Group 18

Solar Energy Production Forecasting

Ender Purcu - 2020402000
Arda Selçuk - 2021402222

Introduction

In our project for IE360 course Spring 2024, we were asked to make daily predictions for hourly solar power generation of Edikli GES from a given data set.

The data provided included several weather condition datas for 25 coordinates which are as follows:
DSWRF_surface : Downward shortwave radiation flux
USWRF_surface, USWRF_top_of_surface: Upward shortwave radiation flux
DLWRF_surface: Downward longwave radiation flux
TCDC_low.cloud.layer, TCDC_middle.cloud.layer, TCDC_high.cloud.layer, TCDC_entire.atmosphere: Total cloud cover
CSNOW_surface: Categorical snow variable
TMP_surface: Temperature

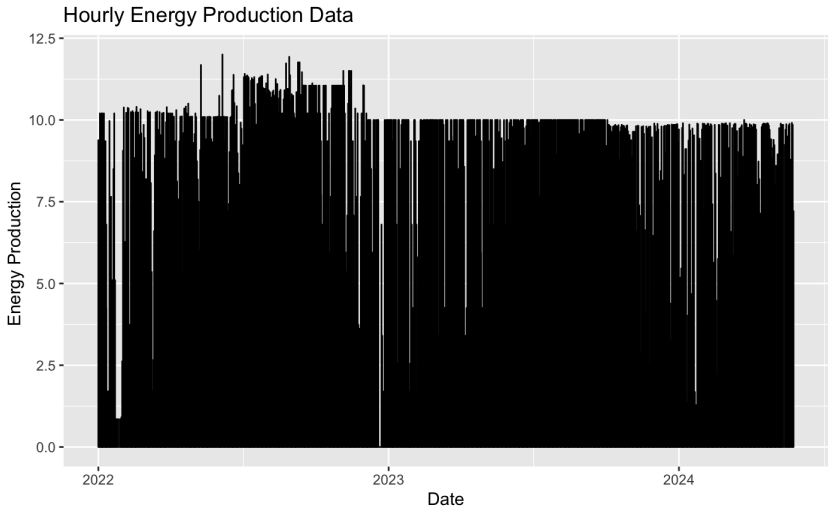
In addition to these, we were also provided with the hourly production values for these plants.

Our data set started from 01-01-2021 00:00 and the data we will be reporting will end on 15-05-2024 23:00 as we were tasked to evaluate our prediction approaches from the 1st of February to 15th of May 2024.

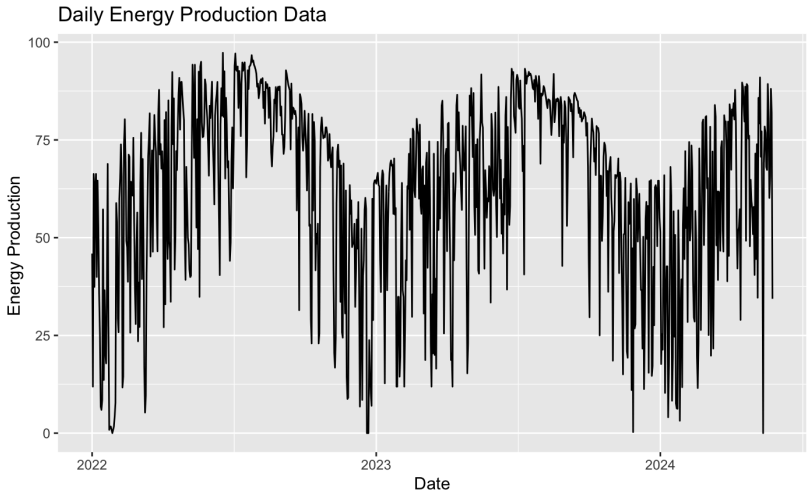
The approaches that we'll be using include time series regression after the preprocessing of the data given and ARIMA modelings for the predictions.

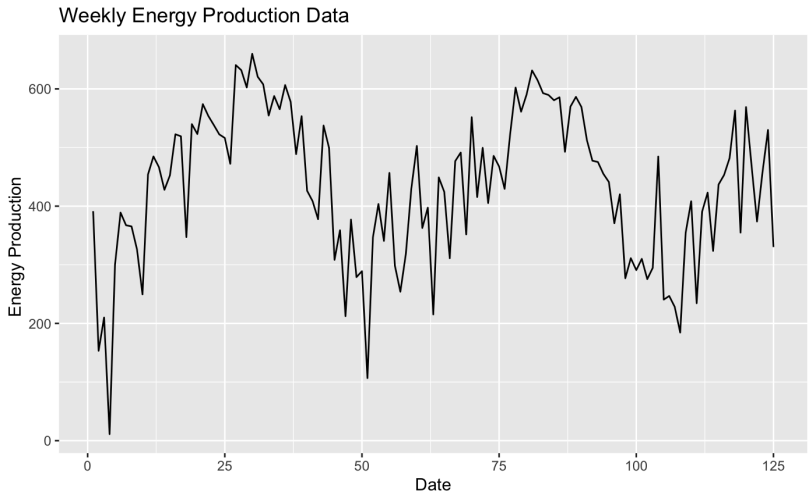
Data Analysis

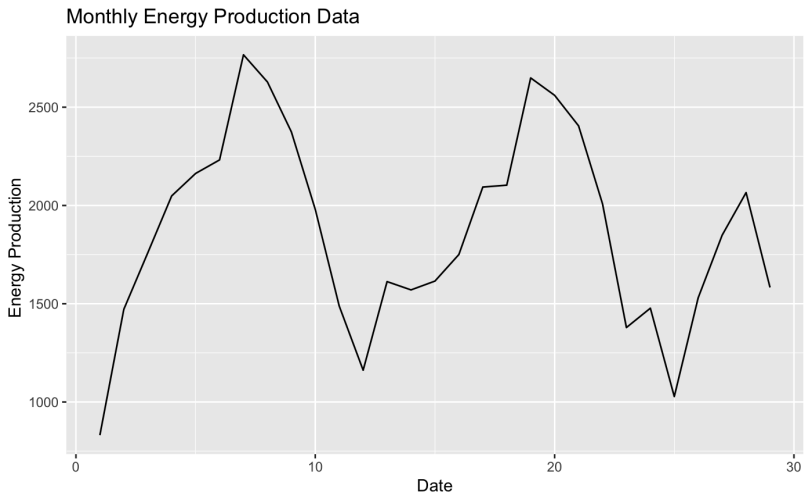
First, we plotted the hourly production data provided to us to start analyzing our dependent variable.



Unfortunately, because of the data amount, the visualization of the hourly data doesn't give much insight to be used. Thus, we adjust the data to give us total daily, weekly and monthly productions which may help to see some trends or seasonal changes.



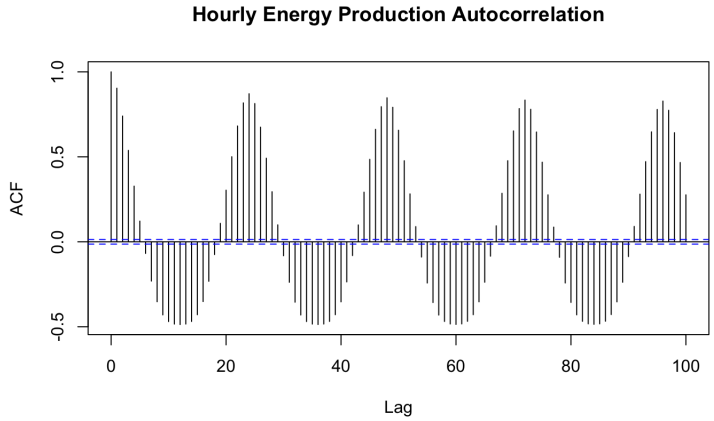




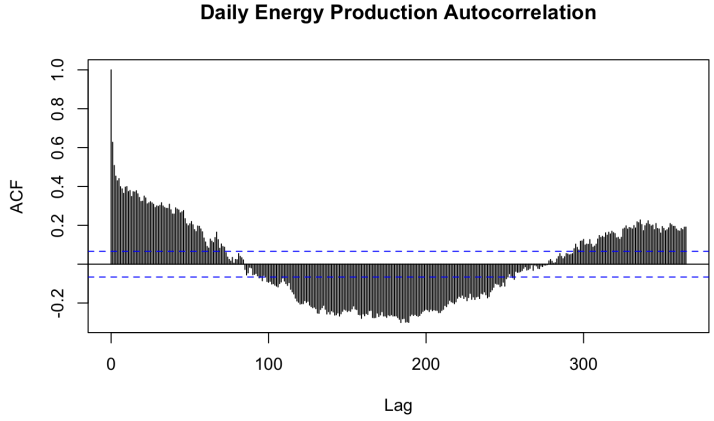
By observing our aggregated production data, we can make some deductions which will be useful in the model building phase.

