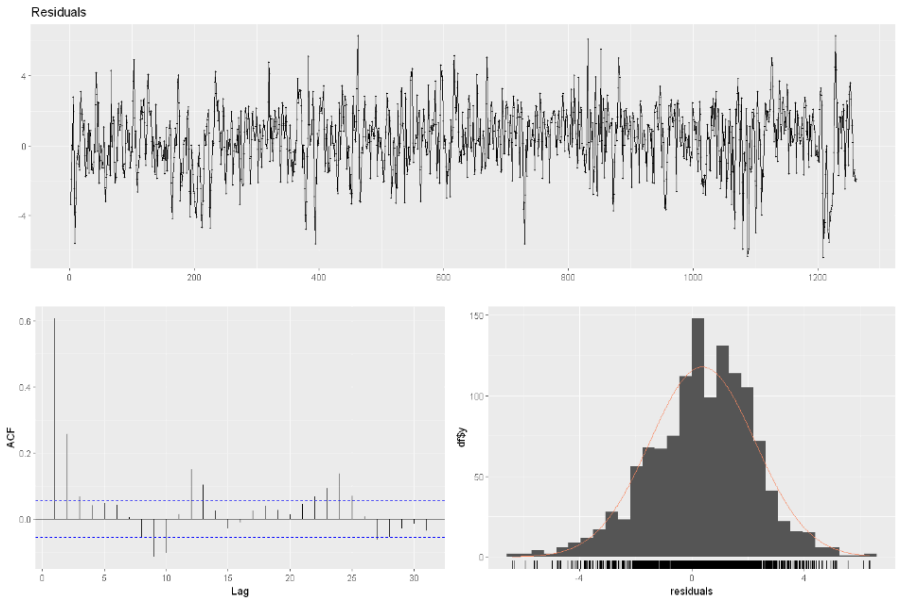
A data.frame: 2 × 11

model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.38983	0.6699049	0.1048255	4469.314	1.993499	1.564115	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.38983	0.6699049	0.1088095	4058.146	1.994581	1.573241	0.3109070	0.3109070

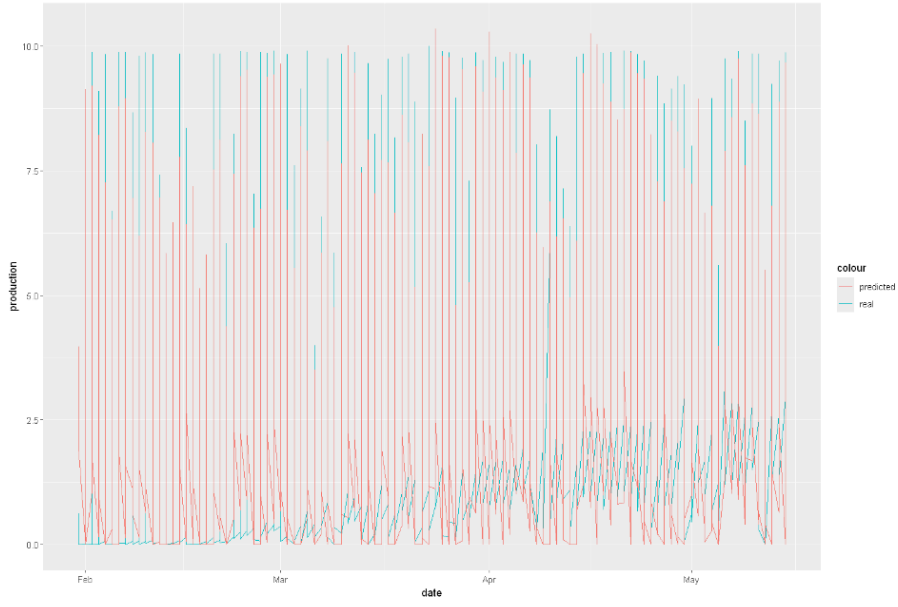
Summary of performance metrics for the first model compared with the baseline linear model

<u>Second Approach : Linear Regression with Lasso Variables + Transformed Hour + Lag 10:</u>

In our second approach, we enhanced the initial model by including lag 10 values, as indicated by the PACF plot. To do this, a new feature called prod_lag_10—which shows the production value lag by 10 intervals. Then, using Lasso regression, we updated our feature set to along with the previously chosen significant variables. After fitting the linear regression model, we evaluated its performance by analyzing the residuals and performing the Ljung-Box test.



Residuals plot , ACF of residuals , and histogram with density plot of residuals for the second approach




```
        Ljung-Box test

data:  Residuals
Q* = 597.35, df = 10, p-value < 2.2e-16

Model df: 0.   Total lags used: 10
```

Comparison of predicted and real production values over time for second approach

```
accu2 <- accu("2. Lasso(1sd) - LM_lag10 (IM)", second_model_evaluation$production, second_model_evaluation$predicted)
comparisons <- rbind(comparisons, accu2)
comparisons
```

A data.frame: 3 × 11

model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.389830	0.6699049	0.10482547	4469.314	1.993499	1.564115	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.389830	0.6699049	0.10880953	4058.146	1.994581	1.573241	0.3109070	0.3109070
2. Lasso(1sd) - LM_lag10 (IM)	1262	5.044160	3.386026	0.6712765	0.07450837	4407.538	1.908707	1.505325	0.2984292	0.2984292

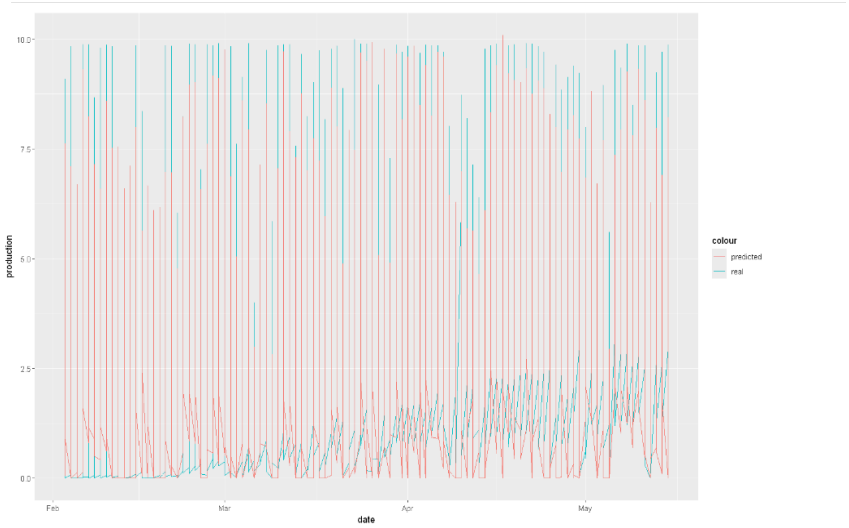
Summary of performance metrics for the second model compared with the previous models

The results indicate that the model with lag 10 values improves accuracy, showing lower RMSE and MAPE compared to both the Baseline-LM and Lasso(1sd)-LM models. This analysis reveals that the underprediction issue is somewhat resolved in this model. However, we agree that this model is not feasible because it is not practical to use lag 10 values in a real-time forecasting setting.This led us to develop a more practical forecasting model using lag 24 and lag 36 values.