We can observe from different periods of aggregation that there is an oscillating seasonality in the productions which indicates an increased solar energy production in the summer and lower production in winter months. We can also see from the daily production plot that the variance in the winter months are much more than the ones in summer months. We can relate this with the changing weather conditions that can be observed in winter and the consistent sunny weathers in summer.

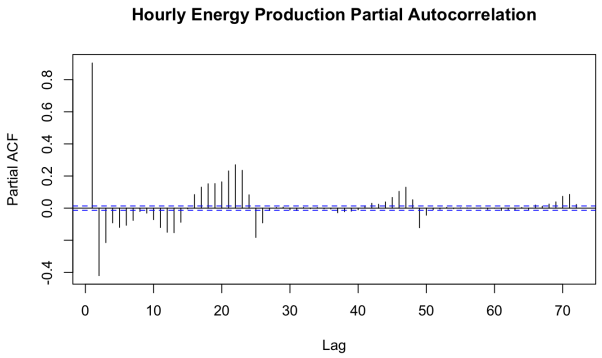
Next, we can check the autocorrelation plots to inspect the seasonalities more closely.

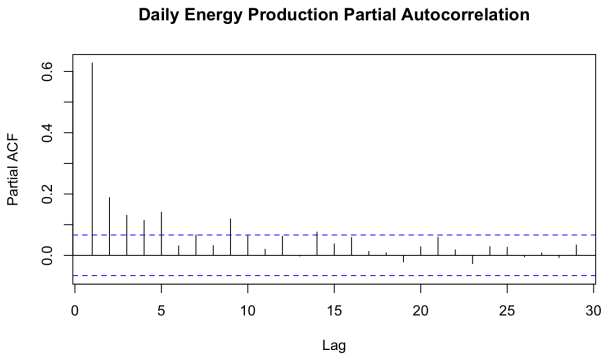


From the hourly autocorrelation function, we can observe that the data has a repeating pattern in every 24 intervals, indicating that there is an hourly seasonality within a day which we have to handle in the model building phase.



Similarly, the daily production autocorrelation function has a sinusodial pattern after 365 days, indicating a seasonal pattern within each year.
Next, we can check the partial autocorrelation functions for further inspection on the data.

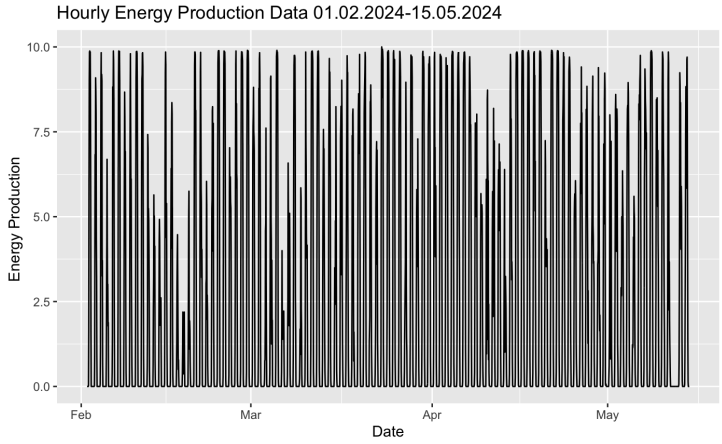




From the partial autocorrelation of the hourly data, we can see spikes in every 24 interval, further proving that our data has a daily seasonality. Also the spike at lag1 indicates that there may be a trend involved with our production data.

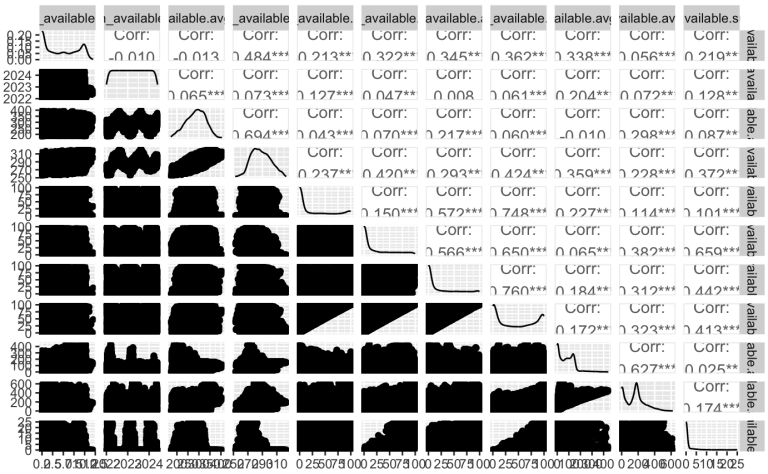
The daily data doesn't contain much information but we can observe that the high spikes go on till the 5th interval and then drop significantly.

Before moving forward, we can also categorize our production data to simplify our models. To do this, we added a categorical variable of "is_sun" where we categorize the production values for the actively producing hours and passive hours where no production occurs. We chose the interval of 6 AM and 7 PM to be our active hours and rest the 0 production hours.



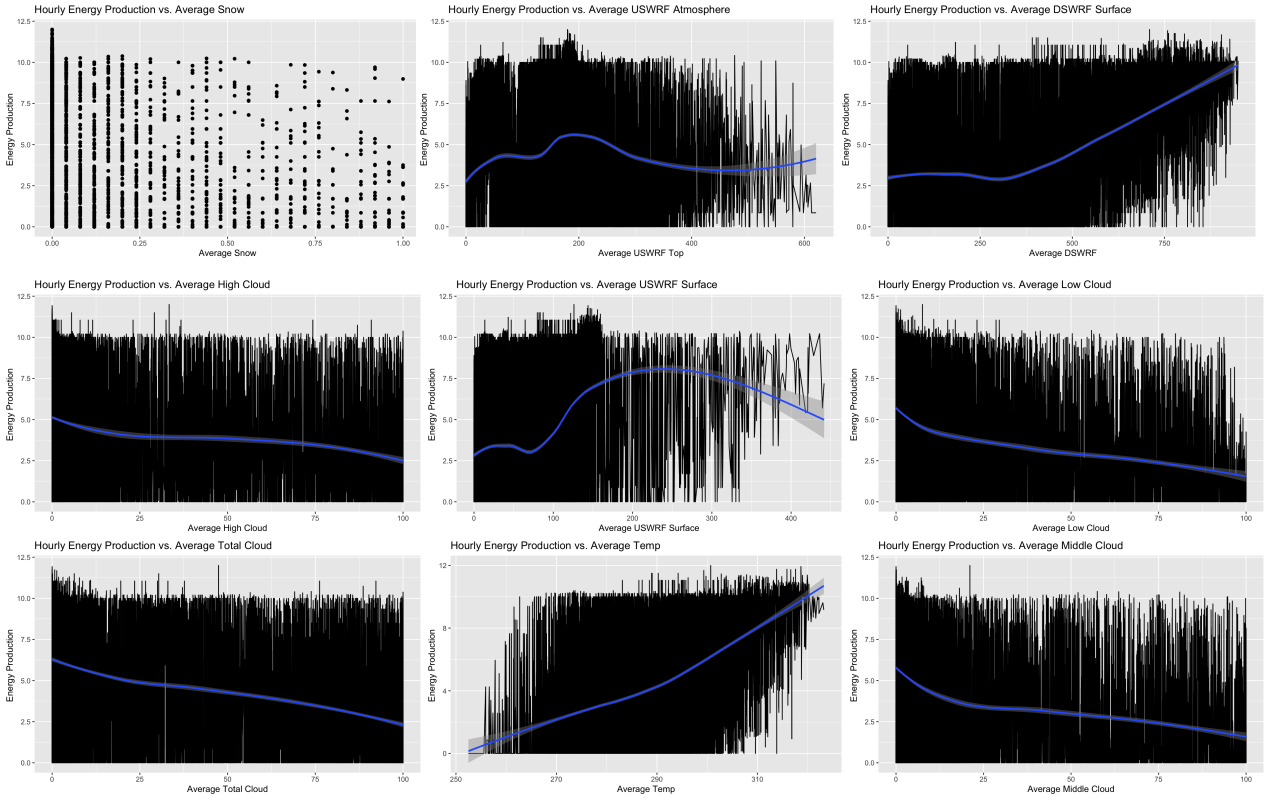
Next up, we can check our weather data to look for the correlations between the productions and the possible predictor variable candidates.

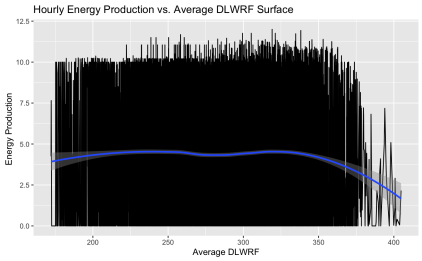
Before doing that, as we have 25 locations for each of these variables, we took the averages of all the weather data categories.



From the correlation plot, we can see that almost all of the variables came out to be significant for the correlation between production data and the weather data except the average DLWRF_surf. We will be later checking their correlations and the significance to our model again on our model building phase.

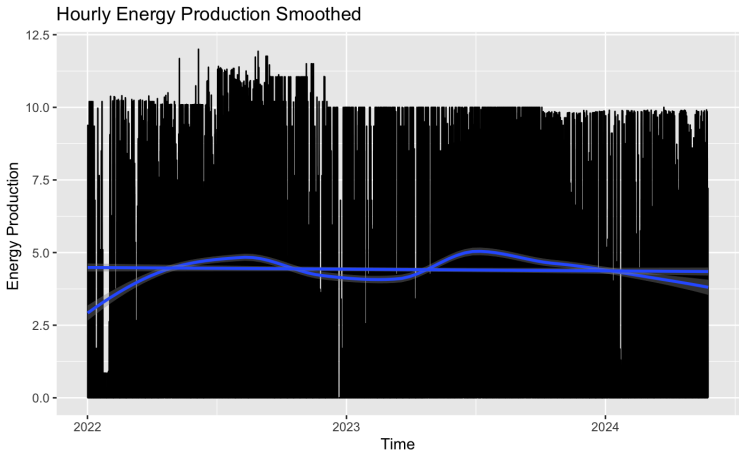
We can also plot the production versus the variable to see the relations between them.





Although it is hard to observe the relations because of the amount of data we have, we can see that if we smooth the data they are mostly linear except USWRF data and the snow categorical variable.

Lastly, let's look at the production with respect to time to see if we can see an observable trend in our data.



From this plot, we can observe the seasonality but a trend can't be seen directly from the graph.

Approach

Time Series Regression

Model 1

We start our modeling by dividing our test and training data. As we were asked to report on our approaches from February 1st 2024 till May 15th 2024, our test data will consist of data within this period. Our training data will consist of data before February 1st 2024.

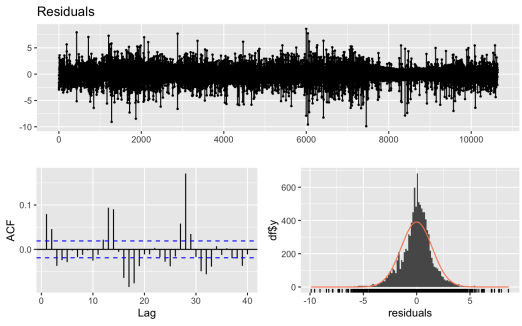
After dividing our data, we initiate a model with our already known properties. We know that the data shows 2 types of seasonality. On a monthly basis and a daily basis, we can add those two terms to our

model. In addition to the seasonal components, we also saw in our PACF graph that in lag 1 and in lag 24, there were significant spikes which can be added to our model as well. Since we removed hours 20,21,22,23,00,01,02,03,04 and 05, the adjusted days will consist of 14 hours and PACF will show spikes in lag 14. Thus we add lag14 instead of lag24.

The resulting model is as follows:

```
Call:
lm(formula = production ~ mon + as.factor(hour_id) + Lag14 +
    Lag1, data = model_data_tr)

Residuals:
    Min      1Q  Median      3Q     Max
-9.8897 -0.6897  0.0730  0.7902  8.6023

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.488431   0.071663   6.816 9.89e-12 ***
monAug               0.237178   0.071359   3.324 0.000891 ***
monDec              -0.223461   0.071485  -3.126 0.001777 **
monFeb              -0.077109   0.072529  -1.063 0.287740
monJan              -0.300394   0.066204  -4.537 5.76e-06 ***
monJul               0.274782   0.071707   3.832 0.000128 ***
monJun               0.112372   0.071191   1.578 0.114489
monMar              -0.089938   0.070668  -1.273 0.203158
monMay               0.071019   0.070524   1.007 0.313946
monNov              -0.177527   0.071579  -2.480 0.013148 *
monOct               0.014136   0.070471   0.201 0.841023
monSep               0.187090   0.071629   2.612 0.009016 **
as.factor(hour_id)7  1.435912   0.076003  18.893  < 2e-16 ***
as.factor(hour_id)8  2.187998   0.081315  26.908  < 2e-16 ***
as.factor(hour_id)9  1.995202   0.089157  22.379  < 2e-16 ***
as.factor(hour_id)10 1.196201   0.094637  12.640  < 2e-16 ***
as.factor(hour_id)11 0.879963   0.096433   9.125  < 2e-16 ***
as.factor(hour_id)12 0.720929   0.096501   7.471 8.60e-14 ***
as.factor(hour_id)13 0.396919   0.095199   4.169 3.08e-05 ***
as.factor(hour_id)14 -0.210469  0.091806  -2.293 0.021893 *
as.factor(hour_id)15 -0.962148  0.086275 -11.152  < 2e-16 ***
as.factor(hour_id)16 -1.579514  0.080379 -19.651  < 2e-16 ***
as.factor(hour_id)17 -1.361516  0.076353 -17.832  < 2e-16 ***
as.factor(hour_id)18 -1.020975  0.075107 -13.594  < 2e-16 ***
as.factor(hour_id)19 -0.530419  0.074803  -7.091 1.42e-12 ***
Lag14                0.163682   0.006824  23.988  < 2e-16 ***
Lag1                 0.677217   0.006832  99.123  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.456 on 10613 degrees of freedom
  (14 observations deleted due to missingness)
Multiple R-squared:  0.8649,    Adjusted R-squared:  0.8646
F-statistic:  2613 on 26 and 10613 DF,  p-value: < 2.2e-16
```

We observe that even with the seasonal and difference variables we got a pretty high R^2 value which is 0.8646, meaning that our model will be able to explain 86.46% of the variation in the actual values. We can also check the residuals.



We can observe that the residuals resemble the normal distribution, which is a good direction to move forward but we can still see that correlation between the residuals exists. Thus, we can improve our model with further addition of variables.