Third Approach : Linear Regression with Lasso Variables + Transformed Hour + Lag 24+ Lag 36 (1sd):

In the third approach, we transformed the data to include lag 24 and lag 36 of the production values. This means the model uses the production values from 24 and 36 hours before the current hour as predictors. We also transformed the hour variable to a quadratic form: −(hour−12)2−(hour−12)2, ensuring the hour column peaks around noon when solar production is typically highest. For feature selection, we identified the feature columns and used them to create the predictor matrix XX and response vector yy. We

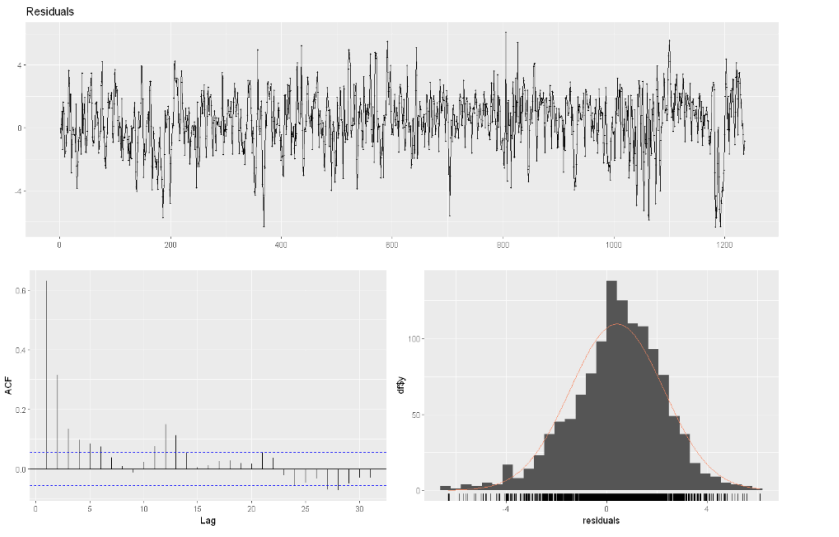
then fitted a linear regression model using these selected features.To evaluate the model, we calculated various performance metrics and conducted residual analysis to check for autocorrelation and normality.




```
        Ljung-Box test

data:  Residuals
Q* = 667.49, df = 10, p-value < 2.2e-16

Model df: 0.   Total lags used: 10
```

Comparison of predicted and real production values over time for third approach

A very low p-value at Ljung-Box test (< 2.2e-16) suggests significant autocorrelation, indicating that the model has not fully captured all temporal dependencies.



Residuals plot , ACF of residuals , and histogram with density plot of residuals for the second approach

The residuals roughly follow a normal distribution but with some deviations, especially in the tails. This suggests that while the model errors are somewhat normally distributed, there may be some outliers or non-normality.
The ACF plot of residuals checks for autocorrelation, and there are significant spikes at certain lags, indicating that the model has not fully accounted for temporal dependencies in the data.

In the third model, we experienced an underprediction issue again. To determine which variables are significant or not for the predictions in a linear regression model, the p-values associated with the coefficients of each variable in the model summary are typically examined. The model summary indicated that many variables were not significant for the predictions. Based on that and the results of our tests indicating that the model is insufficient, we will employ Lasso regression with a higher penalty parameter when constructing our fourth model.

```
accu3 <- accu("3. Lasso(1sd)-LM_lag24_lag36", third_model_evaluation$production, third_model_evaluation$predicted)
comparisons <- rbind(comparisons, accu3)
comparisons
```

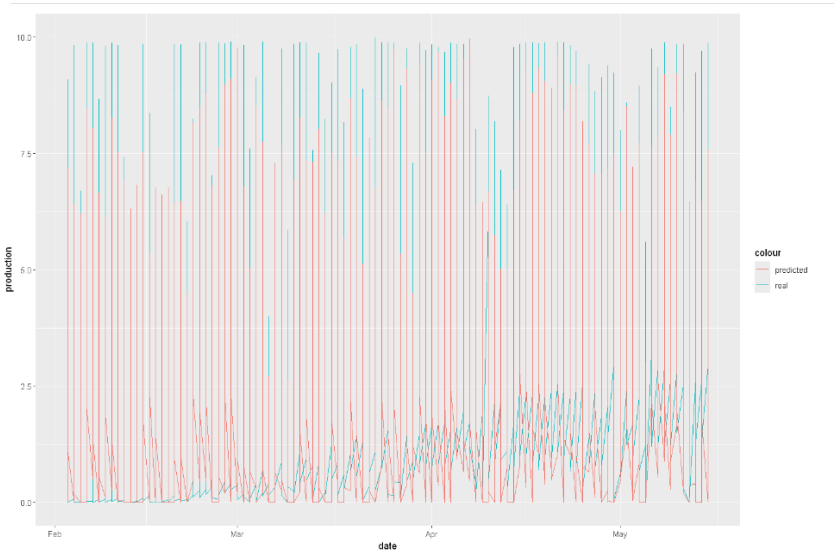
A data.frame: 4 × 11

model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.389830	0.6699049	0.10482547	4469.314	1.993499	1.564115	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.389830	0.6699049	0.10880953	4058.146	1.994581	1.573241	0.3109070	0.3109070
2. Lasso(1sd) - LM_lag10 (IM)	1262	5.044160	3.386026	0.6712765	0.07450837	4407.538	1.908707	1.505325	0.2984292	0.2984292
3. Lasso(1sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.08091652	4201.848	1.906118	1.493561	0.2954455	0.2954455

Summary of performance metrics for the third model compared with the previous models

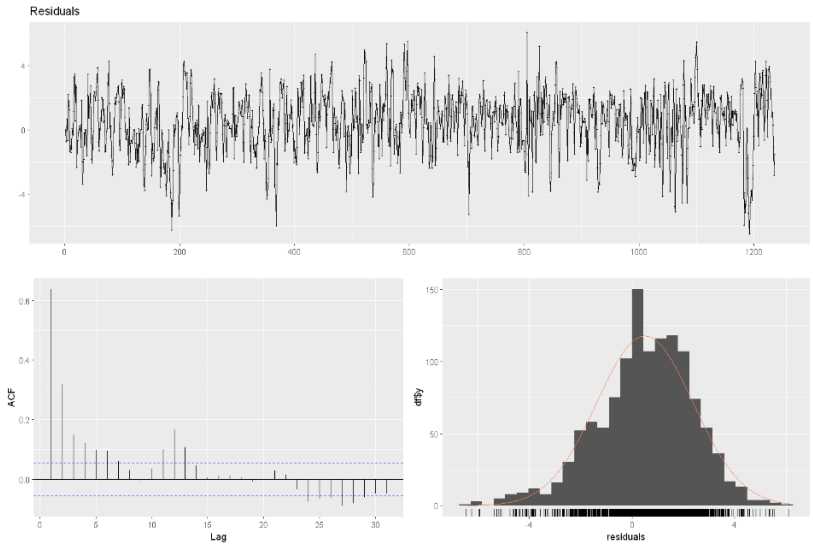
Fourth Approach : Linear Regression with Lasso Variables + Transformed Hour + Lag 24+ Lag 36 (3sd):