Model 2

Next up, we added all the averaged variables to see how they affect our model knowing that some of these variables will be unrelated and we may need to remove them afterwards. We check for their significance in this model.

```
Call:
lm(formula = production ~ -1 + mon + as.factor(hour_id) + Lag14 +
    Lag1 + avg_temp + avg_DSWRF + avg_lcloud + avg_mcloud + avg_hcloud +
    avg_tcloud + avg_USWRFtop + avg_DLWRFsurf + avg_USWRFsurf +
    avg_CSNOW, data = model_data_tr)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0197  -0.7158   0.0781   0.7668   8.0608

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
monApr                -6.2325540  1.5064692  -4.137 3.54e-05 ***
monAug                -6.0827269  1.5438259  -3.940 8.20e-05 ***
monDec                -6.9003819  1.4722342  -4.687 2.81e-06 ***
monFeb                -6.5021370  1.4640603  -4.441 9.04e-06 ***
monJan                -6.9197600  1.4682702  -4.713 2.47e-06 ***
monJul                -5.9998300  1.5300240  -3.921 8.86e-05 ***
monJun                -5.9474117  1.5146957  -3.926 8.67e-05 ***
monMar                -6.3202323  1.4862227  -4.253 2.13e-05 ***
monMay                -6.0845687  1.5117701  -4.025 5.74e-05 ***
monNov                -6.7954957  1.4878781  -4.567 5.00e-06 ***
monOct                -6.5232630  1.5070606  -4.328 1.52e-05 ***
monSep                -6.3087997  1.5296954  -4.124 3.75e-05 ***
as.factor(hour_id)7    1.4339621  0.0748292  19.163  < 2e-16 ***
as.factor(hour_id)8    2.2528233  0.0833386  27.032  < 2e-16 ***
as.factor(hour_id)9    2.1429245  0.0992002  21.602  < 2e-16 ***
as.factor(hour_id)10   1.9946090  0.1505606  13.248  < 2e-16 ***
as.factor(hour_id)11   1.7473705  0.1655432  10.555  < 2e-16 ***
as.factor(hour_id)12   1.6441303  0.1761312   9.335  < 2e-16 ***
as.factor(hour_id)13   1.3667894  0.1816623   7.524 5.75e-14 ***
as.factor(hour_id)14   0.7955475  0.1818304   4.375 1.22e-05 ***
as.factor(hour_id)15   0.0448128  0.1771898   0.253 0.800344
as.factor(hour_id)16  -0.7263122  0.1547507  -4.693 2.72e-06 ***
as.factor(hour_id)17  -0.6585855  0.1364411  -4.827 1.41e-06 ***
as.factor(hour_id)18  -0.4325847  0.1181362  -3.662 0.000252 ***
as.factor(hour_id)19  -0.0285958  0.1037890  -0.276 0.782923
Lag14                  0.1591265  0.0068441  23.250  < 2e-16 ***
Lag1                   0.6224335  0.0078475  79.316  < 2e-16 ***
avg_temp               0.0348264  0.0061725   5.642 1.72e-08 ***
avg_DSWRF             -0.0018410  0.0002163  -8.513  < 2e-16 ***
avg_lcloud             0.0014841  0.0011342   1.308 0.190736
avg_mcloud            -0.0016616  0.0008028  -2.070 0.038511 *
avg_hcloud             0.0003151  0.0008407   0.375 0.707853
avg_tcloud            -0.0040692  0.0010503  -3.874 0.000108 ***
avg_USWRFtop          -0.0002658  0.0003787  -0.702 0.482779
avg_DLWRFsurf         -0.0094946  0.0011758  -8.075 7.47e-16 ***
avg_USWRFsurf          0.0010828  0.0005541   1.954 0.050699 .
avg_CSNOW             -0.2428656  0.1122221  -2.164 0.030475 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.423 on 10599 degrees of freedom
  (18 observations deleted due to missingness)
Multiple R-squared:  0.9427,    Adjusted R-squared:  0.9425
F-statistic:  4713 on 37 and 10599 DF,  p-value: < 2.2e-16
```

In this model, we can see that our R^2 value increased to 0.9425, which is a good direction for our model but we can also see that some of the variables we added were insignificant to our model such as avg_lcloud, avg_hcloud and avgUSWRFtop. Thus, we create our next model by excluding these variables.

Model 3

After removing the mentioned variables from our model, the new model looks like this:

```
Call:
lm(formula = production ~ -1 + mon + as.factor(hour_id) + Lag14 +
    Lag1 + avg_temp + avg_DSWRF + avg_mcloud + avg_tcloud + avg_DLWRFsurf +
    avg_USWRFsurf + avg_CSNOW, data = model_data_tr)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0155  -0.7242   0.0792   0.7732   8.0814

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
monApr               -5.8846140  1.3035009  -4.514 6.42e-06 ***
monAug               -5.7429828  1.3455591  -4.268 1.99e-05 ***
monDec               -6.5311525  1.2882731  -5.070 4.05e-07 ***
monFeb               -6.1402056  1.2729887  -4.823 1.43e-06 ***
monJan               -6.5499829  1.2812006  -5.112 3.24e-07 ***
monJul               -5.6573904  1.3270224  -4.263 2.03e-05 ***
monJun               -5.6116927  1.3103816  -4.282 1.86e-05 ***
monMar               -5.9629716  1.2871068  -4.633 3.65e-06 ***
monMay               -5.7412767  1.3046463  -4.401 1.09e-05 ***
monNov               -6.4302950  1.3005722  -4.944 7.76e-07 ***
monOct               -6.1679612  1.3153529  -4.689 2.78e-06 ***
monSep               -5.9588546  1.3324405  -4.472 7.82e-06 ***
as.factor(hour_id)7   1.4363436  0.0744875  19.283  < 2e-16 ***
as.factor(hour_id)8   2.2580297  0.0808278  27.936  < 2e-16 ***
as.factor(hour_id)9   2.1497321  0.0915056  23.493  < 2e-16 ***
as.factor(hour_id)10  1.9578034  0.1081488  18.103  < 2e-16 ***
as.factor(hour_id)11  1.7081429  0.1148161  14.877  < 2e-16 ***
as.factor(hour_id)12  1.5980308  0.1195390  13.368  < 2e-16 ***
as.factor(hour_id)13  1.3216388  0.1217332  10.857  < 2e-16 ***
as.factor(hour_id)14  0.7461996  0.1207838   6.178 6.73e-10 ***
as.factor(hour_id)15 -0.0094597  0.1170861  -0.081   0.9356
as.factor(hour_id)16 -0.7783211  0.1030807  -7.551 4.69e-14 ***
as.factor(hour_id)17 -0.7084445  0.0947727  -7.475 8.32e-14 ***
as.factor(hour_id)18 -0.4777300  0.0885034  -5.398 6.89e-08 ***
as.factor(hour_id)19 -0.0678920  0.0840929  -0.807   0.4195
Lag14                 0.1581962  0.0068244  23.181  < 2e-16 ***
Lag1                  0.6226021  0.0078272  79.544  < 2e-16 ***
avg_temp              0.0332317  0.0052299   6.354 2.18e-10 ***
avg_DSWRF            -0.0017787  0.0002011  -8.846  < 2e-16 ***
avg_mcloud           -0.0018571  0.0007863  -2.362   0.0182 *
avg_tcloud           -0.0036635  0.0007093  -5.165 2.45e-07 ***
avg_DLWRFsurf        -0.0090989  0.0009695  -9.386  < 2e-16 ***
avg_USWRFsurf         0.0008952  0.0004951   1.808   0.0706 .
avg_CSNOW            -0.1613856  0.0972504  -1.659   0.0970 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.423 on 10603 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9427,    Adjusted R-squared:  0.9425
F-statistic:  5127 on 34 and 10603 DF,  p-value: < 2.2e-16
```

From this removal, we can see that our R^2 value didn't change, meaning that the variables removed were in fact insignificant and didn't add or take from our model.

Model 4

Next up, we can try to combine the monthly and daily seasonal dummy variables as different hours in different months may affect our model each in a unique way since the position of the sun is also related to the month as well as what time of day it is. So, we give seasonal dummy variables as month:hour and build our final time series regression model.

```
Call:
lm(formula = production ~ -1 + mon:as.factor(hour_id) + Lag14 +
    Lag1 + avg_temp + avg_DSWRF + avg_lcloud + avg_tcloud + avg_DLWRFsurf +
    avg_USWRFsurf + avg_CSNOW, data = model_data_tr)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7083 -0.5642  0.0647  0.6801  8.5590

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
Lag14                     9.524e-02  6.821e-03  13.963  < 2e-16 ***
Lag1                      6.371e-01  7.797e-03  81.712  < 2e-16 ***
avg_temp                  9.127e-02  7.887e-03  11.573  < 2e-16 ***
avg_DSWRF                -4.300e-03  3.990e-04 -10.777  < 2e-16 ***
avg_lcloud                1.926e-03  9.408e-04   2.047   0.0407 *
avg_tcloud               -3.565e-03  6.315e-04  -5.646 1.68e-08 ***
avg_DLWRFsurf            -1.521e-02  1.276e-03 -11.919  < 2e-16 ***
avg_USWRFsurf             5.044e-03  6.704e-04   7.523 5.78e-14 ***
avg_CSNOW                -1.887e-01  1.043e-01  -1.810   0.0704 .
monApr:as.factor(hour_id)6  -2.060e+01  1.937e+00 -10.634  < 2e-16 ***
monAug:as.factor(hour_id)6  -2.049e+01  1.952e+00 -10.498  < 2e-16 ***
monDec:as.factor(hour_id)6  -2.082e+01  1.917e+00 -10.860  < 2e-16 ***
monFeb:as.factor(hour_id)6  -2.052e+01  1.894e+00 -10.831  < 2e-16 ***
monJan:as.factor(hour_id)6  -2.073e+01  1.906e+00 -10.878  < 2e-16 ***
monJul:as.factor(hour_id)6  -1.984e+01  1.961e+00 -10.117  < 2e-16 ***
monJun:as.factor(hour_id)6  -1.985e+01  1.959e+00 -10.133  < 2e-16 ***
monMar:as.factor(hour_id)6  -2.057e+01  1.915e+00 -10.743  < 2e-16 ***
monMay:as.factor(hour_id)6  -2.060e+01  1.948e+00 -10.571  < 2e-16 ***
monNov:as.factor(hour_id)6  -2.090e+01  1.929e+00 -10.836  < 2e-16 ***
monOct:as.factor(hour_id)6  -2.077e+01  1.941e+00 -10.699  < 2e-16 ***
monSep:as.factor(hour_id)6  -2.051e+01  1.948e+00 -10.530  < 2e-16 ***
monApr:as.factor(hour_id)7  -1.827e+01  1.948e+00  -9.379  < 2e-16 ***
monAug:as.factor(hour_id)7  -1.925e+01  1.984e+00  -9.703  < 2e-16 ***
monDec:as.factor(hour_id)7  -2.042e+01  1.916e+00 -10.660  < 2e-16 ***
monFeb:as.factor(hour_id)7  -1.970e+01  1.893e+00 -10.406  < 2e-16 ***
monJan:as.factor(hour_id)7  -2.053e+01  1.905e+00 -10.777  < 2e-16 ***
monJul:as.factor(hour_id)7  -1.843e+01  1.987e+00  -9.273  < 2e-16 ***
monJun:as.factor(hour_id)7  -1.828e+01  1.978e+00  -9.243  < 2e-16 ***
monMar:as.factor(hour_id)7  -1.884e+01  1.915e+00  -9.839  < 2e-16 ***
monMay:as.factor(hour_id)7  -1.825e+01  1.970e+00  -9.267  < 2e-16 ***
monNov:as.factor(hour_id)7  -1.940e+01  1.926e+00 -10.073  < 2e-16 ***
monOct:as.factor(hour_id)7  -1.809e+01  1.938e+00  -9.332  < 2e-16 ***
monSep:as.factor(hour_id)7  -1.833e+01  1.959e+00  -9.356  < 2e-16 ***
monApr:as.factor(hour_id)8  -1.905e+01  1.968e+00  -9.680  < 2e-16 ***
monAug:as.factor(hour_id)8  -1.820e+01  2.026e+00  -8.984  < 2e-16 ***
monDec:as.factor(hour_id)8  -1.804e+01  1.914e+00  -9.429  < 2e-16 ***
monFeb:as.factor(hour_id)8  -1.776e+01  1.896e+00  -9.365  < 2e-16 ***
monJan:as.factor(hour_id)8  -1.855e+01  1.903e+00  -9.743  < 2e-16 ***
monJul:as.factor(hour_id)8  -1.837e+01  2.011e+00  -9.136  < 2e-16 ***
monJun:as.factor(hour_id)8  -1.867e+01  1.992e+00  -9.372  < 2e-16 ***
monMar:as.factor(hour_id)8  -1.768e+01  1.927e+00  -9.177  < 2e-16 ***
monMay:as.factor(hour_id)8  -1.838e+01  1.985e+00  -9.257  < 2e-16 ***
monNov:as.factor(hour_id)8  -1.797e+01  1.932e+00  -9.301  < 2e-16 ***
monOct:as.factor(hour_id)8  -1.774e+01  1.959e+00  -9.056  < 2e-16 ***
monSep:as.factor(hour_id)8  -1.821e+01  1.998e+00  -9.111  < 2e-16 ***
monApr:as.factor(hour_id)9  -1.845e+01  1.985e+00  -9.292  < 2e-16 ***
monAug:as.factor(hour_id)9  -1.927e+01  2.057e+00  -9.369  < 2e-16 ***
monDec:as.factor(hour_id)9  -1.780e+01  1.929e+00  -9.226  < 2e-16 ***
monFeb:as.factor(hour_id)9  -1.723e+01  1.915e+00  -8.995  < 2e-16 ***
monJan:as.factor(hour_id)9  -1.825e+01  1.917e+00  -9.521  < 2e-16 ***
monJul:as.factor(hour_id)9  -1.876e+01  2.029e+00  -9.247  < 2e-16 ***
monJun:as.factor(hour_id)9  -1.870e+01  2.003e+00  -9.335  < 2e-16 ***
monMar:as.factor(hour_id)9  -1.820e+01  1.943e+00  -9.364  < 2e-16 ***
monMay:as.factor(hour_id)9  -1.895e+01  1.998e+00  -9.487  < 2e-16 ***
monNov:as.factor(hour_id)9  -1.839e+01  1.956e+00  -9.401  < 2e-16 ***
monOct:as.factor(hour_id)9  -1.861e+01  1.989e+00  -9.357  < 2e-16 ***
monSep:as.factor(hour_id)9  -1.863e+01  2.031e+00  -9.169  < 2e-16 ***
monApr:as.factor(hour_id)10 -1.793e+01  1.911e+00  -9.381  < 2e-16 ***
monAug:as.factor(hour_id)10 -1.795e+01  1.970e+00  -9.114  < 2e-16 ***
monDec:as.factor(hour_id)10 -1.833e+01  1.919e+00  -9.548  < 2e-16 ***
monFeb:as.factor(hour_id)10 -1.809e+01  1.907e+00  -9.483  < 2e-16 ***
monJan:as.factor(hour_id)10 -1.872e+01  1.911e+00  -9.794  < 2e-16 ***
monJul:as.factor(hour_id)10 -1.790e+01  1.931e+00  -9.273  < 2e-16 ***
monJun:as.factor(hour_id)10 -1.785e+01  1.902e+00  -9.386  < 2e-16 ***
monMar:as.factor(hour_id)10 -1.827e+01  1.909e+00  -9.568  < 2e-16 ***
monMay:as.factor(hour_id)10 -1.798e+01  1.903e+00  -9.449  < 2e-16 ***
monNov:as.factor(hour_id)10 -1.867e+01  1.929e+00  -9.679  < 2e-16 ***
monOct:as.factor(hour_id)10 -1.840e+01  1.941e+00  -9.480  < 2e-16 ***
monSep:as.factor(hour_id)10 -1.832e+01  1.962e+00  -9.341  < 2e-16 ***
monApr:as.factor(hour_id)11 -1.829e+01  1.914e+00  -9.558  < 2e-16 ***
monAug:as.factor(hour_id)11 -1.841e+01  1.978e+00  -9.307  < 2e-16 ***
monDec:as.factor(hour_id)11 -1.883e+01  1.927e+00  -9.774  < 2e-16 ***
monFeb:as.factor(hour_id)11 -1.822e+01  1.914e+00  -9.518  < 2e-16 ***
monJan:as.factor(hour_id)11 -1.913e+01  1.919e+00  -9.971  < 2e-16 ***
monJul:as.factor(hour_id)11 -1.801e+01  1.936e+00  -9.305  < 2e-16 ***
monJun:as.factor(hour_id)11 -1.803e+01  1.903e+00  -9.478  < 2e-16 ***
monMar:as.factor(hour_id)11 -1.846e+01  1.915e+00  -9.641  < 2e-16 ***
monMay:as.factor(hour_id)11 -1.840e+01  1.906e+00  -9.655  < 2e-16 ***
monNov:as.factor(hour_id)11 -1.892e+01  1.938e+00  -9.765  < 2e-16 ***
monOct:as.factor(hour_id)11 -1.865e+01  1.951e+00  -9.557  < 2e-16 ***
monSep:as.factor(hour_id)11 -1.854e+01  1.972e+00  -9.397  < 2e-16 ***
monApr:as.factor(hour_id)12 -1.831e+01  1.914e+00  -9.566  < 2e-16 ***
monAug:as.factor(hour_id)12 -1.816e+01  1.981e+00  -9.168  < 2e-16 ***
monDec:as.factor(hour_id)12 -1.920e+01  1.931e+00  -9.944  < 2e-16 ***
monFeb:as.factor(hour_id)12 -1.893e+01  1.917e+00  -9.873  < 2e-16 ***