As the fourth model, we built a time series linear regression model with Lasso variables, which were acquired using a different lambda value. In this case, lambda was chosen as the one within three standard deviations of the minimum MSE. Subsequently, as we did in the previous approaches, we plotted the residual plots, performed the Ljung-Box test, and compared its performance with the previous models.




```
        Ljung-Box test

data:  Residuals
Q* = 707.24, df = 10, p-value < 2.2e-16

Model df: 0.   Total lags used: 10
```

Comparison of predicted and real production values over time for fourth approach



Residuals plot , ACF of residuals , and histogram with density plot of residuals for the fourth approach

```
accu4 <- accu("4. Lasso(3sd)-LM_lag24_lag36", fourth_model_evaluation$production, fourth_model_evaluation$predicted)
comparisons <- rbind(comparisons, accu4)
comparisons
```

A data.frame: 5 × 11

model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.389830	0.6699049	0.10482547	4469.314	1.993499	1.564115	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.389830	0.6699049	0.10880953	4058.146	1.994581	1.573241	0.3109070	0.3109070
2. Lasso(1sd) - LM_lag10 (IM)	1262	5.044160	3.386026	0.6712765	0.07450837	4407.538	1.908707	1.505325	0.2984292	0.2984292
3. Lasso(1sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.08091652	4201.848	1.906118	1.493561	0.2954455	0.2954455
4. Lasso(3sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.09916113	4422.851	1.946731	1.549387	0.3064887	0.3064887

Summary of performance metrics for the fourth model compared with the previous models

The presence of patterns in the residuals, significant autocorrelations, and deviations from normality suggest that the model could be improved by addressing these issues. As we see at the accuracy metrics, imposing a higher penalty unfortunately did not help in reducing errors.

Using ggpairs, we examined each variable's link to production in more detail. We were unable to identify an unmodeled association between the factors and the response variable using ggpair plots. The ggpair graph of every variable appears to show very little relationship. As a result, we now stop using linear regression models and switch to ARIMA models.

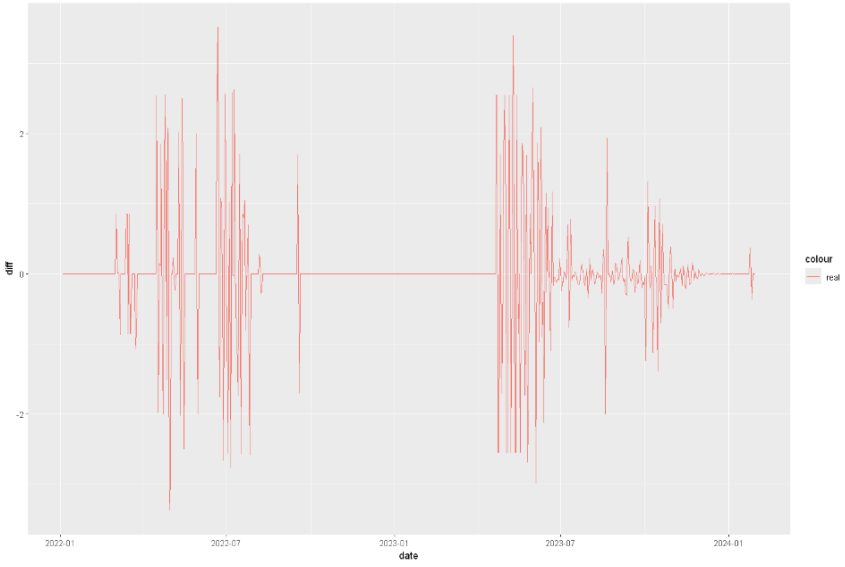
Fifth Approach : ARIMA models for each hour:

In this approach, we start by grouping the data by hour. During the testing phase created ARIMA models with auto.arima for every test day by iterating over the days.This method allows us to update our ARIMA model daily with the latest available information.

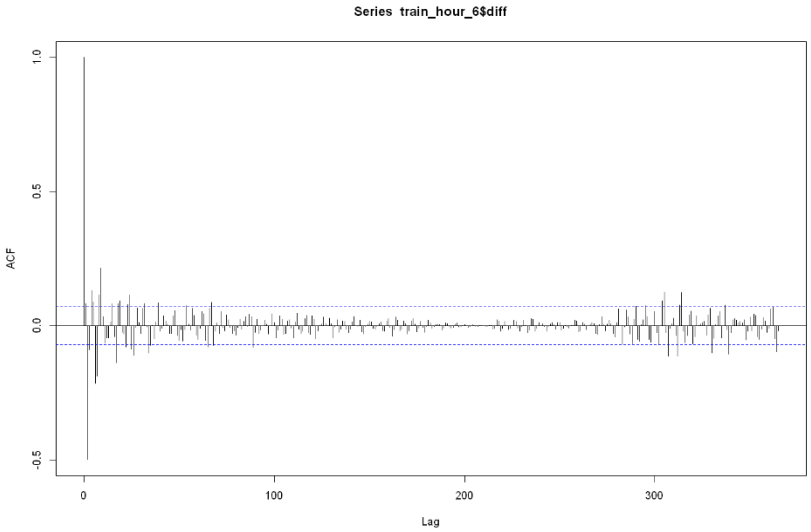
For Hour 6:

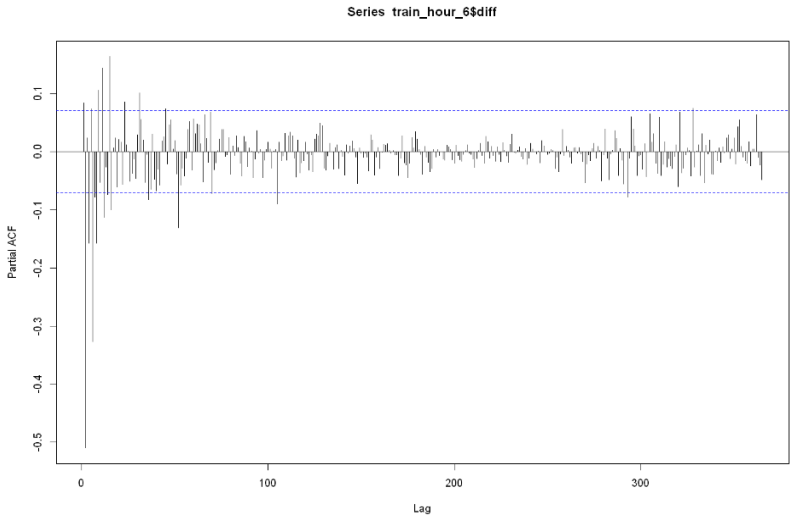
Geom plot between date and production data for hour 6

The geom plot provides a clear visualization of the seasonal and daily variations in solar power production for hour 6. We first did differencing on the non-stationary data before building an ARIMA model. It is significant to highlight that using a 1-day difference is not practical in our forecasting scenario because the most recent information available is from 2 days prior.