monJan:as.factor(hour_id)12 -1.942e+01  1.922e+00 -10.101  < 2e-16 ***
monJul:as.factor(hour_id)12 -1.798e+01  1.937e+00  -9.281  < 2e-16 ***
monJun:as.factor(hour_id)12 -1.810e+01  1.900e+00  -9.526  < 2e-16 ***
monMar:as.factor(hour_id)12 -1.859e+01  1.917e+00  -9.698  < 2e-16 ***
monMay:as.factor(hour_id)12 -1.780e+01  1.904e+00  -9.351  < 2e-16 ***
monNov:as.factor(hour_id)12 -1.932e+01  1.942e+00  -9.945  < 2e-16 ***
monOct:as.factor(hour_id)12 -1.896e+01  1.955e+00  -9.697  < 2e-16 ***
monSep:as.factor(hour_id)12 -1.863e+01  1.976e+00  -9.427  < 2e-16 ***
monApr:as.factor(hour_id)13 -1.881e+01  1.911e+00  -9.844  < 2e-16 ***
monAug:as.factor(hour_id)13 -1.864e+01  1.975e+00  -9.435  < 2e-16 ***
monDec:as.factor(hour_id)13 -1.993e+01  1.930e+00 -10.329  < 2e-16 ***
monFeb:as.factor(hour_id)13 -1.878e+01  1.917e+00  -9.798  < 2e-16 ***
monJan:as.factor(hour_id)13 -1.956e+01  1.921e+00 -10.182  < 2e-16 ***
monJul:as.factor(hour_id)13 -1.798e+01  1.933e+00  -9.302  < 2e-16 ***
monJun:as.factor(hour_id)13 -1.794e+01  1.895e+00  -9.471  < 2e-16 ***
monMar:as.factor(hour_id)13 -1.899e+01  1.916e+00  -9.910  < 2e-16 ***
monMay:as.factor(hour_id)13 -1.813e+01  1.896e+00  -9.562  < 2e-16 ***
monNov:as.factor(hour_id)13 -2.001e+01  1.942e+00 -10.303  < 2e-16 ***
monOct:as.factor(hour_id)13 -1.934e+01  1.953e+00  -9.905  < 2e-16 ***
monSep:as.factor(hour_id)13 -1.858e+01  1.972e+00  -9.423  < 2e-16 ***
monApr:as.factor(hour_id)14 -1.869e+01  1.905e+00  -9.811  < 2e-16 ***
monAug:as.factor(hour_id)14 -1.894e+01  1.963e+00  -9.646  < 2e-16 ***
monDec:as.factor(hour_id)14 -2.115e+01  1.926e+00 -10.982  < 2e-16 ***
monFeb:as.factor(hour_id)14 -1.956e+01  1.914e+00 -10.220  < 2e-16 ***
monJan:as.factor(hour_id)14 -2.026e+01  1.917e+00 -10.567  < 2e-16 ***
monJul:as.factor(hour_id)14 -1.836e+01  1.926e+00  -9.532  < 2e-16 ***
monJun:as.factor(hour_id)14 -1.855e+01  1.886e+00  -9.836  < 2e-16 ***
monMar:as.factor(hour_id)14 -1.901e+01  1.912e+00  -9.943  < 2e-16 ***
monMay:as.factor(hour_id)14 -1.861e+01  1.888e+00  -9.860  < 2e-16 ***
monNov:as.factor(hour_id)14 -2.045e+01  1.936e+00 -10.565  < 2e-16 ***
monOct:as.factor(hour_id)14 -2.042e+01  1.944e+00 -10.504  < 2e-16 ***
monSep:as.factor(hour_id)14 -1.957e+01  1.961e+00  -9.978  < 2e-16 ***
monApr:as.factor(hour_id)15 -1.923e+01  1.893e+00 -10.159  < 2e-16 ***
monAug:as.factor(hour_id)15 -1.953e+01  1.946e+00 -10.036  < 2e-16 ***
monDec:as.factor(hour_id)15 -2.182e+01  1.919e+00 -11.375  < 2e-16 ***
monFeb:as.factor(hour_id)15 -2.090e+01  1.911e+00 -10.938  < 2e-16 ***
monJan:as.factor(hour_id)15 -2.114e+01  1.910e+00 -11.071  < 2e-16 ***
monJul:as.factor(hour_id)15 -1.930e+01  1.915e+00 -10.077  < 2e-16 ***
monJun:as.factor(hour_id)15 -1.867e+01  1.875e+00  -9.955  < 2e-16 ***
monMar:as.factor(hour_id)15 -1.994e+01  1.905e+00 -10.464  < 2e-16 ***
monMay:as.factor(hour_id)15 -1.852e+01  1.878e+00  -9.861  < 2e-16 ***
monNov:as.factor(hour_id)15 -2.174e+01  1.926e+00 -11.288  < 2e-16 ***
monOct:as.factor(hour_id)15 -2.137e+01  1.932e+00 -11.058  < 2e-16 ***
monSep:as.factor(hour_id)15 -2.054e+01  1.947e+00 -10.550  < 2e-16 ***
monApr:as.factor(hour_id)16 -2.034e+01  1.904e+00 -10.685  < 2e-16 ***
monAug:as.factor(hour_id)16 -2.077e+01  1.951e+00 -10.645  < 2e-16 ***
monDec:as.factor(hour_id)16 -2.168e+01  1.926e+00 -11.254  < 2e-16 ***
monFeb:as.factor(hour_id)16 -2.167e+01  1.907e+00 -11.364  < 2e-16 ***
monJan:as.factor(hour_id)16 -2.169e+01  1.910e+00 -11.352  < 2e-16 ***
monJul:as.factor(hour_id)16 -2.064e+01  1.919e+00 -10.759  < 2e-16 ***
monJun:as.factor(hour_id)16 -2.018e+01  1.903e+00 -10.603  < 2e-16 ***
monMar:as.factor(hour_id)16 -2.124e+01  1.907e+00 -11.133  < 2e-16 ***
monMay:as.factor(hour_id)16 -2.022e+01  1.901e+00 -10.641  < 2e-16 ***
monNov:as.factor(hour_id)16 -2.189e+01  1.941e+00 -11.274  < 2e-16 ***
monOct:as.factor(hour_id)16 -2.219e+01  1.951e+00 -11.373  < 2e-16 ***
monSep:as.factor(hour_id)16 -2.201e+01  1.956e+00 -11.254  < 2e-16 ***
monApr:as.factor(hour_id)17 -2.081e+01  1.904e+00 -10.931  < 2e-16 ***
monAug:as.factor(hour_id)17 -2.168e+01  1.944e+00 -11.149  < 2e-16 ***
monDec:as.factor(hour_id)17 -2.075e+01  1.913e+00 -10.847  < 2e-16 ***
monFeb:as.factor(hour_id)17 -2.136e+01  1.901e+00 -11.235  < 2e-16 ***
monJan:as.factor(hour_id)17 -2.076e+01  1.901e+00 -10.916  < 2e-16 ***
monJul:as.factor(hour_id)17 -2.170e+01  1.918e+00 -11.315  < 2e-16 ***
monJun:as.factor(hour_id)17 -2.073e+01  1.905e+00 -10.883  < 2e-16 ***
monMar:as.factor(hour_id)17 -2.114e+01  1.907e+00 -11.086  < 2e-16 ***
monMay:as.factor(hour_id)17 -2.071e+01  1.901e+00 -10.891  < 2e-16 ***
monNov:as.factor(hour_id)17 -2.097e+01  1.931e+00 -10.858  < 2e-16 ***
monOct:as.factor(hour_id)17 -2.144e+01  1.944e+00 -11.029  < 2e-16 ***
monSep:as.factor(hour_id)17 -2.214e+01  1.948e+00 -11.364  < 2e-16 ***
monApr:as.factor(hour_id)18 -2.124e+01  1.902e+00 -11.168  < 2e-16 ***
monAug:as.factor(hour_id)18 -2.183e+01  1.934e+00 -11.284  < 2e-16 ***
monDec:as.factor(hour_id)18 -2.072e+01  1.911e+00 -10.844  < 2e-16 ***
monFeb:as.factor(hour_id)18 -2.028e+01  1.884e+00 -10.764  < 2e-16 ***
monJan:as.factor(hour_id)18 -2.042e+01  1.892e+00 -10.793  < 2e-16 ***
monJul:as.factor(hour_id)18 -2.124e+01  1.914e+00 -11.094  < 2e-16 ***
monJun:as.factor(hour_id)18 -2.090e+01  1.906e+00 -10.970  < 2e-16 ***
monMar:as.factor(hour_id)18 -2.032e+01  1.900e+00 -10.694  < 2e-16 ***
monMay:as.factor(hour_id)18 -2.130e+01  1.902e+00 -11.203  < 2e-16 ***
monNov:as.factor(hour_id)18 -2.077e+01  1.927e+00 -10.780  < 2e-16 ***
monOct:as.factor(hour_id)18 -2.075e+01  1.931e+00 -10.747  < 2e-16 ***
monSep:as.factor(hour_id)18 -2.121e+01  1.935e+00 -10.964  < 2e-16 ***
monApr:as.factor(hour_id)19 -2.016e+01  1.897e+00 -10.623  < 2e-16 ***
monAug:as.factor(hour_id)19 -2.052e+01  1.915e+00 -10.712  < 2e-16 ***
monDec:as.factor(hour_id)19 -2.078e+01  1.914e+00 -10.856  < 2e-16 ***
monFeb:as.factor(hour_id)19 -2.026e+01  1.882e+00 -10.766  < 2e-16 ***
monJan:as.factor(hour_id)19 -2.050e+01  1.896e+00 -10.810  < 2e-16 ***
monJul:as.factor(hour_id)19 -2.052e+01  1.906e+00 -10.766  < 2e-16 ***
monJun:as.factor(hour_id)19 -2.020e+01  1.905e+00 -10.603  < 2e-16 ***
monMar:as.factor(hour_id)19 -2.009e+01  1.892e+00 -10.618  < 2e-16 ***
monMay:as.factor(hour_id)19 -2.012e+01  1.899e+00 -10.596  < 2e-16 ***
monNov:as.factor(hour_id)19 -2.082e+01  1.929e+00 -10.792  < 2e-16 ***
monOct:as.factor(hour_id)19 -2.082e+01  1.934e+00 -10.764  < 2e-16 ***
monSep:as.factor(hour_id)19 -2.058e+01  1.918e+00 -10.733  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.355 on 10460 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9488, Adjusted R-squared:  0.9479
F-statistic:  1094 on 177 and 10460 DF,  p-value: < 2.2e-16
```

Our final model increased it's R^2 value up to 0.9479, which means that our time series regression model will be able to explain 94.79% of the production values.

As the performance of our predictions were tested via their WMAPE values, we also looked at the WMAPE values to see the performances of the different models we built. For our time series regressions, WMAPE values are as follows. For that, first we predict the values in the period February 1st 2024 and May 15th 2024 and then calculate each models WMAPE's according to the forecast values we got.

model <chr>	**WMAPE** <dbl>	**Model_Type** <chr>
Model 1	0.2100034	Time Series Regression
Model 2	0.2026244	Time Series Regression
Model 3	0.2023981	Time Series Regression
Model 4	0.2043825	Time Series Regression

We can see that the best performing model that we had was model 3 according to the WMAPE values which shows that adding monthly changing hour seasonality didn't work for forecast accuracy even though the R^2 value showed an increase

Model 5

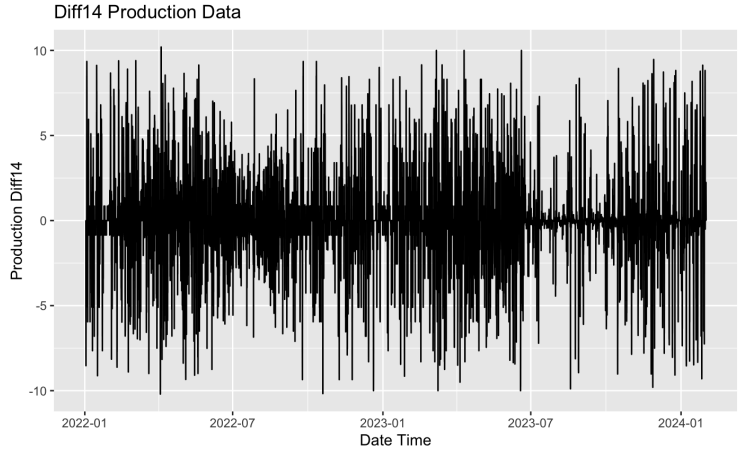
Next up, we continue with the ARIMA models. For that, first we look for the stationarity of our data and later eliminate them if there exists any.

We use KPSS unit root test to look for the stationarity.

```
        KPSS Test for Trend Stationarity

data:  model_data_tr$production
KPSS Trend = 2.4359, Truncation lag parameter = 12, p-value = 0.01
```

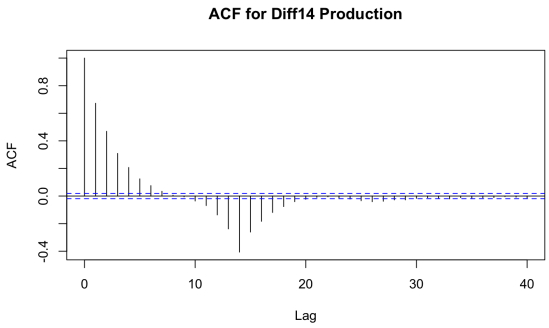
We see that p-value is lower than 0.05, thus we reject the null hypothesis and say that our model is not stationary. To eliminate that, we differentiate our data and apply the test again.

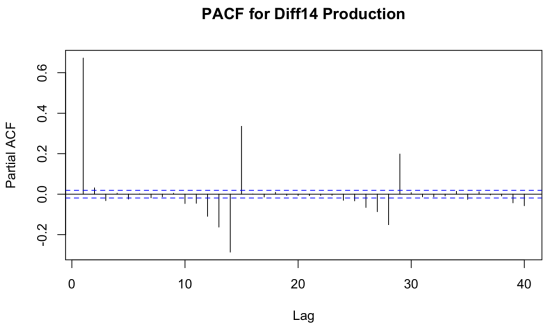
```
        KPSS Test for Trend Stationarity

data:  model_data_tr$diff14
KPSS Trend = 0.0022202, Truncation lag parameter = 12, p-value = 0.1
```



This time, we are able to say that our data is stationary as we can see p-value is greater than 0.1. Now, we can continue to build our ARIMA model.

First, we look our ACF and PACF plots to see if we can come up with the p,d,q values for our ARIMA model.

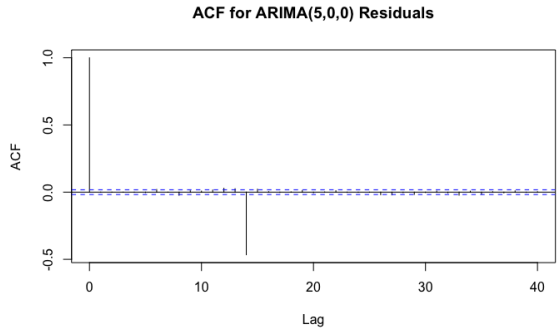


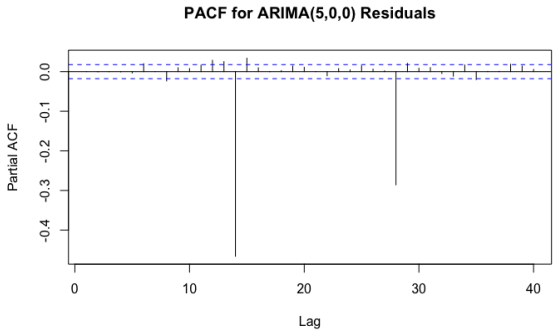


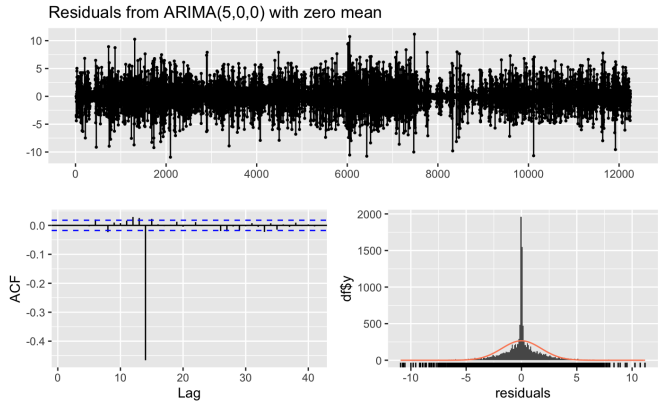
Although we differentiated our data and have a stationary one, we can still see spikes in the PACF plot in lag14, lag28 and in every 14 time periods. We can also see that ACF plot have a negative spike in the lag14, which indicates that there may still be seasonality involved with our data. But since we had the KPSS test indicate that our data is stationary, let's continue to build an ARIMA model to observe the results.

```
 Best model: ARIMA(5,0,0) with zero mean

Series: production_available$diff14
ARIMA(5,0,0) with zero mean

Coefficients:
        ar1     ar2      ar3     ar4      ar5
      0.674  0.0358  -0.0316  0.0226  -0.0209
s.e.  0.009  0.0109   0.0109  0.0109   0.0090

sigma^2 = 3.17:  log likelihood = -24419.11
AIC=48850.22   AICc=48850.23   BIC=48894.69
```

Our auto.arima() function gave us ARIMA(5,0,0) to be used for this forecasting. Let's check the residuals ACF, PACF and distribution to get a better sense on how our ARIMA(5,0,0) model did.