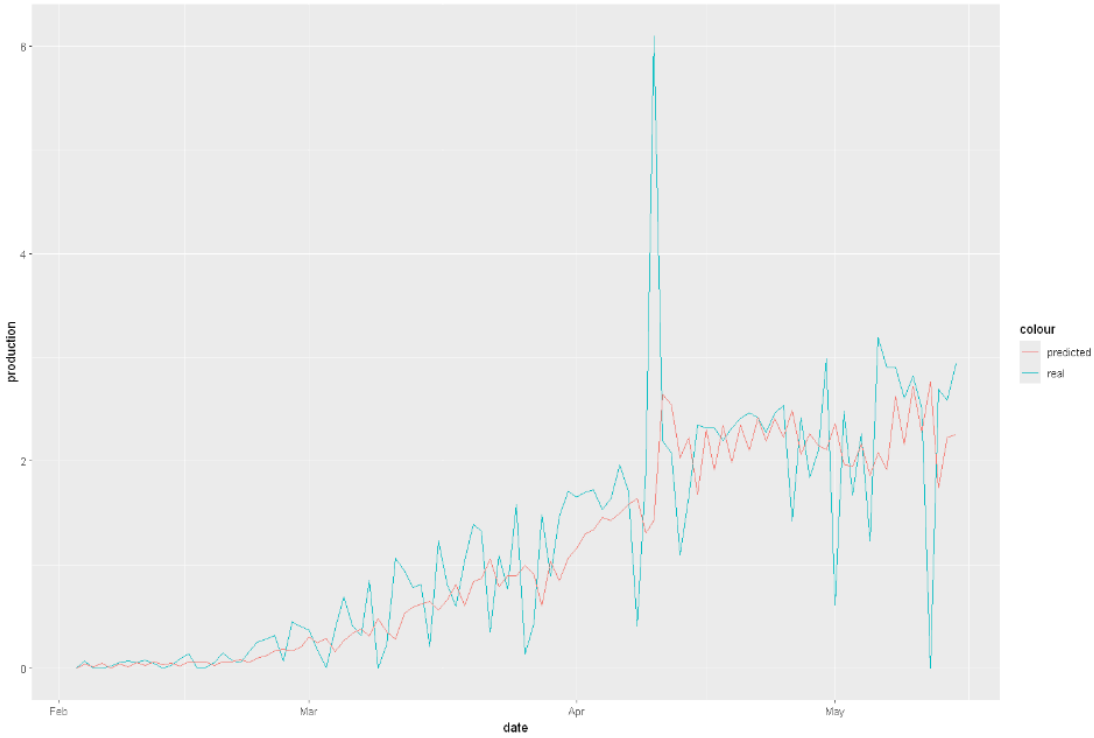


Differenced production data over time for hour 6



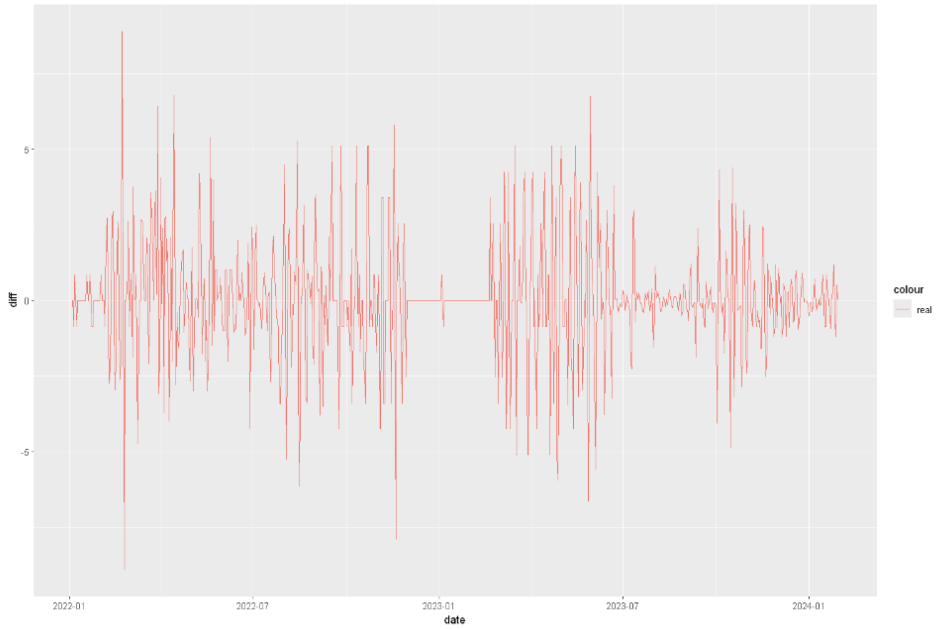


ACF and PACF plots of differenced production data for hour 6

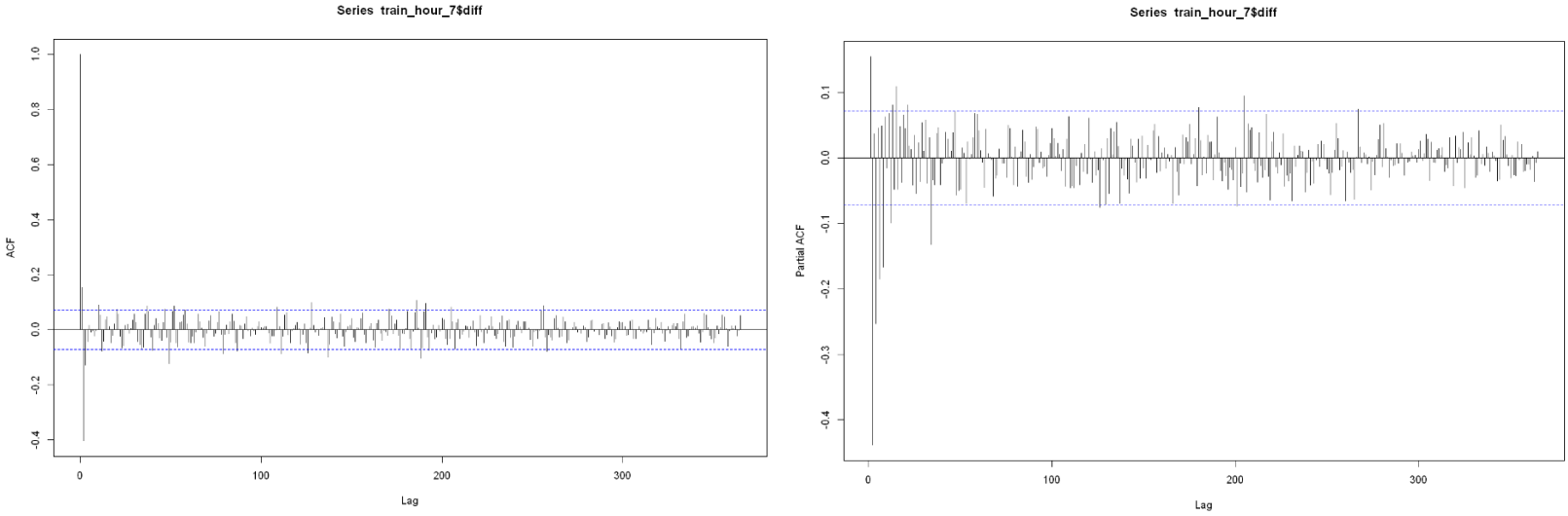


Predicted vs. actual production values for hour 6

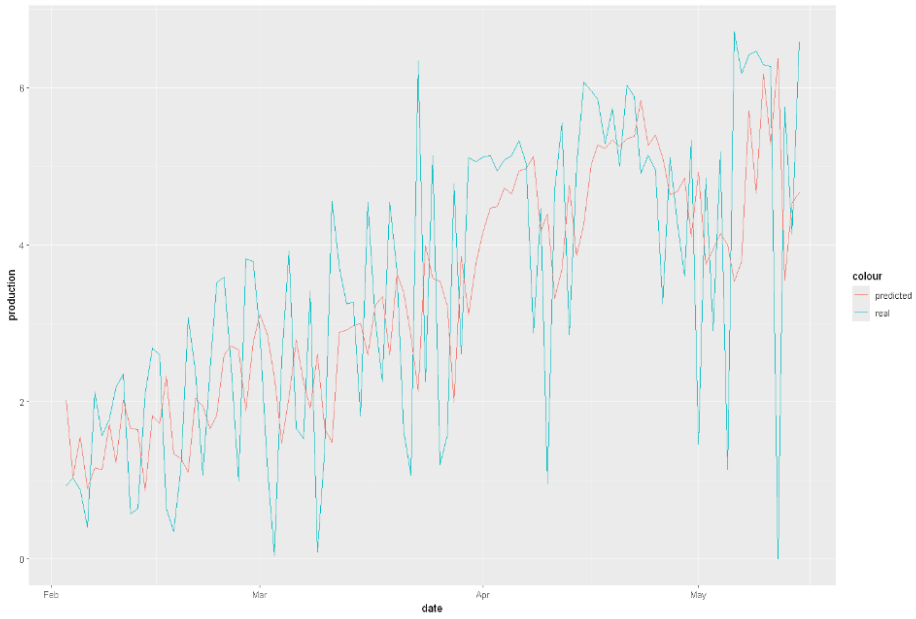
For Hour 7:



Geom plot between date and differenced production data for hour 7



ACF and PACF plot of differenced production data for hour 7



Predicted vs. actual production values for hour 7

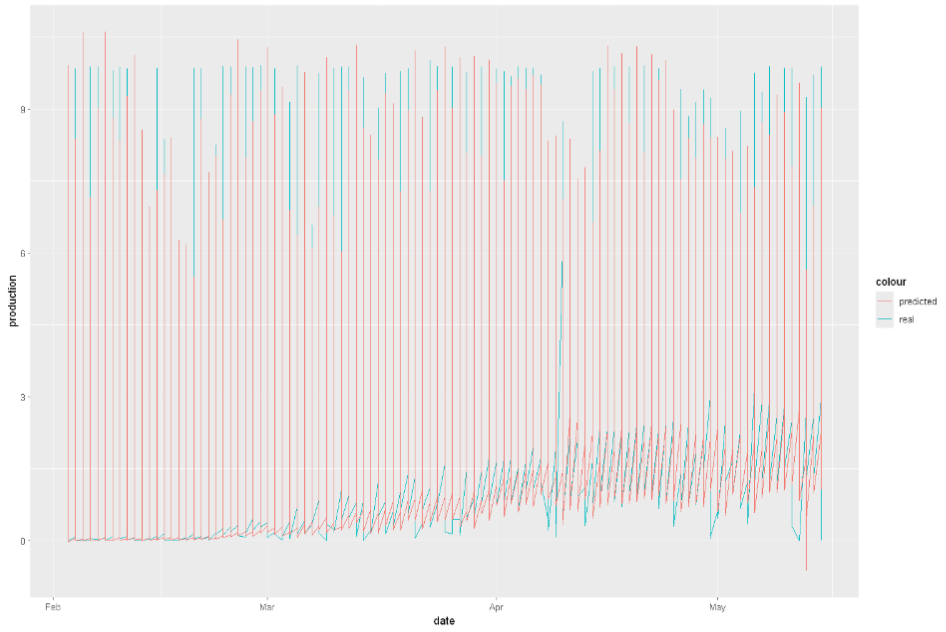
```r
# Fit a simple ARIMA model without external regressors
simple_model <- auto.arima(train_hour_7$diff)
print(summary(simple_model))
```

```
Series: train_hour_7$diff
ARIMA(1,0,2) with zero mean

Coefficients:
         ar1     ma1      ma2
      0.1603  0.1911  -0.7010
s.e.  0.0537  0.0375   0.0318

sigma^2 = 2.386:  log likelihood = -1400.91
AIC=2809.81   AICc=2809.87   BIC=2828.32

Training set error measures:
                     ME     RMSE      MAE MPE MAPE      MASE         ACF1
Training set 0.05620919 1.541624 1.040589 NaN  Inf 0.6586888 -0.006162245
```

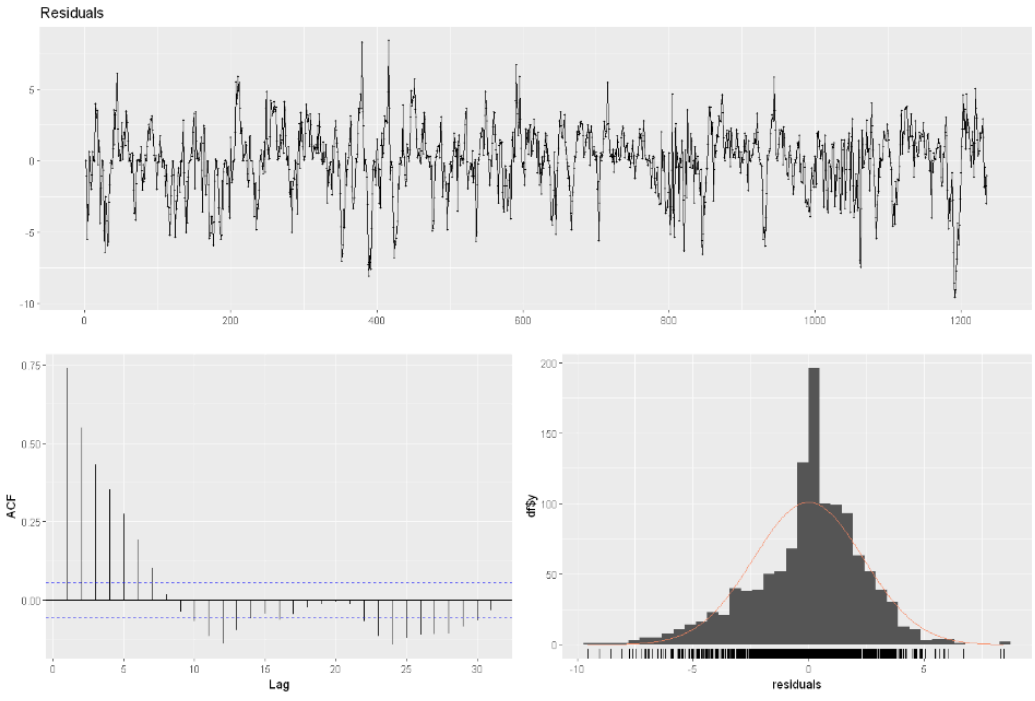
Summary of the simple ARIMA model for hour 7

Summary of fifth approach: To create separate ARIMA models for each hour, we followed the same steps shown above for hours 6 and 7. For each hour, we first examined the geom plot and ACF/PACF of our data to assess trends and stationarity. We then applied differencing to remove trends and achieve stationarity, analyzed the resulting plots, and compared the actual and predicted values. Then, we combined these, and the outcomes were as follows:

```
ggplot(arima_results,aes(x=date)) + geom_line(aes(y=production,color='real')) + geom_line(aes(y=predicted,color='predicted'))
```



Predicted vs. actual production values for unified all hours



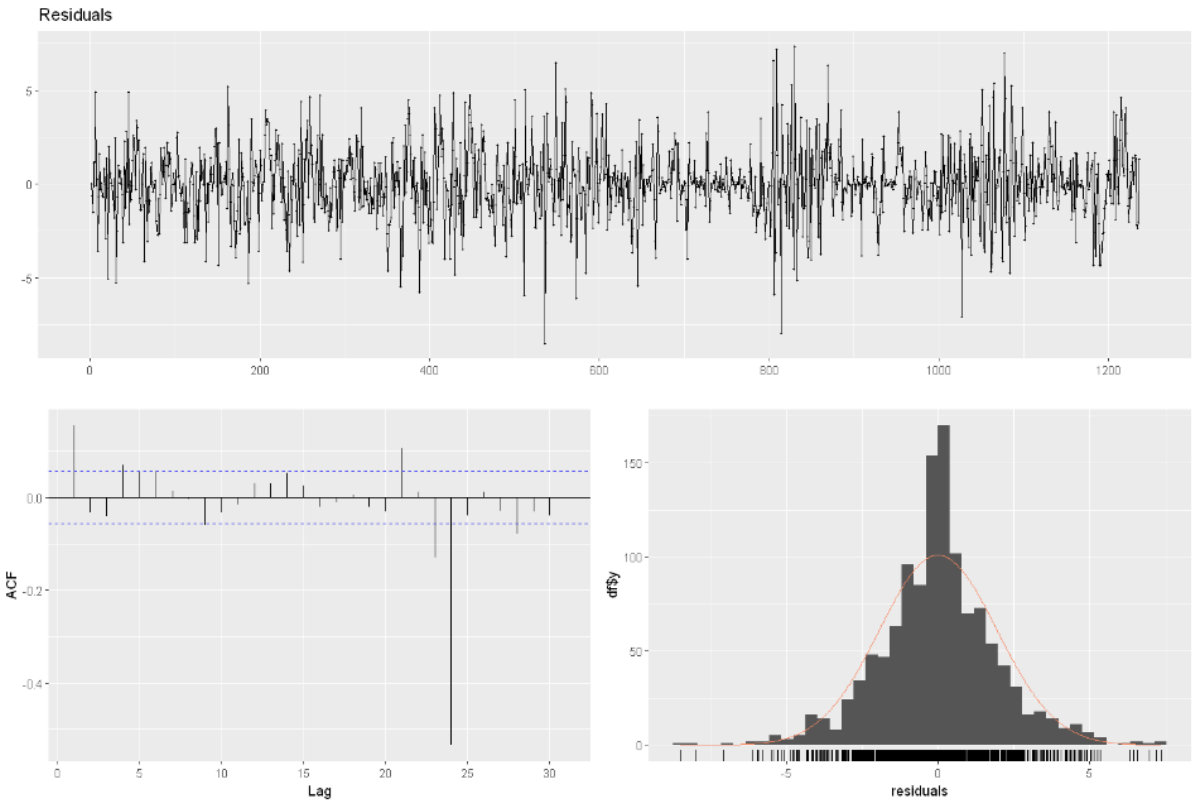
Residuals plot , ACF of residuals , and histogram with density plot of residuals for fifth model

model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.389830	0.6699049	0.1048254650	4469.314	1.993499	1.564115	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.389830	0.6699049	0.1088095279	4058.146	1.994581	1.573241	0.3109070	0.3109070
2. Lasso(1sd) - LM_lag10 (IM)	1262	5.044160	3.386026	0.6712765	0.0745083684	4407.538	1.908707	1.505325	0.2984292	0.2984292
3. Lasso(1sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.0809165157	4201.848	1.906118	1.493561	0.2954455	0.2954455
4. Lasso(3sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.0991611324	4422.851	1.946731	1.549387	0.3064887	0.3064887
5. ARIMA for each hour	1236	5.055283	3.381461	0.6688964	0.0007428499	6006.657	2.373454	1.752375	0.3466422	0.3466422

Summary of performance metrics for the fifth model compared with the previous models

Sixth Approach : ARIMA for overall data: Finally, we attempted to create an ARIMA model using all of our data.While doing this, we used the training set, and the results were as follows:

Predicted vs. actual production values for fifth model



Predicted vs. actual production values for fifth model

Final Approach & Results:

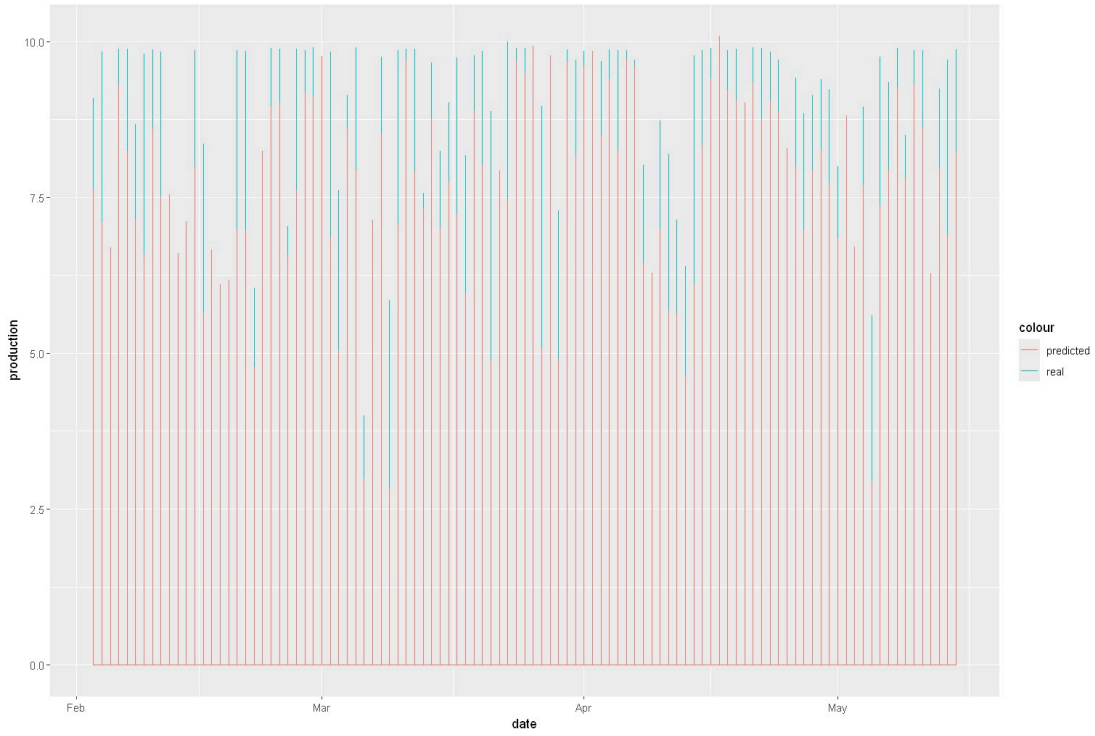
Now that we have elaborated on all of our approaches in detail, let's focus on the best performing approach that we have developed. In the table below, the comparison of all the developed approaches' forecasting performances can be seen. In the table, it is clearly seen that the 3rd approach (Linear Regression with Lasso Variables + Transformed Hour + Lag 24+ Lag 36 (1sd)) is the best performing approach in the sense that it yields the most accurate forecasts.

A data.frame: 8 × 11

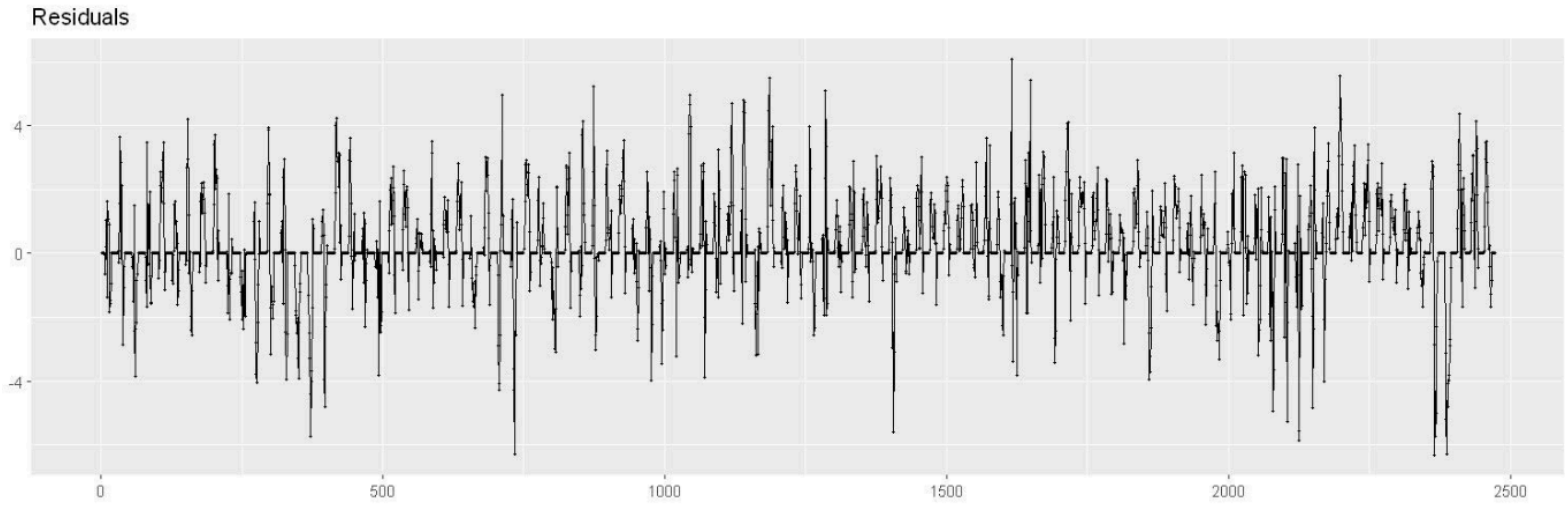
model	n	mean	sd	CV	FBias	MAPE	RMSE	MAD	MADP	WMAPE
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0. Baseline-LM	1269	5.060165	3.389830	0.6699049	0.1048254650	4469.314	1.993499	1.5641154	0.3091036	0.3091036
1. Lasso(1sd)-LM	1269	5.060165	3.389830	0.6699049	0.1088095279	4058.146	1.994581	1.5732406	0.3109070	0.3109070
2. Lasso(1sd) - LM_lag10 (IM)	1262	5.044160	3.386026	0.6712765	0.0745083684	4407.538	1.908707	1.5053248	0.2984292	0.2984292
3. Lasso(1sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.0809165157	4201.848	1.906118	1.4935608	0.2954455	0.2954455
4. Lasso(3sd)-LM_lag24_lag36	1236	5.055283	3.381461	0.6688964	0.0991611324	4422.851	1.946731	1.5493871	0.3064887	0.3064887
5. ARIMA for each hour	1236	5.055283	3.381461	0.6688964	0.0007428499	6006.657	2.373454	1.7523747	0.3466422	0.3466422
6. ARIMA for overall (IM)	1236	5.055283	3.381461	0.6688964	0.0012032262	3022.291	1.935345	1.4082776	0.2785754	0.2785754
Final Model (3rd model with all hours)	2472	2.537112	3.473294	1.3689953	0.0843471019	2100.963	1.349820	0.7562505	0.2980754	0.2980754