We can see in our model that there still exists a high autocorrelation in lag14, and we also observe that the distributions of the differences are gathered mostly around 0. It resembles the normal distribution except the high spike around 0 value. Let's check the WMAPE for performance measure on forecasts.

model <chr>	**WMAPE** <chr>	**Model_Type** <chr>
Model 1	0.210003375586762	Time Series Regression
Model 2	0.202624443172111	Time Series Regression
Model 3	0.202398090600185	Time Series Regression
Model 4	0.20438247408606	Time Series Regression
Model 5	0.388694412541202	ARIMA(5,0,0)

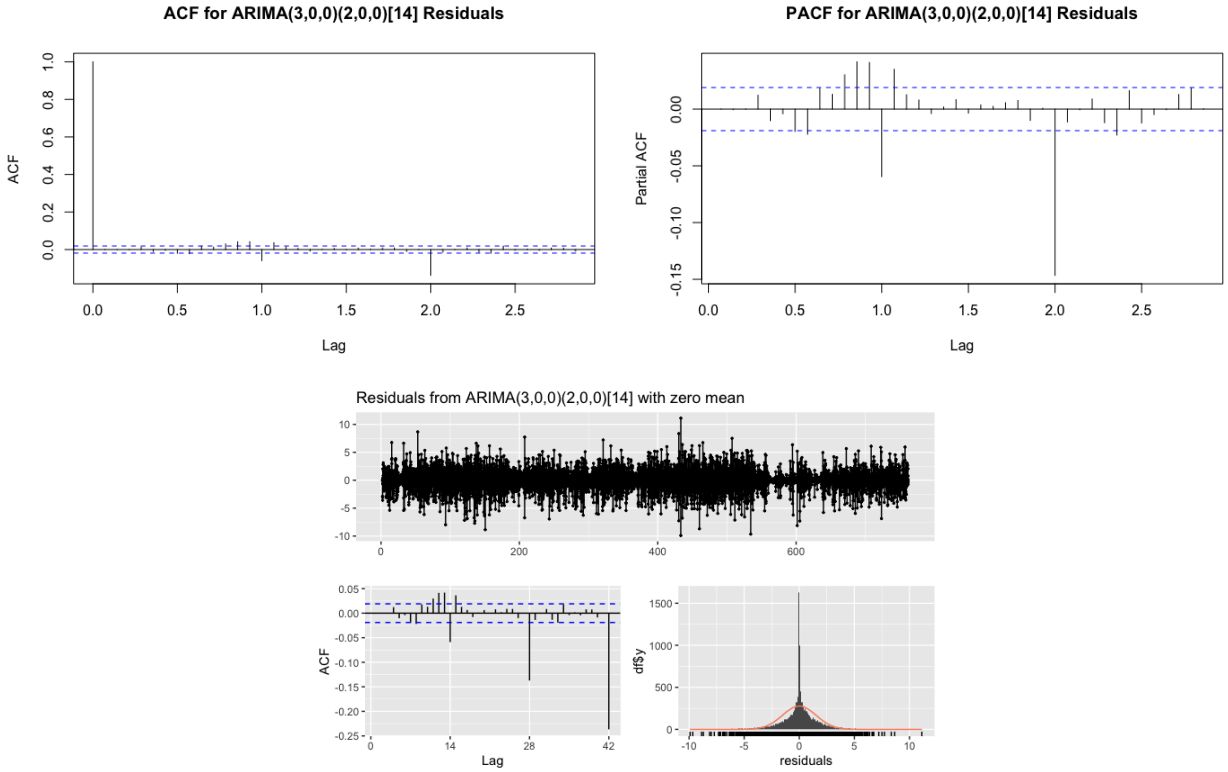
Unfortunately, our ARIMA model didn't work as well as our time series regression models.

Model 6

After observing the spikes at lag14 on both ACF and PACF plots, let's build a SARIMA model to see if it improves our model or not.

```
 Best model: ARIMA(3,0,0)(2,0,0)[14] with zero mean

Series: ts_data
ARIMA(3,0,0)(2,0,0)[14] with zero mean

Coefficients:
         ar1     ar2      ar3     sar1     sar2
      0.6916  0.0509  -0.0337  -0.6013  -0.2745
s.e.  0.0097  0.0118   0.0097   0.0094   0.0093

sigma^2 = 2.311:  log likelihood = -19555.48
AIC=39122.97   AICc=39122.98   BIC=39166.6
```

After adding a seasonality with period 14 to our auto.arima() function, we obtained ARIMA(3,0,0)(2,0,0)[14] as our SARIMA model. We can also observe that AIC and BIC values are decreased compared to the ARIMA(5,0,0) model, which is a good sign. Let's check the ACF and PACF for residuals once again.



We can see from the ACF and PACF plots that our pikes at still shows the similar spikes and the distribution of the residuals are still similar to our initial ARIMA(5,0,0) model, meaning that in terms of the residuals and the predictions, it didn't improved as significantly as we desired and still have some seasonal component that wasn't explained via the SARIMA model.

Now let's look at the WMAPE values for one last time with our 6th model added.

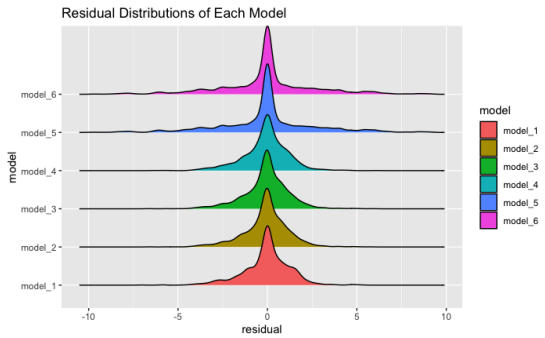
model <chr>	**WMAPE** <chr>	**Model_Type** <chr>
Model 1	0.210003375586762	Time Series Regression
Model 2	0.202624443172111	Time Series Regression
Model 3	0.202398090600185	Time Series Regression
Model 4	0.20438247408606	Time Series Regression
Model 5	0.388694412541202	ARIMA(5,0,0)
Model 6	0.387105724373435	ARIMA(3,0,0)(2,0,0)[14]

We can see that there was a slight improvement upon using SARIMA model instead of the ARIMA model.

Result

When we look at our 6 models that we built, 4 time series regression, 1 ARIMA and 1 SARIMA, we can compare their WMAPE values and the residual distributions to decide on a model to use for forecasting.

model <chr>	**WMAPE** <chr>	**Model_Type** <chr>
Model 1	0.210003375586762	Time Series Regression
Model 2	0.202624443172111	Time Series Regression
Model 3	0.202398090600185	Time Series Regression
Model 4	0.20438247408606	Time Series Regression
Model 5	0.388694412541202	ARIMA(5,0,0)
Model 6	0.387105724373435	ARIMA(3,0,0)(2,0,0)[14]



We can see that the best resulting WMAPE value comes from model 3, which we could use for forecasting but as we don't have time weather condition data for future time periods it would be hard to forecast both the weather conditions and the production amounts. Thus, we decide on whether to use ARIMA or SARIMA for our predictions. In this case, we used ARIMA for our predictions since we believed that already seasonal differenced productions we had would be enough of a seasonal component for forecasting future values even though using SARIMA method would decrease the WMAPE values slightly.

Upon looking at the distribution graph, we can also see that time series regression residuals look more like a normal distribution which is consistent with the white noise series assumption whereas model 5 and 6 look more gathered around the 0 mean value.

Conclusion and Future Work

Through the project, we worked on production and weather data from 01.01.2022 till 15.05.2024 and made daily forecasting afterward for the time period we were asked to. We build 6 models, 4 time series regression, 1 ARIMA and 1 SARIMA and trained them with the data from 01.01.2022 till 01.02.2024.

Following the model building phase, we tested their performances on the data ranging from 01.02.2024 to 15.05.2024. This testings showed us that our best model was model 3, but since other variables may also be needed for time series regression forecasting, we used ARIMA approach for the forecasts.

One of our shortcomings was that we didn't incorporate the possibly related weather data to our ARIMA model, which could've improved our forecasts and performance of our models significantly, so it is one of the future improvements that we can make. Other is that we could seperate each hour and apply our model to every single hour seperately, which could've given us a better result and could solve the seasonal spikes we got from our ACF and PACF plots of ARIMA models. Another point that we could further work on is that the time limits we took. Although for most of the months between 8PM and 5 AM, the production is mostly 0, there are still some hours in a year where solar energy production may be observed within these limits. We can work on a model in which this behavior is worked upon and considered in the models.

In general, forecasting required a long process and detailed analysis of the data. Adding too few or too much data can weaken the performance of the models while adding unrelated or highly correlated data can also have the same effect too. Thus, choosing, using and applying the right transformations to each data added to the models play a significant role in a strong model building.

Code

Our R Code for the project