Comparison of All Developed Approaches

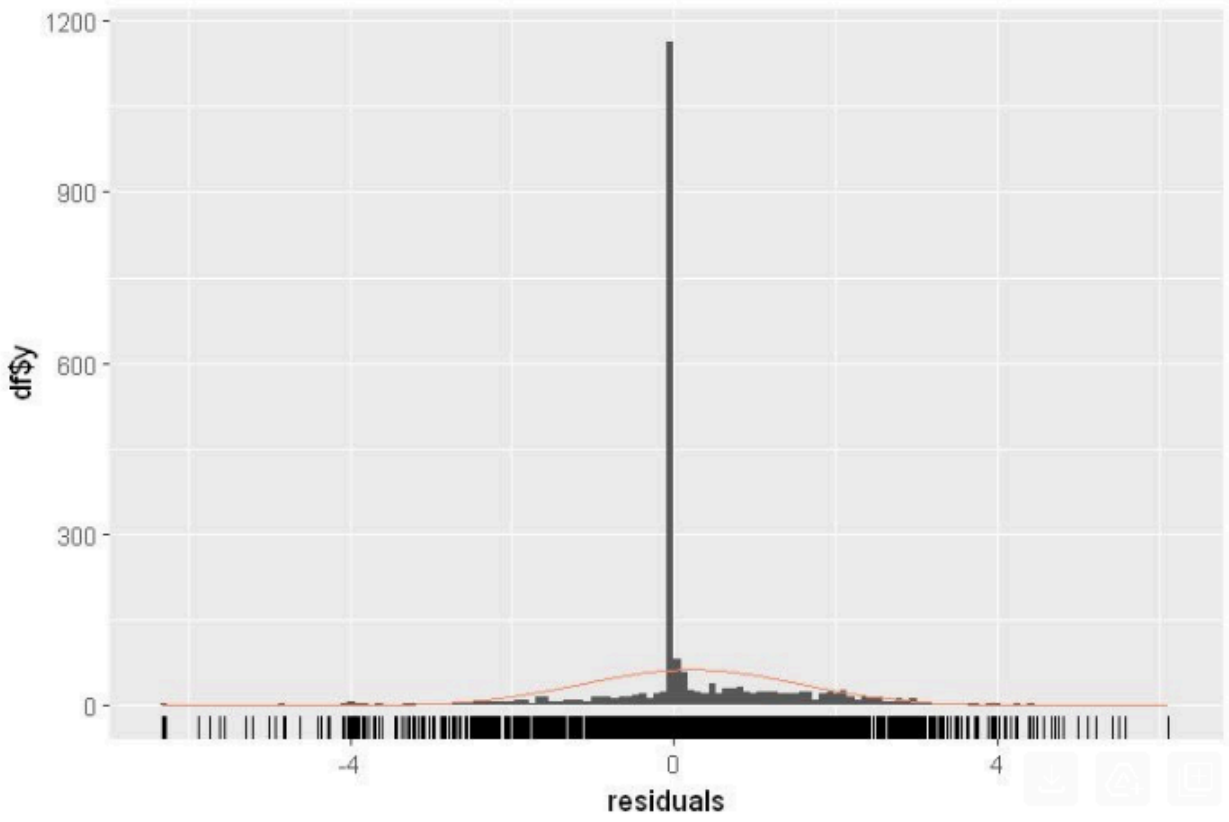
In the table, we see that the final model performs the best. After determining the best approach, we moved on with testing it. Our testing time horizon was 1st of February to 15th of May 2024. Between these dates, we tested our model by producing forecasts for each date and comparing these forecasts to the actual realisations of solar power production in these dates. This time, for the sake of a thorough testing, we also included the hours of the day which we had previously determined as zero-production hours. In the above table, it is seen that this model performed the best based on MAPE comparison. More elaborative forecast results yielded by our final approach can be visually seen in the figures below:



Our Final Approach Performance over the Test Period (February 1st - May 15th 2024)



Residual Analysis of Our Final Approach



Residual Analysis of Our Final Approach

In the residual analyses, we can examine our models in terms of the behaviour of its residuals. We see that the residuals have a constant mean and variance, which is highly required in a good time series model. In the second chart, residuals clearly show a normal distribution pattern with no visible skew or autocorrelation. These findings strengthen our belief in the accuracy of the final model, and show us that there is little to no remaining unaccounted for information that is required for the model.

Conclusions:

In this study, we addressed the challenge of forecasting the hourly solar power production at the Edikli Solar Power Plant in Niğde, Turkey. Given that solar energy production is inherently dependent on meteorological factors, we aimed to develop a model that leverages these variables to predict solar power generation accurately.Our approach involved several stages. First, we organized our data in Python and applied a general regression analysis. After this initial step, we decided to continue our work in R. We then applied Lasso regression to select the most significant variables, refining our feature set. We evaluated models with different sets of lagged values (Lag 10, Lag 24, and Lag 36) and lambda values. Next, we developed ARIMA models for each hour and an overall ARIMA model, accounting for temporal dependencies and seasonality. The best-performing approach was a linear regression model with Lasso-selected variables, a transformed hour, and lagged production values (Lag 24 and Lag 36). The resulting model had the best performance metrics and showed the highest forecasting accuracy. and thorough residual analysis showing no significant autocorrelation or non-normality.

In conclusion, our study highlights the importance of incorporating meteorological variables, appropriate feature transformations, and lagged production values in developing effective forecasting models for solar power production. The performance of the final model underscores its potential for real-world applications in electricity pricing and capacity planning, providing a valuable tool for optimizing operations at the Edikli Solar Power Plant.

Future Work & Improvements:

Expanding the variable base would be a good improvement for our model. Although our current variable data provide a solid foundation for our forecasts, a good additional variable information would be the weather forecast for the solar power plant's location. By using this information, instead of using past weather condition data we can use much more flexible and on point weather forecast data. This is especially essential in today's world, where the climate changes rapidly and each year shows more and more extreme weather conditions. According to sector analyses, with the ongoing climate change trend, the solar panels will be more and more valuable in terms of electricity generation (Putol, 2024). Therefore, its impact on seasonality of production should be considered as well, which may also lead to the creation of a new seasonal climate change variable.

Links to the Project Codes:

R Codes:
https://github.com/BU-IE-360/spring24-melistuncer/blob/61a337aef120383994cade7978ea2059617264f4/IE360_Project_RCodes_Group28.ipynb

Python Codes:
https://github.com/BU-IE-360/spring24-melistuncer/blob/61a337aef120383994cade7978ea2059617264f4/IE360_Project_Data_processing_Group28.ipynb

References:

1. Adams, S. & Lambert, D. (2006). Earth Science: An Illustrated Guide to Science. The Diagram Group.
2. Desonie, D. (2012). CK-12 Earth Science for High School. CK-12 Foundation. Retrieved from: https://archive.org/details/ost-earth-sciences-ck_12_earth_science_for_high_school/page/n1/mode/2up
3. Montgomery et al. (2008). Introduction to Time Series Analysis and Forecasting
4. Hyndman, R. J. & Athanasopoulos, G. (2018). Forecasting: Principles and Practice
5. Putol, R. (2024). Climate change will increase the value of solar panels. *Earth.com*. https://www.earth.com/news/climate-change-will-increase-the-value-of-solar-panels/